Newcastle University

LEEDS BECKETT UNIVERSITY

KTH VETENSKAP OCH KONST

UNIVERSIDADE D COIMBRA

ЧНУ

1920

MCMXVIII

ТНЕУ

ЧДУ

НТУ

Vivere est cogitare

ІПМЕ НАН України

IT Альянс

SmartMe UNIVERSITY

Internet of Things
for Industry and Human Applications

VOLUME 1
FUNDAMENTALS AND TECHNOLOGIES

ALIoT

Internet of Things
for Industry and Human Applications

Volume 1
Fundamentals and Technologies

Funded by the
Erasmus+ Programme
of the European Union

Ministry of Education and Science of Ukraine
National Aerospace University "Kharkiv Aviation Institute"

Internet of Things for Industry and Human Applications

Volume 1

Fundamentals and Technologies

Edited by V. S. Kharchenko

Project ERASMUS+ ALIOT
"Internet of Things:
Emerging Curriculum for Industry and Human Applications"
(573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP)

2019

UDC 62:004=111

I73

Reviewers: Dr. Mario Fusani, ISTI-CNR, Pisa, Italy
Dr. Olga Kordas, KTH University, Stockholm, Sweden
Viktor Kordas, KTH University, Stockholm, Sweden

I73 Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 1. Fundamentals and Technologies / V. S. Kharchenko (ed.) - Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019. - 605p.

ISBN 978-617-7361-80-9
ISBN 978-617-7361-81-6

Three-volume book contains theoretical materials for lectures and training modules developed in frameworks of project "Internet of Things: Emerging Curriculum for Industry and Human Applications /ALIOT" (Project Number: 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP, 2016-2019) funded by EU Program ERASMUS+. Volume 1 describes challenges, principles and technologies of Internet of Things (IoT) and Internet of Everything (IoE). The book consists of 4 parts for corresponding MSc courses: fundamentals of IoT, IoE and Web of Things (sections 1-4), data science for IoT (sections 5-8), mobile and hybrid IoT computing (sections 9-11), IoT technologies for cyber physical systems (sections 12-15). The book prepared by Ukrainian university teams with support of EU academic colleagues of the ALIOT consortium.

The book is intended for MSc and PhD students studying IoT technologies, software and computer engineering and science. It could be useful for lecturers of universities and training centers, researchers and developers of IoT systems.

Fig.: 172. Ref.: 606. Tables: 30.

Approved by Academic Council of National Aerospace University "Kharkiv Aviation Institute" (record № 4, December 19, 2018).

ISBN 978-617-7361-81-6

Міністерство освіти і науки України
Національний аерокосмічний університет
ім. М. Є. Жуковського «Харківський Авіаційний Інститут”

Інтернет речей для індустріальних і гуманітарних застосунків

Том 1

Основи і технології

Редактор Харченко В.С.

Проект ERASMUS+ ALIOT
“Інтернет речей: нова освітня програма для потреб
промисловості та суспільства”
(573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP)

2019

УДК 62:004=111
І73

Рецензенти: Др. Маріо Фузані, ISTI-CNR, Піза, Італія Др.
Ольга Кордас, KTH University, Стокгольм, Швеція
Віктор Кордас, KTH University, Stockholm, Sweden

І73 **Інтернет речей для індустріальних і гуманітарних застосунків. У трьох томах. Том 1. Основи і технології** / За ред. В. С. Харченка. – Міністерство освіти і науки України, Національний аерокосмічний університет ХАІ, 2019. – 605 с.

ISBN 978-617-7361-80-9
ISBN 978-617-7361-81-6

Книга, що складається з трьох томів, містить теоретичні матеріали для лекцій та тренінгів, розроблених в рамках проекту Internet of Things: Emerging Curriculum for Industry and Human Applications / ALIOT, 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP, 2016-2019, що фінансується програмою ЄС ERASMUS +. Том 1 описує проблеми, принципи і технології Інтернету речей (IoT) і Інтернету всього (IoE). Книга складається з 4 частин для відповідних магістерських курсів: основи IoT, IoE і Веб речей (розділи 1-4), наука про дані для IoT (розділи 5-8), мобільні і гібридні IoT обчислення (розділи 9-11), IoT технології для кіберфізіческіх систем (розділи 12-15).

Книга підготовлена українськими університетськими командами за підтримки колег з академічних закладів країн ЄС, що входять до консорціуму проекту ALIOT.

Книга призначена для магістрантів і аспірантів, які вивчають технології IoT, програмну і комп'ютерну інженерію, комп'ютерні науки. Може бути корисною для викладачів університетів і навчальних центрів, дослідників і розробників систем IoT.

Рис.: 172. Посилань: 606. Таблиць: 30.

Рекомендовано до видання вченою радою Національного аерокосмічного університету імені М.Є. Жуковського «Харківський авіаційний інститут» (протокол № 4 від 19 грудня 2018 г.).

УДК 62:004=111

ISBN 978-617-7361-81-6

CONTENTS

PREFACE

ALIOT ERASMUS+ project. Three-volume book contains material for lectures and training modules developed during carrying out of project "**I**nternet **o**f **T**hings: Emerging Curriculum for Industry and Human **A**pp**l**ications /**ALIOT**[1]" (Project Number: 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP, 2016-2019) funded by EU Program ERASMUS+. Main ALIOT project objectives are development and transfer of innovative Internet of Things (IoT) and Internet of Everything (IoE) related research ideas and practices between the academic and industrial sectors and for society as whole.

The tasks of the ALIOT project are the following:

1) to introduce a Multi-domain and Integrated Internet of Things (IoT) programme and develop 4 courses for MSc students:

- MC1 Fundamentals of IoT and IoE,
- MC2 Data science for IoT and IoE,
- MC3 Mobile and hybrid IoT-based computing,
- MC4 IoT technologies for cyber physical systems;

2) to introduce a Multi-Domain and Integrated IoT programme and develop 4 courses for doctoral students:

- PC1 Simulation of IoT and IoE-based systems,
- PC2 Software defined networks and IoT,
- PC3 Dependability and security of IoT,
- PC4 Development and implementation of IoT-based systems;

3) to establish multi-domain IoT cluster network and develop 6 training courses for human and industry applications:

- ITM1 IoT for smart energy grid,
- ITM 2 IoT for smart building and city,
- ITM 3 IoT for intelligent transport systems,
- ITM 4 IoT for health systems,

[1] *The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

- ITM 5 IoT for ecology monitoring systems,
- ITM 6 IoT for industrial systems.

The tasks of the project have been solved by ALIOT consortium of Ukraine and EU countries universities and organizations:

- Newcastle University (NU), United Kingdom (grant holder and EU coordinator);
- National Aerospace University "Kharkiv Aviation Institute" (KhAI), Ukraine (national coordinator);
- Leeds Beckett University (LBU), United Kingdom;
- Coimbra University (CU), Portugal;
- University KTH, Stockholm, Sweden;
- lnstitute of lnformation Science and Technologies ISTI-CNR, Pisa, Italy;
- Chernivtsi National University (ChNU), Ukraine;
- East Ukraine National University (EANU), Ukraine;
- Odesa National Polytechnic University (ONPU), Ukraine;
- Ternopil National Economic University (TNEU), Ukraine;
- Petro Mogila Black Sea National University (PMBSNU), Mykolaiv, Ukraine;
- Zaporizhzhya National Technical University (ZNTU), Ukraine;
- Pukhov lnstitute for Modelling in Energy Engineering (IPME), National Academy of Science of Ukraine, Kyiv, Ukraine;
- IT-Alliance (ITA), Ukraine;
- Smart.ME company (SM), Ukraine.

ALIOT books. To assure the ALIOT courses the following books are edited:

- Three volume multi-book “Internet of Things for Industry and Human Applications” for theoretical/lecture part of courses:

Volume 1. Fundamentals and Technologies (MSc study),

Volume 2. Modelling and Development (PhD study),

Volume 3. Assessment and Implementation (training modules);

- 4 practicum books for MSc courses:
- 4 practicum books for PhD courses;
- 6 books for domain oriented training modules.

The volumes consists of 14 parts according with list of MSc (Parts 1-4), PhD (Parts 5-8) and training (Parts 9-14) courses. Parts are called according with corresponding courses (Parts 1-4 as MC1-MC4, Part 5-8 as PC1-PC14, Parts 9-14 as ITM1-ITM6).

Parts consist of the sections 1-56 (4 sections for courses MC1-MC2, MC4, PC1-PC4, ITM1-ITM5; 3 sections MC3, 5 sections for ITM6). Section 0 introduces into the multi-book.

Contents and authors of the Volume 1. Volume 1 consists of Introduction (section 0), parts 1-4, sections 1-15.

The need of new curriculum in the Internet of things (IoT) for MSc, PhD and engineering levels of education is described in the section 0 (Introduction). IoT publication dynamics is analysed. The joint project on curriculum development ALIOT, financed in the frame of Erasmus+ programme, is discussed. The project ensures adaptation of academic programs in Ukraine and other countries to the needs of the European labor market, thus enhancing the opportunities of academic and labor abundant. The ALIOT covers hot domains of IoT applications such as health systems, intellectual transport systems, ecology and industry 4.0 systems, smart grid, smart buildings and city. The description of interdisciplinary multi-domain and transnational programmes of MSc and PhD levels is introduced with the mechanisms of intensive capacity building measures as well as the establishment of multi-domain IoT cluster network in Ukraine is given.

Authors of the section 0 are Assoc. Prof., Dr A. V. Boyarchuk (KhAI), Dr. O. O. Illiashenko (KhAI), Prof., DrS. V. S. Kharchenko (KhAI), Prof. DrS., D. A. Maevsky (ONPU), Prof., Dr. Chris Phillips (NU), Dr. A. P. Plakhteev (KhAI), L. V. Vystorobska (ONPU).

PART I. FUNDAMENTALS OF IOT AND IOE.

Section 1 analyses concept of IoT, challenges, methodological issues and solutions in area of IoT application, an extended concept of IoT is suggested. It discusses the methodological and practical issues of implementing Big Data Analytics (BDA) and IoT systems and tools in context of cyber security and safety assessment and assurance. Concepts of safety and cyber safety considering security and cyber security attributes for critical application are analysed. The benefits and limitations of application of BDA and IoT based technologies in safety critical systems are discussed. Industrial cases such as Internet of Drones based NPP accident monitoring system, IoT based monitoring and control system, system for prediction of software dependability are described. The recommendations and limitations of BDA and IoT application in safety critical systems are formulated.

Authors of the section 1 are Prof., DrS. V. S. Kharchenko (KhAI), Dr. O. O. Illiashenko (KhAI), Prof., Dr. Andrzej Rucinski (UNH), Dr. Ah-Lian Kor (LBU).

Section 2 considers is devoted to review of IoT/WoT technologies, curriculum for future specialists on IoT/WoT technologies development and experience of teaching of discipline: “Technologies and tools of WoT application development” in different variations. The review of existing WoT/IoT technologies demonstrates that server-side programming language of Web Thing API built with consideration of architecture of REST is JS environment in Node.JS. Considering Web Thing API programming language, the curriculum suggests the stack of technologies for development of WoT applications using JavaScript/Node.JS. The discipline “Technology and tools for developing Web applications”, supplemented by the sections of “Cloud Computing”, “IoT/WoT” and related reviewed curriculum is taught in NTU “KhPI” at the Department of Information Systems and NAU “KhAI” at the Department of Computer Systems, Networks and Cyber Security under the courses “Web of Things” and “Industrial Internet of Things”.

Authors of the section 2 are Assoc. Prof., Dr. V. A. Tkachenko (NTU KhPI), Senior Researcher, DrS. E. V. Brezhniev (KhAI).

The analysis of the available solutions and performance regarding the IoT systems has been conducted in the section 3. The standards and recommendations in the area of IoT systems architecture, security, technologies have been analyzed. The features of the presentation of models of systems of the Internet of things, the requirements for their organization are considered. The analysis is carried out and the basic metrics applicable to the evaluation of the criteria of the Internet of things systems are described. The features of the Internet of things domains are considered.

Authors of the section 3 are Assoc. Prof., Dr M. O. Kolisnyk, Prof., DrS. V. S. Kharchenko (KhAI).

Section 4 considers communication techniques of IoT with emphasis on protocols enabled. Such topics as architecture of networks used in IoT, delays in wireless sensor networks, Bluetooth 5.0 capabilities and analysis of applicability in the area of IoT of such protocols as HTTP/HTTPS, MQTT, AMQP and cloud architecture for IoT are studied. Comparative analysis of mentioned protocols has carried out. Methods for determining the effective speed of error-correcting codes and comparing their speeds are considered. Comprehensive indicator of the energy efficiency of error-correcting codes for IoT devices is suggested.

Authors of the section 4 are Senior Researcher, DrS. O. A. Chemeris, Dr, Senior Researcher S. Ya. Hilgurt (IPME), Assoc. Prof., Dr. V. Y. Pevnev (KhAI), Senior Lecturer M. V. Tsuranov (KhAI).

PART II. DATA SCIENCE FOR IOT AND IOE.

Section 5 presents a short introduction to data science for IoT. An overview of data characteristics, appropriate approaches and data science techniques and algorithms applicable to IoT data is provided. IoT and IoE ecosystem is described. Scientific analytics models used in IoT verticals, as well as data fusion and data processing from IoT devices, are discussed. One of the primary goals of this chapter to give insight on how IoT data from devices – from sensors to end devices can be extracted and analyzed to uncover information.

Section 6 discusses the principles and technologies of mining and data processing for IoT systems. The features of the use of data mining for IoT data, models and data mining methods for IoT, data mining of stream data and massive data are described.

Authors of the sections 5,6 are Prof., DrS. I. S. Skarga-Bandurova, Dr. T. O. Biloborodova (EUNU).

Section 7 presents the progress and practical achievements that were conducted at the intersection area of the Internet of Things technology and Artificial Intelligence. The principles of Deep Neural Networks are considered in details, including given examples, the main aspects of design, usage and implementation for Internet of Things.

Authors of the section 7 are Assoc. Prof., Dr. V. S. Koval, Dr. M. P. Komar (TNEU).

Section 8 discusses big data issue in the context of IoT and studies the fundamental trade-off between data consistency and delays in distributed data storages. The primary focus of the paper is on investigating the interplay between performance (response time and throughput) of the Cassandra NoSQL database and consistency settings. The chapter reports the results of benchmarking the read and write performance of a replicated Cassandra cluster, deployed in the Amazon EC2 Cloud. We present quantitative results showing how different consistency settings affect Cassandra performance under different workloads. Finally, we generalize our experience by proposing a benchmarking-based methodology for run-time optimization of consistency settings to achieve the maximal Cassandra performance and still guarantee the strong data consistency under mixed workloads.

Authors of the section 8 are Assoc. Prof., Dr. O. M. Tarasyuk, Prof., DrS. A. V. Gorbenko (KhAI).

PART III. MOBILE AND HYBRID IOT-BASED COMPUTING.

Section 9 covers the basic of Android and iOS development and connection of IoT devices, mainly wearable's, to the mobile applications. As this book is intended for MSc-, PhD- students and engineers in information technologies, this section covers various topics starting with nowadays mobile and IoT standards development, user interface design essentials, recommended by Google and Apple application architecture for mobile and wearable devices and endings with uploading developed application to the App Store and Google Play markets.

Section 10 describes a merging of two cutting edge technologies - Cloud computing and Internet of Things, and how the benefits of this connection can be applied for mobile applications development. As this study material is intended for MSc-, PhD- students as well as for experienced IT engineers, here we provide an analysis of nowadays Cloud services economics and overview of most widely known Cloud architectures and infrastructures that can support the Android and iOS applications development.

Authors of the sections 9, 10 are Dr V. O. Butenko (KhAI), D. A. Butenko (KhAI), Dr O. B. Odarushchenko (PSAA).

This section describes the integration of two from the most discussed information technologies Big Data and IoT. Big Data Platform of IoT Services is described. Big Data platform stack and tools are analysed. Architectures of Big Data systems such as Lambda, Kappa and requirements to systems are discussed. An engineering aspects of the Big Data from an IoT perspective are analysed.

Authors of the section 11 are Assoc. Prof., Dr .O. M. Odarushchenko (KhAI), Assoc. Prof., Dr O. Y. Strjuk (KhAI).

PART IV. IOT TECHNOLOGIES FOR CYBER PHYSICAL SYSTEMS.

Section 12 describes the results of the analysis of the current state of development of cyber-physical systems (CPS) and the role of Internet of Things technologies in their formation. It is shown that the synergy processes of achievements in the field of CPS and IoT allows solving complex issues of modern industry and humanitarian sphere. They are the basis for the development of IoE, SNSS, and self-organized cybernetic technologies. One of the approaches for system analysis and synthesis of modern CPS is proposed and the decisive role of IoT technologies in this process is substantiated. The models of CPS and IoT and the possibilities of their improvement using methods of system analysis, Petri nets, theory of queuing systems are considered.

Section 13 describes the tasks of CPS analysis and synthesis. The place of IoT technologies in these tasks are considered in three aspects: analysis of the modern software/hardware base for the development of cyber components, providing a reliable interface between system components at conceptual model levels, and the capabilities of CPS modeling and development software.

Authors of the sections 12, 13 are Assoc. Prof., Dr H. I. Vorobets, Dr O. I. Vorobets, PhD student V. E. Horditsa (ChNU).

Section 14 discusses the research on creating a network architecture concept that uses Power over Ethernet (PoE) as a method for transferring data and power over a single medium, conjoined with the principals of neural networks as well as decentralized and remote computing for data processing. The concept used a Cisco Catalyst 4507R+E switch and utilized cloud and on-board computing to provide an easily scalable and adaptable architecture that can be modularly integrated into existing solutions. The prototype set-up was tested on RaspberryPi microcontroller boards as sensor hubs, and used DigitalOcean as the cloud computing service of choice. The Movidius Neural Compute chip was used to deploy the neural networks for the relevant data processing and deep learning. The proposed architecture shows great promise on the feasibility of creating modular systems that unite the concepts of IoT, PoE and Neural Network approaches. These approaches can be used to develop cyber physical systems for different human and industry domains.

Authors of the section 14 are PhD student I. M. Lobachev (ONPU), Assoc. Prof., Dr M.V. Lobachev (ONPU), Prof., DrS. V.S. Kharchenko (KhAI).

Section 15 focuses on model-based design of cyber-physical systems and how to apply two methodologies within this approach. The first methodology is based on the MARTE profile of UML, which is intended to model embedded real-time systems. The second one uses the SysML language, which allows to model the physical and computational parts of cyber-physical systems. The combined use these methodologies in model-based design allow to comprehensively model and analyze the functional and non-functional properties of cyber-physical systems.

Author of the section 15 is Assoc. Prof., Dr. R. K. Kudermetov (ZNTU).

Volumes 1-3 edited by Prof., DrS. V. S. Kharchenko (KhAI). Camera-ready versions of Volumes 1-3 were prepared by Dr. O. O. Illiashenko (KhAI).

Acknowledgements. The editor and authors of this book would like to express their appreciation and gratitude to all colleagues from partner universities and organizations for regular discussion, advises and support.

We thank colleagues who develop the project ERASMUS+ ALIOT “Internet of Things: Emerging Curriculum for Industry and Human Applications” http://aliot.eu.org/ and participate in discussions of topics related to IoT during a few meetings and schools in Sweden (Stockholm, December 2016), Ukraine (February 2017, 2018, Chernivtsi; May 2017, Mykolaiv; May 2018, Kyiv; February 2019, Ternopil; May 2019, Zaporizhzhya), Portugal (Coimbra, October 2017), United Kingdom (Newcastle-Leeds, July 2018).

We thank participants of International Workshops on Cyber Physical Systems and Internet of Things Dependability (WS CyberIoT-DESSERT) at the conferences IDAACS (September 2017, Bucharest, Romania), DESSERT (May 2018, Kyiv, Ukraine) and monthly Seminar on Critical Computer Technologies and Systems (CriCTechS, KhAI, 2017-2019) at the Department of Computer Systems, Networks and Cybersecurity for discussion of preliminary project results in point of view research, development and education issues.

We would like to thank reviewers of the multi-book:

- Dr. Mario Fusani (ISTI-CNR, Pisa, Italy);

- Dr Olga Kordas (KTH University, Stockholm, Sweden)

- Senior Project Manager Viktor Kordas (KTH University, Stockholm, Sweden)

for very helpful advises and valuable recommendations.

0. INTRODUCTION. STATE OF THE ART AND ALIOT BASED EDUCATION ON INTERNET OF THINGS

Dr A. V. Boyarchuk (KhAI), Dr. O. O. Illiashenko (KhAI),
Professor, DrS V. S. Kharchenko (KhAI), D. A. Maevsky (ONPU),
Professor, Dr. C. Phillips (NU), Dr. A. P. Plakhteev (KhAI),
L. Vystorobska (ONPU)

Contents

Abbreviations

ECTS – European Credit Transfer and Accumulation System

EQF – European Qualifications Framework

LLL – Life Long Learning

IITN – International Innovation Transfer Network

IoT – Internet of Things

IoE – Internet of Everything

MC – Master course

PC – PhD course

ITM – Industrial Training Module

0.1 State of the art and an approach

0.1.1 Motivation

The International collaboration between organizations for research and education purposes plays a key role in establishing and obtaining of practical and useful results in both areas. Modern economics presumes fast time-to-market and constant changes and development of new technologies. There are numbers of international programs supporting cooperation between universities, and between university and IT industry companies. The most known and experienced are former Tempus, Erasmus+ programme. The aim of Erasmus+ is to contribute to the Europe 2020 strategy for growth, jobs, social equity and inclusion, as well as the aims of ET2020, the EU's strategic framework for education and training in modern technologies [1].

The Internet of Things is an emergency topic of technical, social and economic significance. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way people work and live.

Nowadays we are told that the machine-to-machine M2M connections will represent 46% of connected devices by 2020 as well as 73% of companies use IoT project data to improve their business and 95% of execs surveyed plan to launch an IoT business within 3 years [2,3]. A number of worldwide companies and research organizations have offered a range of projections about the potential impact of IoT on the Internet and the economy during the next ten years: Cisco predicts more than 24 billion Internet–connected objects by 2019 [4], Huawei forecasts 100 billion IoT connections by 2025 [5]. McKinsey Global Institute suggests that the financial impact of IoT on the global economy may be as much as $3.9 to $11.1 trillion by 2025 [6].

Main challenges and questions in this technology are related to security and privacy issues, device interoperability, regulatory and rights domain, emerging economy and development issues. This section is based on material of the [7].

0.1.2 State of the Art and publication statistics

In this section we present a brief history of the origin and development of the Internet of things. Wikipedia says that the first publication, in which the idea of the Internet of things is presented, was published in 2000 [8]. This is an article by a large group of authors (13 authors): “People, places, things: Web presence for the real world” [9]. It is based on their report at the 2000 Third IEEE Workshop on Mobile Computing Systems and Applications. Authors wrote that “…the convergence of Web technology, wireless networks, and portable client devices provides new design opportunities for computer/communications systems. … we have been exploring these opportunities through an infrastructure to support Web presence for people, places and things. Using URLs for addressing, physical URL beaconing and sensing of URLs for discovery, and localized Web servers for directories, we can create a location-aware but ubiquitous system to support nomadic users. On top of this infrastructure we can leverage Internet connectivity to support communications services. Web presence bridges the World Wide Web and the physical world we inhabit, providing a model for supporting nomadic users without a central control point”. Despite the fact that the words "Internet of things" do not appear in this annotation, one can see that the basic concept of the future IoT here is already clearly indicated.

It should be noted that the article "That 'Internet of Things' Thing" [12], which was published in January 2009 by Kevin Ashton, affirms another. The author shifts the date of the appearance of the term "Internet of things" one year ago. In particular he writes: “I could be wrong, but I'm fairly sure the phrase "Internet of Things" started life as the title of a presentation I made at Procter & Gamble (P&G) in 1999. Linking the new idea of RFID in P&G's supply chain to the then-red-hot topic of the Internet was more than just a good way to get executive attention. It summed up an important insight—one that

10 years later, after the Internet of Things has become the title of everything from an article in Scientific American to the name of a European Union conference, is still often misunderstood."

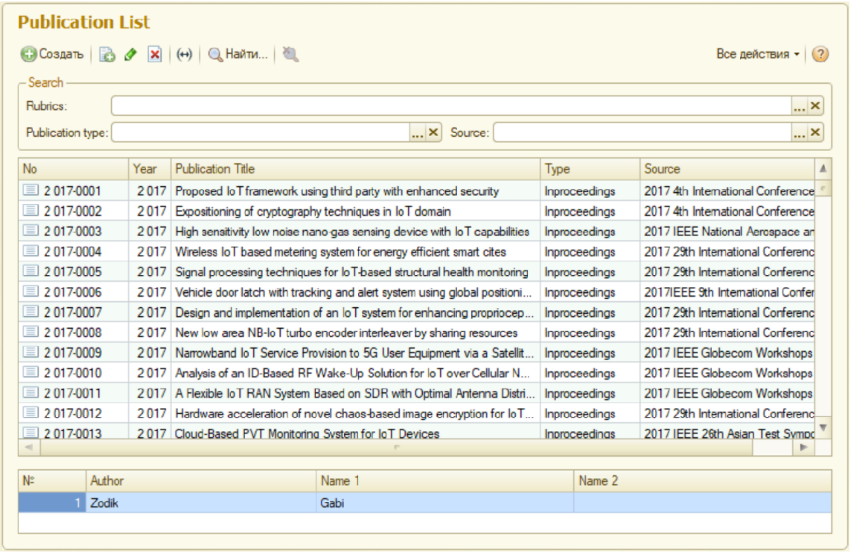
The presentation of which Kevin Ashton says is not published anywhere. The first publication, in the name of which the abbreviation "IoT" occurs, appeared in March 2008 [10]. It was presented in Zurich, at one of the workshops of the conference "Internet of Things 2008 International Conference for Industry and Academia". The workshop was named "Workshop on Designing the Internet of Things for Workplace Realities". It is noteworthy that initially at the same conference a workshop "Sketchtools - Creative Tools for Prototyping Smart Devices" was announced, which did not take place due to a lack of interest of the participants in this problem [11].

Current state of the art and dynamics of IoT development can be analyzed on the basis of processing of statistical data about publications. To process such data the software tool "Accounting Publication Manager" (APM) has been developed using 1C:Enterprise 8.2 system.

The capabilities of the 1C: Enterprise system allow us to accumulate and organize this information according to the chosen criteria. APM collects and present information about the name, publication year, authors and type of publication such as "Article", "Book", "Proceeding", "Techreport" etc. APM allows you to upload a list of publications in the BibTeX format [13] similar to the IEEE Explore Digital Library, The ACM Digital Library, ELSEVIER and others. The ability to import data from a BibTeX file allows you to reduce time to enter information in APM and reduce the number of errors made when entering.

The IEEE Explore libraries and the ACM Digital Library were used to compile a list of IoT references. According to the key words "IoT", 12,406 publications were found in the library of IEEE Explore, and 1,714 publications were found in the ACM Digital Library. Thus, only 14,120 publications are downloaded in the APM. Figure 0.1 shows part of the list of publications received.

A large number of publications allows us to judge the status and main trends of IoT development.



Rubrics:

Publication type:

Source:

No	Year	Publication Title	Type	Source
2 017-0001	2 017	Proposed IoT framework using third party with enhanced security	Inproceedings	2017 4th International Conference
2 017-0002	2 017	Expositioning of cryptography techniques in IoT domain	Inproceedings	2017 4th International Conference
2 017-0003	2 017	High sensitivity low noise nano-gas sensing device with IoT capabilities	Inproceedings	2017 IEEE National Aerospace ar
2 017-0004	2 017	Wireless IoT based metering system for energy efficient smart cites	Inproceedings	2017 29th International Conferenc
2 017-0005	2 017	Signal processing techniques for IoT-based structural health monitoring	Inproceedings	2017 29th International Conferenc
2 017-0006	2 017	Vehicle door latch with tracking and alert system using global positioni...	Inproceedings	2017IEEE 9th International Confer
2 017-0007	2 017	Design and implementation of an IoT system for enhancing propriocep...	Inproceedings	2017 29th International Conferenc
2 017-0008	2 017	New low area NB-IoT turbo encoder interleaver by sharing resources	Inproceedings	2017 29th International Conferenc
2 017-0009	2 017	Narrowband IoT Service Provision to 5G User Equipment via a Satellit...	Inproceedings	2017 IEEE Globecom Workshops
2 017-0010	2 017	Analysis of an ID-Based RF Wake-Up Solution for IoT over Cellular N...	Inproceedings	2017 IEEE Globecom Workshops
2 017-0011	2 017	A Flexible IoT RAN System Based on SDR with Optimal Antenna Distri...	Inproceedings	2017 IEEE Globecom Workshops
2 017-0012	2 017	Hardware acceleration of novel chaos-based image encryption for IoT...	Inproceedings	2017 29th International Conferenc
2 017-0013	2 017	Cloud-Based PVT Monitoring System for IoT Devices	Inproceedings	2017 IEEE 26th Asian Test Sympc

№	Author	Name 1	Name 2
1	Zodik	Gabi	

Figure 0.1. Publication list in Accounting Publication Manager

First of all, the dynamics of the growth in the number of publications by years is of interest. The data obtained from the APM program for changing the number of publications are presented in Table 0.1.

Table 0.1. Number of IoT-devoted publications by years

No	Year	Number of publications
1	2008	5
2	2009	7
3	2010	127
4	2011	249
5	2012	396
6	2013	508
7	2014	1135

8	2015	2152
9	2016	3885
10	2017	5520

As can be seen from this table, the explosive growth in the number of publications began in 2014 and since then, the number of publications has been increasing from year to year. This explosive growth is clearly visible in Figure 0.2.

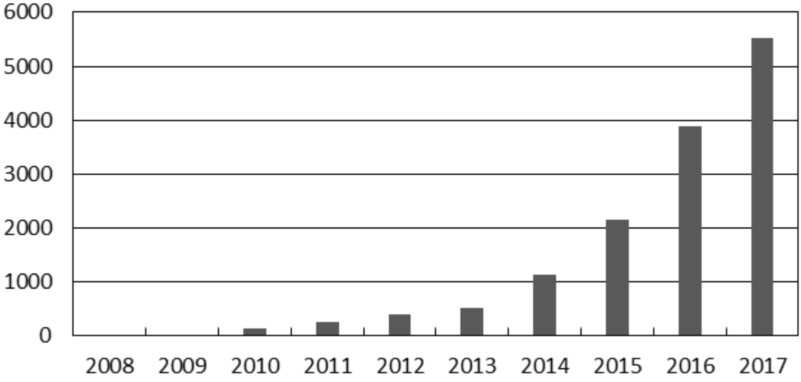


Figure 0.2. The explosive growth of IoT-devoted publications by years

The average growth rate is 1250 publications per year. If this trend continues, then about 6,800 publications should be expected in 2018, and in 2020 the number of publications should exceed 10,000. Whether this is true or not, time will tell. However, we can conclude that IOT is becoming more and more popular every year and can completely change our life. But can we be sure that this will be a change for the better?

A more complete picture of IoT development trends can be obtained by analyzing how the number of types of publications has changed over the years (Table 0.2).

Table 0.2. Types number of IoT-devoted publications by years

Year	Article	Book	In book	In proceedings	Techreport
2 008	0	0	0	5	0

2 009	0	0	0	7	0
2 010	7	0	0	120	0
2 011	11	0	0	238	0
2 012	8	0	0	388	0
2 013	51	0	6	451	0
2 014	120	0	3	1 012	0
2 015	236	1	23	1 892	0
2 016	447	1	19	3 417	1
2 017	1 167	3	77	4 273	0

Most of the work was published in the conferences’ proceedings. It is caused by fast feedback on presented results during conferences and workshops, possibilities of discussion in real time and face to face in comparing with journal paper.

With the help of the APM software, the distribution by years of the number of conferences in which IoT reports were presented was obtained. This distribution is shown in Table 0.3 and in Figure 0.3.

Table 0.3. Distribution of conferences with IoT presentations

Year	Conferences
2008	5
2009	7
2010	64
2011	124
2012	193
2013	229
2014	446
2015	697
2016	1 048
2017	982

Figure 0.3 shows that the number of IoT conferences held in the world shows the same explosive growth as the number of publications. It's caused by, firstly, an increase in the number of conferences

automatically leads to an increase in the number of publications, and secondly, an increase in the number of publications reflects the growing interest of researchers in the IoT problem. This, in turn, leads to an increase in the number of conferences. Hence there is a kind of positive feedback explaining nature dynamics on Fig. 0.3 and Fig.4.0.

However, we should pay attention to a small reduction (by 66) in the number of conferences in 2017. Indirectly this may indicate a kind of "saturation" of the conference market. It can lead to a decrease in the growth rate of the number of publications. However, the increasing number of publications from year to year can be predicted certaintly.

It is interesting to analyze the distribution by year of the number of authors of publications (Fig.0.4).

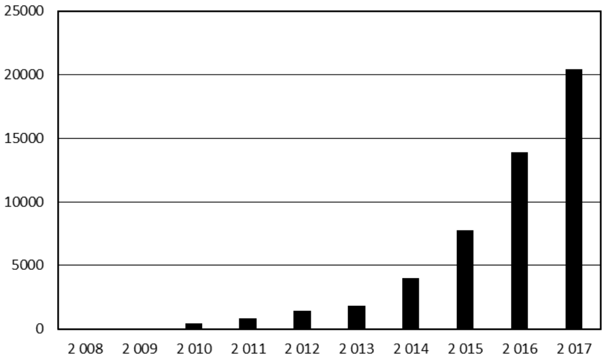


Figure 0.3. The diagram of the distribution of conferences by year

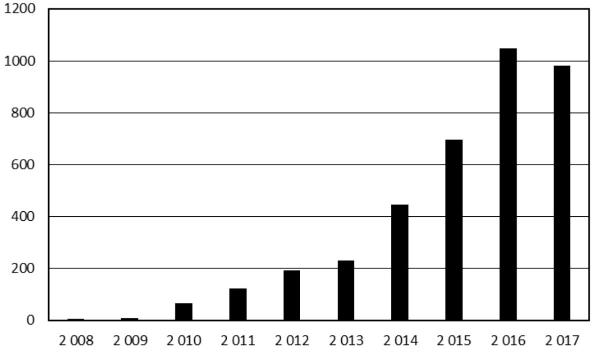


Figure 0.4. The diagram of the distribution of authors by year

In constructing this diagram, only "authors" were taken into account. That is, if the author published several works within a year, he was counted only once. This diagram clearly demonstrates the growing popularity of IoT among the scientific community. So, in 2008, 23 authors published works on IoT subjects. In nine years, in 2017, the number of authors increased to 20440 (more than 880 times).

Thus, summarizing the short historical analysis, we can draw the following conclusions:

– The history of the Internet of things does not last more than ten years, except for the first work of this year related to this topic. This is probably the youngest area of research.

– Internet of things demonstrates explosive growth in all indicators: the number of people involved in development and research, the number of scientific publications and the number of innovative developments.

– The results of work in the field of IoT will be global in nature. In this regard, the consequences of implementing IoT in various areas can surpass the consequences of the appearance of a personal computer.

– In accordance with the previous paragraph it is clear that these consequences will not only be positive. Already now, a positive answer to the question "But whether your phone is spying on you" does not surprise anyone. Who knows, maybe in a few years we will be interested, is our electric light spying on us?

0.1.3 Objectives and approach

The objectives of the next part of introduction are the following:

– to analyze challenges in IoT related education;

– to describe the Erasmus+ ALIOT project (project reference number 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP) which ensures adaptation of academic programs in Ukraine and other countries to the needs of the European labor market in context of IoT;

– to overview an experience of EU and USA universities and content of modules and courses for some IoT related domains such as smart building and smart cities, simulation of the IoT systems and their dependability and security assessment and assurance;

– to present some preliminary results of the ALIOT project and compare with international experience.

We suggest vertically integrated program of education on IoT technologies and application of IoT based systems. This program covers basing on BSc Computer Engineering (or other IT Engineering curriculum) three level of education and training:

- MSc program;
- PhD program;
- Engineering training for several key domains.

0.2 ALIOT project for vertically integrated education

0.2.1 Challenges in area of IoT education

The wise and intensive application of IoT on regional and country levels for key domains (health and ageing, energy grid and smart cities, transport and industry) will help to intensify the effort taken to overcome the pointed problems.

For their successful decision it is extremely important to combine related efforts in IoT education, research and engineering. Currently the subject area of IoT is almost absent from BSc and MSc programmes of Ukrainian universities, while the national and international IT-markets require specialists in this area. There is no single specialty on computing science or engineering that covers all aspects of IoT research, development and production implementation. The available learning laboratories, computer, network equipment and software are not completely suitable for the educational processes in the described specialties due to the absence of specialized software applications/instrumental tools/subscriptions on specialized databases for testing and modeling of smart IoT-based infrastructures, such as advanced traffic management systems, smart lighting, forest fire detection and smart-grid systems.

To sum up the existing challenges, it is important to provide the system analysis of the problems on the basis of universities' internal surveys:

– An innovative approach on IoT studies is not present within the MSc curriculum for the target and related specialties;

– There is a complete absence of doctoral courses in the area of IoT research and development;

– Urgent demand from most of the industrial actors for certified specialists in IoT area;

– Total absence of public awareness in the field of modern concepts and current approaches in IoT engineering.

International cooperation provides creation of a modernized EU innovative learning system for training and professional development in the emerging field of IoT, robotics, computer networks and microcontrollers including the development of smart devices for traffic system with adapted academic programs to the requirements of UA and EU employers. The training resources (MSc, PhD, industrial training modules) will be developed by leading experts in the EU and Ukraine. The EdX-based platform will have an internationalization dimension and will be made accessible to both ICT community in Ukraine and the EU. The multi-domain IoT cluster network will help strengthen relations between the academia and industries (e.g. Samsung, Microsoft, Cisco, etc.) through academic and research collaborations, knowledge exchange and transfer.

Through the national projects and programs the following results cannot be obtained and efficiently introduced:

– It is not possible to provide fast response to changing market conditions without the development of a cutting edge IoT curriculum which has extensive applications for the industries and societal benefits;

– Without mobility, it will be difficult for the transfer and exchange of knowledge, skills, competences, and best practices amongst experts;

– The European Union has implemented credit-transfer system of accumulation of academic credits ECTS (European Credit Transfer and Accumulation System) [14] long ago and it is tied to the EQF

(European Qualifications Framework) [15]. The EU has extensive experience in social adaptation of persons with disabilities and this experience is very important for Ukraine and other countries.

0.2.2 Innovative character of the ALIOT

The Joint Project on Curriculum Development implements a new approach to the delivery of educational services through ongoing feedback from employers and correction of the educational process, methodological and logistical support of the educational process. It also provides creation of professional community in IoT, robotics, computer networks and microcontrollers. The project ensures adaptation of academic programs to the needs of the European labor market, thus enhancing the opportunities of academic and labor abundant. The other novelty is usage of the concept of ECTS and the concept of learning throughout life (LLL). In Ukraine, these concepts do not work in fact.

Among main innovating elements the following positions are taking place:

– Training courses on development and implementation of techniques of IoT, robotics, networks and microcontrollers for the social adaptation of persons with disabilities (such programs in Ukraine are not available);

– Training courses on development and implementation of IoT, robotics, networks and microcontrollers for the smart and safe traffic systems (such programs are not available in Ukraine);

– New interdisciplinary, and transnational MSc programme on IoT (which is an emerging field) adapted to the modern Ukrainian and European labor market ensure the labor mobility;

– Providing the social adaptation of unemployed people and people with inadequate qualifications, by obtaining a new qualification on the principles of accumulation and ECTS, this is especially important for people with disabilities.

In order to achieve the indicated objectives the consortia members from 5 countries (UK, Ukraine, Sweden, Portugal, Italy) agreed to apply for a joint project in curriculum development named “Internet of Things: Emerging Curriculum for Industry and Human Applications” (ALIOT).

Detailed description of the project consortia could be found on the official website of the ALIOT in section “Project Consortium” [16]. The consortia members had a mutual collaboration last years through other educational projects funded under the Tempus programme – GREENCO [17], CABRIOLET [18], SEREIN [19].

The wider objective of Curriculum Development project ALIOT financed under Erasmus+ programme is to provide studies in the emerging field of IoT according to the needs of the modern society; to bring the universities closer to changes in global ICT labour market and world education sphere; to give students an idea of various job profiles in different IoT domains. ALIOT will strengthen the internationalization dimension of the postgraduate programme for higher education systems through the incorporation of Bologna objectives which ensure the transparency of the quality assurance systems, governance and management systems.

The specific objectives of Curriculum Development project ALIOT were established as:

– Introduction of a Multi-domain and Integrated IoT programme for master students in UA universities by September 2019;

– Introduction a Multi-Domain and Integrated IoT programme for doctoral students in UA universities by September 2019;

– Providing of the mechanism for intensive capacity building measures for UA CT tutors by September 2019;

– Establishing of the Multi-Domain IoT Cluster Network in Ukraine by September 2019. This network will provide an environment for knowledge sharing and transfer as well as cross-fertilization of innovative IoT-related research ideas and practices between the academic and industrial sectors.

The aim of Multi-Domain IoT Cluster Network is to integrate all available and produced curriculum, methods and tools for providing training and consultancy services in the area of IoT-based systems for different application domains: human, business-critical, safety-critical. The network will be a means for knowledge sharing, exchange, and transfer. It will also promote public awareness of the cutting edge IoT-related concepts, technologies, and applications.

0.2.3 Project activities and methodology

In order to reach the described goal the new MSc programme, PhD programme and in-service training programme on IoT will be developed and introduced, IoT cluster network offices will be established in 7 Ukrainian universities, intensive capacity building scheme for course developers, lectures and IoT Cluster offices will be launched. This will be achieved through the implementation of the following activities (named as key workpackages, or WPs):

– WP1: Development of master Curriculum on IOT – strategy for the UA needs analysis, development of master curriculum, lecture books and teaching plan, purchase and installation of needed hardware and software, scheme for the implementation of curriculum and delivery of guest lectures.

– WP2: Development of doctoral Curriculum on IOT – strategy for the development and introduction of doctoral modules which is similar to master one, described in WP1, with some specific measures applied to doctoral level curriculum.

– WP3: Capacity building measures – system for comprehensive training in the relevant theoretical and analytical skills needed to design and introduce the above approach for involved E&C engineering departments.

– WP4: Establishment of Multi-Domain IOT Cluster Network on the base of involved ICT departments of 7 Ukrainian universities. Each office will be specialized for the specific application domain and thus be responsible for networking and cooperating of R&D, academic and industrial partners acting in the respective domain.

0.2.4 Expected impact of the project

Who will benefit from the project:

– Enrollees and students who wishes to get an education in the field of IoT, computer networks, microcontrollers and robotics (training with using new adapted to the modern UA and EU labor market training programs) – approximately 120 customers per year at one university.

– Recent graduates, young professionals on manufacturing processes automation, computer networks, microcontrollers and robotics who wishes to adapt to the requirements of the employers (the possibility of studying individual disciplines) –approximately 80 customers per year.

– Regional and National industrial, profit and non-profit organizations and companies that are interested in improving the skills of their employees (training in the IoT centers, platform for on-line teaching and web-conferencing) – approximately 40 customers per year at one region where universities are located.

– Individual customers (development of individual training modules on request) – approximately 60 customers per year.

How these results will be useful for these target groups:

– *Local level* – training of specialists who meet the modern requirements in ICT industry, development of students start-up projects related to smart university infrastructure and E-identification;

– *Regional level* – improvement of national industrial and agricultural standards, quality and production, implementation to E-government program, medical and social services;

– *European level* – employment in the EU, development of joint educational and R&D projects with EU enterprises, establishment subsidiary of EU enterprises in Ukraine.

– Before the recruitment of the target groups, prospective students will be reached through: publicising ALIOT programme to all partners’ undergraduate and master’s degree students; via ALIOT website [10] and social media accounts (e.g. Facebook, Twitter, etc.); via Erasmus Mundus Plus interest groups.

– The target groups also will be reached with the help of electronic means: IoT web-platform and a project web-site presenting full information about the project will be developed. Tempus networks (e.g. IITN [20], IoT and GREENCO [17], etc.) and another Erasmus Mundus programmes (e.g. PERCCOM [21]).

After the end of the project target groups will be reached with internal and external dissemination events: academic and promotional seminars, on-line meetings and workshops of IoT Cluster Network, regular face-to-face meetings at departmental and faculty level to guarantee support for project activities.

Talented youth: Thanks to the nature of IoT and IoE (Internet of Everything) such kind of teaching directions is considered as highly attractive for young generation of university graduates and in some years – generation of secondary school graduates. This will be resulted in more deep penetration of IoT ideas and methodology to different groups of active youth of Ukraine and Europe.

ICT business sector: Feedback from businesses and engineering companies who are interested in improving the skills of their employees, with the help of KhAI-maintained CIDECS by Tempus ECOTESY web-platform [22] to adjust curricula of training specialists according to the new requirements; cooperation agreements with universities and centers of career and technology transfer.

Wide ICT community: Tempus networks, established in Ukraine and supported in last 5 years (International Technology Transfer Network by Tempus UNI4INNO [20] and CIDECS [22]) will be also used to reach Ukrainian ICT community with the help of graphical dissemination materials.

0.2.5 ALIOT curriculum

Structure of the curricula consists of fourteen courses (Figure 0.5) for three level of education: master courses, doctoral courses and industrial training.

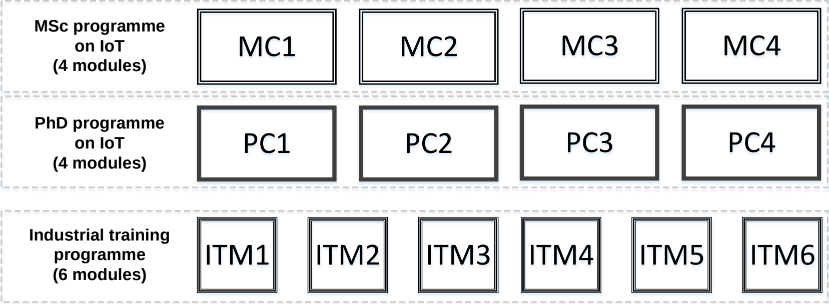

MSc programme on IoT (4 modules)
MC1
MC2
MC3
MC4
PhD programme on IoT (4 modules)
PC1
PC2
PC3
PC4
Industrial training programme (6 modules)
ITM1
ITM2
ITM3
ITM4
ITM5
ITM6

Figure 0.5. Structure of the curriculum

Target master courses to be developed (MC1..MC4): MC1 Fundamentals of IoT and IoE (Internet of Everything); (initial technologies, synergy of technologies, architectures, communications, standards); MC2 Data science for IoT and IoE; MC3 Mobile and hybrid IoT-based computing; MC4 IoT technologies for cyber physical systems.

Target doctoral courses to be developed (PC1..PC4): PC1 Simulation of IoT and IoE-based system; PC2 Software Defined Networks and IoT; PC3 Dependability and Security of IoT; PC4 Development and implementation of IoT-based systems (sensors, actuators, networking).

List of target modules for industrial trainings (ITM-1 .. ITM-6): ITM1: IoT for Smart energy grid; ITM2: IoT for Smart building and city; ITM3: IoT for UAV fleet; ITM4: IoT for automotive & intelligent transport systems; ITM5: IoT for ecomonitoring; ITM6: IoT for industrial systems.

The development of tailor-made curriculum is based on knowledge transfer from ALIOT European partners, thus enabling development of European up-to-date curriculum, in accordance with all current standards, including the Bologna process.

Master courses were developed on the input made by partners from Royal Institute of Technology KTH, Sweden, University of Coimbra, Portugal.

Doctoral curriculum reflects theoretical knowledge and cases given by Leeds Beckett University and University of Newcastle upon Tyne, UK.

Training modules for business sector and curriculum a whole take into consideration contribution from the Institute of Information Science and Technology, Italian National Research Council and all participants of consortium from EU, USA, Ukraine and other countries.

0.3 Overview of the IoT courses in Europe and the United States

Before the discussion about opportunities in field of IoT for students and the future experts in the west countries, we could not ignore Ukrainian experience, thus we attempted to search for any courses in the Universities of our country. However, the result was predictable and, unfortunately, there is an apparent defect what is related to courses in IoT.

0.3.1 Overview of IoT courses in ALIOT project partners

In this situation, we decided to consider an alternative European study experience in this subject. For sure, first of all the members of consortium ALIOT were examined, among which there were Newcastle University, Leeds Beckett University, Royal Institute of Technology and University of Coimbra.

As we talk about study programs, it is relatively complicated to intend some kinds of compulsory methods of examination in a wide range of characteristics. So the table format was selected for the systematization and generalization of the obtained information.

It is also necessary to explain the outline of our table, and to give a comment to a few rows in it. In the beginning, we name the course with the University and the level of competence for this course (Master, PhD etc.). After that, it is important to give some numbers about a amount of hours or credits (ECTS). The more hours require higher aims, which signals what and where we aspire for students to be by the end, what we want the students know.

As the followed part, we cite as an example two courses, among which are Internet of Things with Sensor Networks and Sensor Based Systems, where it worth to emphasize the importance of the selected courses. Since a various number of connected devices is already added to the Internet, a multitude of sensors and mobile users' terminals are designed to interact in order to offer novel services in smart cities and territories in general. These devices, in the so-called Internet of Things (IoT), have very specific characteristics both in terms of hardware (a very little amount of memory and computational power), software and management (few system updates). Being able to understand and to simulate the IoT became essential. The only source of data, we collected, was performed by information on the web pages of the universities. However, the total amount of revealed course information fluctuates from announcements about opening to full bunch of disciplines and required references. One of the most informative site outline revealed in Newcastle University, where visitors could easily reach the learning outcomes, graduate skills framework, teaching activities, reading lists, assessment methods and timetable.

Below we provide the results for both courses in tables 0.4, 0.5.

Table 0.4. Course characteristic for Master Study (Wireless Embedded Systems)

Name	Hours/Credits	Course Objective	Competence of the graduates
EEE8092 : Internet of Things and Sensor Networks (NC)	100/5.0	Practical experience of wireless networking for computers, embedded devices or sensors, building upon the complementary taught module “Wireless Networks”.	Wireless network protocols. Technologies for the implementation of wireless networks and sensor networks. Sensor systems and circuit design.

Table 0.5. Course characteristic for Master Study (Embedded Systems)

Name	Hours/Credits	Course Objective	Competence of the graduates
II2302 Sensor Based Systems (KTH)	150/7.5	An introduction to sensor enabled systems, with an emphasis on embedded platforms; broad sensor technologies, the physical properties of measurement, the usage in embedded designs.	Design a network topology for communicating sensor nodes that satisfies stated requirements of robustness, security, performance and cost; the usage of sensor based architectures to design advanced applications that use context awareness, personalization, augmented and virtual spaces.

0.3.2 Metrics-based approach of IoT courses analysis

Unfortunately, the table-organized information is not pictured a broad look at the research we occupied with. Thus, for the main objective as ubiquitous analysis of courses, it was convenient to use numerical indicators and metrics. Therefore we developed metric system based on works and articles on this subject. Our set of metrics represented the following:

1. Course duration. Collecting the actual amount of credits from each course, we took the highest number, and divided all other credits on it, so that there was a range 0-1, after that we converted these credits according to the schema below: 5 points – 0.8…1 ; 4 points – 0.6…0.8 ; 3 points – 0.4…0.6 ; 2 points – 0.2…0.4 ; 1 point – 0...0.2

2. Form of education: 5 points – full-time courses; 4 points – remote; 3 points - extramural studies.

If it is mixed, then the corresponding points summarized.

3. Basics of a course (coverage width). It is estimated at quantity of other courses and disciplines which are based on this course. When we found the highest number of connections, we assessed the basics the same way as in the first clause. (Division and range conversion)

4. Completeness of a course on the website:

- 5 points - on the website are completely available the program, the abstract of lectures, control questions and instructions for independent work;
- 4 points - are the program, the abstract of lectures, control questions;
- 3 points - the program, the abstract of lectures;
- 2 points - only the program;
- 1 point - only the announcement of a course.

5. Availability of a course: 5 points - all materials in free access, record on a course are not required; 4 points - all materials in free access (free of charge) after record on a course; 3 points - materials paid; 2 points - materials given by the professor personally.

Tables 0.6, 0.7 provide the summary of metrics for two most widely known courses - Simulation of IoT and IoE-based systems and IoT for Smart building and city.

Thus, summing up the results of the metric analysis it is possible to draw the following conclusions.

Table 0.6. Metric for Simulation of IoT and IoE-based systems course

University code/ Course Name	**Credits**	**Form**	**Basics**	**Compl eteness**	**Avail-ability**	**Sum**
Internet of Things and Sensor Networks (NC)	2	5	5	3	3	18
Sensor Based Systems (KTH)	2	5	3	2	3	15
M2M Technology Internet of Things (NC)	2	5	1	3	3	14
Human-Computer Interaction (UC)	1	5	1	2	3	12
Intelligent Sensors (UC)	2	5	3	2	3	15
Smart Grid Communications (UC)	2	5	1	2	3	13
Simulation and Modelling (LBU)	5	5	1	1	2	14
Internet of Things (UPU)	2	5	2	3	3	15
Sensor Data Fusion (PU)	3	5	3	2	3	16
Wireless, Sensor and Actuator Networks (RLU)	5	5	2	2	2	16
Smart Cards, RFIDs and Embedded Systems Security (RLU)	5	5	1	2	2	15
Interconnected devices (RLU)	3	5	1	2	2	13
Wireless, Sensor and Actuator Networks (BU)	5	5	2	2	2	16

Table 0.7. Metric for IoT for Smart building and city course

University code/ Course Name	**Credits**	**Form**	**Basics**	**Comple-teness**	**Avail-ability**	**Sum**
Power Distribution Engineering (NC)	3	5	1	3	3	15
Planning and forecasting in energy sector; Energy for Smart Cities (KTH)	1	5	1	2	2	11
Energy Simulation of Buildings (UC)	2	5	1	2	3	13
Energy Planning and Sustainable Development (UC)	2	5	1	2	3	13
Building Energy Management Systems and Intelligent Buildings (LBU)	5	5	1	1	2	14

For the IoT programs it is possible to find around 100 opportunities related to this subject area, moreover in the various faculties among which are Science and Technology, Computer Science, Wireless Embedded Systems, Advanced Computing, Engineering and Built Environment.

However, at the European universities there is no abundance and a variety of direct courses on IoT for today. The education system, as the most inertial part of a sheaf the science-university-industry, is late here.

The direct program with full-course performed only in 15 Universities in Europe. The lead country with the biggest number of opportunities for those, who interests in being graduate in IoT, is the UK.

The most common modules for IoT courses are Wireless, Sensor and Actuator Networks, and Embedded-Systems Security, Engineering, Intelligent Systems, Robotics, Introduction to smart grids, Human-Computer Interaction, Systems Engineering, Data Processing, Signals and Systems, C Programming.

0.4 Case studies on ALIOT based project education

0.4.1 Control unit for mini plotter

The technology of building IoT elements is based on the use of typical circuit-based solutions (platforms) based on microcontrollers and software-implemented functions - sensor interrogation, control of executive devices, information display, etc.

The control of actuators is usually realized by forming a sequence of combinations of control signals and is very often used in various projects. For project-based learning, a mini plotter is used with two-coordinate control of drawing simple closed contours (Figure 0.6a, b).

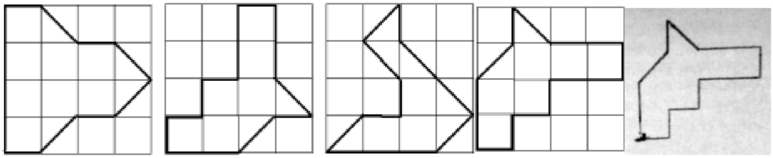
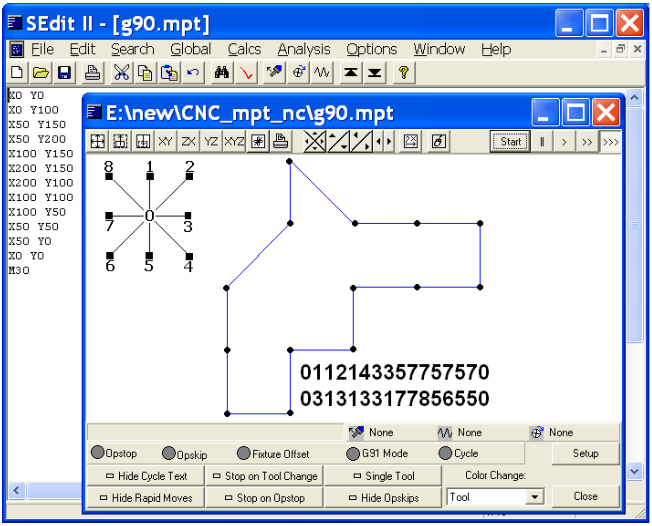


Figure 0.6. Examples of options for setting the plotter control (a, b) and the final result (c)

A universal way of describing the movement of the plotter's writing node is to build a G-file (Figure 0.7). The construction of the G-file interpreter based on the microcontroller is a rather difficult task for the subsequent stages of the project development.



```
X0 Y0
X0 Y100
X50 Y150
X50 Y200
X100 Y150
X200 Y150
X200 Y100
X100 Y100
X100 Y50
X50 Y50
X50 Y0
X0 Y0
M30
```

```
0112143357757570
0313133177856550
```

Figure 0.7. G-file for the variant of the task (Figure 0.6b)

At the initial stage a number of simplifications are introduced:

For the contour in Figure 0.7 when moving clockwise, we get the line "0112143357757570", and when moving counter-clockwise - the line "0313133177856550". These lines can be used to control the progress of the project.

The structure of the mini plotter (Figure 0.8) includes the target microcontroller MCU1, for which the program for generating signals (x +, x-, y +, y-) is being developed.

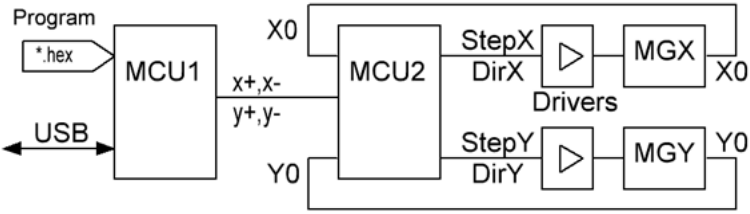


Figure 0.8. Block diagram of a mini plotter

Direct control of stepper motors MGX, MGY with sensors of initial position X0 and Y0 is carried out by microcontroller MCU2 by means of drivers. The MCU2 is assigned the function of returning the writing node to the initial state (X0, Y0).

Figure 0.9 shows the mini-plotter view, for which various MCU1 target microcontrollers and software development tools can be used. The project implementation includes the following stages:

a. Analysis of an individual variant of the task and its formalization is the representation in the form of a row of vectors, tables and time diagrams of output states;

a. Selecting the target microcontroller and control lines (x +, x-, y +, y-);

b. Construction of an algorithm for the functioning of MCU1;

c. Selection of development tools for MCU1;

d. Debugging the program using simulators (Proteus, etc.);

e. Loading the program into MCU1;

f. Set the initial state of the mini plotter, connect MCU1 and start the program;

g. The representation of the final result is the image of the specified contour.

An intermediate monitoring of project implementation is carried out (stages A and E).

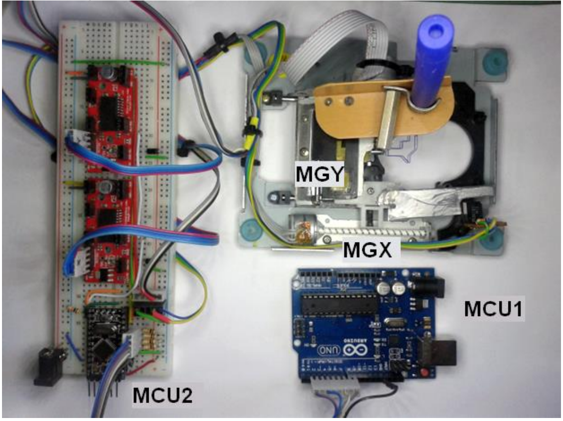


Figure 0.9.Type of mini-plotter (Homemade DIY CNC)

Further development of the project can include the use of more complex two- or three-axis drives, the implementation of a prototype drawing service (cutting, engraving, burning) with the transmission of vector lines or G-files via the Internet.

0.4.2 Control Unit for the LED ribbon with pixel addressing

LED strip with pixel addressing based on WS2812b is used in "smart" lighting, lighting, advertising, etc. Single-wire control of series-connected pixels is used. Each pixel receives 3 bytes - RGB. By setting the levels of 0..255 colour components of the RGB of each pixel, you can implement various static and dynamic colour effects. Matrix colour panels can be constructed from the strip segments.

The microcontroller module (Figure 0.10a), which can work independently, or accept and execute commands via standard wired (USB, RS485, Ethernet) or wireless (Bluetooth, WiFi) interfaces, manages the LED tape. Various proprietary interfaces can be used.

For the proposed projects, the asynchronous serial interface of the target microcontroller is used, which, with the help of converters, is converted to USB, Bluetooth. The microcontroller is able to receive commands:

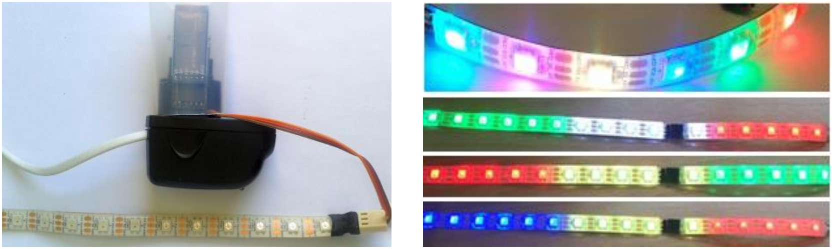
- Select the length of the tape (N <1..30> n);
- Brightness setpoints of red (R <0..255> r), green (G <0..255> r), or blue (B <0..255> r) colour components (RGB).

The length N can refer to the total number of pixels in the tape, or to determine the length of the initial portion of the tape. To this range of pixels will have the action of commands to change the brightness of the RGB.

Three-color coloring (green-white-red) of a 16-pixel LED strip (Figure 0.10b) is implemented using a sequence of command lines:

```
N16nR0rG255gB0b
N10nR255rG255gB255b
N5nR255rG0gB0b
```

The prepared command standards for job variants allow for a formal check of the progress of the task.



a) b)

Figure 0.10. LED ribbon control module (a) and colouring options (b)

The commands are transmitted to the module using a terminal program via USB, or an application - a Bluetooth terminal for a mobile device. To develop variants of the task of varying complexity, one can change the length of the tape, the colour scheme (splitting into groups of pixels and group colours).

Further directions of tasks may include addition of a set of instructions executed by the microcontroller and the implementation of dynamic colour schemes.

Conclusion and questions

Market research and analysis has shown that there are bottlenecks in the education system and industry in Ukraine which brings this country to the lack of specialists in IoT area, and it could be covered via introduction of the new curriculum for students and developed staff in the field of constantly accelerating IoT technology. The description of an Erasmus+ funding programme and lecture courses, which supports the ALIOT project, will be given in the next sections and parts of the three volume book.

The teaching courses for MSc, PhD students as well as capacity building in the field of training modules that have been developed in frame of Erasmus+ project ALIOT form an integrated vertical structure of education, training and research space in different areas of IoT-based systems industry and human applications.

Successful project implementation will ensure sustainable and comprehensive staff provision in IoT education and engineering for EU and Ukrainian enterprises and institutions. The project team expresses confidence that the obtained results will be useful both for Ukrainian universities and other countries' higher educational establishments acting in the field of training specialists in the area IoT-technologies.

In order to better understand material that is presented in this section, we invite you to answer the following questions.

1. What are the main challenges in IoT and IoE sectors?
2. What are the meanings of IoT and IoE in current academic and research environment?
3. What tools can be used for analyzing dynamics and IoT developments?
4. What are the reasons of exponential growth of IoT-related publications in 2014?
5. What are the key trends in IoT-related publications?
6. What are the current challenges in IoT education?
7. What are the innovative features of ALIOT project?
8. What is the main elements of ALIOT project methodology?
9. For whom ALIOT project developments are intended?
10. What is the structure of ALIOT curriculum?
11. What are the described case studies for IoT-related curriculum within ALIOT?

References

[1] Erasmus+ programme http://ec.europa.eu/programmes/erasmus-plus/node_en

[2] Annual Security Report, Cisco http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf

[3] Internet of Things (IoT). The IoT links objects to the Internet, enabling data and insights never available before. http://www.cisco.com/c/en/us/solutions/

[4] Cloud and Mobile Network Traffic Forecast - Visual Networking Index (VNI). Cisco, 2015. http://cisco.com/c/en/us/solutions/serviceprovider/visual-networking-index-vni/index.html

[5] Harnessing the Power of Connectivity. Huawei, 2017 http://www.huawei.com/minisite/gci/files/gci_2017_whitepaper_en.pdf?v=20170

[6] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, D. Aharon, “Unlocking the potential of the Internet of Things” http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/

[7] A. Boyarchuk, O. Illiashenko, V. Kharchenko, D. Maevsky, C. Phillips, A. Plakhteev, L. Vystorobska. Internet of Things for Industry and Human Applications: ALIOT Based Vertically Integrated Education. In Dependable Internet of Things for Human and Industry: Modeling, Architecting, Implementation / V. Kharchenko, Ah Lian Kor, A. Rucinski (editors). River Publishers, Demark-Netherland, 2018, p. 535-560.

[8] Web of Things. Available at: https://en.wikipedia.org/wiki/Web_of_Things

[9] T. Kindberg, J.Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, M. Spasojevic. ‘People, places, things: Web presence for the real world’, 2000. doi:10.1109/MCSA.2000.895378

[10] G. Ramirez, M. Munoz, C. Delgado. ‘IoT early possibilities in learning scenarios’, Workshop on Designing the Internet of Things for Workplace Realities: Social and Cultural Aspects in Design and Organization (Social-IoT). Zurich, Switzerland. March 26. 2008.

[11] Internet of Things 2008. Workshops. Available at: http://www.iot-conference.org/iot2008/cfp/workshops.html [accessed March 2008].

[12] K. Ashton. ‘That 'Internet of Things' Thing’, Available at: http://www.rfidjournal.com/articles/view?4986 [accessed January 2009].

[13] O. Patashnik. Designing BIBTEX Styles Available at: https://pctex.com/files/downloads/manuals/btxhak.pdf [accessed February 2018].

[14] European Credit Transfer and Accumulation System (ECTS) http://ec.europa.eu/education/resources/european-credit-transfer-accumulation

[15] Descriptors defining levels in the European Qualifications Framework (EQF) http://ec.europa.eu/ploteus/content/descriptors-page

[16] Erasmus+ ALIOT project “Internet of Things: Emerging Curriculum for Industry and Human Applications” website http://aliot.eu.org/

[17] Tempus GREENCO project “Green computing and communications” website http://my-greenco.eu/

[18] Tempus CABRIOLET project “Model-oriented approach and intelligent knowledge-based system for evolvable academia-industry cooperation in electronic and computer engineering” website http://my-cabriolet.eu/

[19] Tempus SEREIN project “Modernization of postgraduate studies on security and resilience for human and industry related domains” website http://serein.eu.org/

[20] Tempus project “Innovation Offices in Ukrainian Higher Education Institutions” http://www.uni4inno.eu/

[21] Erasmus Mundus project “Pervasive computing & communications for sustainable development” PERCCOMM project website http://perccom.univ-lorraine.fr/

[22] Centers of Innovations in Ecosystems Technosphere http://cidecs.net/about/

PART 1. FUNDAMENTALS OF INTERNET OF THINGS AND INTERNET OF EVERYTHING

DrS, Prof. V. S. Kharchenko (KhAI), Dr. O. O. Illiashenko (KhAI),
Dr. Prof Andrzej Rucinski (UNH), Dr. Ah-Lian Kor (LBU)

Contents

Abbreviations

AI — Artificial Intelligence

BD — Big Data

BDA — Big Data Bases Analytics

CAGR — Compound Annual Growth Rate

CEO — The Chief Executive Officer

ICT — Information and Communications Technology

IEEE — The Institute of Electrical and Electronics Engineers

IoD — Internet of Drones

IoT — Internet of Things

IT — Information Technologies

MIT — Massachusetts Institute of Technology

NPP — Nuclear Power Plant

RBD — Reliability Block Diagrams

SD — Software Defined

SDD — Software Defined Data Bases

SDE — Software Defined Every Thing

SDN — Software Defined Networks

SLS — Space Launch System

SW — Software

TQM — Total Quality Management

VLSI — Very-large-scale integration

WoT — Web of Things

1. CONCEPTS AND CHALLENGES OF INTERNET OF THINGS IMPLEMENTATION

1.1 Internet of Important Things

1.1.1 Concept of IoT

There are numerous publications which introduce and discuss the Internet of Things (IoT). IoT is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. In the midst of these, this work has several unique characteristics which should change the reader's perspective, and in particular, provide a more profound understanding of the impact of the IoT on society. These salient points may be summarized as follows:

- IoT is characterized as a disruptive innovation and/or technology as defined by Christensen and others https://www.postscapes.com/internet-of-things-definition/;
- IoT is one of the drivers of the high-tech market with an emphasis the significance of the IoT market prediction;
- IoT has a global range, the bulk of chapters in this publication have been derived from the different conferences such as DESSERT, DepCoS, IDAACS and others, workshops such as CyberIoT, TheRMIT, CrISS which has become an East-West Catalyst for IoT based innovation;
- IoT is one of the disruptive technology milestones in a technology development roadmap: beginning with the first transistor, on through era of VLSI and vital electronics, 5G and Grand Challenges.

The history of information and communication technology "ICT" starting with the development of the first transistor is sufficiently rich to formulate general laws of technology development. The well-known and popular Moore's Law is based on empirical observation over many technological generations. However, it is too simplistic to become the scientific base of logology in the ICT sphere.

In contrast, the editors of this publication recognize the existence of parallel dual worlds: one is biology based, and the other one is technology based. In general, there is an urgent need for the study of the interaction between the two. The urgency is driven in part by the observation that the technology-based, virtual sphere of: smart phones, the Internet, and software Applications "Apps" is no longer controllable. On a daily basis, several disparate digital media allow several billion individuals to interact in various unknown, and unknowable ways. Marshall McLuhan's Global village really exists.

This realization generates fundamental challenges facing humanity. A classic example is restricting the access of a teenager to the Internet, and a smart phone. This implies that we not only have to establish the existence rules between the real, and virtual worlds; but that we also need to re-establish the supremacy of humans over the emerging "Cyber-World" of robots, and personal assistants. The vision outlined in the 40's of the 20th Century by Norbert Wiener has materialized.

Moore's Law is a special case of a more generalized, but still empirical observation, i.e. a "Generalized More's Law", which notes that the development of ICT is ruled by Total Quality Management (TQM) style cycles. Christensen at Harvard Business School, identified how a cycle of innovation originates with a Disruptive Innovation, initially only accessible to scientific, governmental, or business elites. This is followed by the contractual phase where the disruptive technology becomes available to society generally.

The overall development process is governed by the growing complexity of microelectronic systems, and continuous integration of the technology world on all scales. The latter represents a heterogeneous ecosystem of ICT entities with virtual components targeting specific humans as users. Virtual components are data bases, Social Media such as Facebook, Search Tools such as Google, community knowledge bases such as Wikipedia, personal assistants, and so on.

The integration process was initiated by the invention of a single transistor, which was followed by the first integrated device, the first embedded system, the first integrated system, the first network, the first constellation of networks and so on. There is no upper boundary in the open-ended development process at this point. Each disruptive innovation has been associated with an application such as VLSI, the personal computer, WWW, the IoT, and so on. This process has made microelectronics globally available, with fully fledged computers as small as a grain of salt and so inexpensive as to be truly disposable. As a result of this development, an exponential function which models Moore’s Law in Cartesian coordinates can be replaced by an evolving spiral representation in polar coordinates.

The anticipated development of the 5G generation wireless technology, the next phase of IoT, can be viewed as an evolutionary stage of ICT technology following the Generalized Moore’s Law. However, the fundamental difference is that 5G is “a priori technology”, and needs to be designed before it can disrupt. The design process will be multifaceted and will affect both the real and virtual worlds introduced above, in ways both profound and unpredictable.

Children of the “α – generation cyber society” who will live their entire adult life in this new era have been already been born. A member of this new global society is going to experience multiple disruptive innovation revolutions during his or her life time with profound impact on redefining the professional personal and social aspects of his and/or her life.

1.1.2 Challenges and solutions of IoT

Global Internet of Things (IoT) market reached USD 598.2 Billion in 2015 and the market is expected to reach USD 724.2 Billion at a CAGR of 13.2% by 2023. One of the pioneering predecessors of the IoT revolution has been the concept of Vital Electronics introduced by Dr. Ted Kochanski [1-3]. His approach illustrates extremely well the ubiquitous character of IoT.

Vital Electronics is the study and use of electrical components, circuits, networks, and systems to achieve a design goal of protecting, saving, and improving critical infrastructure, and hence the quality of life. Vital Electronics’ domain is a heterogeneous computing environment derived from sensors networks, embedded systems, and ambient intelligence with intelligent, robust, and trustworthy nodes capable of building Application-Centric Embedded Computers from “off-the-shelf” virtual computational and networking parts. Vital Electronics makes Embedded Computers more capable, reliable, energy-efficient, and optimized to their tasks. These Embedded Computer inhabit our critical *infrastructure and other key applications, at increasingly low-levels, and with increasing interconnectedness with their peers. At the same time, Vital Electronics enhance the ease, and speed of the design of reliable Embedded Computers, and their associated Embedded Systems through the reuse of proven and certified "design elements," and other "virtual components." Vital Electronics is founded on the synergistic interaction between Moore's Law, Metcalf's Law, High-Level System Design Tools and MEMS Sensors and Actuators. The increasingly capable Programmable Systems on a Chip (PSoC) such as Cypress Semiconductor's PSOC family with its companion PSoC Creator tools are the key building blocks of Vital Electronics.*

However, while the original Vital Electronics was an academic international conference topic for a few years – It never had the critical mass to make a major impact on society. Thusly – Vital Electronics has been revived in the context of the on-rushing IoT tide to perhaps shape the impact.

*New Vital Electronics should be synonymous with t**he Internet of Important Things** [IOIT] applied to realms such as: Health, Housing, Transportation Utility Infrastructure Etc., previously only peripherally and superficially affected by electronics.The "forever" problem" has been that in general, the "electro-technical community" with the exception of people and organizations devoted to a particular market, or specialized field of endeavor didn't have the "subject matter expertise" to know where to contribute to solving "important problems," of a local or global extent. Meanwhile, the "Subject Matter Experts" who knew what was needed didn't have the knowledge of the specialized electro-technologies which could provide the core of a solution to a challenging problem. Attempts to bridge the gap have typically failed due to lack of a common terminology, vernacular or even a common context with which to discuss the issues. Basic premise of the new Vital Electronics is to build a "Technical Ethos" to support the application of modern electro-technology to important problems at the "grass roots level" – i.e. to provide the tools [hard and soft tools] to enable the people at the subject matter expertise and "problem facing" level to define specific tasks which would "make life: Safer, More Secure, Healthier, More Efficient.*

The New Vital Electronics is premised on the ability to take maximum advantage of: tremendous recent advances in core electronics technologies *driven by consumer and other high-volume products, such as: smart cell phones, tablets, watches, cars, robots, etc. Explosion of "Open Source" hardware and software driven by the renaissance in "hobbyist" hardware such as Arduino and Raspberry Pi Revolutionary enhancements in remote learning fostered and disseminated by initiatives such as MIT Open Courseware, IEEE on-line courses, etc.*

Democratization and global spread of supporting technologies such as: modeling and simulation, computer based design of a myriad of things, sophisticated visualization and Augmented Reality, and the ability to "print stuff" often again spread by hobbyist / amateur interests such as video gaming.

Democratization of technical knowledge at the "grass roots" level enhanced by Wikipedia and similar on-line sources of relatively reliable knowledge. Global spread of technical infrastructure such as cell phone networks located in the middle of African hinterlands. Global spread of package delivery on prompt basis driven by Amazon and similar on-line suppliers, distributors and expediters.

Key improvements in electronics technologies: Absolute Performance, Cost Performance, Reliability, Size, Weight, and Efficiency [i.e. the generalized Moore's Law] for: Processing, Signals, Databases, AI, Sensing, Communications (Wireless – e.g. 5G, Free-Space and Guided Optical), Information Display, Augmented Reality, Compact Multivariable Multimedia, Power Supply Technologies (Rechargeable and Disposable Batteries, Wireless, Energy Harvesting, Control of Physical Objects (MEMS, Biomorphic and Biofunctional manipulators and actuators).

Incredible improvement in availability of high performance and high function hardware and software (i.e. supercomputers and super bandwidth communications) which had traditionally been restricted to major corporate, big universities and Federal-level governmental entities.

Hobbyists and startups are today building mechatronic systems based on the above concepts with an investment comparable to buying an SUV which in all ways, outperform systems funded by leading nation states a decade ago at the level of Billions. On a larger scale, SpaceX has delivered a functioning "Falcon Heavy," a heavy-lift booster, for a fraction of NASA's budget allocation for similar performance [SLS Block-1, smaller version of Space Launch System], and at a pace inconceivable by NASA. Falcon Heavy also lands vertically and is reusable.

The concept of Grand Challenges was introduced to counterpart the Japanese program of the 5th generation of computing and since then the definition has gained popularity and recognition in many branches of human activities. Inspection of different grand challenges indicates the absence of a common consensus and standard taxonomies. However, many grand challenge solutions are enabled by and include in designing for the Internet of Things (IoT).

Thus, the IoT becomes a grand challenge fabric from hard computer engineering point of view. The IoT impact is so profound and hard to estimate today that this new computer technology may be categorized as "disruptive innovation". Yet another observation can be made related to the lack of commonly recognized and accepted collaboration schemata. One new but pragmatic collaboration approach is based on the theory of service science [5]. Service science assumes, among other things, so called value co-creation, a truly disruptive and somewhat utopian vision of collaboration.

Based on the above observation, it is proposed to consider the following hypothesis: The collaboratory concept, which is relying upon service science principles, becomes feasible because of the Internet of Things. Thus, service science and the 5G/IoT is serving as an enabler to address Grand Challenges. In other words, the Cartesian product of the Internet of Things and service science is the key concept presented in this paper.

Grand Challenge ideas are formalized our current into several service systems related e-categories: Health as a Service (HaaS), Learning as a Service (LaaS), Business as a Service (BaaS) and Government as a Service (GaaS). Each has a role in Smart City ecosystem evolution/revolution leveraging ICT/5G/IoT. Our concept is to focus on the individual and his/her security in each of these areas such as patient/caregiver for HaaS, student/parent for LaaS, employee/customer for BaaS and citizen/taxpayer/visitor for GaaS. By leveraging advances in ICT 5G and beyond and IoT, it is important pursuing the development of a Common Open Standards-Based International Innovation Digital Infrastructure for collaborative research and development. Besides, it is also important developing a disruptive innovative approach on how individuals of all ages from around the world (patient/caregiver, student/parent, employee/customer and citizen/taxpayer/visitor) can safely and easily utilize and benefit from the Digital Ecosystem World IoT of the future to benefit and advanced themselves.

Dr. Sumit Chowdhury is a key ICT driver behind the India 100 Smart City Program and founder and CEO of Gaia Smart Cities. The Gaia Smart Cities effort integrates IoT, industrial automation and digitalization solutions for enterprises and smart cities. Gaia's suite of ICT applications bring together sensors, hardware, software and analytics on a cloud-based platform and allow businesses and cities to automate processes, track metrics and improve performance. The India 100 Smart City program is an innovative initiative by the Government of India. The India Smart Cities Mission is to improve the quality of life of people by harnessing technology as a means to create a smart ecosystem for their citizens. The 100 Smart City objective is to promote diverse cities around the country (large, small, villages, coast line to interior) and for them to provide a core infrastructure and give a decent quality of life to its citizens, a clean and sustainable environment with the application of 'Smart' Solutions. A strong cooperative development is required between government, civic groups, industry and the citizens.

The India 100 Smart City program has defined eight critical pillars – Smart Governance, Smart Energy, Smart Environment, Smart Transportation, Smart IT and Communications, Smart Buildings, Smart Health Facilities and Smart Education. Pan City is an example of one city with development using three overriding frameworks: Smart Integration, Shared Data, Shared Networks. These have three inter-related components: Integrated City Management, Security Systems Management, Energy and Environment Management.

Pan City is currently focusing these in five major areas with examples of some sub areas being considered:

1. Smart Energy – Smart Grid, Metering, Renewables, Flexible Response
2. Smart Water – Distribution and Storm Management, Maintenance, Health
3. Smart Mobility – Public Transit, Real Time Information, Traffic Management, EV Stations
4. Smart Public Services – Public Safety, Lighting, Emergency Management, Internet Availability, Health, Learning
5. Smart Structures – Energy Management, Safety, Connectivity, Efficiency, Maintainability

1.2 Big Data and Internet of Things safety and security

1.2.1. Introduction in Big Data and Internet of Things safety and security

Information and communication technologies (IT) are, on the one hand, mean of dependability (reliability, availability, safety, security) assurance for systems for critical and commercial domains, and, on the other hand, they are source of vulnerabilities, faults and failures causing new security and safety related challenges and fatal effects for critical infrastructures and business applications.

Influence of modern ITs and IT related paradigms becomes more and more challengeable, first of all, for safety critical systems such as:

- instrumentation and control systems (I&Cs) of nuclear power plants (NPPs),
- on-board and ground control and navigation systems of piloted aerospace and aviation complexes,
- railway signalling and blocking systems,
- automotive systems including vehicle to vehicle, vehicle to infrastructure,
- health monitoring and control systems and so on.

Failures and emergencies of safety critical systems as a rule are caused by several reasons, combination of physical, design and interaction faults and human errors [21,22]. Physical faults are characteristic for hardware, design faults are characteristic for software (and programmable logics), interactive faults are consequences of physical and information intrusions on hardware and software respectively.

To ensure dependability we have to analyze related possibilities and risks at the all levels of a hierarchy “element-component-system-infrastructure” taking into account interaction and interdependency in the vertical and horizontal dimensions [22-24]. Von Neumann’s paradigm “reliable systems out of unreliable elements” [25] should be transformed considering challenges caused by application of modern ITs. Paradigm “dependable and safe infrastructure/system/component out of undependable and unsafe (or not enough dependable and safe) systems/components/elements” is becoming more and more important [26].

Besides, concept "IT for safety and security" should be added by "safe and secure IT". New technologies such as Internet of Things (IoT), Big Data (BD) and others can create new positive possibilities and challengeable deficits of cyber security and safety and it's required thorough analysis to search balance of key attributes and to take into account limitations for their application.

There are a lot publications dedicated to aspects of safety and security in context of Big Data (or BD bases analytics – BDA), IoT and other new conceptions and technologies. BDA and IoT are close conceptions, because IoT communications can be called a circulatory system for collection and processing of (big) data. These publications related to BDA/IoT can be divided on three groups:

- publications about BDA/IoT where aspects of safety, security, dependability are not defining and mentioned only [27,28];
- publications describing BDA/IoT based technologies as means to assure safety, security, dependability of critical or non-critical systems [29-31];
- publications that analyse aspects of BDA/IoT safety, security, dependability as a key problem. In this case the challenges and solutions for assessment and assurance of BDA/IoT based systems safety, security, dependability are considered [32-33].

Importance of analysing problems which are crossing of "BD/IoT systems" and "safety, security, dependability attributes" is confirmed by increasing of corresponding references during 2017 (N17) and 2018 (N18) years. Table 1 contains parts related to Internet references on pdf documents concerning fuzzy logic and artificial intelligence (as a close domain to BD and IoT), big data and Internet of Things.

Table 1.1. Reference statistics

Keywords, pdf	**Number of Internet references (N17), July 10, 2017**	**Number of Internet references (N18), August 10, 2918**	**N18/N17**
fuzzy logic	28 000 000	50 400 000	1.8
fuzzy safety	2 260 000	9 940 000	4.5

fuzzy logic security	2 760 000	6 030 000	2.2
fuzzy logic dependability	1 020 000	2 670 000	2.6
artificial intelligence	46 500 000	107 000 000	2.3
artificial intelligence safety	5 150 000	21 600 000	4.1
artificial intelligence security	72 000 000	83 800 000	1.2
artificial intelligence reliability	8 240 000	13 700 000	1.7
artificial intelligence dependability	724 000	10 700 000	1.5
big data	148 000 000	362 000 000	2.4
big data reliability	11 200 000	42 900 000	3.9
big data safety big data for safety	17 700 000	172 000 000 123 000 000	9.7 7.2
big data security big data for security	17 700 000	215 000 000 189 000 000	12.8 10.6
big data dependability	154 000	428 000	2.8
Internet of Things	-	350 000 000	-
Internet of Things reliability	-	23 800 000	-
Internet of Things safety	-	116 600 000	-
Internet of Things security	-	130 700 000	-

The following conclusions can be done basing on Table 1.1:

- number of references “BD/IoT – safety, security,…” has increased by a factor N18/N17 = 1.2-12.8 during 2017-2018 years;
- the hottest topics are “BD safety” and “BD for safety”, “BD security” and BD for security”;
- topics “IoT safety”, “IoT security” have metrics values similar “BD safety” and “BD security”.

Basing on analysis of publications it should conclude that systematic researches of positive possibilities, restrictions and deficits of safety, security and dependability connected with application of BD and IoT are much needed.

1.2.2. Safety and cyber safety via cyber security

Safety (Fig.1.1,a) is an attribute defining how IT based (for example I&C) system (via controlled object) influences on environment (other systems and objects with high value of failure and people health or life), decreases risks and consequences of emergencies. On the other side, failures of safety critical I&C can increase these risks, i.e. cause unsafe influence (red arrow) on information or/and physical environment [34].

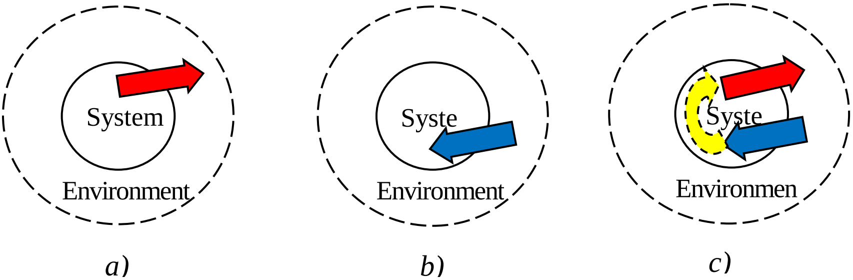


Fig.1.1. General models of safety (a), (cyber) security (b) and cyber safety (c)

Security (computer security, cyber security) defines the degree of influence of information or physical environment on system (blue arrow, Fig.1.1,b). Insecure influence of environment on safety critical system can cause unsafe influence of system on environment (yellow arrow, Fig.1.1,c).

In case when transition of system in unsafe state can be caused by attack via open or partially open cyber space a concept “cyber safety” can be used. Other words, if safety of system depends on cyber security (as a part of information security) it’s justifiable using of concept “cyber safety” (as a part of safety).

In general, unsafe behaviour of system can be caused by [34,35]:

- hardware anomalies (physical faults, manufacture design and physical faults, vulnerabilities of hardware components attacked by intruders);

- software anomalies (design faults tolerated by changing data environment, for example, by restart; design faults which have to be eliminated by changing of software code; faults caused by software ageing and vulnerabilities of software components attacked by intruders);

- FPGA anomalies similar hardware physical faults and software design faults, and two types of hardware and software vulnerabilities;

- system anomalies (configuration and system vulnerabilities).

Cyber safety is very important methodological concept safety critical systems which perform in cyber space and can be attacked by intruders.

1.3 Big data for safety and security critical domains

1.3.1. Possibilities and risks of application of BDA for critical domains

Data can be collected in such systems by use of different sensors, storages and other sources of information [36]. Table 1.2 shows possibilities and risks of BDA application in critical domains.

Table 1.2. Possibilities and risks of application of BDA for critical domains

Criti-cality types	**Domains**	**How (where) are BD made available?**	**Why can BD be applied?**	**Risks BDA application**
Safety critical	Nuclear Instrumentation and Control (NPP I&C)	By sensors and I&C storage	To optimize maintenance	Safety (via security) risk
	Aviation on-board systems	By sensors and OBS storage	To support decision making	Safety (via security) risk, Real time mode
	Airport flight control	By sensors and AFC storage	To support decision making	Safety (via security) risk,

	systems (AFC)			Real time mode
	Health (control) systems	By patient e-record database analysis	To support decision making	Safety (via security) risk, Real time mode
Security/ Data critical	Health (monitoring, storage) systems	By patient e-record database analysis	To support decision making	Security (privacy) risk
	Banking (access)	Banking database and other data analysis	To minimize risk of access	Security (privacy) risk
Mission critical	Space (unpiloted)	Data storages (Internet)	To minimize risks and optimize results	Security risk
	Big R&D project	Data storages (Internet)	To get the best results	Security (money loss) risk
Business critical	Banking (charges)	Banking database and other data analysis	To minimize risk	Security (money loss) risk
	E-commerce	Banking database and other data analysis	To optimize services	Security (money loss) risk

BDA is used to achieve the following objectives:

- to minimize risks or avoid potentially dangerous situations;
- to support decision making in pre-accident and post-accident cases;
- to optimize services and maintenance of complex systems (similar NPP I&C systems) and so on.

Main risks of BDA application are caused by two reasons:

- increasing of data capacity and additional possibilities to get unauthorized access to information;
- necessity of real time processing of huge data capacity to make decision or support decision making in time.

1.3.2. Reasons of accidences and application of BDA

Main reasons of accidents are complexity of projects and design anomalies, human errors and environment factors. Severe accidents are occurred if such reasons overlap in time. It confirmed by results of analysis of accident reasons for different severe emergencies beginning of crash of the biggest Swedish ship Vasa in 1668 to Fukushima accident (Table 1.3).

Table 1.3. Causes of accidences and application of BDA

Accidents, years	**Count-ries**	**Comp-lexity issue**	**Design ano-malies**	**Human factors/ errors**	**Environme nt**	**Is it Black Swan?**	**Could BDA help?**
Vasa, 1668	Sweden	Yes/No	Yes	Yes (politics, overloading)	Yes (strong wind)	No/Yes	No/Yes
Titanic, 1912	UK	Yes	Yes	Yes (business)	Yes (iceberg)	Yes/No	No/Yes
Three Mile Island, 1979	USA	No	Yes	Yes (violations of rules and errors)	No	Yes/No	Yes, for recovery
Challen-ger, 1986	USA	No	Yes	Yes (business, prestige)	Yes (wind)	No/Yes	No
Chernobyl, 1986	Ukraine	No	Yes	Yes (violations of rules)	No	Yes/No	Yes, for recovery
Fukushima, 2011	Japan	No	Yes	Yes (imperfect management during recovery)	Yes (tsuna-mi)	Yes/No	Yes, for recovery

More detailed description of accident reasons has been presented in [37]. Two questions and aspects of analysis are most interesting:

- are these accidents Black Swan? [38] Expert assessment of the accidents with priority Yes/No is shown in Table 3;
- could BDA used to help to predict and avoid these accidents? BDA could be used to support decision making for recovery after NPP accidents.

1.3.3. Application of BDA: pro and contra

Preliminary conclusions of application of BDA for safety critical systems are the following.

Search, transmission, collection, processing of big data can be applied:

- to improve maintenance and avoid failures including techniques of predictive analytics [39];
- to predict and minimize risks of emergencies;
- to support decision making and decrease resources/costs for recovery accidents and so on.

However, collecting and processing of huge data capacity can be

- useless, if required information and knowledge haven't been got;
- unsafe/insecure, if additional vulnerabilities resulted from increased capacity of data have been used for attacks and intrusions and cause obtaining secret/private information, fatal failures or accidents;
- energy-intensive, because BDA increases number of sensors, traffic intensity, additional storage and so on.

Implementation of BDA technologies can be a reason of extensive development as:

- the probabilistic/deterministic methods and techniques based on "small" data can provide more "fast" processing and receiving of information;
- "slow" traditional methods of processing of "big" data can be more effective;
- BDA based on artificial intelligence. Deep Learning requires big data to start application. Such situation is similar to "snowball effect" and can be called a rule "big data requires more big data";

- big data can be more unsafe/insecure than “small” data for safety critical (non-critical) systems.

Partial question is the following: what is better more complex (for example, semi-Markov’s) model with inaccurate parameters calculated by use of big/” small” data or simpler (Markov’s) model with accurate parameters calculated by use of “small”/big data?

1.4 Concept extending and limitations of internet of things application

To analyse IoT and IoT based systems safety and security issues definition of Internet of Things has to be specified. There are a lot of definitions [40]. In simplified view they are formulated by the following ways:

IoT is a new technology…

IoT is a mix/joining of existed technologies…

IoT is a new idea joining of known and modern technologies…

The conclusion to be drawn that IoT is a paradigm of joining and parametrization of a few technologies such as sensors, embedded and programmable devices, communications and cloud services).

IoT can be presented in general as

$$\text{IoT} \rightarrow \text{(X)Io(Y)Z}, \tag{1.1}$$

where (X) is an adjective determining main required attribute such as

X = {Dependable, Safe, Secure,…; Industrial,…};

I = Internet or Web; Web of Things (WoT) is known as well as IoT;

(Y) is an adjective determining actual attribute of things (Z),

Y = {Dependable, Safe, Secure,…; Important, Intelligent,…};

Z = {Alphabet: A (Aqua,...), B (Business,...), C (Cars,...), D (Drones,...), ...}.

Expression (X)Io(Y)Z is an example of generalizing application of modern technologies and IT methodologies. Other similar example is application of cloud based services. There are well-known Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and others including Everything as a Service (EaaS).

So, the following expression can be suggested

$$\text{(X)Eaa(Y)S,} \tag{1.2}$$

where X and Y have the same sense as in formula (X.1). The eCV Collaboratory has formalized such idea into several service systems related e-categories: Health as a Service (HaaS), Learning as a Service (LaaS), Business as a Service (BaaS) and Government as a Service (GaaS). Each has a role in Smart City ecosystem evolution/revolution leveraging ICT/5G/IoT. An concept is to focus on the individual and his/hers security in each of these areas such as patient/caregiver for HaaS, student/parent for LaaS, employee/customer for BaaS and citizen/taxpayer/visitor for GaaS [RP-Andrz].

One more example is software defined (SD) networks (SDN), data bases (SDD) and so on (software defined every thing, SDE). Hence, generalized expression is possible

$$\text{SD(Y)E,} \tag{1.3}$$

where Y has the same sense as in (1.1, 1.2).

Considering that application of IoT is accompanied by increasing of nodes and communications, increasing of transmitted data and, hence, increasing of threats, vulnerabilities, potential attacks and failures which can cause emergencies the following expressions, that are not strong mathematical formulas, describe these circumstances:

IoT = IoT (Internet of Things = Internet of Threats),

IoE = IoE (Internet of Everything = Internet of Emergencies).

According with [41] one of the ten main trends of IT development during next five years will be problem of IoT security and safety. Through 2022, half of all security budgets for IoT will go to fault remediation, recalls and safety failures rather than protection.

Hence, Von Neumann paradigm can be formulated for IoT application by following way: safe/secure IoT based systems or computing out of unsafe (or not enough safe)/ insecure (or not enough secure) nodes and communications. There are a few separate options of this expression depending on characteristics of nodes, communications and cloud resources.

1.5 Industry cases of Internet of Things and Big Data application

1.5.1. Internet of drones based post NPP accidence monitoring system

A general structure and underlying principles for creating an IoT based multi-version post-severe NPP accident monitoring system is shown on the Fig. 1.2 [42]. The system consists of an Internet of Things (IoT S) subsystem, a single wired communication subsystem (Wire S), light and wireless communication subsystems (Li-Fi S and Wi-Fi S) and three drone-based wireless subsystems (Drones, DF1, DF2). Drone fleet communicate with private cloud using IoT (DoT S1-S3 and IoT S). Thus sensors subsystems, drone fleet and private cloud form Internet of Drones (IoD) system for accident monitoring with multi-version sensor and communication sybsystems.

System dependability has to be assessed taking into account three issues: reliability, security and survivability.

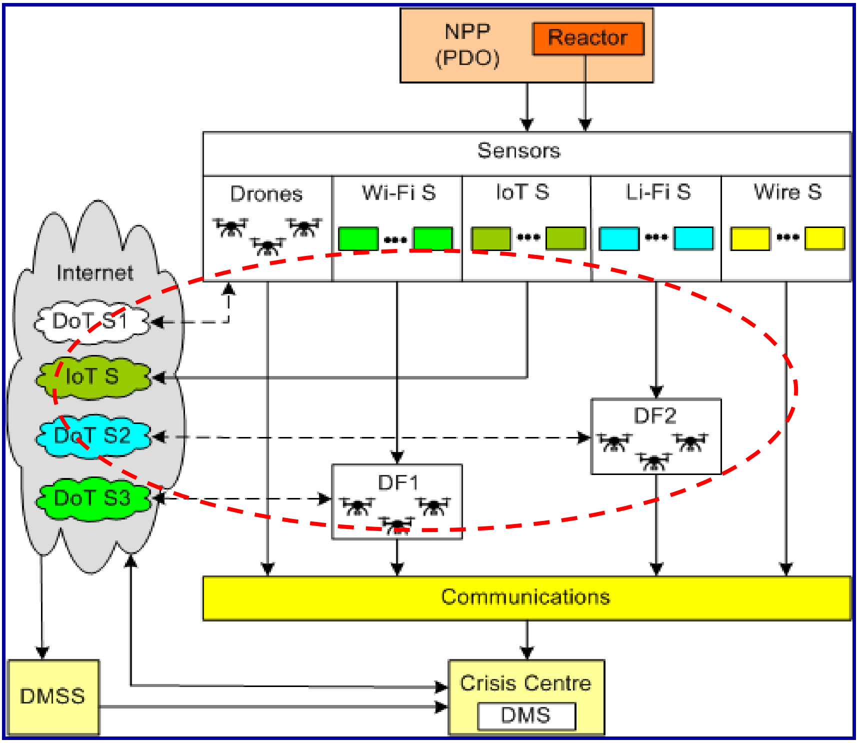


Fig.1.2. Internet of Drones based system for monitoring of severe NPP accident

Reliability block diagrams (RBD) for the system and its subsystems are based on considerations of different variants of sensor, communication and decision-making subsystems [35,43].

The probability of failure-free operation can be estimated and researched considering subsystem failure rates and various system configurations depending on strategy and procedures of drone fleet application [44].

Security assessment is based on vulnerability analysis of IoD subsystem and simulation of attacks on component and system vulnerabilities [45]. Survivability models are described in [46].

1.5.2. Internet of mobile devices based health systems

Other case is a healthcare IoT system (Fig.1.3) [47]. The system has unified structure and is designed to monitor and help to patients with diseases such as diabetics. The system components are a device with a reader, cloud, healthcare provider and communication channel.

Networked healthcare devices sense electrical, thermal, chemical, and other signals from the patient's body and inform about the physical and mental state. Such devices and system as a whole are safety critical because a human's life depends on its performance.

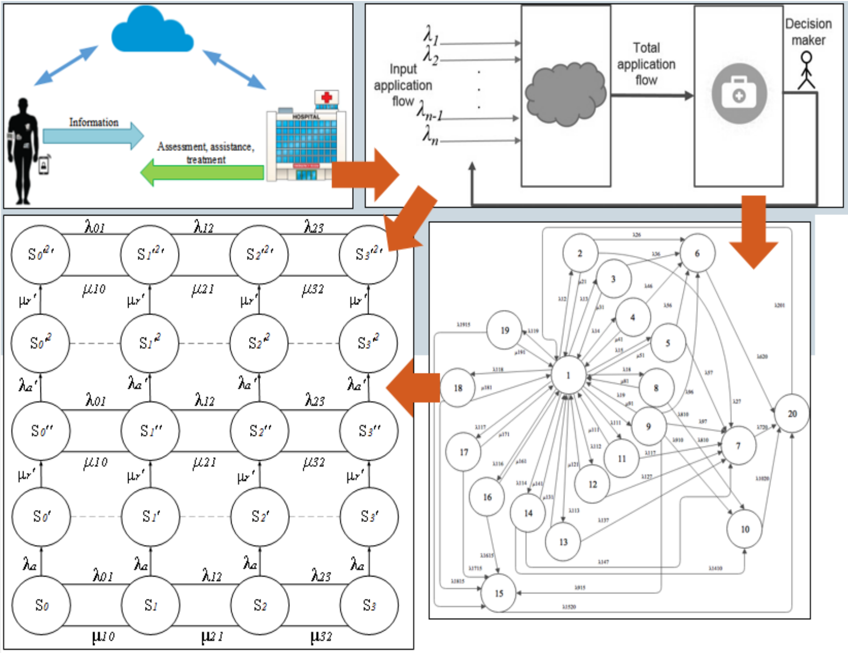


Fig.1.3. Modeling of IoT based health system

To assess safety a few techniques are applied [48]:

- failure/attack trees to identify security problems of the IoT infrastructure;

- a few models of healthcare IoT system based on the queueing theory considering dynamics of requests and publishing of vulnerabilities;
- multi-fragmental Markovian chains with fragments described by availability model of devices.

The models describe streams of the requests, hardware and software faults, attacks on vulnerabilities and procedure of recovery by restart and eliminating of one or more vulnerabilities

.

1.5.3. BDA based prediction of software (SW) reliability and security

To assess safety and security of mentioned and other industrial systems it's required to parametrize developed models. Most complex task is parametrization of software reliability and security. Usually information to evaluate software failure rates is not enough in frame of a company that develops and maintains a system [49].

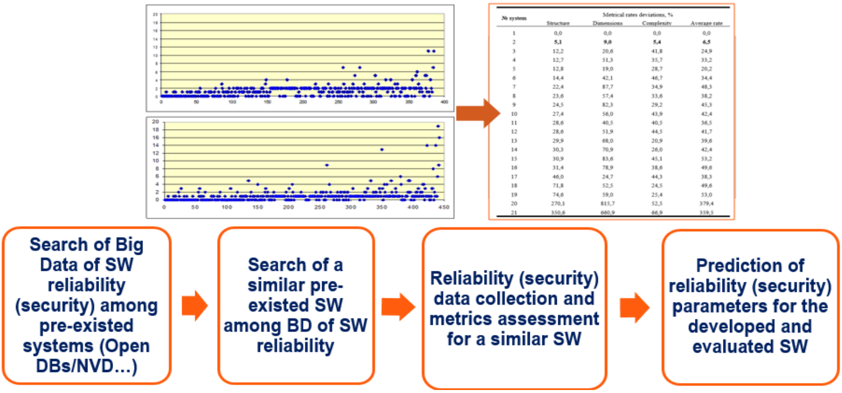
The methodology of software system reliability and security prediction can be based on processing information about software with similar attributes and metrics, which is extracted from BD storages and vulnerability databases [50]. The technique to search of similar programs uses [51]:

- metrics of complexity and structure software, metrics of program language similarity. The metrics assess group and average deviation rates describing the software system similarity;
- software agent tool to search, collect and process data.

The stage of SW reliability and security prediction and screenshots are shown on Fig. 1.4.

1.6 Work related analysis

The main goal of this section was to analyse challenges caused by development and implementation of IoT, Big Data and other modern technologies, first of all, challenges in area of safety and security assurance. This analysis is based on overview of standards, publications and education activities of a lot of universities of USA and EU universities including ALIOT project partners.


0
50
100
150
200
250
300
350
400
450
0
2
4
6
8
10
12
14
16
18
20
Search of Big Data of SW reliability (security) among pre-existed systems (Open DBs/NVD...)
Search of a similar pre-existed SW among BD of SW reliability
Reliability (security) data collection and metrics assessment for a similar SW
Prediction of reliability (security) parameters for the developed and evaluated SW

№ system	Metrical rates deviations, %			
	Structure	Dimensions	Complexity	Average rate
1	0,0	0,0	0,0	0,0
2	**5,1**	**9,0**	**5,4**	**6,5**
3	12,2	20,6	41,8	24,9
4	12,7	51,3	35,7	33,2
5	12,8	19,0	28,7	20,2
6	14,4	42,1	46,7	34,4
7	22,4	87,7	34,9	48,3
8	23,6	57,4	33,6	38,2
9	24,5	82,3	29,2	45,3
10	27,4	56,0	43,9	42,4
11	28,6	40,5	40,5	36,5
12	28,6	51,9	44,5	41,7
13	29,9	68,0	20,9	39,6
14	30,3	70,9	26,0	42,4
15	30,9	83,6	45,1	53,2
16	31,4	78,9	38,6	49,6
17	46,0	24,7	44,3	38,3
18	71,8	52,5	24,5	49,6
19	74,6	59,0	25,4	53,0
20	270,1	815,7	52,5	379,4
21	350,6	660,9	66,9	359,5

Fig.1.4 Principle and stage of BDA for software reliability and security prediction and assessment

These universities conduct research and implement education MSc and PhD programs on the Internet of Things and its applications for different domains such as human, industry, safety and security critical systems. In particular, the following courses and programs have been considered:

- IoT course for MSc in Coimbra University, Portugal [53];

- MSc programs in KTH University, Sweden including:

a) IoT for Information and Network Engineering [54],

b) Communication Systems [55],

c) Embedded Systems [56];

- MSc Programme on Embedded Systems and Internet of Things, Newcastle University, United Kingdom [57].

The courses focus on the Internet of Things for smart transport and cities, industry and smart grid, the development of techniques and tools for support creation of IoT based systems. However, a specific issues of IoT cyber safety and security are not studied. Part of the subsection 1.1 is based on [21].

Conclusions and questions

It's important to evaluate the impact as IoT, Big Data, 5G and other technologies evolves to address Global Grand Challenges. Another activity includes contributing towards the rebirth of the renaissance engineer/architect, a professional who provides deep expertise in a selected set of engineering discipline and a broad outlook in important social and global issues. This is consistent with the T-shape concept.

New technologies create new possibilities for people and society, but bring new deficits of cyber security and safety. This conclusion concerns fully of technologies of Big Data Analysis and Internet of Thing. Concept of cyber safety is important attribute for these and other technologies applied in critical domains.

BDA can be used as a powerful tool for trustworthy assessment of safety and security. Industrial cases illustrate possibilities how IoT and BDA can be used to assure safety and security for critical systems and infrastructures. Besides, big data analysis techniques can tolerate challenges of inaccurate assessment of high availability systems assessment. BDA makes it possible to improve maintenance and minimize risks of (fatal or pre-fatal) failures, support decision making and decrease resources/costs for recovery.

Limitations of BDA application are caused by extensive nature of technologies for collecting and processing big data. There are several challenges for BDA application in critical domains.

Some closure of safety critical domains causes restriction of multi-domain application of BDA. There is a problem “BD are not such big as they could be”; for example, diversity application results and CCF statistics are not enough available for each other [52].

Verification of BDA based techniques application. Independent verification and validation is a strong requirement to safety critical systems creation process in nuclear and other domains with high value of failure. BD based technologies are used for power saved/green applications.

However, BDA requires more and more resources. Hence, BDA has to become greener itself. It is required to search of a balance between traditional “small” data based methods and BDA.

There is common challenge for BDA and IoT: the more data and the more IoT nodes and communications – the less security (confidentiality) and safety of systems.

For IoT and IoT systems Von Neumann’s paradigm should be specified and implemented as “a secure IoT out of unsecure nodes, communications and clouds". Hence, important direction of future research is search of balance between "BDA and IoT for system security and safety" and assurance of "security and safety of BDA and IoT based systems" considering features of developed and operated systems, physical and cyber environment.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions:

1. What is the main idea of IoT and how is it different from other systems?
2. What made the electronics globally available?
3. Why is it possible to consider IoT a breakthrough technology?
4. What is the impact of public and government collaboration on IoT development?
5. What is the role of IoT in Smart City ecosystem?
6. What layers of population get an easier way of interaction with the help of IoT?
7. What issues can be caused by vulnerabilities of the system?
8. Why the security of each system is important?
9. What causes the increasing number of safety and security information resources?
10. What is the difference between safety and security? What do they have in common?
11. What are main vulnerabilities that can damage system's performance?
12. What kind of domain can be under risk while using BD?
13. What type of risk is applied to the domain which uses BD?
14. What are the purposes it is advised to use DBA for? When should the usage of them be avoided?
15. Why IoT security systems are highly important?
16. What are the reasons to expect major investments in the IoT security systems in the nearest future?
17. What are the ways of interaction between data collected by drones and other sensor's data?
18. Why the accuracy of system's measurement is important?

References

[1] "Internet of Things Online Course | MIT Sloan Executive Education", *Executive.mit.edu*, 2019. [Online]. Available: https://executive.mit.edu/openenrollment/program/internet-of-things-business-implications-and-opportunities/#.XTysbuZR3IV. [Accessed: 27- Jul- 2019].

[2] "Define IoT - IEEE Internet of Things", *Iot.ieee.org*, 2019. [Online]. Available: https://iot.ieee.org/definition.html. [Accessed: 27- Jul- 2019].

[3] *Grouper.ieee.org*, 2019. [Online]. Available: http://grouper.ieee.org/groups/2413/April15_meeting_report-final.pdf. [Accessed: 27- Jul- 2019].

[4] ITUIT Study Group 13 leads the work of the ITU on standards for next generation (NGN) and future networks (ITU, SERIES Y, 2005), 2005.

[5] R. Journal, "That 'Internet of Things' Thing - 2009-06-22 - Page 1 - RFID Journal", *Rfidjournal.com*, 2019. [Online]. Available: http://www.rfidjournal.com/article/view/4986. [Accessed: 27- Jul- 2019].

[6] M. Conner, "32-38", *EDN*, 2010. [Online]. Available: https://www.edn.com/design/sensors/4363366/Sensors-empower-the-quot-Internet-of-Things-quot-. [Accessed: 27- Jul- 2019].

[7] A. Gabbai, "Kevin Ashton Describes "the Internet of Things"", *Smithsonian*, 2015. [Online]. Available: https://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/#CCK32kDiIlxVuB1j.99. [Accessed: 27- Jul- 2019].

[8] G. Santucci, "The Internet of Things: Between the Revolution of the Internet and the Metamorphosis of Objects, 2010. Available: http://ec.europa.eu/information_society/policy/rfid/documents/iotrevolution.pdf [accessed June 01, 2018].

[9] "Everything you need to know about IIoT | GE Digital", *Ge.com*, 2018. [Online]. Available: https://www.ge.com/digital/blog/everything-you-need-know-about-industrial-internet-things. [Accessed: 27- Jul- 2019].

[10] J. Gulley, F. Holl, T. Kochanski and A. Rucinski, "A Pilot Course in Vital Electronics", in *8th European Workshop on Microelectronics Education, (EWME2010)*, Darmstadt, Germany, 2010.

[11] K. Bikonis, A. Rucinski and T. Kochanski, "PSOC Embedded Systems and its Potential Application in Wireless Sensors Network", in *2nd International Conference on Information Technology*, Wydzial ETI Politechniki Gdańskiej, 2010, Gdańsk, Poland., 2010, pp. vol. 18, pp. 249.

[12] T. Kochanski, “Technologies and Tools to build the Internet of Things,” IEEE Electron Devices, Solid State Circuits and Computer Societies and GBC/ACM, June 8, 2011 Analog Devices Wilmington, MA 01887.

[13] A. Doboli, T. Kochanski, K. Panetta and A. Rucinski, "Responding to Global Engineering Education Challenges through Vital Electronics", in *2010 ASEE Global Engineering Colloquium*, Marina Bay Sands, Singapore, 2010.

[14] B. Metcalfe, "Metcalfe's Law after 40 Years of Ethernet", *Computer*, vol. 46, no. 12, pp. 26-31, 2013. Available: 10.1109/mc.2013.374.

[15] A. Pellerin, "Easing embedded software development with EDA tools", *Embedded,* 2017. [Online]. Available: https://www.embedded.com/design/programming-languages-and-tools/4443336/Easing-embedded-software-development-with-EDA-tools. [Accessed: 28- Jul- 2019].

[16] R. Bogue, "Recent developments in MEMS sensors: a review of applications, markets and technologies", *Sensor Review,* vol. 33, no. 4, pp. 300-304, 2013. Available: 10.1108/sr-05-2013-678.

[17]]PSoC® 4, *PSoC 4000 Family Datasheet*. Cypress Semiconductor, 2017.

[18] M. Chin, "IBM has created a computer smaller than a grain of salt", *Mashable,* 2018. [Online]. Available: https://mashable.com/2018/03/19/ibm-worlds-smallest-computer/#oSy8ztxvigq7. [Accessed: 28- Jul- 2019].

[19] 5G in Five Minutes 5G Video: Skyworks CTO Peter Gammel explains the critical components for 5G and their importance

in the evolution of 5G technologies, Skyworks. Available: www.skyworksinc.com/5G [accessed May 29, 2018].

[20] "Azure/ai-toolkit-iot-edge", *GitHub*. [Online]. Available: https://github.com/Azure/ai-toolkit-iot-edge. [Accessed: 28- Jul- 2019].

[21] Dependable Internet of Things for Industry and Human Domains / V. Kharchenko, Ah Lian Kor, A. Rusincki (editors), River Publishers, 2019, 622 p.

[22] M. Yastrebenetsky and V. Kharchenko, "Nuclear Power Plant Instrumentation and Control Systems for Safety and Security", in *IGI Global*, USA, 2016, p. 472.

[23] H. Hristov and W. Bo, "Safety Critical Computer Systems: Failure Independence and Software Diversity Effects on Reliability of Dual Channel Structures", *Information Technologies and Control*, vol. 12, no. 2, pp. 9-18, 2014. Available: 10.1515/itc-2015-0011.

[24] B. Schneidhofer and S. Wolthusen, *Pdfs.semanticscholar.org*, 2015. [Online]. Available: https://pdfs.semanticscholar.org/5acb/243d3a7e679148a211ff8ddc78593c05eb42.pdf. [Accessed: 28- Jul- 2019].

[25] J. von Neumann, "Lectures on probabilistic logics and the synthesis of reliable organisms from unreliable components", 1952.

[26] V. Kharchenko and A. Gorbenko, "Evolution of von Neumann's paradigm: Dependable and green computing", in *East-West Design & Test Symposium*, 2013.

[27] A. Zomaya and S. Sakr, *Handbook of Big Data Technologies*. Springer International Publishing AG, 2017, p. 890.

[28] W. Härdle, H. Horng-Shing Lu and X. Shen, *Handbook of Big Data Analytics, Seria Springer Handbooks of Computational Statistics*. Springer International Publishing, 2018, p. 538.

[29] H. Parkinson and G. Bamford, "The Potential for Using Big Data Analytics to Predict Safety Risks by Analysing Rail Accidents", in *Third International Conference on Railway Technology: Research, Development and Maintenance*, 2016.

[30] O. Jaradat, I. Sljivo, I. Habli and R. Hawkins, "Challenges of Safety Assurance for Industry 4.0", in *2017 13th European Dependable Computing Conference (EDCC)*, Geneva, Switzerland, 2019.

[31] R. Toshniwal, K. Ghosh Dastidar and A. Nath, "Big Data Security Issues and Challenges", *International Journal of Innovative*

Research in Advanced Engineering, vol. 2, no. 2, pp. 15-20, 2015. [Accessed 28 July 2019].

[32] G. Walter and K. Bowers, "New Concept for a Big Data Safety Strategy", p. 24, 2018. [Accessed 28 July 2019].

[33] Z. Alam and H. Patel, "Security and Privacy Issues of Big Data in IoT based Healthcare System using Cloud Computing", *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 26-30, 2017. [Accessed 28 July 2019].

[34] K. Fu et al., "Security, and Privacy Threats Posed by Accelerating Trends in the Internet of Things", 2017. [Accessed 28 July 2019].

[35] K. Trivedi and A. Bobbio, "Reliability and Availability Engineering", *Cambridge Book Press*, p. 703, 2017. [Accessed 28 July 2019].

[36] V. Kharchenko, "Diversity for Safety and Security of Embedded and Cyber Physical Systems: Fundamentals Review and Industrial Cases", in *15th Biennial Baltic Electronics Conference*, 2016.

[37] V. Kharchenko, "Critical Computing and Big Data: Challenges and Solutions", in *2nd IEEE Conference Data Stream Mining and Processing*, 2018.

[38] V. Kharchenko, "Big Data and Internet of Things for Safety Critical Domains: Challenges and Solutions", in *International Conference on Information Technologies*, 2018.

[39] D. Galar, "Data Science in Industry and Transport: The black swan effect and the swan song desire", in *4th Annual Conf. on Computational Science and Computational Intelligence*, 2017.

[40] B. Khalid and N. Abdelwahab, "Big Data and Predictive Analytics: Application in Public Health Field", *International Journal of Computer Science and Information Technology and Security*, vol. 6, no. 5, pp. 1-6, 2016. [Accessed 28 July 2019].

[41] "What is the Internet of Things? Internet of Things definitions", *i-SCOOP*, 2016. [Online]. Available: https://www.i-scoop.eu/internet-of-things/. [Accessed: 28- Jul- 2019].

[42] "Tintri Showcases All-Flash Solutions at Gartner Symposium in Barcelona", *wallstreet-online.de*, 2017. [Online].

Available: https://www.wallstreet-online.de/nachricht/10033739-tintri-showcases-all-flash-solutions-at-gartner-symposium-barcelona. [Accessed: 28- Jul- 2019].

[43] V. Kharchenko, A. Sachenko, V. Kochan, H. Fesenko, M. Yanovsky and N. Yastrebenetsky, "NPP post-accident monitoring system based on unmanned aircraft vehicle: concept, design principles", *Nuclear and Radiation Safety*, vol. 173, pp. 24-29, 2017. [Accessed 28 July 2019].

[44] V. Kharchenko, A. Sachenko, V. Kochan and H. Fesenko, "Reliability and survivability models of integrated drone-based systems for post emergency monitoring of NPPs", in *International Conference on Information and Digital Technologies*, 2016.

[45] V. Kharchenko and V. Torianyk, "Cybersecurity of the Internet of Drones: Vulnerabilities analysis and IMECA based assessment", in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies*, 2018.

[46] H. Fesenko, V. Kharchenko and N. Bardis, "An approach to the drone fleet survivability assessment based on a combinatorial model", in *AIP Conference*, 2018.

[47] A. Strielkina, D. Uzun and V. Kharchenko, "Availability models for healthcare IoT systems: Classification and research considering attacks on vulnerabilities", in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies*, 2018.

[48] A. Strielkina, D. Uzun and V. Kharchenko, "Modelling of healthcare IoT using the queueing theory", in *9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2017.

[49] Yaremchuk and Kharchenko, "Big data and similarity-based software reliability assessment: The technique and applied tools", in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies*, 2018.

[50] V. Kharchenko and S. Yaremchuk, "Technology Oriented assessment of software reliability: Big Data based search of similar programs", in *13th International Conference on ICT in Education, research and industrial applications*, 2017.

[51] S. Yaremchuk, V. Kharchenko, A. Gorbenko, “Search of Similar Programs Using Code Metrics and Big Data-Based Assessment

of Software Reliability,. In: Alani M., Tawfik H., Saeed M., Anya O. (editors), Applications of Big Data Analytics. Springer, 2018.

[52] V. Kharchenko, A. Siora, V. Sklyar, A. Volkoviy and V. Bezsaliy, "Multi-diversity versus common cause failures: FPGA-based multi-version NPP I&C systems", in *Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies*, 2010.

[53] Internet Of Things Course - Immersive Programme Master in City and Technology [https://apps.uc.pt/search?q=Internet+of+Things]

[54] Master's programme in Information and Network Engineering [https://www.kth.se/en/studies/master/information-and-network-engineering/master-s-programme-in-information-and-network-engineering-1.673817]

[55] Master's programme in Communication Systems [https://www.kth.se/en/studies/ master/communication- systems/ description - 1.25691]

[56] Master's programme in Embedded Systems [https://www.kth.se/en/ studies/master/embedded-systems /description-1.70455/]

[57] Related Programmes to Embedded Systems and Internet of Things (ES-IoT) MSc [https://www.ncl.ac.uk/postgraduate/courses /degrees/embedded-systems-internet-of-things-msc/relateddegrees.html

2. TECHNOLOGIES OF INTERNET AND WEB OF THINGS

Ph.D., Prof. V.A. Tkachenko (NTU “KhPI”),
DrS, Prof. Y. V. Brezhniev (KhAI)

Contents

Abbreviations

6LoWPAN — IPv6 Low Power Wireless Personal Area Networks

AJAX — Asynchronous JavaScript and XML

ALIOT — Internet of Things: Emerging Curriculum for Industry and Human Applications

AMQP — Advanced Message Queuing Protocol

API — Application Programming Interface

BLE — Bluetooth Low Energy

BSON — Binary JavaScript Object Notation

CoAP — Constrained Application Protocol

DBMS — Database Management System

DDS — Data Distribution Service

DTLS — Datagram Transport Layer Security

ETSI — European Telecommunications Standards

IEEE — Institute Institute of Electrical and Electronics

IETF — Engineers Internet Engineering Task Force

IoE — Internet of Everything

IoP — Internet of People

IoT — Internet of Things

JMS — Java Message Service

jQuery — JavaScript Library

JS — JavaScript

JSON — JavaScript Object Notation

KhAI — Kharkiv Aviation Institute

KhPI — Kharkiv Polytechnic Institute

LoWPAN—Low-Power Wireless Personal Area Networks

LPWAN — Low-Power Wide-area Network

M2M — Machine-to-Machine

MEAN — MongoDB, Express.js, Angular.js, Node.js Mongo DB — Humongous Database

MQTT — Message Queue Telemetry Transport

MySQL — Free relational DBMS

NAU — National Aerospace University

NFC — Near field communication

Node.js — JavaScript run-time environment

NoSQL — Not Only SQL
NTU — National Technical University
OAuth 2.0 — Open Authorization Protocol
OCF — Open Connectivity Foundation
OGC — Open Geospatial Consortium
OpenSSL — Cryptographic Library (SSL/TLS) with Open Source
PaaS — Platform as a Service
PETRAS — Privacy, Ethics, Trust, Reliability, Acceptability and Security
PostgreSQL — Free object-relational DBMS
Pub/Sub — Publisher/Subscriber
REST — Representational State Transfer Radio
RFID — Frequency IDentification
SOA — Service-Oriented Architecture
SQL — Structured Query Language
SSE — Server Side Events
SSL — Secure Sockets Layer
TLS — Transport Layer Security
WLAN — Wireless Local Area Network
WoT — Web of Things
WPAN — Wireless Personal Area Network
WSN — Wireless Sensor Networks,
WSS — WebSocket Security
XML — eXtensible Markup Language
XMPP — Extensible Messaging and Presence Protocol

2.1 Internet and Web of Things

2.1.1 Conception of Web of Things

Development of technology M2M/IoT, information processing tools (Big Data) and decision-making (Cognitive Analytics) lead to changes in the technological, economic and social development models society. Areas of use of IoT are expanding in energy, transport, medicine, agriculture, housing, Smart City, Smart Home, etc. IoT focuses only on connecting physical objects to the network and their interaction with each other. Cisco introduced a new concept - Internet of Everything (IoE), which is based on the integration of people, things, data and processes. Thus, the next stage in the development of IoT / WoT is the Internet of all (IoE). In the future, the orbit of IoT will include technology deep machine learning, artificial intelligence, technology blockchain, robotics, etc. IoT is characterized by large changes in the infrastructure of the Internet and new communication models Smart Things or connections: “Thing-Thing”, “Thing-User” and “Thing-Web Object”. IoT Infrastructure consists of various networks of physical objects based on heterogeneous hardware and software platforms, protocol stacks, which are generally incompatible with each other. So the IoT is a collection of isolated physical networks that cannot communicate with each other via the Internet.

The concept of a WoT based on Web and its new technologies [1], enables the integration of all kinds of Smart Things and applications with which they interact. The concept WoT introduced such a notion as “Web Thing”, which is a digital representation of a physical or virtual object that is accessible through Web API RESTful. One of the major development issues for this new concept is creating efficient hypermedia-enriched application programming interfaces (APIs) [2]. Web API RESTful or Web API built with consideration of the REST architecture for a virtual representation of the physical objects are identified by URL and use application layer protocols such as HTTP, WebSocket, CoAP, MQTT in JSON format, and TLS / DTLS cryptographic streaming protocols. Thus, the virtual equivalent of physical objects (Web Thing), which were assigned a URL via Web API, can communicate with each other or with applications by using application-level protocols and share data in text-based JSON. In addition to the software interface Web Thing can be equipped with custom interfaces to ensure interoperability model “Thing- User”. The WoT reuses existing and well-known Web standards used in the programmable Web (e.g., REST, HTTP, JSON), semantic Web (e.g., JSON-LD, Microdata, etc.), the real-time Web (e.g., Web Sockets) and the social Web (e.g., OAuth or social networks). [3].

Thus, WoT provides the integration of Internet-connected physical devices of different producers on application level regardless of how they are connected on a network level and ensures the creation of a single global ecosystem of the IoT, which is open and compatible.

Currently WoT standardization has engaged WoT community and such international organizations for Standardization, as W3C (https://www.w3.org/WoT/), IETF (https://www.ietf.org/), ETSI (http://www.etsi.org/), OCF (https://openconnectivity.org/) and OGC (http://www.opengeospatial.org/), and supported by European research projects on the IoT, such as Sensei-IoT (http://www.sensei-iot.org/), SmartSantander (http://www.smartsantander.eu/) and IoT-A (https://iota.org/). WoT Interest Group published draft standards [4, 5, 6, 7, 8]. In addition to set out draft standards, Mozilla IoT community published its draft standards [9].

Together with the development of IoT technology and its network service WoT, there is an increasing need for specialists for development (software and hardware) and integration of technical solutions in the field of IoT, maintenance and operation of IoT networks. The problem of training (training and retraining) of IoT / WoT specialists is becoming urgent. The issue of training current and future engineers and researchers, technology application development and integration of modern IoT/WoT-solutions can be solved jointly by the companies that design and manufacture tools for IoT and institutions of higher education. For example, the aim of the project ALIOT [10] is integration of all available and prepared training programs, manuals and tools for the provision of training and advisory services in the field of systems based on the IoT for applications in different areas.

Companies involved into developing and manufacturing tools for IoT, are interested in higher-education-obtained professional skills creation and exploitation of IoT/WoT in a timely manner, in order to remain competitive in the field of development and production of IoT/WoT. In their turn higher education institutions are interested in teaching students the basics of the IoT/WoT design and operation to be competitive on the labor market in the field.

2.1.2 State of Art in Web of Things

It should be noted that companies that develop and produce tools for IoT and universities prepare specialists in the field of development and integration of modern IoT-solutions. For example, the company has created a University for Telit IoT [11]. The program Telit IoT University currently includes six courses, one of which is the IoT for Developers. In IoT University course [12], students look at User Interface and User Experience design strategies common to the industry and apply those strategies to building applications in ThingWorx using the Mash Up Builder. This course is focused on IoT-project, which does consider the WoT technologies. The PTC IoT Academic Program [1] consists of the ThingWorx™ application enablement platform in a PTC hosted environment where students and educators can build their own IoT applications. PTC works with corporate customers as well as market partners to ensure that students from all disciplines are better prepared to meet the needs of today's IoT world.

In the article [14] the information is presented on many bachelor's and master's programs on IoT. WoT technology is not considered. The IoT MSc program [15] is available at the Queen Mary University of London. MSc Internet of Things (Data) is currently available for one-year full-time study, two years' part-time study (Introduction to IOT, Enabling Communication Technologies for IOT). WoT technology is not considered. The article [16] presents the best universities that offer courses in the field of "Internet of Things", and study in detail what they offer their students. WoT technology is not considered.

The project "IoT Academy Samsung" [17] is organized on the base of Moscow Institute of Physics and Technology. In accordance with the experts of the research center of Samsung's teaching materials students will undergo a year-long training course on examining real case studies on Internet of things technologies in various industries and will be able to create their own IoT devices prototypes. WoT technology is not considered. The Cisco Internet of Things (IoT) [18] certifications and training are job-role-based programs designed to help meet the growing need for specialized talent. This education portfolio provides Internet Protocol (IP) networking expertise, with a focus on automation, manufacturing and energy and future expansion to include equally transformative industries. WoT technologies are not considered.

National Aerospace University “KhAI” and other Ukrainian universities prepare specialists on programmable mobile systems and IoT [19]. IoT-based systems are developed and investigated as well. WoT technologies are not considered in the discipline of “Industrial Internet of Things (IoT). Lviv IT Cluster and National University “Lvivska polytechnica” have launched a Bachelor program “Internet of things” [20]. Goal is to prepare specialists in the field of designing elements and applications of IoT, WoT is not considered. The work [21] deals with possible reflection of the theme of the Internet of things (Internet of Things) and machine-to-machine (M2M) in higher education curriculum (programs), but it does not reflect the technology of WoT. The purpose of such a curriculum is to consider issues related to information and communication technologies used in IoT and M2M. The proposed course aimed at listeners acquainted with modern information technologies, which stand behind such directions as inter-machine interaction and Internet of things.

Currently the authors have not seen in open papers mentions of any existing training programs on teaching students about technology for Web Things application development. Practical guide “Building the Web of Things” [1] is a basic training manual, which presents key technologies and concepts necessary to build application-level IoT and Architecture WoT, as well as define the methodology of application development for the Web Things on JS/Node.JS. This manual is intended for trained professionals in the field of Web application development technologies at JS/Node.JS.

Step-by-step tutorial can help professionals use WoT and semantic network to develop applications of Semantic WoT [22], but does not solve the problem of student learning of all the necessary technologies for the development of WoT. The book “Web application development Framework” [23] is designed for inexperienced Web developers, it outlines the creation of user interfaces, discusses ways to develop the server-side application on Node.JS, and methods of cloud services usage for deploying Web applications. The book “Web development with Node and Express” [24] is intended for programmers who want to build Web applications (regular sites that embody REST application programming interfaces) using JavaScript, Node. JS and Express. The curriculum “Technology and development tools WoT applications” was prepared with consideration of all teaching materials, that are prepared for programmers' learning on development technologies of advanced Web applications based on the JS/Node.JS, that will give the listeners the starting point for the WoT concept development and developing real applications Web Things.

2.1.3 Goals of the Web of Things course

The purpose of this work is to review IoT/WoT technology, curriculum (case-studies of WoT) and discus of structures tested in NTU "KhPI", the working curriculum of subjects "WoT application development technologies and tools", drawn up in the light of developing Web Thing API technologies proposed in the concept of WoT. This program offers technology development of WoT-applications, that is, the stack of MEAN technologies (Mongo, Express, Angular, Node) for development of WoT applications using JavaScript/Node.JS. Duration of training - one semester, the course is designed to train professionals (for master students) in "Computer Science", which is intended to form the students' theoretical knowledge and practical framework in design and exploitation of WoT-applications.

The task of the study course "Technology and tools for developing Web applications" is a theoretical and practical training of future specialists on such matters as:

– technologies for the application of markup languages, languages for description and programming in client Web applications;

– technology and tools for creating interactive Web interfaces;

technology application in Node.JS server applications;

– technology exchange messages between Web Apps in a mode of Real Time (Ajax, WebSockets) in XML messaging formats, JSON;

– technology for building applications with SOA architecture (architecture, REST);

– cloud computing technology and application deployment model for cloud platforms;

– architecture and technology of IoT;

– cloud platforms and services for the Web of Things;

– technology for application development based on Web based Things Raspberry Pi using the Node.js;

– security, privacy, and access control to the physical devices in the IoT/WoT.

The structure of this chapter is the following: Introduction; Survey of IoT/WoT Technologies; Structure of the training program “Technologies and tools for developing WoT applications”; Conclusions; References.

2.2 Survey of Internet and Web of Things technologies

Existing M2M-technologies allow machines to exchange information with each other. M2M is a subset of IoT. IoT is the Internet of People (IoP), extended by computing networks of physical items (Smart-Things), which can independently organize various connection models. IOT is a concept of the network infrastructure development (physical basis) online, in which “smart” things without human intervention are able:

- to connect to the network for remote interaction with other devices (Thing-Thing);
- to interact or interaction with autonomous or cloud data processing centers, or DATA centers (Thing-Web Objects) for data transmission, storage, processing, analysis and management decisions aimed at changing the environment Wednesday:
- to interact with user terminals (Thing-User) for the control and management of these devices [25].

The article [26] is the first to present the correlations among machine-to-machine (M2M), wireless sensor networks (WSNs), cyber-physical systems CPS and internet of things (IoT). The authors suggest that CPS is an evolution of M2M by the introduction of more intelligent and interactive operations, under the architecture of IoT.

Cisco believes [27] that The Internet of Everything is the next step in the evolution of smart objects-interconnected things in which the line between the physical object and digital information about that object is blurred. The WoT is a refinement of the IoT by integrating smart things not only into the Internet (network), but into the Web Architecture (application) [28].

2.2.1 IoT global network architecture

The papers provide overviews of the IoT: concepts, architectures, development technologies, physical devices, programming languages, protocols, and application [29, 30, 31, 32, 33]. The Internet of things consists of the networks of physical objects, the traditional network of the Internet and various devices (Gateway, Border router, etc.) that connect these networks. Figure 2.1 presents the components of the IoT architecture, which consists of several computer networks of physical objects connected to the Internet.

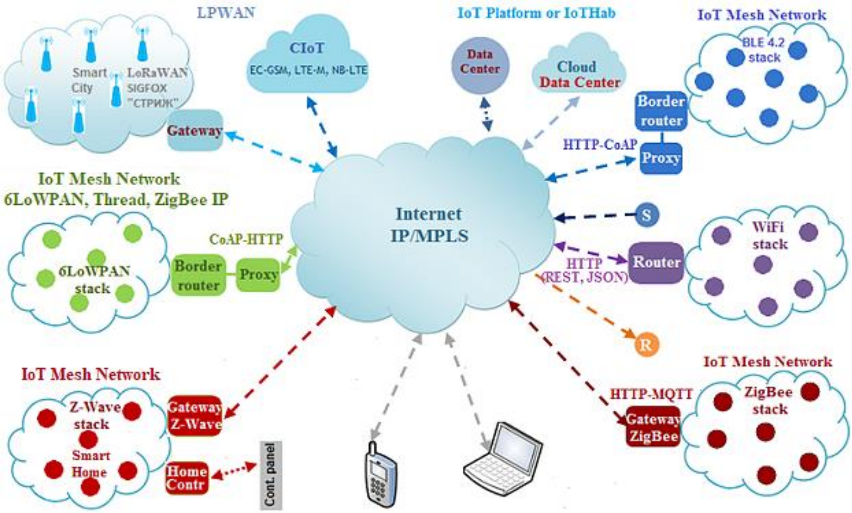


Figure 2.1. Components of IoT Architecture [25]

As seen of Figure 2.1, the network of Internet of things consists of: the computer networks of physical objects (Smart Objects), traditional IP Internet and various devices (Gateway, a Border router, Router), integrating these networks. It should be noted that Smart Objects are the sensors or actuators (sensors or actuators), equipped with a microcontroller with real-time operating system with a stack of protocols, memory, and communication device embedded into various objects, such as in electricity or gas meters, pressure sensors, vibration or temperature switches, etc. Smart Objects can be organized in computer network physical objects that can be connected via gateways (hubs or specialized IoT platform) to the traditional Internet.

In IoT there is not a single universal protocol for the integration of physical objects. Therefore, to create a network of physical devices, one shall acquire all the components of one manufacturer. As a result, the network of physical objects is fragmented and the provision of integration of physical devices connected to the Internet with incompatible protocol stacks is expensive.

Gateways are used to integrate the networks of IoT (for example, Z-Wave, ZigBee etc.), protocol stacks, which are incompatible with the TCP/IP stack of the Internet. Edge routers are used to integrate the Internet with networks of IoT, based on network protocol 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), where IPv6 is a version for wireless personal area sensor networks with low power consumption IEEE 802.15.4.

A proxy is used to harmonize protocols HTTP- CoAP. Technology-based network Thread, ZigBee, 6LoWPAN self-organizing nature are IP networks, and may not have an exit to external IP network using 6loWPAN protocol stack for the Organization of the work of autonomous networks and data transmission between the autonomous network nodes.

Wireless networks, used in LPWAN: IoT; WLAN and WPAN. LPWAN Technology. Key long-range LPWAN networks (Low-power Wide-area Network) technologies include: LoRaWAN; SIGFOX; Swift; CIoT (EC-GSM, LTE-M, NB-IoT). According to experts' estimates, more than 50% of IoT solutions would use LPWAN network. WLAN Technology. Medium-range technology, WLAN refers Wi-Fi (www.wi-fi.org) - a set of wireless standards IEEE 802.11, which can be used to build a wireless local area WLAN based network objects on the TCP/IP stack.

To build local wireless computer networks items, Wi-Fi Alliance has created a new IEEE 802.11 specification, which provides technology to build cellular networks. In addition, new standard Wi-Fi HaLow (IEEE 802.11 specification ah for the IoT) was created with low power consumption. Wireless personal area networks (WPAN). Key WPAN short-range wireless networks technologies: 6LoWPAN, Thread, ZigBee, Wireless IP, Z-Wave, EnOcean, RFID/NFC, BLE 4.2. Controllers and mini computers in the IoT. Today to manage the physical devices, the IoT uses controllers and mini computers: Arduino, Espruino, Tessel, Intel Edison and Galileo, Raspberry Pi, whose applications are created in c/C++, Java, JavaScript, Python, etc.

IoT application layer protocols. In the networks of physical objects, the interaction between components is done using the application layer protocols: DDS [34], CoAP, MQTT, XMPP, AMQP, JMS, REST/HTTP [35], etc. DDS is the core technology for Industrial IoT. CoAP Protocol (Constrained Application Protocol) - limited data transfer protocol similar to HTTP, but adapted to work with “smart” devices. MQTT protocols, XMPP, AMQP, JMS - these messaging protocols are based on broker scheme: publish/subscribe.

Security considerations for IoT. Security of IoT must be addressed at all stages of the development cycle and operation of hardware and software, communication channels, protocols stack, cloud components etc. is currently given a lot of attention in the field of IoT security. In the paper [36], several security and privacy concerns related to IoT are mentioned. The protection of data and privacy of users has been identified as one of the key challenges in the IoT.

The survey presents Internet of Things with architecture and design goals. In addition, a review and analysis of security and confidentiality issues at different levels in the IoT was performed. It should be noted that for the security of the IoT, standards [37] and guidance [38] have been created that provide manufacturers of tools with a set of guidelines for improving IoT security.

The document "State-of-the-Art and Challenges for the Internet of Things Security" [37] can be used by implementers and authors of IoT specifications as a reference for details about security considerations while documenting their specific security challenges, threat models, and mitigations. The goal of guidance [38] is to help manufacturers build more secure products in the Internet of Things area. Royal Academy of Engineering (London) [39] is a leader in Cybersecurity of the IoT. The PETRAS Cybersecurity of the IoT Research Hub brings together nine leading UK universities.

The development of IoT depends on many factors: technology, low-power wireless networks; Smart Objects technology; the pace of 5G networks adoption; operating systems for microcontrollers sensors and actuators; widespread use of 6LoWPAN/IPv6 protocol stack; M2M technology; effective use of Cloud computing for IoT platforms; Misty technology computing (fog computing) and Software-Defined Networks; ensuring hardware and software cyber resilience.

2.2.2 Technology Web of Things

IoT focuses on the lower layers of the network stack, and the WoT service on the upper layers, application tiers. By using web technologies, protocols, programming languages and formats [40] such as REST, XML, JSON, MQTT, XMPP, Atom, WADL, Open ID and OAuth, the WoT has contributed to reducing the barriers for common understanding and smooth interplay between heterogeneous real world devices, services and data.

The WoT concept, based on the Web and its new technologies [1], provides integration of all types of Smart Things and applications with which they interact. It is known that WoT uses standards applied in such technologies as programmable Web (HTTP, REST, JSON), semantic Web (JSON-LD, Microdata, etc.), Real-time Web (WebSockets) and social Web (OAuth or social networking APIs). The problems of the WoT architecture, development technologies, programming languages, APIs, application-level protocols based on RESTful principles are described in many articles [41, 42, 43, 44].

Thus, WoT provides integration of devices in the Internet. The WoT is a service similar to the IoT infrastructure service, World Wide Web, of the Internet infrastructure. WWW is a distributed information system based on the use of hypertext documents in HTML format, access and transfer of which are achieved using the HTTP application. WoT is an extended service Web.

By analogy with the Web architecture, the architecture of the WoT is the World Wide Web or distributed system of Web Things virtual resources (virtual representations of Things) that provide access to the physical objects, i.e. applications that are hosted on Smart Objects or intermediate IoT network devices through Web Thing API.

The essence of WoT is that Web Thing physical objects or intermediate gateway devices, given that they have their own URL (Web-address) and software interface with the RESTful Web API can communicate in text-based JSON both with each other and with applications based on SOA. To ensure interaction model "Thing- User" Web Things applications must have user interfaces. Due to limited resources, not all Web Things can offer their own Web API is RESTful, based on the concept of WoT.

For integration of Smart Objects in The Internet three different integration templates are provided:

- Direct Connectivity,
- Gateway Based Connectivity,
- Cloud Based Connectivity [1].

Implementation of Web Thing API on its own platform can be performed on the basis of a Web server that is hosted on the controller embedded in Things.

Web application for the Web Things can consist of frontend and backend, i.e. can be implemented as user interfaces for users to interact with Things via Web browsers (for example, site sensors: http://devices.webofthings.io/pi/) and mobile applications and interfaces (API) applications using the RESTful architecture for data exchange between devices. As a controller, you can apply, for example, single board computer Raspberry Pi based on Linux.

The latest version of the computer (Raspberry Pi 3 Model B) has a built-in support for Wi-Fi and Bluetooth 4.1. In addition, Raspberry Pi GPIO ports available for direct connection to devices (e.g. temperature sensors, displacement, etc.). To implement Web Server Node.js can be applied (for example, Node v 7.10.1).

To develop the server-side application it is advisable to choose the programming language JavaScript in Node.js. Software code of client side of the application is developed in HTML, CSS and JavaScript. For of data exchange between devices in application the interfaces of Client APIs and API Server, built with consideration of the REST architecture, are implemented.

Figure 2.2 presents Web API Thing, which can be placed either directly on the device itself, and intermediate Web Things gateway network or in cloud service.

In the case of implementation of a Web Thing API on an intermediate device such as a gateway, you can use the prototype gateway Things Gateway, Figure 2.3.

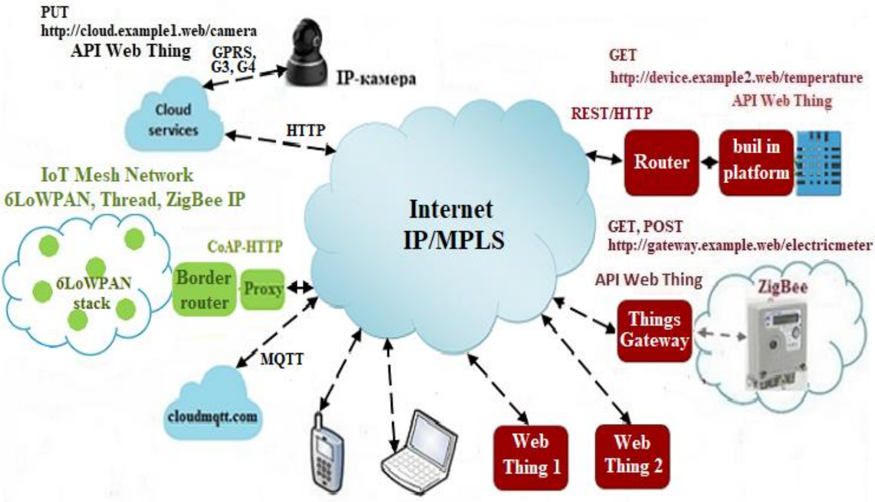


Figure 2.2. Components of WoT Architecture [45]

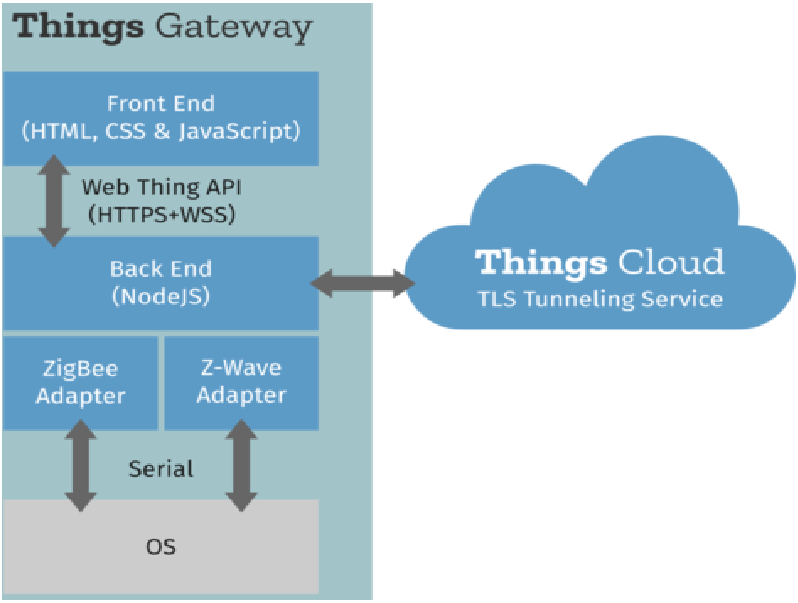


Figure 2.3. Things Gateway [46]

Things Gateway, Figure 2.3, is created by the developers of the Mozilla community in JavaScript using Node.js server platform, and is available as a ready to install on the Raspberry Pi board assemblies. If you implement a Web API Thing on cloud server, Web Thing Clients (Web devices or users) communicate to cloud-based server (cloud-based server analogous Things Gateway) by domain address of devices, which runs the application hosted on that server, and the application accesses devices, such as a camcorder, and manages them.

The EVRYTHNG Platform [47] is a cloud Platform-as-a-Service (PaaS) for storing, sharing, and analyzing data generated by physical objects. The Platform gives a unique and permanent digital identity (also known as ADIs) to each individual object and allows authorized applications and users to access it via REST and Pub/Sub (MQTT) APIs.

Security in WoT is provided by certificates, encryption and authentication. Cryptographic streaming protocols TLS/DTLS [48, 49] are the basis of secure HTTP protocols (HTTPS), WebSocket (WSS), MQTT (MQTTS) and CoAP, which are used in WoT. To do this, you can install the OpenSSL library on the server (sudo apt-get install openssl). In addition, you can apply recommendations and methods for authorizing and controlling access to the server.

Authentication is one of the means of protecting WoT applications [50, 51, 52, 53]. You can set the API token with Node.js (install Node.js: node-oauth2-server). Authentication OAuth2 is designed to protect the Web API using a token-based authentication process. The token will be used to authenticate Smart Objects for each request to the server. You can use the OAuth 2.0 social media tools for WoT authentication.

2.3 Training on technologies and tools for developing WoT applications

All of the technologies outlined in the proposed themes are used in the development of Web Thing applications in line with the concept of WoT.

Thus, the training program proposed in Table 2.1 is designed to prepare future specialists develop real Web Thing applications.

Table 2.1. Curriculum structure

The topic	Content
Development tools for Web applications: IDE, browser, version control system.	At present, the only IDE for creating client and server applications on JS/Node.js is WebStorm [54]. But the own IDE (for JS development) could be built based on a text editor Sublime [55] with the plugins. In this program Git [56] is used as a version control system (VCS) of the Web application files, Git is used in many famous projects as VCS. In addition, familiarity with Git gives students the opportunity to explore GitHub, the largest Web service for hosting IT projects and their joint development.

HTML and XML technologies in client-side Web applications.	HTML document structure, logical languages HTML5 markup [57] and XML [58], technologies used in creating layouts or templates on HTML5 for Web sites, technology HTML-layout technology in editor Sublime Text.
CSS Technology And CSS3 in client Web applications and the use of the Bootstrap framework.	CSS [59] - language for describing the appearance of documents (style declaration, selector types, block and line elements, style preprocessors, CSS frameworks and Emmet LiveStyle). The layout technology of a web application with adaptive design based on Bootstrap [60] in the Sublime Text editor.
Technology and tools for creating interactive Web interfaces.	To develop interactive Web-interfaces, one must apply the basic triad HTML technologies, CSS and JavaScript [60, 61], which form the structure, style, and behavior of Web applications. One of the components of the triad technologies is: the JavaScript programming language (syntax, set of technologies for creating interactive Web applications with JavaScript).
Technology of using jQuery to create interactive Web interfaces.	jQuery [63] is a JavaScript-based library that contains ready-made JavaScript functions. jQuery manipulates the html elements of the document and uses the DOM to change its structure. There are two methods for connecting the jQuery library to the client application: local and remote connections. JQuery has a large number of third-party plug-ins with which make it possible to significantly improve the interface of the client side Web-applications or WoT.

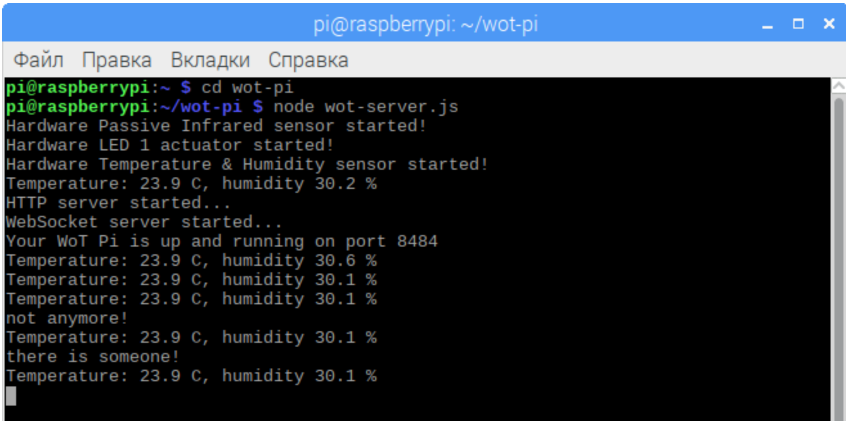
The exchange of messages between Web Apps in a Real Time mode (Ajax).	Four network technologies for interacting Web Apps based on client-side JavaScript scenarios (AJAX, COMET, SSE, WebSocket). This topic is dedicated to AJAX [64] (Ajax technology and data transfer formats, ajax requests for “pure” JavaScript; ajax and jQuery).
Exchange of messages between web apps in Real Time mode (WebSocket).	WebSocket [65] – is a technology of asynchronous interaction between the Web client and the Web server. WebSocket is a protocol of full-duplex communication over a TCP connection, intended for the exchange of messages between a web client and a web server in real time. WebSockets have an API that can be used in web applications and is called the WebSockets API [66, 67].
Web servers and application servers.	It is proposed to consider the HTTP protocol, the client/server model, the architecture of the Web server, application server, as well as to form an idea about the technology of these tools. In addition to traditional Web servers, Node.js technology is considered [68], which enables to create event-driven servers using JavaScript.
DBMS. Technology and software for creating databases.	There are 6 data models: lists (flat) relational databases, hierarchical, network structures, object-oriented databases and document-oriented data model. Currently, they are the most widely used when designing a relational database model (MySQL, PostgreSQL, MSSQL Server). It should be noted that the most popular database management system for Node.js is currently the MongoDB [69], which is a NoSQL. MongoDB is a document-oriented management system (DBMS) open source software. MongoDB is a new approach to build databases without SQL queries, tables, foreign keys, etc. In MongoDB, JavaScript is used as the query language, the data is stored in the BSON format, i.e. binary JSON.

WebRTC - is a technology for creating Web communications applications.	WebRTC [70] is an open source technology for building peer-to-peer networks, which allows to send text and multimedia data directly between browsers. Signaling server is used only for setting up p2p connection between the two browsers. The WebRTC technology is implemented by three JavaScript APIs: RTCPeerConnection, Media Stream (getUserMedia), RTCDataChannel.
Cloud technologies-development tools for Web applications and messaging service in Real Time mode.	Cloud computing [71] is the delivery of computing services - servers, storage, databases, networking, software, analytics and more—over the Internet (“the cloud”). Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage, similar to how you are billed for water or electricity at home.
Technologies for creating applications with SOA architecture and use of SaaS with APIs.	A service-oriented architecture [72] is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or two or more services coordinating some activity. Some means of connecting services to each other is needed.
IoT Technologies.	The Web of Things [29, 30, 31, 32, 33, 34, 35] is a refinement of the Internet of Things by integrating smart things not only into the Internet (network), but into the Web Architecture (application). In this section, it is necessary to consider: the IoT architecture, controllers and mini computers, the IoT platform, application programming languages, wireless network technologies, protocols, IoT security issues.

Protocols and technologies of creating Applications for Web Things based on Raspberry Pi 3 Model B using the Node.js platform.	The Semantic Web [1, 40]. The Web of Things [41, 42, 43, 44] is a high-level application protocol designed to maximize interoperability in the IoT. The WoT architecture stack is not composed of layers in the strict sense, but rather of levels that add extra functionality. Each layer helps to integrate Things to the Web even more intimately and hence making those devices more accessible for applications and humans. The following shall be considered in this section: Linux-based mini computers; versions of Raspbian for Raspberry Pi; implementation the Web Thing API for the Direct Connectivity integration template and for the Gateway Based Connectivity integration template.
Cloud platforms and services for the Web of Things.	Data Analytics. The Cloud Based Connectivity integration template allows Web platform to act as a gateway to implement API Web Thing on the staging device. EVRYTHNG platform [73] is a cloud-based platform-as-a-service (PaaS). The platform provides a unique and permanent identifier for each individual object and enables authorized applications and users to access it via the REST API and Pub/Sub (MQTT). It is proposed to consider the technology for implementing of the Web Thing API for the Cloud Based Connectivity integration template.
Security, privacy, and access control to the physical devices on the Internet.	Security in IoT [36, 37, 38] should be provided at different levels of the network. Security in WoT is provided by certificates, encryption and authentication [48, 49, 50, 51, 52, 53]. The security issues: protecting Web Thing (encryption, enable HTTPS, WSS and TLS on the server); authentication and access control; the use of social networking tools OAuth 2.0 for WoT-authentication.

NTU “KhPI Information Systems Department, trains students in the specialty “Computer Science” according to the curriculum presented in Table 2.1.

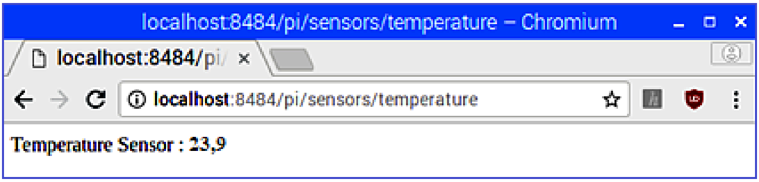
Figure 2.1 and Figure 2.2 shows the implementation materials of the web interface API Web Thing for the integration template Direct Connectivity. From the one shown in Figure 2.1 it follows that the HTTP and WebSocket servers are functioning. The Web Thing application installed in the Node.js environment on the Raspberry Pi 3 Model B provides data on the functioning of the motion sensors (PIR sensors), temperature and humidity (DH22) and the actuator (LED 1). Data on temperature, humidity and movement is constantly updated.



```
pi@raspberrypi:~ $ cd wot-pi
pi@raspberrypi:~/wot-pi $ node wot-server.js
Hardware Passive Infrared sensor started!
Hardware LED 1 actuator started!
Hardware Temperature & Humidity sensor started!
Temperature: 23.9 C, humidity 30.2 %
HTTP server started...
WebSocket server started...
Your WoT Pi is up and running on port 8484
Temperature: 23.9 C, humidity 30.6 %
Temperature: 23.9 C, humidity 30.1 %
Temperature: 23.9 C, humidity 30.1 %
not anymore!
Temperature: 23.9 C, humidity 30.1 %
there is someone!
Temperature: 23.9 C, humidity 30.1 %
```

Figure 2.1. A screenshot of the application's operation Web Thing for the Direct Connectivity integration template

According to the Figure 2.2, on request you can view physical devices parameter values in a Web-browser, for example, the values of the temperature sensor.


localhost:8484/pi/sensors/temperature – Chromium
localhost:8484/pi/
localhost:8484/pi/sensors/temperature

Temperature Sensor : 23,9

Figure 2.2. A screenshot displaying the values of the temperature sensor in the Web-browser

2.4 Work related analysis

This work reviews the existing technologies in the IoT / WoT, for which draft standards are being developed. At the 8th International Workshop on Web Objects (WoT 2017), it was noted that the REST architecture was the de facto basis for building the software interface of intelligent physical objects connected to the Internet. WoT, based on the Web and its new technologies, provides integration of all kinds of Smart Things and applications with which they interact, and transforms the world of physical objects into a distributed information system. Nowadays a creation of RESTful Web API for Web Thing on JS in Node.JS is preferred over other programming languages. In this regard, the proposed curriculum technologies are relevant and aim to prepare future professionals to develop real Web Thing applications. The technologies mentioned are tested in the teaching process of students.

The industrial IoT is part of IoT. The IoT is a network of computers, devices, and objects that collect and share the industrial data. It allows operating of industrial systems in more efficient and safe manner. Besides, industrial IoT is used for smart grid to improve efficiency of power generation and distribution. The many challenges and risks caused by industrial IoT shall be addressed by development of methods and tools. This will be done under the course that is to be developed by National aerospace university.

It should be noted that companies that develop and produce tools for IoT and universities prepare specialists in the field of development and integration of modern IoT-solutions. For example, the company has created a University for Telit IoT [11]. The program Telit IoT

University currently includes six courses, one of which is the IoT for Developers.

In IoT University course [12], students look at User Interface and User Experience design strategies common to the industry and apply those strategies to building applications in ThingWorx using the Mash Up Builder. This course is focused on IoT-project, which does consider the WoT technologies.

The PTC IoT Academic Program [13] consists of the ThingWorx™ application enablement platform in a PTC hosted environment where students and educators can build their own IoT applications.

In the article [14] the information is presented on many Bachelor’s and Master’s programs on IoT. The IoT MSc program [15] is available at the Queen Mary University of London. MSc Internet of Things (Data) is currently available for one-year full-time study, two years’ part-time study (Introduction to IOT, Enabling Communication Technologies for IOT).

The article [16] presents the best universities that offer courses in the field of “Internet of Things”, and study in detail what they offer their students. The project “IoT Academy Samsung” [17] is organized on the base of Moscow Institute of Physics and Technology.

Things (IoT) [18] certifications and training are job-role-based programs designed to help meet the growing need for specialized talent.

National Aerospace University “KhAI” and other Ukrainian universities prepare specialists on programmable mobile systems and IoT [19]. Lviv IT Cluster and National University “Lvivska polytechnica” have launched a Bachelor program “Internet of Things” [20].

Conclusions and questions

In this chapter we proposed and discussed the structure of the curriculum “Technology and development tools WoT applications” designed for training the development of modern web applications based on JS/Node.JS, which will give the listeners a starting point for mastering the WoT concept and developing Web Things applications.

Because in the future the orbit IoT/WoT will include the technologies of deep machine learning, artificial intelligence, technology blockchain and robotics, the curricula for training of future specialists in the field of IoT/WoT will be updated and filled with new content. In the future it is planned to deploy a specialty “Architecture and technology IoT/WoT” and propose the disciplines that will be taught within the framework of this specialty.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What technology is used to create interactive web applications using JavaScript?
2. What is the essence of jQuery technology?
3. What is the essence of a SQL database management system?
4. What is the essence of NoSQL (Not Only SQL) DBMS?
5. MongoDB DBMS Interaction Technologies with Web Applications.
6. What is WebSocket?
7. What is the essence of Ajax technology and data transfer formats?
8. What is the difference between WebSockets and REST?
9. What technology is used for web server and web application interaction?
10. What is the essence of Node.js technology, which allows you to create event-driven servers?
11. What are the main cloud computing models?
12. Web services and cloud computing.
13. What does the IoT architecture consist of?
14. What are the main technologies used in IoT/IIoT?
15. What are the main programming languages used to create applications that implement the API at the first level of IoT/IIoT?
16. What technologies are used to create client IoT applications for smart devices?
17. List the components of the IoT system architecture.
18. What are the main cloud computing used for IoT platforms?

19. Technologies of work with Big Data (MapReduce, Hadoop).

20. IoT/IIoT и интеллектуальный анализ данных (Data Mining).

21. 6LoWPAN protocol stack for IoT/IIoT.

22. Application Layer Protocols for IoT/IIoT.

23. With what networks does the interaction of sensors and actuators at the first level of IoT / IIoT?

24. RFID Technology Concepts.

25. WSN technologies and short-range and long-range networks protocols.

26. M2M technologies and protocols (DDS, LwM2M standards).

27. What are the main standards applied in IoT/IIoT?

28. Basics of WoT. Semantic web and microformats.

29. Stack of architecture Web of Things with different layers.

30. Patterns of Smart Objects Integration to the Internet.

31. Web Thing API implementation technologies for WoT integration patterns.

32. Security, privacy, access control to physical devices WoT

33. Integrating Web of Things and Semantic Web Technologies.

34. Semantic Web Technologies Standards (Semantic Sensor Network XG Final Report, RDF Schema).

35. What standards are used to develop WoT applications?

References

1. Dominique D. Guinard and Vlad M. Trifa, ‘Building the Web of Things’, Manning Publications.: United States, June 2016, 344 p.

2. Jaime A. Martins, Andriy Mazayev, Noélia Correia, ‘Hypermedia APIs for the Web of Things’, IEEE Journals & Magazines, Vol. 5, p.p. 20058 – 20067, September 2017.

3. Mihai Vlad Trifa, ‘Building Blocks for a Participatory Web of Things: Devices, Infrastructures, and Programming Frameworks’, A dissertation submitted to the ETH Zurich for the degree of Doctor of Science, 2011, 190 p.

4. Kazuo Kajimoto, Matthias Kovatsch, Uday Davuluru, ‘Web of Things (WoT) Architecture’. W3C First Public Working Draft, 14 September, 2017. Available at: https://w3c.github.io/wot-architecture/. [accessed May 3, 2018].

5. Sebastian Kaebisch, Takuki Kamiya, ‘Web of Things (WoT) Thing Description’. W3C Editor's Draft 27 April, 2018. Available at: https://w3c.github.io/wot-thing-description/. [accessed May 3, 2018].

6. Zoltan Kis, Kazuaki Nimura, Daniel Peintner, Johannes Hund, ‘Web of Things (WoT) Scripting API’. W3C Editor's Draft 04 April, 2018. Available at: https://w3c.github.io/wot-scripting-api/. [accessed May 3, 2018].

7. Michael Koster, ‘Web of Things (WoT) Protocol Binding’, W3C Editor's Draft 04 April, 2018. Available at: https://w3c.github.io/wot-binding-templates/. [accessed May 3, 2018].

8. Vlad Trifa, Dominique Guinard, David Carrera, ‘Web Thing Model’, W3C Member Submission 24 August, 2015. Available at: https://www.w3.org/Submission/wot-model/. [accessed May 3, 2018].

9. Ben Francis, ‘Web Thing API’, Unofficial Draft 02 May, 2018. Available at: https://iot.mozilla.org/wot/. [accessed May 3, 2018].

10. The name of project is ALIOT, which is acronym from official name ‘Internet of Things: Emerging Curriculum for Industry and Human Applications’. (n.d.). Available at: http://aliot.eu.org/. [accessed May 3, 2018].

11. Telit IoT University, ‘Things-to-Apps Made Easy’. (n.d.). Available at: https://www.telit.com/iot-university/. [accessed May 3, 2018].

12. IoT University, ‘IoT UI Development with ThingWorx: Course Summary, Course Milestones’. (n.d.). Available at: https://www.iotu.com/enrollment/student/iot-ui-development-with-thingworx. [accessed May 3, 2018].

13. IoT Academic Program, ‘The PTC IoT Academic Program is a “passport” to the future for students, makers, and researchers’. (n.d.). Available at: https://www.ptc.com/-/media/Files/PDFs/Academic/iot_academic-program_EN.pdf?la=en&hash=6AAA287FED90220A9B4104D100CBA9BF94B1CF3B. [accessed May 3, 2018].

14. G. Kortuem, A. K. Bandara, N. Smith, M. Richards, M. Petre, ‘Educating the internet-of-things generation’, Computer, vol. 46, no. 2, pp. 53-61, Feb., 2013.

15. Queen Mary University of London, ‘MSc Internet of Things - Queen Mary University of London’. (n.d.). Available at: https://www.qmul.ac.uk/postgraduate/taught/coursefinder/courses/173148.html. [accessed May 3, 2018].

16. IoT India Magazine, ‘10 Leading University Courses on IoT (Worldwide)’, August, 2016. Available at: https://curiousdose.com/2016/08/10-leading-university-courses-iot-worldwide/. [accessed May 3, 2018].

17. MIPT, ‘The IOT Academy of Samsung opens in MFTI’. (n.d.). Available at: https://mipt.ru/news/v_mfti_otkryvaetsya_iot_akademiya_samsung. [accessed May 3, 2018].

18. Cisco, ‘Internet of Things Specialists’. (n.d.). Available at: https://www.cisco.com/c/en/us/training-events/training-certifications/certifications/specialist/internet-of-things.html. [accessed May 3, 2018].

19. Department of Computer Systems, Networks and Cybersecurity, ‘Programmable mobile systems and Internet of Things’. (n.d.). Available at: https://csn.khai.edu/speciality/programovani-mobilni-sistemi-i-internet-rechej . [accessed May 3, 2018].

20. Internet of Things, ‘Bachelor program in Internet of Things’ (n.d.). Available at: http://iot.lviv.ua/. [accessed May 3, 2018].

21. Namiot D.E. ‘About the Internet of Things training programs’, International Journal of Open Information Technologies ISSN: 2307-8162 vol. 3, no. 5, pp. 35-38, 2015.

22. Amelie Gyrard, Pankesh Patel, Soumya Kanti Datta and Muhammad Intizar Ali, ‘Semantic Web Meets Internet of Things and Web of Things: [2nd Edition]’, WWW 2017 - 26th International World Wide Web Conference, Perth, Australia, pp. 917-920, Apr 2017.

23. Semmy Purewal, ‘Learning Web App Development’, O'Reilly Media, February 2014, 306 p.

24. Ethan Brown, ‘Web Development with Node and Express’, O'Reilly Media, July 2014, 336 p.

25. V. Tkachenko, ‘IoT - modern telecommunication technologies’, August, 2016. Available at: http://www.lessons-tva.info/articles/net/013.html. [accessed May 3, 2018].

26. Jiafu Wan, Min Chen, Feng Xia, Di Li, and Keliang Zhou, “From Machine-to-Machine Communications towards Cyber-Physical Systems”, Computer Science and Information Systems, Vol. 10, No. 3, pp. 1105-1128, 2013

27. M. Selinger, A. Sepulveda, and J. Buchan, ‘Education and the internet of everything: How ubiquitous connectedness can help transform pedagogy’, Cisco Consulting Service and Cisco EMEAR Education Team, Oct., 2013. Available at: https://www.cisco.com/c/dam/en_us/solutions/industries/docs/education/education_internet.pdf. [accessed May 3, 2018].

28. Dominique Guinard, ‘What is the Web of Things’? April 8, 2017. Available at: https://webofthings.org/2017/04/08/what-is-the-web-of-things/. [accessed May 3, 2018].

29. Fleisch, Elgar, ‘What is the Internet of Things’? An Economic Perspective. Economics, Management, and Financial Markets, 5 (2), pp. 125-157, ISSN 1842-3191, January, 2010.

30. Luigi Atzori, Antonio Iera, Giacomo Morabito, ‘The Internet of Things: A survey’, Computer Networks, Vol. 54, Issue 15, pp. 2787-2805, 28 October, 2010.

31. R. Parashar, A. Khan, and Neha, ‘A survey: The internet of things’, International Journal of Technical Research and Applications, vol. 4, Issue 3, pp. 251-257, May-June, 2016.

32. Pallavi Sethi and Smruti R. Sarangi, ‘Internet of Things: Architectures, Protocols, and Applications’, Journal of Electrical and Computer Engineering, Vol. 2017, Article ID 9324035, January, 2017, 25 p.

33. Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, Wei Zhao, ‘A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications’, IEEE Internet of Things Journal, Vol, 4, Issue 5, pp**,** 1125 – 1142, October, 2017.

34. DDS the proven data connectivity standard for the IoT, ‘What is DDS’? (n.d.). Available at: http://portals.omg.org/dds/what-is-dds-3. [accessed May 3, 2018].

35. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, Moussa Ayyash, ‘Internet of Things: A Survey on Enabling Technologies, Protocols and Applications’, IEEE Communications Surveys & Tutorials, Vol. 17(4): Fourthquarter 2015, pp. 2347 – 2376, November, 2015.

36. Sathish Kumar and Dhiren Patel, ‘A Survey on Internet of Things: Security and Privacy Issues’, International Journal of Computer Applications 90(11), pp. 20-26, March, 2014.

37. O. Garcia-Morchon, S. Kumar, M. Sethi, ‘State-of-the-Art and Challenges for the Internet of Things Security draft-irtf-t2trg-iot-seccons-14’, April, 21, 2018. Available at: https://datatracker.ietf.org/doc/draft-irtf-t2trg-iot-seccons/. [accessed May 3, 2018].

38. OWASP Internet of Things Project, ‘IoT Security Guidance’. (n.d.). Available at: https://www.owasp.org/index.php/IoT_Security_Guidance. [accessed May, 3, 2018].

39. Oxford e-Research Centre, ‘The Internet of Things: realising the potential of a trusted smart world’. (n.d.). Available at: http://www.oerc.ox.ac.uk/news/Centre-contribution-IoT-reports. [accessed May 3, 2018].

40. A. Kamilaris, S. Yumusak, and M. I. Ali, ‘WOTS2E: A search engine for a Semantic Web of Things’, 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, pp. 436-441, 2016.

41. Dominique Guinard, Vlad Trifa, and Erik Wilde, ‘A Resource Oriented Architecture for the Web of Things’. In Proc. of the 2nd International

Conference on the Internet of Things (IoT 2010), LNCS, Tokyo, Japan, Springer Berlin / Heidelberg, November, 2010.

42. Dominique Guinard, Vlad Trifa, Erik Wilde, ‘Architecting a Mashable Open World Wide Web of Things’, Technical Report No. 663, Department of Computer Science, ETH Zürich, February, 2010.

43. Yuchao Zhou, Suparna De, Wei Wang, Klaus Moessner, ‘Search Techniques for the Web of Things: A Taxonomy and Survey’, Sensors 16(5), 600; doi:10.3390/s16050600, Apr., 2016.

44. Andreas Kamilaris, Andreas Pitsillides, Francesc X. Prenafeta-Bold, Muhammad Intizar Ali, ‘A Web of Things based eco-system for urban computing - towards smarter cities’, Telecommunications (ICT) 2017 24th International Conference on, pp. 1-7, 2017.

45. V. Tkachenko, ‘Web of Things - IoT Network Service’, October 2017. Available at: http://www.lessons-tva.info/articles/net/014.html. [accessed May 3, 2018].

46. Ben Francis, “Building the Web of Things”, June, 2017. Available at: https://hacks.mozilla.org/2017/06/building-the-web-of-things. [accessed May 3, 2018].

47. EVRYTHNG Developer Hub, ‘Welcome’. (n.d.). Available at: https://developers.evrythng.com/docs. [accessed May 3, 2018].

48. S. Friedl, A. Popov, A. Langley, E. Stephan, 'Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension’, July, 2014. Available at: https://tools.ietf.org/html/rfc7301. [accessed May 3, 2018].

49. M. Peck, K. Igoe, ‘Suite B Profile for Datagram Transport Layer Security / Secure Real-time Transport Protocol (DTLS-SRTP) draft-peck-suiteb-dtls-srtp-02’, March, 2013. Available at: https://tools.ietf.org/html/draft-peck-suiteb-dtls-srtp-02. [accessed May 3, 2018].

50. Eugene Ferry, John O’Raw, Kevin Curran, ‘Security Evaluation of the OAuth 2.0 Framework’, Information Management and Computer Security, Vol. 23, Iss. 1, pp. 73, October, 2014.

51. OAuth Community Site, ‘OAuth 2.0’. (n.d.). Available at: https://oauth.net/2/. [accessed May 3, 2018].

52. T. Borgohain, A Borgohain, U Kumar, S Sanyal, ‘Authentication systems in Internet of Things’, arXiv preprint arXiv:1502.00870, 2015.

53. Elena Reshetova, Michael McCool, ‘Web of Things (WoT) Security and Privacy Considerations’, 14 December, 2017. Available at: https://www.w3.org/TR/wot-security/. [accessed May 3, 2018].

54. WebStorm, ‘The smartest JavaScript IDE. Powerful IDE for modern JavaScript development’. (n.d.). Available at: http://www.jetbrains.com/webstorm/. [accessed May 3, 2018].

55. Sublime text, 'A sophisticated text editor for code, markup and prose'. (n.d.). Available at: https://www.sublimetext.com/. [accessed May 3, 2018].

56. Git, 'Git-local-branching-on-the-cheap'. (n.d.). Available at: https://git-scm.com/. [accessed May 3, 2018].

57. Resources for developers, 'HTML5 - Web developer guides'. (n.d.). Available at: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5. [accessed May 3, 2018].

58. World Wide Web Consortium (W3C), 'Extensible Markup Language (XML)', October 2016. Available at: https://www.w3.org/XML/. [accessed May 3, 2018].

59. Resources for developers, 'CSS3 - CSS: Cascading Style Sheets'. (n.d.). Available at: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3. [accessed May 3, 2018].

60. Bootstrap, 'The most popular HTML, CSS, and JS library in the world'. (n.d.). Available at: https://getbootstrap.com/. [accessed May 3, 2018].

61. Resources for developers, 'JavaScript'. (n.d.). Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript. [accessed May 3, 2018].

62. JavaScript, 'Ready to try JavaScript'? (n.d.). Available at: https://www.javascript.com/. [accessed May 3, 2018].

63. jQuery, 'What is jQuery'? (n.d.). Available at: https://jquery.com/. [accessed May 3, 2018].

64. Resources for developers, 'Ajax'. (n.d.). Available at: https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX. [accessed May 3, 2018].

65. Websocket.org - Powered by Kaazing, 'About HTML5 WebSocket'. (n.d.). Available at: http://websocket.org/aboutwebsocket.html. [accessed May 3, 2018].

66. Socket.IO, 'Socket.io 2.0 is here. Featuring the fastest and most reliable real-time engine'. (n.d.). Available at: https://socket.io/. [accessed May 3, 2018].

67. World Wide Web Consortium (W3C), 'The WebSocket API'. (n.d.). Available at: https://www.w3.org/TR/websockets/. [accessed May 3, 2018].

68. Node.js®, 'Node.js v8.11.1 Documentation'. (n.d.). Available at: https://nodejs.org/en/. [accessed May 3, 2018].

69. MongoDB for GIANT Ideas, 'MongoDB Atlas. Database as a Service'. (n.d.). Available at: https://www.mongodb.com/. [accessed May 3, 2018].

70. Alan B. Johnston, Daniel C. Burnett, ‘WebRTC: APIs and RTCWEB Protocols of the HTML5 Real - Time Web’, St. Louis, USA: Digital Codex LLC, Smashwords Edition, 2013, 247 p.

71. Microsoft Azure, ‘What is cloud computing’? (n.d.). Available at: https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/. [accessed May 3, 2018].

72. Service Architecture, ‘Service-Oriented Architecture (SOA) Definition’. (n.d.). Available at: https://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html. [accessed May 3, 2018].

73. EVERYTHNG Developer Hub, ‘Standard API Introduction’. (n.d.). Available at: https://developers.evrythng.com/v3.0/reference. [accessed May 3, 2018]

3. STANDARDS AND METRICS OF IOT BASED SYSTEMS

Dr., Assoc. Prof. M. O. Kolisnyk,
DrS. Prof. V.S. Kharchenko (KhAI)

Contents

Abbreviations

API — Application Program Interface
DSRC — Dedicated Short-Range Communication
FPGAs — Field-Programmable Gate Array
HTTP — Hyper-Text Transfer Protocol
IoT — Internet of Things
IoT-A — Internet of Things-Architecture
IIoT — Industrial Internet of Things
ITU-T — International Telecommunication Union sector Telecommunications
LoWPAN_IPHC — Low-Power Wireless Personal Area Network
LWM2M — LightweightM2M
M2M — Machine-To-Machine
MCU — microcontrollers
NBMA — Non-Broadcast Multiple Access
OS — Operating System
PLC — Programmable Logic Controllers
PPDR — Public Protection Disaster Relief
PS — Public Safety
REST — Representational State Transfer
RTU — Remote Terminal Units
SCADA — Supervisory Control Computers
SG — Smart Grid
SOCs — System-on-a-chip
SOA — Service-oriented Architecture
XML — eXtensible Markup Language

3.1 Standards overview and harmonization in IoT context

3.1.1 IoT standards classification

IoT is widely used in multiple applications such as are home monitoring, HealthCare, smart cities control, smart devices, smart vehicles and smart grid [1].

Modern IoT technologies have been created under a totally different scenario [2,3]:

1) Internet and cellular networks have become the world standard, with very high levels of coverage, reliability and availability;

2) smaller and smarter devices are constantly hit the industrial and consumer markets, to better understand and present the new after-sales and remote-controlled services;

3) software development and system interoperability standards such as XML, web services and SOA are converging to create fertile ground for M2M communications technologies, that makes it easier to use them in a variety of industries.

The current state of the IoT is characterized by a diverse set of initiatives, standards and implementations. Standardization and interoperability remain a challenge. Initial applications have been developed in vertical domains like logistics or energy, with their own protocols and architectures. Current efforts are converging into standard specifications like OMA LWM2M and reference architectures like the European Internet of Things-Architecture (IoT-A). These efforts are bringing together key players and enabling implementations that are reusable across different application domains [2]. Connected things all share five key components: the need for smarter power consumption, storage, and management; the need for stronger safeguards for privacy and security; high-performance microcontrollers (MCUs); sensors and actuators; and the ability to communicate. Without them, all governed by standards, there will be no IoT [3].

IoT standards are still a work in progress. Standards for IoT Systems are divided into: common things for the internet (they differ, as each standardization organization offers its own definition), then there is a division according to the standards for each kind of IoT (Smart Transport, Smart Hospital, Smart City, Smart Office etc.). Then there is a division into standards for networks of information transfer between devices of the IoT, and standards describing rules of information transfer (protocols). Also standards divided into standards for hardware and software of IoT-based systems.

Basic standards for IoT:

1. ISO/IEC AHG1 produced the following definition of IoT which was adopted by SWG 5 [4]:

- "An infrastructure of interconnected objects, people, systems and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react."

2. IEEE Internet of Things (IoT) Ecosystem Study:

- **Internet of Things (IoT)** is a system consisting of networks of sensors, actuators, and smart objects whose purpose is to interconnect "all" things, including everyday and industrial objects, in such a way as to make them intelligent, programmable, and more capable of interacting with humans and each other.

- The **IoT paradigm** promises to make any electronic devices part of the Internet environment. This new paradigm opens the doors to new innovations and interactions between people and things that will enhance the quality of life and utilization of scarce resources [3].

- The **IoT** is a concept and a paradigm. It considers pervasive presence in an environment of things/objects that interact with each other and cooperate with other things/objects in order to allow/provide new applications/services and reach common goals [4].

The **IoT** has been defined in Recommendation ITU-T Y.2060 (06/2012) as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

Home Plug Powerline communication technology (IEEE 1901 and IEEE 1901.2) transforms any power line into a communication cable. Wi-Fi (IEEE 802.11.x) offers a convenient form of connectivity for our smartphones and tablets. ZigBee (IEEE 802.15.4) is well known for its role in home automation.

Recommendation provides:

1) a basic model for updating IoT software/firmware;

2) a common update procedure (sequences) for IoT firmware (including software);

3) the requirements and capabilities for updating IoT firmware. A common software/firmware update procedure is defined with general requirements. With these, IoT secure updates can be securely implemented in common among stakeholders in IoT context comprising IoT device developer and IoT system/service providers. The network architecture of IoT devices may differ, but four functional entities are required in all the cases, i.e., Device core, Communicator, Status tracker, and Firmware server. A Device core stores and uses firmware on an IoT device. A Communicator checks the - 2 - SG17-LS084 firmware status of the IoT device and initiate firmware update procedure upon needed. A Status tracker keeps tabs on the status of IoT devices under its administration. For instance, it checks the list of IoT devices that has already completed the update. A Firmware server distributes firmware packages. The list of capabilities of these functional entities are elaborated in the later section. A Device core communicates with a Communicator; multiple Communicators communicate with a Status tracker, which may communicate with multiple Firmware servers [5].

3.1.2 ISO/IEC standards

IEC 61131-3:2013 specifies the syntax and semantics of a unified suite of programming languages for programmable controllers (PCs). This suite consists of two textual languages, Instruction List (IL) and Structured Text (ST), and two graphical languages, Ladder Diagram (LD) and Function Block Diagram (FBD). It includes the following significant technical changes: main extensions are new data types and conversion functions, references, name spaces and the object oriented features of classes and function blocks.

ISO/IEC 14543-3-10:2012(E) Information technology - Home electronic systems (HES) architecture - Part 5-11: Intelligent Grouping and Resource Sharing for HES Class 2 and Class 3 - Remote user interface specifies a wireless protocol for low-powered devices such as energy harvested devices in a home environment. This wireless protocol is specifically designed to keep the energy consumption of such sensors and switches extremely low. The WSP protocol system consists of two and optionally three types of components that are specified in this standard. These are the transmitter, the receiver and optionally the repeater. Repeaters are needed when the transmitter and the receiver are located in such a way that no good direct communication between them can be established.

ISO/IEC 29177:2016 Information technology - prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 31, Automatic identification and data capture techniques, in collaboration with ITU-T. The identical text is published as ITU-T H.642.3 (06/2012). Automatic identification and data capture technique - Identifier resolution protocol for multimedia information access triggered by tag-based identification [6].

ISO/IEC JTC 1/SC 41 Scope - standardization in the area of Internet of Things and related technologies.

1. Serve as the focus and proponent for JTC 1's standardization programme on the Internet of Things and related technologies, including Sensor Networks and Wearables technologies.

2. Provide guidance to JTC 1, IEC, ISO and other entities developing Internet of Things related applications.

IEC 62443 Cyber Security of IoT. General group of standards:

IEC 62443-1-1 – concepts &models used throughout of series.

IEC 62443-1-2 - master glossary of terms & abbreviations.

IEC 62443-1-3 - describes a series of quantitive metrics (draft).

IEC 62443-1-4 – life cycle for IACS security.

Policies & Procedures

IEC 62443-2-1 - requirements to define & implement an effective IACS cyber security management systems.

IEC 62443-2-2 - specific guidance (draft).

IEC 62443-2-3 - guidance on the specific subject of patch management to IACS.

IEC 62443-2-4 - requirements for suppliers of IACS.

System Requirements

IEC 62443-3-1 - application of various security technologies to an IACS.

IEC 62443-3-2 - security risk assessment & system design for IACS.

IEC 62443-3-3 - the foundational system security assurance levels.

Components Requirements:

IEC 62443-4-1 - the derived requirements that are applicable to the development of products.

IEC 62443-4-2- sets of derived requirements that provide a detailed mapping of the system requirements to subsystems components of the system under consideration.

The IEC 61508 standard provides for the direct involvement of process personnel in the provision of safety functions, which determines the training requirements and qualifications of professionals who determine the level of safety requirements for a particular process.

3.1.3 ITU-T standards

ITU-T standards for IoT [7]:

ITU-T SG 2 Operational aspects of service provision and telecommunications management – numbering, naming, addressing.

ITU-T SG 3 Tariff and accounting principles including related telecommunication economic and policy issues.

ITU-T SG 5 Environment and climate change.

ITU-T SG 9 Television and sound transmission and integrated broadband cable networks.

ITU-T SG 11 Signalling requirements, protocols and test specifications – testing architecture for tag-based identification systems and functions.

ITU-T SG 12 Performance, QoS and QoE.

ITU-T SG 13 Future networks including mobile and NGN, Y.2016 – Functional requirements and architecture of the NGN for applications and services using tag-based identification.

ITU-T SG 15 Optical transport networks and access network infrastructures.

ITU-T SG 16 Multimedia coding, systems and applications requirements and architecture for multimedia information access triggered by tag-based ID, H.6621 – architecture of a system for multimedia information access trigged by tag-based identification, F.771 – Service description and requirements for multimedia information access trigged by tag-based identification.

ITU-T SG 17 Security – security and privacy of tag-based applications. ITU-T SG 17 would like to congratulate the establishment of the new working group on Software Updates for Internet of Things, i.e., SUIT WG.

X.1171 – Threats and requirements for protection of personally identifiable information in applications using tag-based identification.

Recommendation ITU-T Y.4114 "Specific requirements and capabilities of the IoT for Big Data" [8]. This Recommendation complements the developments on common requirements of the IoT and functional framework of the IoT in terms of the specific requirements and capabilities that the IoT is expected to support in order to address the challenges related to Big Data. Recommendation ITU-T Y.4114 specifies requirements and capabilities of the Internet of things (IoT) for big data. This Recommendation also constitutes a basis for further standardization work such as functional entities, application programming interfaces (APIs) and protocols concerning big data in the IoT.

Recommendation ITU-T Y.4500.1 "oneM2M – Functional architecture". This Recommendation describes the end-to-end oneM2M functional architecture, including the description of the functional entities and associated reference points. OneM2M functional architecture focuses on the service layer aspects and takes underlying network-independent view of the end-to-end services. The underlying network is used for the transport of data and potentially for other services.

Recommendation ITU-T Y.4201 “High-level requirements and reference framework of smart city platforms”. Recommendation ITU-T Y.4201 presents the high-level requirements and reference framework of smart city platforms (SCPs). The SCP is a fundamental platform supporting all the services and applications of a smart city, with the objective to improve quality of life, provide urban operation and services for the benefit of citizens while ensuring city sustainability [9].

Most of the ITU-T SGs have responsibilities for standardizing specific security aspects (TMN security, IPCablecom security, future networks security, multimedia security, disaster management, electromagnetic environment and climate change security issues, etc.)

ITU-T SG 17 provides security coordination within ITU-T SGs, ITU sectors and externally with the ISO/IEC JTC 1/SC 27, ETSI, IETF, Liberty Alliance/Kantara Initiative, FIDIS, OASIS and others through SAG-S, 5thETSI Security Workshop, 20-22 January 20105of 20 Alliance/Kantara Initiative, FIDIS, OASIS and others through SAG-S, projects, workshops, JCA-IdM, JCA-CIT, LSs, common texts of Recommendations, etc.

Most of the ITU-T SGs have responsibilities for standardizing specific security aspects (TMN security, IPCablecom security, future networks security, multimedia security, disaster management, electromagnetic environment and climate change security issues, etc.).

ITU-T Y.4113 “Requirements of the network for the Internet of Things”

ITU-T Y.4114 “Specific requirements and capabilities of the IoT for Big Data”

ITU-T Y.4115 “Reference architecture for IoT device capability exposure"

ITU-T Y.4451 “Framework of constrained device networking in the IoT environments”

ITU-T Y.4452 “Functional framework of Web of Objects”

ITU-T Y.4453 “Adaptive software framework for IoT devices”

ITU-T Y.4553 “Requirements of smartphone as sink node for IoT applications and services”

ITU-T Y.4702 “Common requirements and capabilities”

ITU-T Y.4805 “Identifier service requirements for the interoperability of Smart City applications”.

ITU-T Y.Supp.42 to ITU-T Y.4100 series.

“Use cases of User-Centric work Space (UCS) Service:

▪ ITU-T Y.Supp.34 to ITU-T Y.4000 series "Smart Sustainable Cities - Setting the stage for stakeholders' engagement"

▪ ITU-T Y.Supp.33 to ITU-T Y.4000 series "Smart Sustainable Cities - Master plan"

▪ ITU-T Y.Supp.32 to ITU-T Y.4000 series "Smart sustainable cities - a guide for city leaders“

▪ ITU-T Y.Supp.31 to ITU-T Y.4550 series "Smart Sustainable Cities - Intelligent sustainable

buildings“.

ITU-T Y.Supp.28 to ITU-T Y.4550 series ”Integrated management for smart sustainable cities”;

ITU-T Y.Supp.29 to ITU-T Y.4250 series “Multi-service infrastructure for smart sustainable cities in new-development areas”;

ITU-T Y.Supp.30 to ITU-T Y.4250 series “Overview of smart sustainable cities infrastructure”

ITU-T Y.Supp.27 to ITU-T Y.4400 series “Setting the framework for an ICT architecture of a smart sustainable city”.

Under study:

Y.Accessibility-IoT - Accessibility requirements for the Internet of things applications and services.

Y.del-fw - Framework of delegation service for the IoT devices.

Y.IoT-DA-Counterfeit - Information Management Digital Architecture to combat counterfeiting in IoT.

Y.IoT-Interop - An Interoperability framework for IoT.

Y.IoT-IoD-PT - Identity of IoT devices based on secure procedures and ensures privacy and trust of IoT systems.

Y.ODI - Open Data Indicator in smart cities.

Y.smartport – Requirement of smart managements of supply services in smart port.

Y.frame-scc - Framework and high-level requirements of smart cities and communities.

Y.fsn - Framework and Service scenarios for Smartwork.

ITU-T Study Group 16 initiated activities in this area in 2006 and already published two Recommendations:

- ITU-T F.771: Service description and requirements for multimedia information access triggered by tag-based identification.
- ITU-T H.621: Architecture of a system for multimedia information access triggered by tag-based identification.

ITU-T Study Group 16 is developing three draft Recommendations jointly with ISO/IEC JTC 1/SC 31:

- ITU-T H. IDscheme | ISO/IEC 29174-1: Information technology – UII scheme and encoding format for Mobile AIDC services – Part 1: Identifier scheme for multimedia information access triggered by tag-based identification.
- ITU-T H.ID-RA | ISO/IEC 29174-2: Information technology – UII scheme and encoding format for Mobile AIDC services – Part 2: Registration procedures.
- ITU-T H.IRP | ISO/IEC 29177: Information technology – Automatic identification and data capture technique – Identifier resolution protocol for multimedia information access triggered by tag-based identification.
- ITU-T SG 16 is working in close collaboration with ITU-T Study Group 17 to use the OID Resolution System in H.IRP. The work is coordinated by ITU-T SG 16 is working in close collaboration with ITU-T Study Group 17 to use the OID Resolution System (ITU-T X.672 | ISO/IEC 29168-1) in H.IRP **[**10**].**
- ITU-T FG M2M - Machine-to-machine (M2M) communication is considered to be a key enabler of applications and services across a broad range of vertical markets (e.g., health-care, logistics, transport, utilities, etc.). A common M2M service layer, agreed at the global level involving stakeholders from the M2M and vertical market communities, would provide a cost-efficient platform, which can be easily deployed in hardware and software, in a multi-vendor environment, and across sectors. FG M2M identified a minimum set of common requirements of vertical markets, focusing initially on the health-care market and application programming interfaces (APIs) and protocols supporting e-health applications and services, and drafted technical reports in these areas.

ITU-T H.621 defines the system architecture for the multimedia information access triggered by tag-based identification on the basis of Recommendation ITU-T F.771, and serves as a technical introduction to subsequent definition of detailed system components and protocols. The services treated provide the users with a new method to refer to the multimedia content without typing its address on a keyboard or inputting the name of objects about which relevant information is to be retrieved. This is one of the major communication services using identification (ID) tags such as radio frequency identifications (RFIDs), smart cards and barcodes. It contains the functional model, its constituent components as well as its workflow.

ITU-T Y.2213 describes high-level service requirements and NGN capability requirements needed to support applications and services using tag-based identification. Several examples of applications and services using tag-based identification are also described with scenarios. The scope of this Recommendation is limited to applications and services using tag-based identification and they are distinguished by the following three mandatory elements: ID tag, ID terminal and identifier.

3.1.4 IEEE Standards

IEEE P2413, "IEEE Standard for an Architectural Framework for the Internet of Things (IoT)".

IEEE 802.15.4 defines four types of frames: beacon frames, MAC command frames, acknowledgement frames, and data frames. IPv6 packets MUST be carried on data frames. Data frames may optionally request that they be acknowledged. In keeping with [11], it is recommended that IPv6 packets be carried in frames for which acknowledgements are requested so as to aid link-layer recovery. Each of these efforts shares a mutual goal: to take the multitude of discrete communications, processing, programming, and other protocols and approaches now competing to clog the IoT and turn them into a single, unified approach to developing its underlying foundational systems and infrastructures [12].

IEEE 802.24 Vertical Applications TAG focuses on application categories that use IEEE 802 technology and are of interest to multiple IEEE 802 WGs and have been assigned to IEEE 802.24 by the IEEE Executive Committee.

For those application categories, IEEE 802.24:

– Acts as a liaison and point of contact with regulatory agencies, industry organizations, other SDOs, government agencies, IEEE societies, etc., for questions regarding the use of 802 standards in those emerging applications.

– Develops white papers, guidelines, presentations and other documents that do not require a PAR that describe the application of

802 standards to those emerging applications – Acts as a resource for understanding 802 standards for certification efforts by industry bodies [13].

A great example of the convergence specified in a fairly recent IEEE 1905.1 standard.

Below is a partial listing of IEEE standards related to the Internet of Things.

IEEE 802.1AS™-2011 - IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.

IEEE 802.1Q™-2011 - IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks.

IEEE 802.3™-2012 - IEEE Standard for Ethernet.

IEEE 802.3.1™-2011 - IEEE Standard for Management Information Base (MIB) Definitions for Ethernet.

IEEE 802.11™-2012 - IEEE Standard for Information Technology - Telecommunications and information exchange between systems-Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking.

IEEE 802.11ad™-2012 - IEEE Standard for Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band.

IEEE 802.15.1™-2005 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).

IEEE 802.15.2™-2003 - IEEE Recommended Practice for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.2: Coexistence of Wireless Personal

Area Networks With Other Wireless Devices Operating in Unlicensed Frequency Bands.

IEEE 802.15.3™-2003 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs) Amendment 1: Mac Sublayer.

IEEE 802.15.3c™-2009 - IEEE Standard for Information technology- Local and metropolitan area networks- Specific requirements- Part 15.3: Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension.

IEEE 802.15.4™-2011 - IEEE Standard for Local and metropolitan area networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs).

IEEE 802.15.4e™-2012 - IEEE Standard for Local and metropolitan area networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer.

IEEE 802.15.4f™-2012 - IEEE Standard for Local and metropolitan area networks- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 2: Active Radio Frequency Identification (RFID) System Physical Layer (PHY).

IEEE 802.15.4g™-2012 - IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks.

IEEE 802.15.4j™-2013 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs) Amendment: Alternative Physical Layer Extension to support Medical Body Area Network (MBAN) services operating in the 2360-2400 MHz band.

IEEE 802.15.5™-2009 - IEEE Recommended Practice for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs).

IEEE 802.15.6™-2012 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 15.6: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)used in or around a body.

IEEE 802.15.7™-2011 - IEEE Standard for Local and Metropolitan Area Networks-Part 15.7: Short-Range Wireless Optical Communication Using Visible Light.

IEEE 802.16™-2012 - IEEE Standard for Air Interface for Broadband Wireless Access Systems.

IEEE 802.16p™-2012 - IEEE Standard for Air Interface for Broadband Wireless Access Systems Amendment: Enhancements to Support Machine-to-Machine Applications.

IEEE 802.16.1b™-2012 - IEEE Standard for WirelessMAN-Advanced Air Interface for Broadband Wireless Access Systems - Amendment: Enhancements to Support Machine-to-Machine Applications.

IEEE 802.22™-2011 - IEEE Standard for Information Technology-Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN)-Specific requirements Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands.

IEEE 802.22.1™-2010 - IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 22.1: Standard to Enhance Harmful Interference Protection for Low-Power Licensed Devices Operating in TV Broadcast Bands.

IEEE 802.22.2™-2012 - IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 22.2: Installation and Deployment of IEEE 802.22 Systems.

IEEE 1284™-2000 - IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers.

IEEE 1285™-2005 - IEEE Standard for Scalable Storage Interface (S/SUP 2/I).

IEEE 1301.3™-1992 - IEEE Standard for a Metric Equipment Practice for Microcomputers - Convection-Cooled With 2.5mm Connectors.

IEEE 1377™-2012 - IEEE Standard for Utility Industry Metering Communication Protocol Application Layer (End Device Data Tables).

IEEE 1394™-2008 - IEEE Standard for a High-Performance Serial Bus.

IEEE 1451.0™-2007 - IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats.

IEEE 1547™-2003 - IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems.

IEEE 1547.1™-2005 - IEEE Standard Conformance Test Procedures for Equipment Interconnecting Distributed Resources with Electric Power Systems.

IEEE 1547.2™-2008 - IEEE Application Guide for IEEE Std 1547™, IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems.

IEEE 1547.3™-2007 - IEEE Guide for Monitoring, Information Exchange, and Control of Distributed Resources Interconnected with Electric Power Systems.

IEEE 1547.4™-2011 - IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems with Electric Power Systems.

IEEE 1547.6™-2011 - IEEE Recommended Practice for Interconnecting Distributed Resources with Electric Power Systems Distribution Secondary Networks.

IEEE 1609.2™-2013 - IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages.

IEEE 1609.3™-2010 - IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services.

IEEE 1609.4™-2010 - IEEE Standard for Wireless Access in Vehicular Environments (WAVE)--Multi-channel Operation.

IEEE 1609.11™-2010 - IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-- Over-the-Air Electronic Payment Data Exchange Protocol for Intelligent Transportation Systems (ITS).

IEEE 1609.12™-2012 - IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Identifier Allocations.

IEEE 1675™-2008 - IEEE Standard for Broadband Over Powerline Hardware 1900.1-2008 IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management.

IEEE 1701™-2011 - IEEE Standard for Optical Port Communication Protocol to Complement the Utility Industry End Device Data Tables.

IEEE 1702™-2011 - IEEE Standard for Telephone Modem Communication Protocol to Complement the Utility Industry End Device Data Tables.

IEEE 1703™-2012 - IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to complement the Utility Industry End Device Data Tables.

IEEE 1775™-2010 - IEEE Standard for Power Line Communication Equipment-Electromagnetic Compatibility (EMC) Requirements-Testing and Measurement Methods.

IEEE 1815™-2012 - IEEE Standard for Electric Power Systems Communications - Distributed Network Protocol (DNP3) 2200-2012 IEEE Standard Protocol for Stream Management in Media Client Devices.

IEEE 1888™-2011 - IEEE Standard for Ubiquitous Green Community Control Network Protocol.

IEEE 1900.1™-2008 - IEEE Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management.

IEEE 1900.2™-2008 - IEEE Recommended Practice for the Analysis of In-Band and Adjacent Band Interference and Coexistence Between Radio Systems.

IEEE 1900.4™-2009 - IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks.

IEEE 1900.4a™-2011 - IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks Amendment 1: Architecture and Interfaces for Dynamic Spectrum Access Networks in White Space Frequency Bands.

IEEE 1901™-2010 - IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications.

IEEE 1902.1™-2009 - IEEE Standard for Long Wavelength Wireless Network Protocol.

IEEE 1905.1™-2013 - IEEE Draft Standard for a Convergent Digital Home Network for Heterogeneous Technologies.

IEEE 2200™-2012 - IEEE Standard Protocol for Stream Management in Media Client Devices.

IEEE 2030™-2011 - IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads

IEEE 2030.5™-2013 - IEEE Adoption of Smart Energy Profile 2.0 Application Protocol Standard.

IEEE 11073-00101™-2008 - IEEE Standard for Health Informatics - PoC Medical Device Communication - Part 00101: Guide--Guidelines for the Use of RF Wireless Technology.

IEEE 11073-10102™-2012 - IEEE Standard for Health informatics - Point-of-care medical device communication - Nomenclature - Annotated ECG.

IEEE 11073-10103™-2012 - IEEE Standard for Health informatics - Point-of-care medical device communication - Nomenclature - Implantable device, cardiac.

IEEE 11073-10201™-2004 - IEEE Standard for Health Informatics - Point-Of-Care Medical Device Communication - Part 10201: Domain Information Model.

IEEE 11073-10404™-2010 - IEEE Standard for Health informatics-Personal health device communication Part 10404: Device specialization-Pulse oximeter.

IEEE 11073-10406™-2011 - IEEE Standard for Health informatics-Personal health device communication Part 10406: Device specialization-Basic electrocardiograph (ECG) (1- to 3-lead ECG).

IEEE 11073-10407™-2010 - IEEE Standard for Health informatics Personal health device communication Part 10407: Device specialization Blood pressure monitor.

IEEE 11073-10408™-2010 - IEEE Standard for Health informatics Personal health device communication Part 10408: Device specialization Thermometer.

IEEE 11073-10415™-2010 - IEEE Standard for Health informatics Personal health device communication Part 10415: Device specialization Weighing scale 11073-10420-2010 IEEE Standard for Health informatics - Personal health device communication Part 10420: Device specialization - Body composition analyzer

IEEE 11073-10417™-2011 - IEEE Standard for Health informatics Personal health device communication Part 10417: Device specialization Glucose meter.

IEEE 11073-10418™-2011 - IEEE Standard for Health informatics - Personal health device communication - Device specialization - International normalized ratio (INR) monitor.

IEEE 11073-10420™-2010 - IEEE Standard for Health informatics - Personal health device communication Part 10420: Device specialization - Body composition analyzer.

IEEE 11073-10441™-2008 - IEEE Standard for Health Informatics - Personal Health Device Communication - Part 10441: Device Specialization - Cardiovascular Fitness and Activity Monitor.

IEEE 11073-30300™-2004 - IEEE Standard for Health informatics - Point-of-care medical device communication - Transport profile – Infrared.

IEEE 11073-30400™-2010 - IEEE Standard for Health informatics-Point-of-care medical device communication Part 30400: Interface profile-Cabled Ethernet.

IEEE 14575™-2000 - IEEE Standard for Heterogeneous Interconnect (HIC) (Low-Cost, Low-Latency Scalable Serial Interconnect for Parallel System Construction).

IEEE 21450™-2010 - IEEE Standard for Information technology - Smart transducer interface for sensors and actuators - Common functions, communication protocols, and Transducer Electronic Data Sheet (TEDS) formats.

IEEE 21451-1™-2010 - IEEE Standard for Information technology - Smart transducer interface for sensors and actuators --Part 1: Network Capable Application Processor (NCAP) information model.

IEEE 21451-2™-2010 - IEEE Standard for Information technology - Smart transducer interface for sensors and actuators - Part 2: Transducer to microprocessor communication protocols and Transducer Electronic Data Sheet (TEDS) formats.

IEEE 21451-4™-2010 - IEEE Standard for Information technology - Smart transducer interface for sensors and actuators - Part 4: Mixed-mode communication protocols and Transducer Electronic Data Sheet (TEDS) formats.

IEEE 21451-7™-2011 - IEEE Standard for Smart Transducer Interface for Sensors and Actuators-Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats.

IEEE P2413 is to develop a standard for the architectural framework for the Internet of Things, which includes descriptions of various IoT domains, definitions of IoT domain abstractions, and identification of commonalities between different IoT domains. The architectural framework defined in this standard will promote cross-domain interaction, aid system interoperability and functional compatibility.

Harmonization and security of IoT:

IEEE 1451-99 is focused on developing a standard for harmonization of Internet of Things (IoT) devices and systems. This standard defines a method for data sharing, interoperability, and security of messages over a network, where sensors, actuators and other devices can interoperate, regardless of underlying communication technology.

Sensor Performance and Quality: Sensors are fundamental to IoT ecosystem with large volume of different sensors integrated into a complex framework. IEEE 2700 proposes a common framework for sensor performance specification terminology, units, conditions and limits is provided.

IEEE P2510-2018 defines quality measures, controls, parameters and definitions for sensor data related to Internet of Things (IoT) implementations.

3.1.5 Other standards

FG M2M - Machine-to-machine (M2M) communication is considered to be a key enabler of applications and services across a broad range of vertical markets (e.g., health-care, logistics, transport, utilities, etc.). A common M2M service layer, agreed at the global level involving stakeholders from the M2M and vertical market communities, would provide a cost-efficient platform, which can be easily deployed in hardware and software, in a multi-vendor environment, and across sectors. The Focus Group on the M2M service layer (FG M2M) studied activities undertaken by various standards developing organizations in the field of M2M service layer specifications to identify key requirements for a common M2M service layer. Identified a minimum set of common requirements of vertical markets, focusing initially on the health-care market and application programming interfaces (APIs) and protocols supporting e-health applications and services, and drafted technical reports in these areas.

EN 50090-1:2011 Home and Building Electronic Systems (HBES) - Part 1: Standardization structure. This European Standard concentrates on control applications for Home and Building HBES Open Communication System and covers any combination of electronic devices linked via a digital transmission network. Home and Building Electronic System as provided by the HBES Open Communication System is a specialized form of automated, decentralized and distributed process control, dedicated to the needs of home and building applications. The EN 50090 series concentrates on HBES Open Communication System Class 1 and includes a specification for a communication network for Home and Building for example for the control of lighting, heating, food preparation, washing, energy management, water control, fire alarms, blinds control, different forms of security control, etc. This European Standard gives an overview of the features of the HBES Open Communication System and provides the reader with references to the different parts of EN 50090 series. This European Standard is used as a product family standard. It is not intended to be used as a stand-alone standard.

CoAP is an Internet Engineering Task Force (IETF) standard. The stable specification is defined in RFC 7252. CoAP aligns itself with the web paradigm by providing URIs to locate resources, Internet media types to describe them and a stateless mapping to HTTP verbs. CoAP has low overhead and is compatible with existing IP infrastructure with an UDP binding. Other bindings for SMS, TCP, TLS and Websockets exist as IETF drafts, with some of them being offered in implementations. The API of the message broker for Amazon AWS is supports two communication options [14]: MQTT 3.1.1; HTTP / REST API; MQTT over Websockets.

BS EN 61508-2:2002 - Functional safety of electrical/ electronic/ programmable electronic safety-related systems. Requirements for electrical/ electronic/ programmable electronic safety-related systems

UNE EN 61508-5:2011 - Functional Safety Of Electrical/Electronic/Programmable Electronic Safety-Related Systems - Part 5: Examples Of Methods For The Determination Of Safety Integrity Levels

BS EN 61508-4:2002 - Functional safety of electrical/ electronic/ programmable electronic safety-related systems. Definitions and abbreviations

ISA TR84.00.02-5:2002 - Safety Instrumented Functions (sif) - Safety Integrity Level (sil) Evaluation Techniques - Part 5: Determining The Pfd Of Sis Logic Solvers Via Markov Analysis

BS EN 61508-1:2002 - Functional safety of electrical/ electronic/ programmable electronic safety-related systems. General requirements

BS EN 61508-3:2002 - Functional safety of electrical/ electronic/ programmable electronic safety-related systems. Software requirements

NASA STD 8719.9:2002 - Standard for Lifting Devices And Equipment

PD R009-004:2001 - Railway specifications. Systematic allocation of safety integrity requirements

NFPA 318:2015 - Protection of Semiconductor Fabrication Facilities

NFPA 318:2006 - Protection of Semiconductor Fabrication Facilities

CEI 65-186 Ed. 1 (2010) - Guideline on The Application Of The Standard Series Cei En 61511 Functional Safety - Safety Instrumented Systems For The Process Industry Sector

08/30193470 DC - BS EN 61508-6. Functional safety of electrical/electronic/programmable electronic safety-related systems. Part 6. Guidelines on the application of IEC 61508-2 and IEC 61508-3.

SR CLC/TR 50451:2007 - Railway Applications - Systematic Allocation of Safety Integrity Requirements

CLC/TR 50451:2007 - Railway Applications - Systematic Allocation Of Safety Integrity Requirements

IEC 61508-2 Ed. 1.0 - Functional safety of electrical/electronic/programmable electronic safety-related systems Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems

NFPA 654:2013 - Prevention Of Fire And Dust Explosions From The Manufacturing, Processing, And Handling Of Combustible Particulate Solids

API 2350 Ed. 4 (2012) - Overfill Protection For Storage Tanks In Petroleum Facilities

ISA TR84.00.02-4:2002 - Safety Instrumented Functions (sif) - Safety Integrity Level (sil) Evaluation Techniques - Part 4: Determining The Sil Of A Sif Via Markov Analysis

PD CLC/TR 50451:2007 - Railway applications. Systematic allocation of safety integrity requirements

NFPA 318:2009 - Protection Of Semiconductor Fabrication Facilities

CEI CLC/TR 50451 Ed. 1 (2008) - Railway Applications - Systematic Allocation Of Safety Integrity Requirements

NFPA 318:2018 - Protection Of Semiconductor Fabrication Facilities

ISA TR84.00.02-1:2002 - Safety Instrumented Functions (sif) - Safety Integrity Level (sil) Evaluation Techniques - Part 1: Introduction

ISA TR84.00.04-1:2005 - Guidelines for the Implementation of Ansi/isa-84.00.01-2004.

NFPA 654:2006 - The Prevention of Fire and Dust Explosions from the Manufacturing, Processing, and Handling of Combustible Particulate Solids.

ISA TR84.00.02-3:2002 - Safety Instrumented Functions (sif) - Safety Integrity Level (sil) Evaluation Techniques - Part 3: Determining the Sil of a Sif Via Fault Tree Analysis.

ISA TR84.00.02-2:2002 - Safety Instrumented Functions (sif) - Safety Integrity Level (sil) Evaluation Techniques - Part 2: Determining the Sil of a Sif Via Simplified Equations

ISA TR91.00.02:2003 - Criticality Classification Guideline for Instrumentation.

BS EN 61508-6:2002 - Functional safety of electrical/ electronic/ programmable electronic safety-related systems. Guidelines on the application of IEC 61508-2 and IEC 61508-3.

PIP PCEDO001:2015 - Guidelines for Control Systems Documentation.

ISA TR12.21.01:2004 (R2013) - Use of Fiber Optic Systems In Class 1 Hazardous (classified) Locations.

CEI EN 61508-5 Ed. 2 (2011) - Functional Safety Of Electrical/Electronic/Programmable Electronic Safety related Systems - Part 5: Examples of Methods for the Determination of Safety Integrity Levels.

3.2 Metrics and measurement of attributes

Metrics, measurement, and metrology are different but related concepts that are essential for creating standards for physical systems, virtual systems, financial institutions, medical care, first responders, governance, and others. Metrics use measurement and other information to describe a product or process. Metrics can also have static or dynamic characteristics. Syntactic measures are static; semantic measures are usually dynamic. Environment and context give semantics to static syntax. Environment and context provide the notion of “dynamic.” As we mentioned with reliability, dynamic measures also include one other very important variable: time.

Fundamental characteristics of the IoT are as follows [15]:

– Interconnectivity: With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

– Things-related services: The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.

– Heterogeneity: The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

– Dynamic changes: The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

– Enormous scale: The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. The ratio of communication triggered by devices as compared to communication triggered by humans will noticeably shift towards device-triggered communication. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

3.2.1 IoT performance

A Reference Model is an abstract framework for understanding significant relationships among the entities of some environment [16].

The IoT Reference Model provides the highest abstraction level for the definition of the IoT Architectural Reference Model. Up to now, few standard committees have been researched in IoT reference model. Among them, the International Telecommunication Union (ITU) is one of the best organizations that proposed a comprehensive reference model in IoT environment. In this regard, an overview of the Internet of things (IoT) has provided by ITU-T Y.2060. It clarified the concept and scope of the IoT, identified the fundamental characteristics and high-level requirements of the IoT and described the IoT reference model [5]. The reference model tries to establish a common grounding for IoT architectures and IoT systems. The ITU recommended reference model for IoT. It is composed of four layers as well as management and security capabilities which are associated with the four layers.

IoT represents the convergence of several interdisciplinary domains [10-14]: networking, embedded hardware, radio spectrum, mobile computing, communication technologies, software architectures, sensing technologies, energy efficiency, information management, and data analytics. The rapid growth of IoT is driven by four key advances in digital technologies. Requirements for IoT reference architecture IoT is emerging as a major horizontal activity which will impact the work of many JTC 1 SCs.

The four layers are as follows [5]: Application layer; Service support and application support layer; Network layer; Device layer.

Application layer: which contains IoT applications.

Service and application support layer: consists of common capabilities which can be used by different IoT applications and various detailed capability groupings, in order to provide different support functions to different IoT applications.

Network layer: provides relevant control functions of network connectivity and IoT services and applications transportation.

Device layer: includes direct/indirect device interaction with the gateway and communication network.

Management capabilities: how to manage the devices, traffic and etc.

Security capabilities includes authorization, authentication, application data confidentiality and integrity protection, privacy protection, security audit, anti-virus and etc.

A Reference Architecture maps onto software elements that implements the functionality defined in the Reference Model [14]. Actually, the architectural elements of models are in the domain of the technologies, protocols, and products which used to implement the domain. A reference architecture tries to show the most complete picture of what is involved in realizing the modeled entities [4]. It is possible to define Reference Architectures at many levels of detail or abstraction and for many different purposes. Architecture handles requirements and forms a superset of functionalities, information structures, mechanisms and protocols [17].

IoT Reference Architecture Projects. Few proposals have been introduced so far in IoT Reference Architecture model. There are 4 IoT models: IoT Architectural Reference Model (IoT-A) proposed by European Commission (FP7); IoT Reference Architecture developed by the WSO2 company; Korean IoT Reference Model; Chinese IoT Reference Model.

IoT-A Architectural Reference Model

European Commission within the Seventh Framework Program (FP7) has supported the proposed project; IoT_A1 by Martin Bauer and et.al. The recommended reference architecture provided high-level architectural views and perspectives for constructing IoT systems presented in Fig. 3.1.

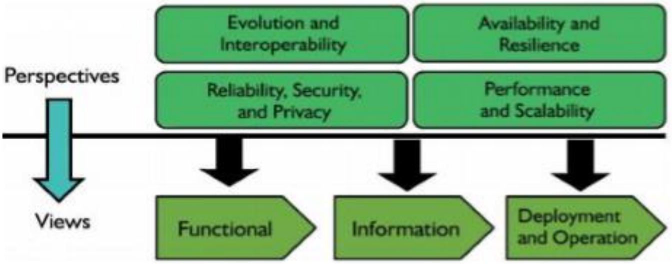

Evolution and Interoperability
Availability and Resilience
Perspectives
Reliability, Security, and Privacy
Performance and Scalability
Views
Functional
Information
Deployment and Operation

Fig. 3.1 - IoT-A's Views and perspectives

Views: different angles for viewing an architecture that can be used when designing and implementing it.

Internet of Things Architecture (IoT-A):

Perspectives: set of tasks, tactics, directives, and architectural decisions for ensuring that a given concrete system accomplishes one or more quality attributes.

Architectural views concludes:

- Functional view: Fig. 3.2 depicts the Functional View. It consists nine functionality groups, each one with one or more functional components.

- Information view: it describes the components that handle information, the static and dynamic information flows through the system.

- Deployment and operation view: this view investigates how the IoT component communicate with each other.

Each perspective encompasses: a desired quality level; relevant IoT requirements; applicability to (types of) IoT systems; activities to achieve the desired qualities; architectural tactics to be used by architects.

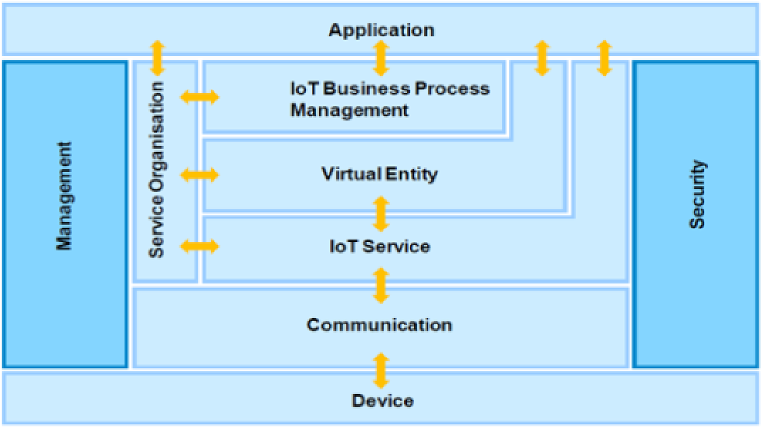


Fig. 3.2 - IoT-A Functional Model

Korean Architectural Reference Model

The Korean Study group has specified IoT reference architecture from a communication viewpoint and a functional viewpoint. Figure 5 illustrates high level functional blocks. It consists of six blocks which represent Infrastructure, Core Functions, Application and Services Functions, Applications and Services, Tools, and Test & Deployment.

Specifically it specifies details of Core Functions in Functional view of IoT RA in Fig. 3.3. Core Functions consists of Connectivity & Underlying Network Management, Resource & Service Management, Semantics & Knowledge, and Security & Privacy.

Chinese Architectural Reference Model

China Communications Standards Association (CCSA) has proposed a reference architecture model for the IoT, which consists of sensing layer, network and business layers, and application layer, presented in Table 3.1. It shows its open and general architecture, which is layered, open, and flexible.

The architecture includes functional platforms as follows:

- Sensing layer: connects sensors, controllers, RFID readers, and location sensing device to IoT network layer;
- Network and service layer: includes backbone networks and resource administration platforms;
- Application layer: includes various applications in IoT system.

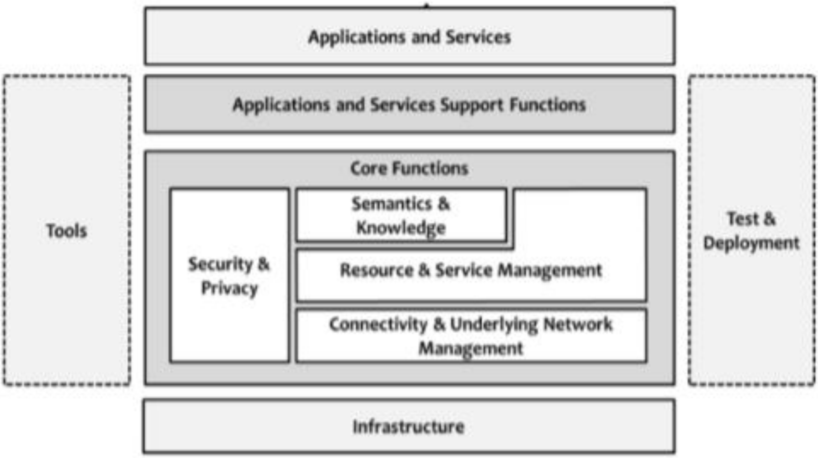

Applications and Services
Applications and Services Support Functions
Tools
Core Functions
Semantics & Knowledge
Security & Privacy
Resource & Service Management
Connectivity & Underlying Network Management
Test & Deployment
Infrastructure

Figure 3.3 - IoT Platform detailed architecture [5]

Tale 3.1 – CCSA reference architecture model for the IoT

Application layer	IoT applications	
Network and service layer	IoT application and support platform	
	IoT backbone network (telecommunication, Internet, private communication)	
Sensing layer	IoT terminals with gateway functions	IoT access gateway
		IoT sensing network
		IoT tip node

WSO2 Architectural Reference Model

Model based on its expertise in the development of IoT solutions. It consists of five layers:

Device Layer: each device should have a unique identifier and direct or indirect communication with the Internet.

Communications Layer: supports device connectivity with multiple potential protocols.

Aggregation/Bus Layer: supports, aggregates, and combines communications from several devices, as well as bridges and transforms data among different protocols.

Event Processing and Analytics Layer: processes and reacts upon events coming from the Aggregation/Bus Layer, as well as can perform data storage.

External Communications Layer: through which users can interact with devices and access data available at the system.

Device Management Layer: communicates with devices through different protocols and allows remotely managing them.

Identity and Access Management Layer: responsible for access control and security directives.

Requirements to IoT:

- Availability. Must be realized in the hardware and software levels to provide anywhere and anytime services for customers. Availability of software refers to the ability of the IoT applications to provide services for everyone at different places simultaneously. Hardware availability refers to the existence of devices all the time that are compatible with the IoT functionalities and protocols.

- Reliability. Aims to increase the success rate of IoT service delivery. It has a close relationship with availability as by reliability, we guarantee the availability of information and services over time. Reliability is even more critical and has more stringent requirements when it comes to the field of emergency response applications.
- Mobility - connecting users with their desired services continuously while on the move is an important premise of the IoT.
- Performance – many components as well as performance of the underlying technologies.
- Management – manage the Fault, Configuration, Accounting, Performance and Security of smart devices. It is necessary to develop new light-weight protocols to handle the potential management.
- Security and Privacy – in heterogeneous networks as in the case of the IoT, it is not easy to guarantee the security and privacy of users.
- Scalability – refers to the ability to add new devices, services, functions for customers without negatively affecting the quality of existing services.
- Interoperability – should be considered by both application developers and IoT device manufactures to ensure the delivery of services for all customers regardless of the specification. End-to-end interoperability need to handle a large number of heterogeneous things that belong to different platforms.

3.2.2 Power consumption

To IoT for office solutions (SBC) are presented such basic requirements:

a) in order to save energy in SBC may to perform the installation of temperature control automatic systems, connection to the mobile network of intelligent systems Smart Metering accounting (electricity, gas and water), which allows you to make decisions on the use of certain energy modes in the office, as well as to save staff time through the use of remote water consumption data collection, electricity, gas, etc.;

b) the possibility of using the various sensors and control units. It is necessary not just to automate certain functions (control of lighting, heating, ventilation and air conditioning - HVAC, etc.), but to integrate virtually any IoT equipment into a single system, works on the algorithm which will set the installer and designer IoT;

c) a complete feedback, which will allow to operate virtually all IoT systems, analyze the situation, make conclusions and to be able to control the IoT without external intervention (without pressing the control panel button), but only upon the occurrence of an event (for example should be provided, the emergence of the human in the corridor include of lighting, on-off ventilation and air conditioning system, power source switching to an alternative power supply, etc.);

d) IoT system should give staff full control over their offices and to provide protection against emerging new threats and threats due to the fact that new computer technologies with connection to the internet allow attackers to connect to the system;

e) physical theft of office equipment and data carriers; theft software; run the executable code for the damage to the systems, for the destruction or corruption of data; modification data; identity theft; execution of actions that do not allow users to access network services and resources; execution of actions that reduce network resources and bandwidth. The basis of any IoT system - is the server on which the control software is stored;

f) increase of the number of computers and servers resulting in significant power consumption, it is necessary to provide greater flexibility and adaptability of the infrastructure of power facilities.

To reduce the load on the power supply can to use a variety of methods, including active implementation of alternative energy sources. Alternative energy helps to improve the economic situation in the country and contribute to environmental improvement. Appropriate use of renewable or locally generated energy in IoT: solar, wind, hydro, geothermal, fuel cell, heat pumps, incorporating liquid cooling in a data center environment will reduce the consumption of electric energy; using a virtualization to reduce the number of computers and servers. Using this method can reduce the number of servers, which will decrease the load on the power supply and reduce the release of thermal energy; using of energy-efficient chips at designing UBTS management systems. One of the advantages of the chip - technology of adaptive dynamic power management. Ultra-low consumption and can be achieved in the operating mode, and a standby mode; using, whenever possible, the low-speed, but reliable data transmission. The use of traditional cellular technology in this area is too expensive, it can use a network of Low-Power Wide-area Network - energy-efficient network of long-range - wireless small data volume transmission technology over long distances, providing environment data collection from sensors, meters and sensors; with sleep mode, these devices may not work for a while.

There are modes of power for the network equipment, which is used when creating IoT: Active - sending packages with high power consumption. Normal Idle (N_IDLE) - no packets (less energy). Low-Power Idle (LP_IDLE) - no packets, less energy-intensive. Power consumption is reduced by turning off unused circuitry during LP_IDLE (part of the PHY, MAC, interconnects, memory, CPU), and only the necessary circuits (for example, clock recovery, alarm) should be included; using of standby mode. This mode is implemented in servers, workstations, in some models of routers. In standby mode, the power consumption of each individual device IoT is minimal, and power consumption increases when attacks to the server and network equipment are successful. The operating system of server has the following modes of reduced energy consumption [18]:

S1 (Power On Suspend, POS, Doze) - Power Saving mode, which turns off the monitor, hard drive, but the central processing unit (CPU) and RAM (memory modules) power is applied, reduced the frequency of the system bus. CPU cache is cleared, the CPU does not perform the instructions from the generator CPU.

S2 (Standby Mode) - reduced power consumption mode. In this mode, the monitor and the hard drive disable. From the CPU turns off the power supply. They stop clocks (continue to operate only those devices that are necessary for memory). Power is supplied only to the system memory (it contains information about the system status).

S3 (Suspend to RAM, STR, Suspend) - Standby. With this power saving mode, power is supplied only to the RAM (it stores information about the system status). All other CPU components are disabled.

S4 (Suspend to Disk, STD, Suspend to Hard Drive, S4-Hibernation) - a deep sleep. With this power saving mode, the current state of the system is written to the hard drive, the power to all components of the is turned off.

3.2.3 Reliability

An important indicator of IoT reliability is the availability function. Availability function - the probability that the object will be in working condition at an arbitrary point in time, except for the planned periods during which the intended use of the object is not provided. In chapter 18 and chapter 34 will be used availability function to estimate the IoT system's reliability.

Reliability factors: P(t): – probability function that a system will operate correctly in [0,t);

• Mean time to failure (MTTF):

• Maintainability: a measure of ability to restore a device to specified condition when maintenance is performed – if we maintain systems at an interval T, with the total number of maintenance events N, the reliability of the maintained system during the time $NT \leq t < (N + 1)T$ is:

$$\mathrm{P}(t) = \mathrm{P}(T)\ NR(t - NT).$$

The fault tolerance of the functioning of the internal components of the IoT system is achieved by applying the following technologies: redundancy of power supplies for server equipment, data storage systems; redundant server network adapters; optical server adapter redundancy; redundancy of cable connection lines of server switching and data transmission network and data storage network; duplication of blade chassis modules: power supplies, control modules, fans, switching modules; placing information on disk storage systems using fail-safe disk groups (RAID).

3.2.4 Cyber security and safety

Security metrics maturity levels:

1. Ad hoc: Security metric reporting is performed organically. Typically, security metrics are not validated. They are communicated in standalone reports, and they are inconsistent and nontransparent.

2. Reactive: Security metric reporting is structured, and metrics have been defined. Security metrics are consistently delivered to decision makers and are acted upon. The scope of the metrics is toward vulnerability measurements. Communications to IT staff aim to ensure system hardening and compliance.

3. Proactive: Security metrics include people, process, risk, user behavior and cost. Besides the IT group, business and data owners are also provided with consistent security metrics and influenced in their security decision making, and they apply security best practices and efficiency throughout the organization. This level of maturity is out of scope for this paper.

4. Predictive: Security metrics include business behaviors and industry trends. They predict risks before they occur.

Risk is the key ingredient for the new solution, this concept is referred to as operational security rating (OSR). The OSR is derived from the likelihood of a vulnerability being exploited and the impact an exploit will cause to the organization.

$$\text{OSR} = \text{Likelihood} * \text{Impact}.$$

In this risk based OSR model each vulnerability is assigned an OSR rating.

The two main metrics on this scorecard are application risk scores for that service, and the on-time closure rates per risk level. The application risk score informs the executive about the current security posture of the applications delivering the service, whereas the closure rate indicates the degree to which vulnerabilities have been closed on time over the past quarters.

Accelerated security initiative led to the creation of Unified Security Metrics (USM). The industry defines several variations of information risk:

$$\text{Risk} = (\text{Vulnerability} * \text{Threat}) * \text{Impact}.$$

Time to detection (TTD) - time passed between when the incident first occurred and when the threat.

Time to remediation (TTR) - how long does it take their incident response team to resolve the problem and remove it from their system the impact of the exploit of a given vulnerability

Impact = f(Data sensitivity, system criticality).

For operational security, likelihood is commonly defined as the ease of exploit of a vulnerability by a threat. ‘Threat’ itself has proven difficult to measure, especially for large and complex businesses that operate across the globe. From a feasibility and implementation point of view it is impracticable to automatically and consistently measure internal and external threat factors and weight them in order to derive likelihood. We found the more feasible option was to measure the opportunity to exploit a given vulnerability in terms of the technological sophistication required and the degree to which that vulnerability was exposed to potential adversaries. Therefore we decided to use ease of exploit and degree of exposure in deriving the likelihood measure in the OSR model.

Likelihood = f(Ease of exploit, Degree of exposure).

Fig.3.4 shows privacy metrics, which described in [21].

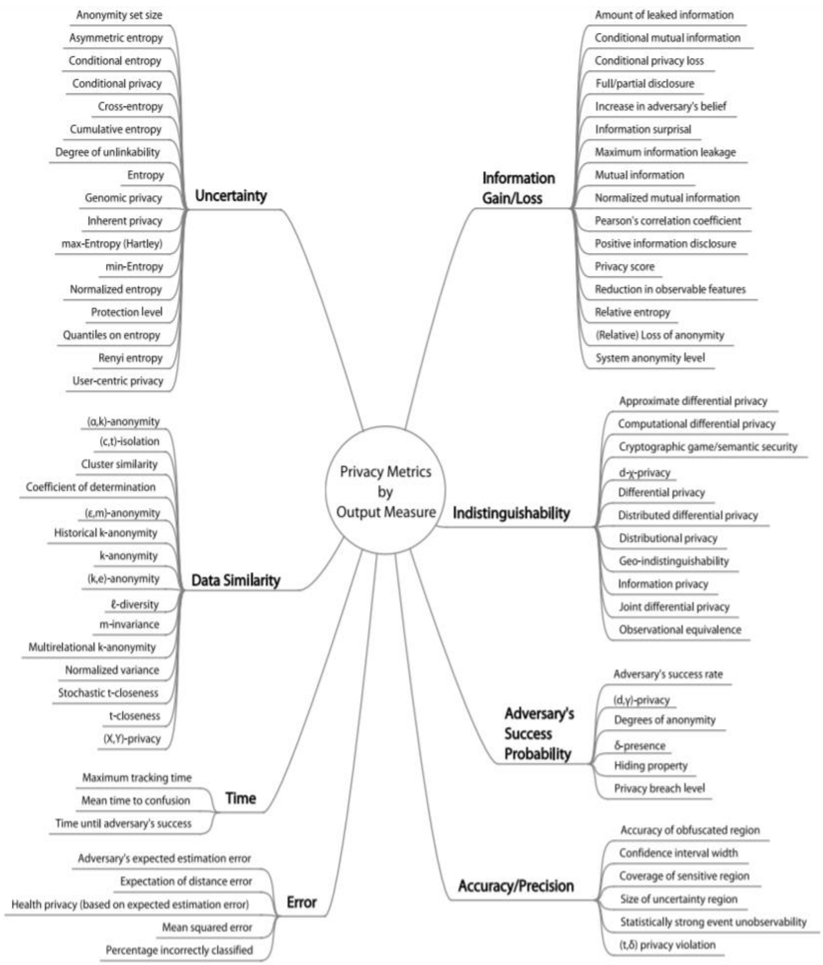


Fig. 3.4 – Privacy metrics

3.2.5 Availability

Availability deals with the duration of up-time for operations and is a measure of how often the system is alive and well. It is often expressed as (up-time)/(up-time + downtime) with many different variants. Up-time and downtime refer to dichotomized conditions. Up-time refers to a capability to perform the task and downtime refers to not being able to perform the task, i.e., uptime or not downtime. Also availability may be the product of many different terms such as:

A = Ahardware * Asoftware * Ahumans * Ainterfaces * Aprocess.

and similar configurations.

Standard ITU-T G.827 Availability performance parameters and objectives for end-to-end international constant bit-rate digital paths defines network performance parameters and objectives for the path elements and end-to-end availability of international constant bit-rate digital paths. These parameters are independent of the type of physical network supporting the end-to-end path, e.g., optical fibre, radio relay or satellite.

Availability issues deal with at least three main factors for: 1) increasing time to failure; 2) decreasing downtime due to repairs or scheduled maintenance; 3) accomplishing items 1 and 2 in a cost effective manner.

As availability grows, the capacity for making money increases because the equipment is in service a larger percent of time. Three frequently used availability terms are explained below. Inherent availability, as seen by maintenance personnel, (excludes preventive maintenance outages, supply delays, and administrative delays) is defined as:

Ai = MTBF/(MTBF + MTTR).

Achieved availability, as seen by the maintenance department, (includes both corrective and preventive maintenance but does not include supply delays and administrative delays) is defined as:

Aa = MTBM/(MTBM + MAMT),

where MTBM is mean time between corrective and preventive maintenance actions and MAMT is the mean active maintenance time

Operational availability, as seen by the user, is defined as:

$$Ao = MTBM/(MTBM + MDT),$$

where MDT is mean down time.

Measure of the ability of power plants, a unit or a plant section to perform its operational function. A distinction is to be made between equipment availability and energy availability:

- Equipment availability is the ratio of available time (operating and standby time) to the calendar period. Equipment availability characterizes the reliability of a plant.
- Energy availability is the ratio of available energy to theoretically possible energy in the period under report. Characterizes the reliability of a plant in general considering all complete and partial outages.

Estimation and prediction naturally use numerical measures. Estimation tells you approximately what you have today with respect to a fixed environment and context.

Effectiveness is defined by an equation as a figure-of-merit judging the opportunity for producing the intended results. The effectiveness equation is described in different formats (Blanchard 1995, Kececioglu 1995, Landers 1996, Pecht 1995, Raheja 1991). Each effectiveness element varies as a probability. Since components of the effectiveness equation have different forms, it varies from one writer to the nextю The major (and unarguable economic issue) is finding a system effectiveness value which gives lowest long term cost of ownership using life cycle costs, (LCC) (Barringer 1996a and 1997) for the value received:

System effectiveness = Effectiveness/LCC.

Effectiveness varies from 0 to 1 and rarely includes all value elements as many are too difficult to quantify. One form is described by Berger (1993):

Effectiveness = availability * reliability * maintainability * capability

The effectiveness equation is the product of: the chance the equipment or system will be available to perform its duty; it will operate for a given time without failure; it is repaired without excessive lost maintenance time; it can perform its intended production activity according to the standard. Each element of the effectiveness equation requires a firm datum which changes with name plate ratings for a true value that lies between 0 and 1.

Berger's effectiveness equation (availability * reliability * maintainability * capability) is argued by some as flawed because it contains availability and components of availability (reliability and maintainability).

Blanchard's effectiveness equation (availability*dependability*performance) has -3- similar flaws. For any index to be successful, it must be understandable and creditable by the people who will use it. Few can quantify reliability or maintainability in terms everyone can understand. The effectiveness equation is simply a relative index for measuring "how we are doing" Availability, Reliability, Maintainability, and Capability [19].

3.3 IoT Domains

IoT decisions:

Smart City: Safe City; Waste Management; Fleet Management; Smart Streetlights; Smart Parking; Irrigation Management; Connected City Lighting; Emergency Response Systems.

Energy & Utilities: Water Metering; Gas Metering; IP-based Metering; Smart Grid.

Smart Agriculture: Irrigation field monitoring; Elevators Connection; Engineering Truck Predictive Maintenance; Internet of Vehicles (Smart Logistics).

Internet of buildings (Smart Home; Smart hospital; Smart hotel; Smart Office; Smart Manufacturing; and the other).

IoT include: Email; Information; Entertainment; Internet of Service (Participation/Trade) include E-commerce; Productivity tools; Integrated chains Internet of People (Collaboration/Share); Voice and video collaboration; Social media and docs; Web logs/boards; Internet of Things (Integration/Control) unites Indexing and tracking; Control and connectivity; Autonomous operations.

3.3.1 Smart Energy Grid

"The grid," refers to the electric grid, a network of transmission lines, substations, transformers and more that deliver electricity from the power plant to your home or business. The digital technology that allows for two-way communication between the utility and its customers, and the sensing along the transmission lines is what makes the grid smart. Like the Internet, the Smart Grid (SG) will consist of controls, computers, automation, and new technologies and equipment working together, but in this case, these technologies will work with the electrical grid to respond digitally to our quickly changing electric demand.

The SG represents an unprecedented opportunity to move the energy industry into a new era of reliability, availability, and efficiency that will contribute to our economic and environmental health. During the transition period, it will be critical to carry out testing, technology improvements, consumer education, development of standards and regulations, and information sharing between projects to ensure that the benefits we envision from the SG become a reality.

Today, an electricity disruption such as a blackout can have a domino effect - a series of failures that can affect banking, communications, traffic, and security. A SG will add resiliency to our electric power system and make it better prepared to address emergencies such as severe storms, earthquakes, large solar flares, and terrorist attacks. Because of its two-way interactive capacity, the SG will allow for automatic rerouting when equipment fails or outages occur. When a power outage occurs, SG technologies will detect and isolate the outages, containing them before they become large-scale blackouts. The new technologies will also help ensure that electricity recovery resumes quickly and strategically after an emergency - routing electricity to emergency services first, for example. In addition, the SG will take greater advantage of customer-owned power generators to produce power when it is not available from utilities. By combining these "distributed generation" resources, a community could keep its health center, police department, traffic lights, phone system, and grocery store operating during emergencies. In addition, the SG is a way to address an aging energy infrastructure that needs to be upgraded or replaced.

3.3.2 Ecological monitoring

The major characteristics of effective monitoring programs typically include: (1) Good questions. (2) A conceptual model of an ecosystem or population. (3) Strong partnerships between scientists, policy-makers and managers. (4) Frequent use of data collected.

Monitoring programs classified into three categories:

1) Passive monitoring, which is devoid of specified questions or underlying study design and has limited rationale other than curiosity.

2) Mandated monitoring where environmental data are gathered as a stipulated requirement of government legislation or a political directive. The focus is usually to identify trends.

3) Question-driven monitoring, which is guided by a conceptual model and by a rigorous design that will typically result in a priori predictions that can be tested.

While mandated monitoring can be useful for producing coarse level summaries of temporal changes in a target population or resource condition they may not identify the mechanism influencing a change in an ecosystem or an entity. A key remaining challenge is to develop much improved mandated monitoring programs through more widespread adoption of the features of successful question-driven monitoring programs in efforts to enhance biodiversity conservation and environmental management.

Ecological monitoring (EM) is not meant to limit the use of natural resources and to limit options for development but is a way of wise long-term development planning. It is a pre-requisite for adaptive natural resource and ecosystem Community-based ecological monitoring. Ecological monitoring can be understood as the collection, analysis and interpretation of data on the natural environment, above all on changes that occur in a certain ecosystem. It attempts to observe living and non-living aspects of the biosphere, the response of the environment to human interventions and to predict the actual or likely impacts. It enables project implementers and target groups, e. g. villagers, to recognize negative ecological effects of their activities at an early stage and to adapt their action. Recently, EM became also more and more recognized as a helpful method in the conservation of nature and natural resource management (NRM). It was proven in several surveys that EM has positively contributed to programs in conservation and development. Ecological monitoring has become an important component of projects and initiatives relating to agriculture, forestry and fishery and measures that focus on nature conservation, management of natural resources and rural development. Any EM process enables an actor or an institution to enter a learning process by providing a base to adapt action. In this context an institution can be any social system, as a community group, a community-based enterprise, a development organization, a rural government body, a rural or urban community or even an individual. As the figure 3.5 below shows, the impacts of this action is observed and assessed by monitoring.

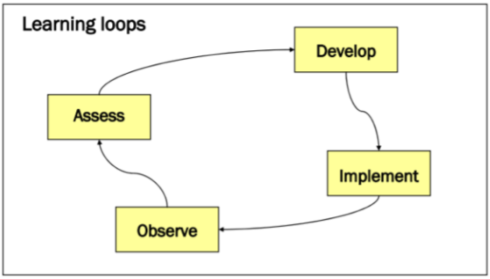
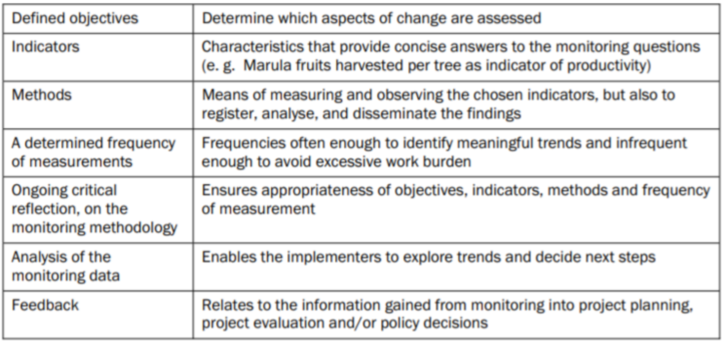


Fig. 3.5 – Learning loops in EM

Data can also be obtained from aerial photographs, satellite images, maps, graphics, statistics, or field work. Furthermore, remote sensing tools can be included, though in many cases they are not necessary. The objectivities of EM presented in Fig. 3.6.



Defined objectives	Determine which aspects of change are assessed
Indicators	Characteristics that provide concise answers to the monitoring questions (e. g. Marula fruits harvested per tree as indicator of productivity)
Methods	Means of measuring and observing the chosen indicators, but also to register, analyse, and disseminate the findings
A determined frequency of measurements	Frequencies often enough to identify meaningful trends and infrequent enough to avoid excessive work burden
Ongoing critical reflection, on the monitoring methodology	Ensures appropriateness of objectives, indicators, methods and frequency of measurement
Analysis of the monitoring data	Enables the implementers to explore trends and decide next steps
Feedback	Relates to the information gained from monitoring into project planning, project evaluation and/or policy decisions

Fig. 3.6 - Main objectivities of EM

If necessary it can be changed as a result of the assessment. It is not being changed if there is no need to do so. Then it would be implemented and then observed again. This way, a harvesting scheme or an erosion control measure can be developed, which is adapted to the needs of the environment and the community.

3.3.3 Smart Vehicle

The transport field has noticed particular development in recent years thanks to the application of intelligent systems. Traditional transport arrangements have been supplanted by Intelligent Transport Systems (ITSs). These new technologies are assisting in solving the main problems of transport engineering, i.e., traffic congestion and accidents. The Connected Vehicles infrastructure can be of various models such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2E).

Nevertheless, these systems must be able to cooperate, for instance, allowing the communication with and among vehicles [19-21]. The interaction between the various involved entities requires the information exchange to use proper communication protocols, such as the IEEE 802.11p and LTE-V2V standards, designed to support vehicle transmissions. Other protocols that can be used in vehicular communications are Bluetooth and IEEE 802.15.4/ZigBee, adequately revised.

1. **Vehicle-to-Vehicle** (V2V) technology consists of wireless data transmissions between motor vehicles. The primary purpose of

this communication is to prevent possible accidents, allowing vehicles in transit to transfer data on their position and their speed within an ad-hoc mesh network [27].

3. **Vehicle-to-Infrastructure (V2I)**. Unlike the V2V communication model, which allows the exchange of information only among vehicles, the V2I enables vehicles in transit to interface with the road system. These components include RFID readers, traffic lights, cameras, lane markers, street lamps, signage, and parking meters. Commonly, V2I communications are wireless, bidirectional, and similarly to V2V, using Dedicated Short-Range Communication (DSRC) frequencies to transfer data. This information is sent from the elements of the infrastructure to the vehicle, or vice versa, through an ad-hoc network.

4. **Vehicle-to-Everything (V2X)**. The V2V and V2I communication models mentioned above are completed in the V2X, which represents a generalization. The latter consists in the data transfer from a vehicle to any entity that can influence it, or vice versa, and incorporates other more specific types of communication including Vehicle-to-Pedestrian (V2P), Vehicle-to-Roadside (V2R), Vehicle-to-Device (V2D), and Vehicle-to-Grid (V2G). One of the main purposes of the V2X technology is precisely to support the possible and efficient communication mechanisms between vehicles and pedestrians aimed at limiting accidents, sometimes fatal.

3.3.4 Smart buildings

True connected home would at the very least mean end-to-end interoperability and security, several of the above mentioned ‘applications’ such as HVAC, light control, room control, some form of energy consumption monitoring and control, the connection of smart home appliances and a more or less stabilized market, making that smart home vision come true.

However, with the Internet of Things existing standards have been joined by several other standards and communication forms. Just think about the various evolutions in connectivity standards *(short range like Bluetooth 5.0, Bluetooth Mesh, the next Wi-Fi and long range too for some applications)*, proprietary standards in the vendor ecosystems and platforms *(the wars of the big players such as Google and Apple)* and the different alliances with home automation standards *(the previously mentioned Z-Wave, ZigBee etc.)*.

If you look at the – longer existing – high-end of the smart home and home automation market there is obviously more maturity and these vendors have solutions that support several of the existing standards such as KNX, as well as IP. In home automation and building automation, a few standards will remain but IoT will replace a lot and act as a converging force.

There is more attention for security *(e.g. the Z-Wave alliance launched a new certification program)*, there are innovations in connectivity (*e.g., finally Bluetooth 5.0 is there, now the products)*, vendors have taken new initiatives *(e.g. Google's Android Things)*, the list goes on.

3.3.5 Smart Health

According to the European Union Agency for Network and Information Security (ENISA), the Smart Hospital is a health organization that provides health care and is based on optimized and automated processes built using the actual IT in accordance with the Health 4.0 concept to improve the existing procedure for patient care and to implement the advanced medical technologies [22]. The strategic objectives of the Smart Hospital are to provide extended patient care including remote medical services, to ensure efficient stream of patients and medical information, to increase diagnostic, surgical and organizational capabilities while maintaining the required level of patient information protection. The Smart Hospital model introduction affects all levels of health organization management and requires re-engineering the health organization architecture as a whole or its separate components.

3.3.6 Industry 4.0. Industrial Internet of Things

Industrial IoT (IIoT) is a system that connects and integrates operational technology (OT) environments, including industrial control systems (ICS), with enterprise systems, business processes, and analytics [23]. The benefits of IIoT are the ability of sensors or connected devices, as part of a closed-loop system, to collect and analyze data and then do something based on what the data reveals. Protocols and performance of IIoT system is presented in Fig. 3.7. The very connectivity, however, also grows the risk of attack — and, increasingly, cyberattacks - by those who may want to bring down the system. a big increase in the number of sensors and devices being connected to each organization's IIoT, forming a huge potential attack surface:

- decades-old OT equipment and control systems never designed for exposure to the internet and, therefore, not designed for security;
- a patchwork of OT and control systems from multiple vendors running proprietary and non-updatable software, including human-machine-interface (HMI) computers with access to remote terminal units (RTUs), SCADAmaster (supervisory control computers), and programmable logic controllers (PLCs);
- poor or absent cyber security practices and technology, including a lack of either designed for the very different ICS/OT environment, not the IT environment;
- lack of budgets, or insufficient budgets, for implementing cyber security awareness, monitoring, and prevention technology;
- a steep escalation in the numbers and types of attackers.

Standards for IIoT:

- Security/IoT ISO/IEC 2700x; JTC1/WG10;
- New IEC/ISO Reference Model IEC/ISO JWG21;
- IEC/ISO 61360 Rules for Properties;
- IEC/ISO 61387 Sensors Prop.;
- IEC/ISO 62683 Switch Gears Prop.;
- IEC/ISO 62832 Digital Factory;
- IEC/ISO 62443 Security in Automation;
- IEC 61987 Sensors; IEC 62683 Switch Gears;
- PLC technologies IEC 61131;
- Engineering data automation ML IEC 62714;
- OPC-Foundation Services IEC 62541 OPC-UA and Companions;
- ProSTEP e.V. Mechanics STEP+APxxx; ITU-T, ITU-R - IoT Standardization Spectrum.

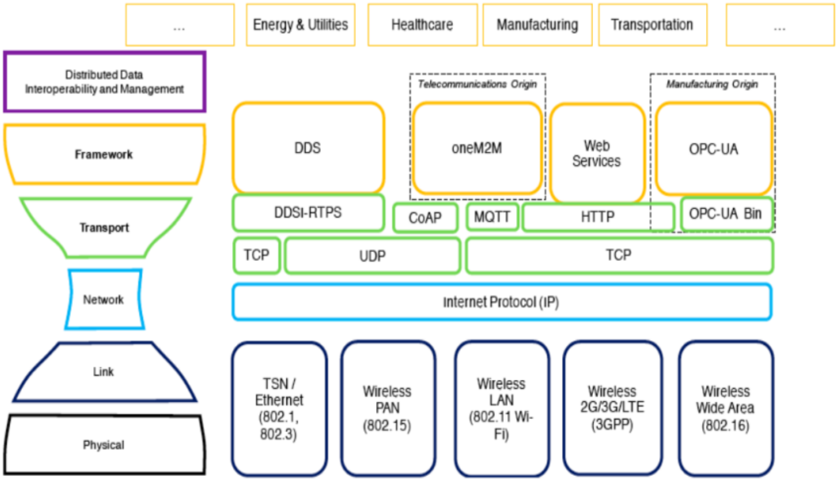


Fig. 3.7 - Industrial Internet Connectivity Stack Model

Enabling Technologies: Sensors/Actuators;
Communication protocols (REST, CoAP, MQTT);
Microservices and Middleware; Data Analytics Engines;
Apps (iOS, Android, Web). Industrial and process systems rely on sensors for reliable and accurate data in all aspects of control and automation. There is a new wave of innovative sensors based on the application of technology to improve performance [24-28].

3.4 Work related analysis

Example of IoT is a project where the information about the state of the road from the individual vehicles incomes into the overall system based on the "cloud." Real-time data about slippery surfaces on the road is transmitted through the mobile communication network to alert the vehicle, which are around. Warning is instantly transmitted to the other vehicles, which are close to the slippery area.

- This enables drivers to take an immediate action to avoid a critical situation.

- Other possible application of this technology is the remote diagnostics. Data can be transferred in advance, thus eliminating the problem in real-time [29].

- Toyota Motor Corp. and Panasonic jointly develop a service that will connect cars and home appliances through the IoT [30].

- The project PRORETA [31] is a research in the area of the cooperative HMIs. The research object is the prototype of the cooperative automobile HMI that implements the scenarios of preventing collisions at the cross-roads.

- The PRORETA HMI system implements a huge number of use scenarios, it does not complicate or irritate and ensures the multimode support.

- The HMI provides 4 support levels – information messages, warnings, actions recommendations, automatic intervention.

- A lot of EU universities including ALIOT project partners conduct research and implement education MSc and PhD programs in the Internet of Things application for transport and other domains. Development of cooperative HMI for cloud and IoT systems based on analysis of these programs and providing some of the educational topics and research directions.

- In particular, the following courses and programs have been considered:

- Coimbra University, Portugal: IoT course for MSc [32]. The courses represents a new stage in the digital evolution and focuses on the Internet of Things for smart transport and cities, and the development of tools to transform city infrastructure;

- KTH University, Sweden: three MSc programs including:

a) IoT related topics in Information and Network Engineering [33],

b) Communication Systems [34],

c) Embedded Systems [34];

- Newcastle University, United Kingdom: MSc Program on Embedded Systems and Internet of Things (ES-IoT) MSc [34].

Conclusions and questions

The Internet of things actively covers various spheres of human activity. In this section, the analysis of the available solutions and performance regarding the IoT systems has been conducted. The standards and recommendations in the area of IoT systems architecture, security, technologies have been analyzed.

The features of the presentation of models of systems of the Internet of things, the requirements for their organization are considered. The analysis is carried out and the basic metrics applicable to the evaluation of the criteria of the Internet of things systems are described. The features of the Internet of things domains are considered.

Questions for self-testing in IoT related standards and methics:

1. What organizations and institutes are developers of IoT related standards?
2. What main groups of the IEEE standards applied for IoT do you know? Please call them.
3. What standards can be used for IoT based V2V systems?
4. What main groups of the ITU-T standards applied for IoT do you know? Please call them.
5. What ITU-T standards can be used for assessment and development of Smart health system?
6. What IEC standards can be used for assessment and development of Smart Grid systems?
7. What the main requirements to IoT system organization?
8. What the availability metrics of IoT system do you know?
9. What security standards can be used for IIoT systems?
10. What the main connection types of Smart Vehicle do you know?
11. What technologies can be used for IoT system organizing?
12. What the performance metrics of IoT system do you know?

References

[1] Internet of Things. IoT Governance, Privacy and Security Issues. European Research Cluster on the Internet of Things. Ovidiu Vermesan, Peter Friess, Coordinators of IERC Cluster. January, 2015. 128 p.

[2] Delivering on the IoT customer experience. Business white paper. Hewlett Packard Enterprise. Available at: http://h20195.www2.hpe.com /v2/GetDocument.aspx?docname =4AA6-5128ENW (accepted at 5.08.2016). 8 p.

[3] Internet of Things and its future. Available at: http://www.huawei.com/ilink/en/about-huawei/newsroom/pressrelease/ HW_080993?dInID=23407&relatedID=19881& relatedName= HW_076569&dInDocName=HW_076557 (access date: 20.11.2017).

[4] IETF Standardization in the Field of the Internet of Things (IoT): A Survey. Isam Ishaq, David Carels, Girum K. Teklemariam, Jeroen Hoebeke, Floris Van den Abeele, Eli De Poorter, Ingrid Moerman and Piet Demeester. J. Sens. Actuator Netw. 2013, 2, 235-287; doi:10.3390/jsan2020235. Journal of Sensor and Actuator Networks ISSN 2224-2708. Available at: http://www.mdpi.com/journal/jsan/.

[5] SG17-LS084. https://www.ietf.org/lib/dt/documents/LIAISON/ liaison-2018-02-06-itu-t-sg-17-tsb-suit-ls-on-iot-secure-update-procedure-attachment-1.pdf.

[6] ISO/IEC 29177:2016. Information technology. [https://www.iso.org/obp/ui/#iso:std:iso-iec:29177:ed-1:v1:en].

[7] Bilel Jamoussi. IoT Prospects of Worldwide Development and Current Global Circumstances. Chief Study Groups Department Telecommunication Standardization Bureau, ITU www.itu.int/ITU-T/go/IoT. 2010, 32 p. https://www.itu.int/en/ITU-T/techwatch/Documents/1010-B_Jamoussi_IoT.pdf.

[8] Ian Lamont, “IoT standards: A starting point, but not the finishing line”, Feb., 2018. https://www.hpe.com/us/ en/insights/articles/iot-standards-a-starting-point-but-not-the-finishing-line-1802.html

[9] RFC 7252. https://tools.ietf.org/html/rfc7252.

[10] ITU-T SG20: Internet of things (IoT) and smart cities and communities (SC&C). 2019. https://www.itu.int/en/ITU-T/studygroups/2017-2020/20/Pages/default.aspx.

[11] RFC3819. Advice for Internet Subnetwork Designers. 2004. https://tools.ietf.org/html/rfc3819.

[12] RFC 4944. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. 2007. https://tools.ietf.org/html/rfc4944.

[13] IEEE 802.24 Smart Grid TAG Procedure for starting a new TG. 2013. 6 p. [https://mentor.ieee.org/802.24/dcn/14/24-14-0016-01-0000-process-for-createing-new-tg.pdf].

[14] RFC 4861. Neighbor Discovery for IP version 6 (IPv6). 2007. https://tools.ietf.org/html/rfc4861.

[15] RFC 5889. IP Addressing Model in Ad Hoc Networks. https://tools.ietf.org/html/rfc5889.

[16] Andrey Dvornikov, Pavel Abramov, Sergey Efremov, Leonid Voskov, “QoS Metrics Measurement in Long Range IoT Networks”. [https://ieeexplore.ieee.org/document/8012393/]

[14] A. Torkaman, M.A. Seyyedi. "Analyzing IoT Reference Architecture Models", International Journal of Computer Science and Software Engineering”, 5, No. 8 (2016): pp. 154-160.

[15] Gerrod Andresen, Zachary Williams. Metrics, KPI, and modeling of long range aircraft availability and readiness. NATO, RTO-MP-AVT-144. 12 p.

[16] Ed Potoczak. Capitalizing on the Convergence of Manufacturing Quality, IoT and Lean. Oct. 17, 2017. [https://www.qualitymag.com/articles/94260-capitalizing-on-the-convergence-of-manufacturing-quality-iot-and-lean].

[17] Automotive IoT Security: Countering the Most Common Forms of Attack. March 7, 2018, Report, GSMA. [https://www.gsma.com/iot/automotive-iot-security-countering-the-most-common-forms-of-attack/].

[18] **GSMA IoT Security Guidelines and Assessment.** 2019. [https://www.gsma.com/iot/iot-security/iot-security-guidelines/].

[19] Industrial Internet of Things Safety and Security Protocol. Center for the Fourth Industrial Revolution Protocol Design Networks. Apr. 2018, 20 p.

[20] IoT Safety/Security Development Guidelines （Second Edition）Information-technology Promotion Agency. 106 p.

[21] Arkadiy Kremer. ITU-T Security work 5th ETSI Security Workshop 20-22 Jan. 2010, Sophia Antipolis, France, ITU-T SG 17. [https://www.itu.int/ITU-T/special-projects/security/presentations/KREMER_ITUTSecurityWork.pdf].

[22] Oksana Ilyashenko, Igor Ilin, Dmitry Kurapeev. Smart Hospital concept and its implementation capabilities based on the incentive extension. SHS Web of Conferences 44, 00040 (2018). [https://doi.org/10.1051/shsconf/20184400040 CC-TESC2018].

[23] Nitin Dahad**,** **“**Designing security into the industrial IoT**”**, Sept. 15, 2018. **[**https://www.embedded.com/electronics-blogs/say-what-/4461109/Designing-security-into-the-industrial-IoT?_mc=EMB_FT_DEV_01].

[24] What is smart Grid?[https://www.smartgrid.gov/the_smart_grid/smart_grid.html].

[25] Alexander Fröde, Christopher Masara. Community-based ecological monitoring. Manual for practitioners. Harare, Aug. 2007. 64 p. SAFIRE - Southern Alliance for Indigenous Resources.

[26] Bill Lydon. Sensors are Fundamental to Industrial IoT. *Nov. 17, 2014. [*https://www.automation.com/automation-news/article/sensors-are-fundamental-to-industrial-iot].

[27] Mr.Radhesh. Session on Role of IoT in Enterprise and Architecture. Nibodha Technologies. Jan 12, 2015. [https://www.slideshare.net/Nibodha/enterprise-architecture-and-iot].

[28] Lauren Robeson.Three Major Protocols Combine to Boost End Users' IIoT Benefits. March 14, 2018. [https://www.prosoft-technology.com/News-Events/Press-Releases/ Three-Major-Protocols-Combine-to-Boost-End-Users-IIoT-Benefits].

[29] Bauer, E. PRORETA 3: An Integrated Approach to Collision Avoidance and Vehicle Automation [Text] / E. Bauer, F. Lotz, M. Pfromm // At – Automatisierungs technik. – 2012. – № 12. – P. 755-765.

[30] Internet Of Things Course - Immersive Program Master in City and Technology [https://apps.uc.pt/search?q=Internet+of+Things].

[31] Master's program in Information and Network Engineering [https://www.kth.se/en/studies/master/information-and-network-engineering/master-s-programme-in-information-and-network-engineering-1.673817].

[32] Master's program in Communication Systems [https://www.kth.se /en/studies/ master/communication- systems/ description - 1.25691]

[33] Master's program in Embedded Systems [https://www.kth.se/en/studies/master/ embedded-systems /description-1.70455/].

[34] Related Programs to Embedded Systems and Internet of Things (ES-IoT) MSc [https://www.ncl.ac.uk/postgraduate/courses /degrees/embedded-systems-internet-of-things-msc/relateddegrees.html].

4. COMMUNICATION, PROTOCOLS AND DATA TRANSMISSION IN IOT

DrS, Senior Researcher O. A. Chemeris,
Dr, Senior Researcher S. Ya. Hilgurt (IPME)
Dr, Assoc. Prof. V. Y. Pevnev, Senior Lecturer M. V. Tsuranov (KhAI)

Contents

Abbreviations

AMQP – Advanced Message Queuing Protocol

BLE – Bluetooth Low Energy

BT5– Bluetooth 5.0

CN – Control Node

ETSI – European Telecommunications Standards Institute

GN – Gateway Node

IEEE – Institute of Electrical and Electronics Engineers

IETF – Internet Engineering Task Force

IoT – Internet of Things

ITU-T – International Telecommunication Union

M2M – Machine to Machine

MQTT – Message Queuing Telemetry Transport

PAN – Personal Area Networks

SIG – Special Interest Group

SN – Sense Node

TCP/IP – Transmission Control Protocol / Internet Protocol

W3C – World Wide Web Consortium

4.1 Communications for IoT

4.1.1 Network architecture

Communication and protocols play a crucial role for IoT. Although the term “Internet of Things” is always used in the literature, a more accurate description would be “Network of Things” [1]. A smart-home installation, for example, consists of numerous things in the home that are interconnected via Wi-Fi or with some central controller. In a factory or farm setting, a network of things may be enabling enterprise applications to interact with the environment and run applications to exploit the network of things. In these examples, remote access over the Internet is usually, but not invariably, available. Whether or not such Internet connection is available, the collection of smart objects at a site, plus any other local compute and storage devices, can be characterized as a network or an Internet of Things.

Most of the literature views the IoT as involving intercommunicating smart objects. The Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) in the published Recommendation Y.2060 entitled “Overview of the Internet of Things” characterizes the IoT as adding the dimension “Any THING communication” to the information and communication technologies that already provide “any TIME” and “any PLACE” communication [2].

Fig. 4.1 depicts the ITU-T IoT Reference Model, which consists of four layers as well as management capabilities and security capabilities that apply across layers [1].

The Device Layer in terms of communications functionality includes, in fact, the OSI physical and data link layers.

The Network Layer performs two basic functions. Networking capabilities refer to the interconnection of devices and gateways. Transport capabilities refer to the transport of IoT service- and application-specific information as well as IoT-related control and management information. In fact, these capabilities correspond to those of the OSI network and transport layers.

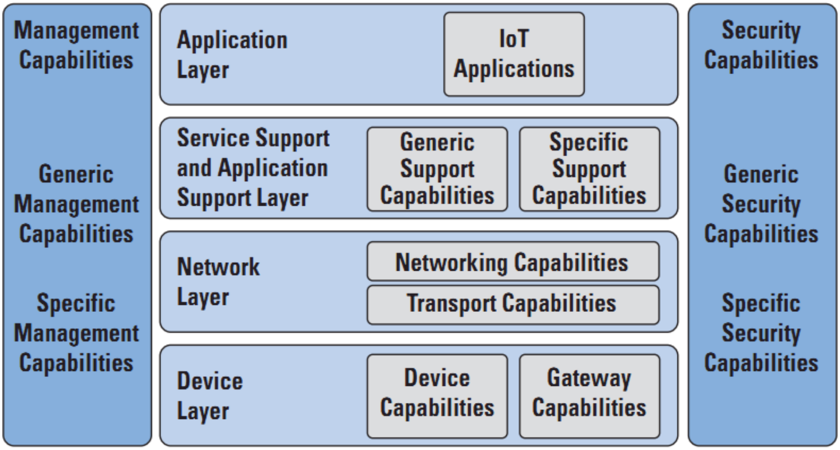

Management Capabilities
Generic Management Capabilities
Specific Management Capabilities
Application Layer
IoT Applications
Service Support and Application Support Layer
Generic Support Capabilities
Specific Support Capabilities
Network Layer
Networking Capabilities
Transport Capabilities
Device Layer
Device Capabilities
Gateway Capabilities
Security Capabilities
Generic Security Capabilities
Specific Security Capabilities

Fig. 4.1 – ITU-T Recommendation Y.2060 IoT Reference Model [1]

The Service Support and Application Support Layer provide capabilities that applications use. Many different applications can use generic support capabilities. Examples include common data processing and database management capabilities. Specific support capabilities are those that cater for the requirements of a specific subset of IoT applications.

The Application Layer consists of all the applications that interact with IoT devices.

The Management Capabilities Layer covers the traditional network-oriented management functions of fault, configuration, accounting, and performance management.

The IoT enables things to see and sense the environment, to make coordinated decisions, and to perform tasks based on these observations [3]. In order to realize the full benefits of the IoT, it will be necessary to provide sufficient networking infrastructure to support low latency and fast response times for IoT applications.

To begin establishing the right networking technology for any IoT application, it is important to first understand the network architecture, or the network topology, that is supported by each technology standard [4].

The consideration of technology is completely dependent on the application specific requirements, for example – range, power consumption, bandwidth, delay and scalability requirements are different for different application. But the networking standards being used today in IoT can be categorized into three basic network topologies: point-to-point, star, and mesh (Fig. 4.2). The current communication technologies such as Wi-Fi, Wi-Fi LP, Bluetooth, Bluetooth LE (BLE), Zigbee, Z-Wave, EnOcean and others support these types of topologies.

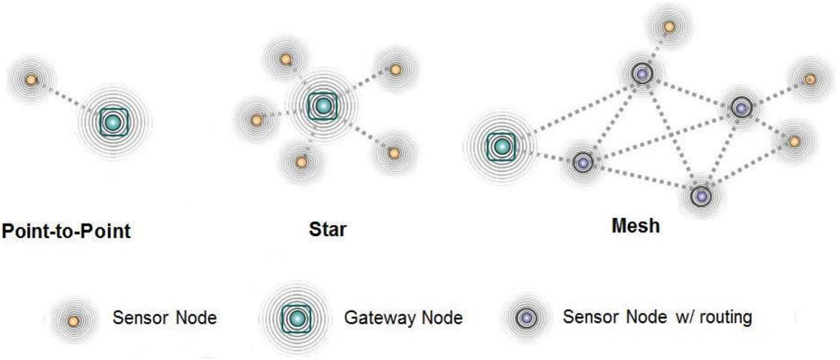


Fig. 4.2 – Basic network topologies

A point-to-point network establishes a direct connection between two network nodes. Communication can take place only between these two nodes, or devices. The advantages of such networking are its simplicity and low cost. The primary limitations are that the network cannot scale beyond these two nodes. The range of the network is therefore limited to one hop, and defined by the transmission range of a single device.

A star topology has some important advantages. The performance of such a network is high, consistent and predictable. In a star network, unlike the mesh network described next, a data packet typically only travels one or two hops, yielding very low network latency. Another advantage of this network type is high reliability. Each device utilizes its own link to the hub. This makes it easy to detect faults and to remove failing network components. The disadvantages of the star topology are similar to the point-to-point network. The range is limited to the transmission range of a single device. Additionally, in a star network, the central component, the gateway is a major factor reducing reliability. In a mesh network, if the gateway loses connectivity, the network can still exchange and store data internally.

A mesh network besides a gateway node and sensor nodes include also sensor nodes with repeater/routing capability, which not only capture their own data, but also serve as relays for other nodes and can propagate the data through the network. This networking topology is used for many applications requiring a long range and broad area coverage.

Because the network range is not limited to the transmission range of a single device, the network range can be very broad, covering large areas and can scale up to thousands of nodes, providing a high density of coverage with a broad assortment of sensors and actuating devices. The main disadvantage of mesh networks is higher complexity compared to the other two types of topology. Additionally, there is higher network latency in mesh networked due to multiple hops from the sensor to gateway.

4.1.2 Delays

Wireless sensor network is one of major IoT applications nowadays as they operate mostly in ad-hoc mode. Ad-hoc networks can communicate without fixed infrastructure. Ease and less time for deployment are the main advantages of ad-hoc networks. Diverse delay requirements are to be consider to support heterogeneous sensor network applications of IoT applications.

Delay in wireless network dependent on the following source of reasons [5].

Delay due to multi-hop. Message traverses several hops before reaching destination in multi-hop networks. Transmission power is analogous to transmission radius, so low power operation of nodes in battery operated sensor networks also increases the number of hops in the network which is another reason for larger delays.

Channel access delay. The channel access mechanisms are mostly CSMA/CA based contention access in wireless networks. Collisions create additional exponential delays in the network. Channel access delays depend on throughput of each node, node density and number of nodes in the network and transmission power.

Aggregation and compression in ad-hoc networks is used to reduce the redundancy of the messages thereby reducing the channel access delays. Aggregation and compression function of throughput and protocols at intermediate nodes can lead to large delays due to processing delays before transmission.

These three sources of delay are tightly coupled and should be considered carefully.

4.1.3 Bluetooth 5.0

Bluetooth is a technology that has been developed more than twenty years ago.

According to related varying requirements a lot of specific enabling communication technologies need to be considered. In this context, Bluetooth Low Energy (BLE), widely spread in consumer hardware, is a key enabler to efficiently connect smartphones with low power sensors in the coverage area of Personal Area Networks (PAN). According to a higher communication range requirements of IoT the Bluetooth Special Interest Group (SIG) provides the next generation Bluetooth 5.0 specification (BT5), which promises increased ranges, speed and broadcast messaging capacity [6].

Fig. 4.3 presents area of Bluetooth 5.0 among the Internet of Things applications, which is now additionally covering Smart Factory (industrial), Smart Home and Smart Building, as well as partly Smart Grid and Smart City applications.

The Bluetooth 5.0 specification is published as the latest version of Bluetooth core specification in the end of 2016 [7].

It is an advancement of the latest BLE version 4.2 and improves data rate, range, broadcast capabilities, as well as fast and seamless pairing processes, allowing even more flexible and versatile deployments.

On the application level, BLE devices use so called pro-files to exchange data based on the upper layer protocol Generic Attribute Profile (GATT).

The position of GATT in the BLE protocol stack is shown in Fig. 4.4 [8].

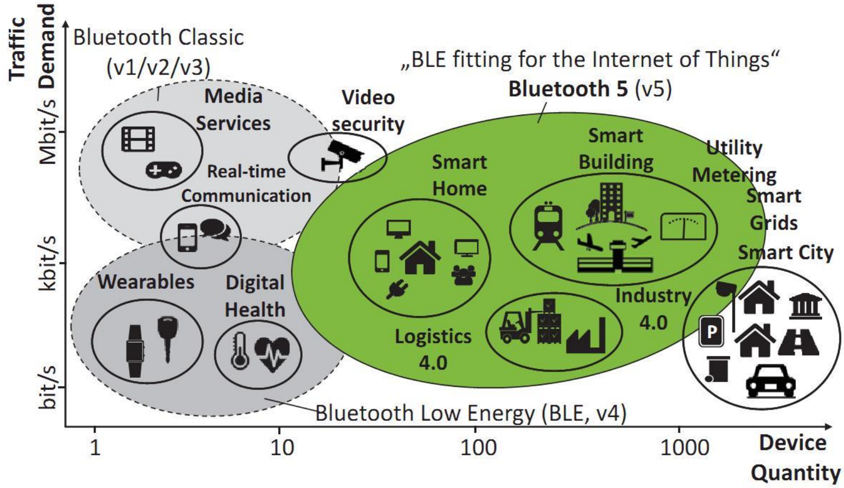


Fig. 4.3 – Expansion of Bluetooth for the Internet of Things

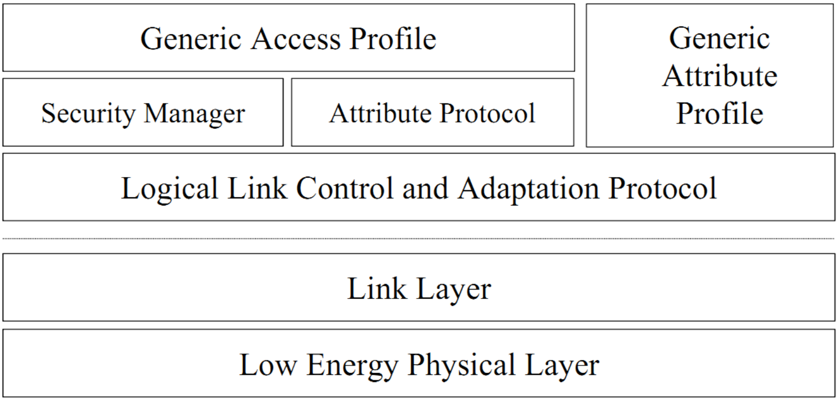

Generic Access Profile
Security Manager
Attribute Protocol
Generic Attribute Profile
Logical Link Control and Adaptation Protocol
Link Layer
Low Energy Physical Layer

Fig. 4.4 – Bluetooth low energy single mode protocol stack

The new physical layer mode LE 2 M PHY allows to operate at 2 M Symbols/s and thus enables higher data rates compared to the well known uncoded LE 1M PHY of Bluetooth 4.0. On the other hand, to achieve higher transmission ranges, a PHY mode with convolutional

FEC coding is added to the specification (LE Coded PHY). The convolutional code is available with a coding rate of 1/2 (S=2) or 1/8 (S=8). Despite the uncoded LE 1M PHY, all improvements are optional and can be implemented based on the considered application requirements. This topic is analyzing the capabilities of BT5 for IoT purposes and thus focuses on the LE Coded PHY modes, as key enabler for a wide range of IoT applications.

4.2 IoT protocols analysis

Many IoT standards are proposed to facilitate and simplify application programmers' and service providers' jobs. Different groups have been created to provide protocols in support of the IoT including efforts led by the World Wide Web Consortium (W3C), Internet Engineering Task Force (IETF), EPC global, Institute of Electrical and Electronics Engineers (IEEE) and the European Telecommunications Standards Institute (ETSI) [3]. Table 4.1 provides a summary of the most prominent protocols defined by these groups. In this part of Multi-book, we classify the IoT protocols into four broad categories, namely: application protocols, service discovery protocols, infrastructure protocols and other influential protocols. However, not all of these protocols have to be bundled together to deliver a given IoT application. Moreover, based on the nature of the IoT application, some standards may not be required to be supported in an application. In the following subsections, we provide an overview of some of the common protocols in these categories and their core functionality.

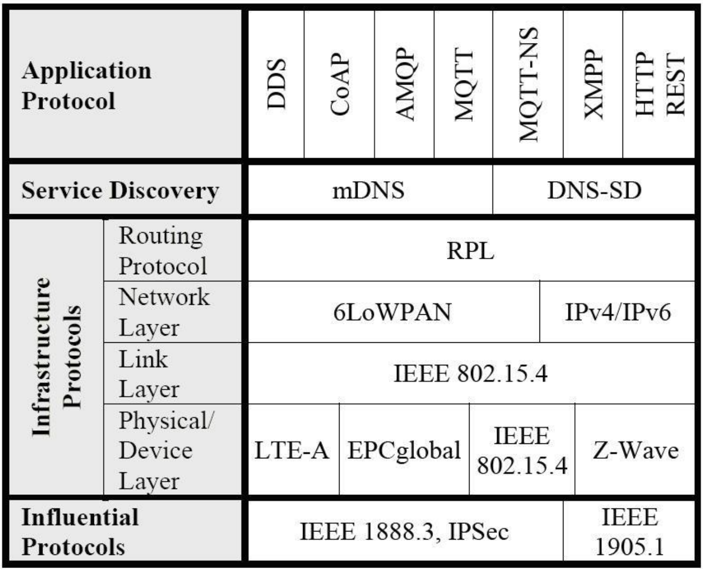
The protocols presented in the Table 4.1 are the most used solution for IoT networks and machine-to-machine (M2M) communication. They enable interaction between the devices and they try to offer solutions to IoT and M2M requirements [9].

Data Distribution Service (DDS) is a Data-Centric Publish/Subscribe (DCPS) service for distributed application communication and integration and is based in a broker-less architecture, using multicast to bring high Quality of Service (QoS) and reliability, that suits for the real-time constraints for IoT.

The Constrained Application Protocol (CoAP) is a web transfer protocol, specialized for constrained nodes and networks, structured for machine-to-machine communication, providing a request/response interaction model, built-in discovery of services and resources, based on Uniform Resource Identifier (URI) and Internet media types.

The Extensible Messaging and Presence Protocol (XMPP) enables a near-real-time exchange of structured yet extensible data between any two or more network entities, using the Extensible Markup Language (XML). It consists in an asynchronous, client-to-client or server-to-server exchanged of Stanzas among a distributed network of globally addressable over TCP, being similar to email's architecture, with useful modifications to allow the communication in close to real time.

Table 4.1 – Standardization efforts in support of the IoT



Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-NS	XMPP	HTTP REST
Service Discovery		mDNS				DNS-SD		
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN					IPv4/IPv6	
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal			IEEE 802.15.4	Z-Wave	
Influential Protocols		IEEE 1888.3, IPSec					IEEE 1905.1	

MQTT for Sensor Network (MQTT-SN), is a protocol that aims to connect embedded devices and networks with applications and middleware, it has one-to-one, one-to-many, many-to-many connection mechanism.

Let us study below more closely such vital standards as MQTT and AMQP, looking first at the more well-known HTTP/HTTPS protocols.

4.2.1 HTTP/HTTPS

Standards HTTP/HTTPS are well known, and there are many libraries that support them. Because these are text-oriented protocol, many small devices such as 8-bit controllers can only partially support them – for example use only POST and GET functionality. Using HTTP is inefficient and costly in terms of network traffic and power usage. So several protocols optimized for IoT use where developed. The two best known are MQTT and AMQP.

4.2.2 MQTT

MQTT (Message Queuing Telemetry Transport) is presented by Andy Stanford Clark of IBM and Arlan Nipper of Arcom (now Eurotech) in 1999 and was standardized in 2013 at OASIS [3]. It aims at connecting embedded devices and networks with applications and middleware. The connection operation uses a routing mechanism (one-to-one, one-to-many many-to-many) and makes MQTT the optimal connection protocol for the IoT and M2M (machine to machine interaction).

To provide transition flexibility and simplicity of implementation to be suitable for resource constrained devices that use unreliable or low bandwidth links MQTT utilizes the publish/subscribe scheme flowing over TCP/IP. The specifications provide three elements: connection semantics, routing, and endpoint. The protocol has bit-wise headers and variable length fields. The packet size is 2 bytes. Fig. 4.5 shows the overall functionality of MQTT.

MQTT simply consists of three components, subscriber, publisher, and broker.

A publisher sends the message on the topic and subscriber consumes a message on a corresponding topic. A message server matches publications to subscriptions. If one or more matches found at the event, the message is delivered to corresponding subscriber and the message is discarded if no matches found. Furthermore, the broker achieves security by checking authorization of the publishers and the subscribers.

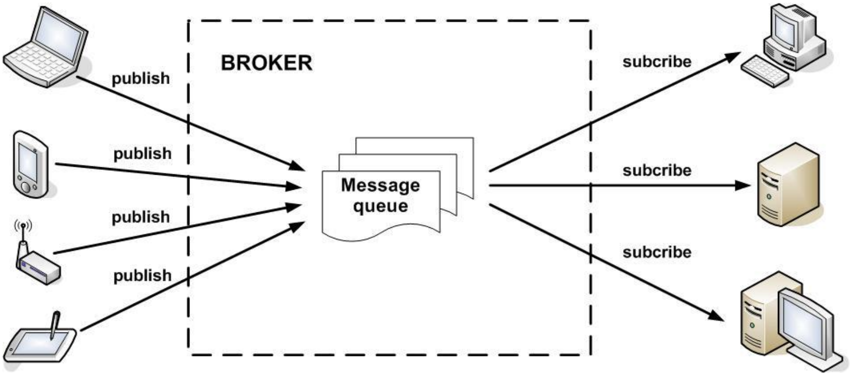


Fig. 4.5 – Architecture of MQTT

Fig. 4.6 illustrates the publish/subscribe process utilized by MQTT and Fig. 4.7 shows the message format used by the MQTT protocol.

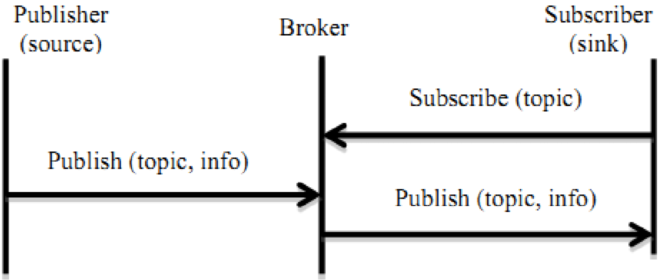
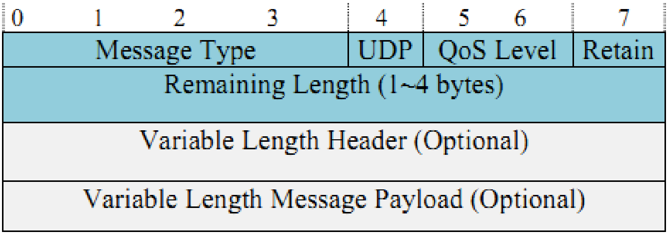


Fig. 4.6 – Publish/subscribe method in MQTT



0	1	2	3	4	5	6	7
Message Type				UDP	QoS Level		Retain
Remaining Length (1~4 bytes)							
Variable Length Header (Optional)							
Variable Length Message Payload (Optional)							

Fig. 4.7 – Publish/subscribe method in MQTT

The first two bytes of message are fixed header. In this format, the value of the Message Type field indicates a variety of messages including CONNECT (1), CONNACK (2), PUBLISH (3), SUBSCRIBE (8) and so on. The UDP flag indicates that the massage is duplicated and that the receiver may have received it before.

The MQTT provides 3 options for selecting the reliability of messaging, which are provided with three levels of quality of service. (QoS) [10]:

– QoS 0 – the message is transmitted only once and does not require confirmation;

– QoS 1 – the message is sent at least once and requires confirmation;

– QoS 2 – for the delivery of communication, a four-stage handshake mechanism is used. In addition, the standard TLS (Transport Layer Security) security level is placed on top of the TCP level. Port 8883 provides security of communication, if the broker's address works with this port, then the traffic is transmitted with encryption.

The QoS Level field indicates the level of assurance for delivery of an application service. Its possible values are listed in Table 4.2.

Table 4.2. – QoS Levels

QoS value	Bit 2	Bit 1	Description
0	0	0	At most once <=1
1	0	1	At least once >=1
2	1	0	Exactly once =1
3	1	1	Reserved

The Retain field informs the server to retain the last received Publish message and submit it to new subscribers as a first message. The Remaining Length field shows the remaining length of the message i.e. the length of the optional parts.

MQTT pros.

1. MQTT is considered as a lightweight messaging protocol because all messages have small code footprint. This protocol is a

bandwidth protocol that was data agnostic with support for multiple levels of QoS.

2. It also provides two-way communication over unreliable networks.

3. MQTT has few methods (publish/subscribe/unsubscribe), quick to learn.

4. The smallest packet of size 2 bytes is possible for an MQTT message.

5. This protocol distributes from one-to-one, one-to-N via the publish/subscribe mechanism.

MQTT cons.

1. MQTT Version 3.x only supports publish / subscribe.

2. MQTT has no advanced features such as flow control.

3. As all the message payloads are binary MQTT protocol lacks interoperability.

4. Problems will arise in open networks because there will be no information about how they are encoded.

4.2.3 AMQP

AMQP (Advanced Message Queuing Protocol) is an open standard message-oriented application layer protocol [11]. To provide M2M functionality AMQP protocol allows its implementations to interoperate with each other. The architecture of AMQP is shown in figure Fig. 4.8.

Communications are handled by two main components: exchanges and message queues. Exchanges are used to route the messages to appropriate queues. Routing between exchanges and message queues is based on some pre-defined rules and conditions. Messages can be stored in message queues and then be sent to receivers. Beyond this type of point-to-point communication, AMQP also supports the publish/subscribe communications model.

Messaging capabilities of AMQP are handled in the layer of messaging, realized on top of its transport layer. AMQP defines two types of messages: bare massages that are supplied by the sender and annotated messages that are seen at the receiver. In Fig. 4.9 the message format of AMQP is shown [3]. The header in this format provides the delivery parameters including durability, priority, time to live, first acquirer, and delivery count.

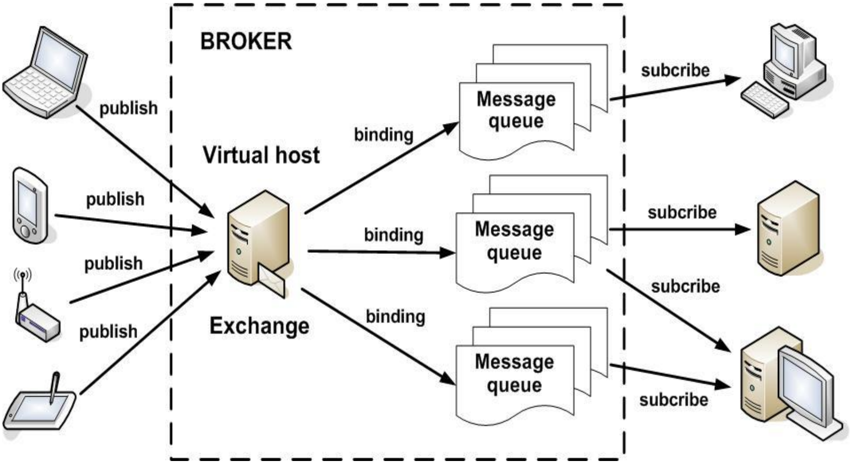


Fig. 4.8 – Architecture of AMQP

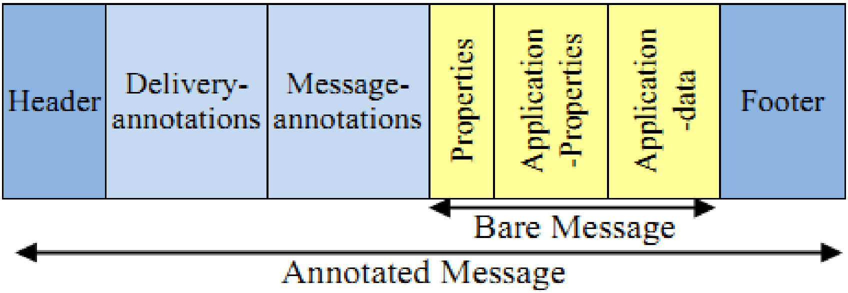


Fig. 4.9 – AMQP message format

The message queue in the AMQP store messages in disk or memory and routes to consumer applications. The message queue is act as storage and distributors of messages. Each message queue is independent to one another. The paramount properties of message queue are private/shared, durable/temporary and client/server.

Predicated on the properties, the user can utilize the message queue to deploy the standard middleware entities: store and forward Queue, private reply Queue and private subscription queue. A store and forward queue holds messages and distribute the messages between consumers on round robin substructure.

These queues are very flexible and durable while messages shared between multiple consumers. Private reply queue holds and forward messages to single consumers. It is an ephemeral queue, server denominated and private to only one consumer. A private subscription queue holds messages collected from amassed sources and forward to a single consumer.

The transport layer of AMQP complements the messaging layer. In this layer, communications are frame-oriented. The structure of a frame is depicted in Fig. 4.10 [3].

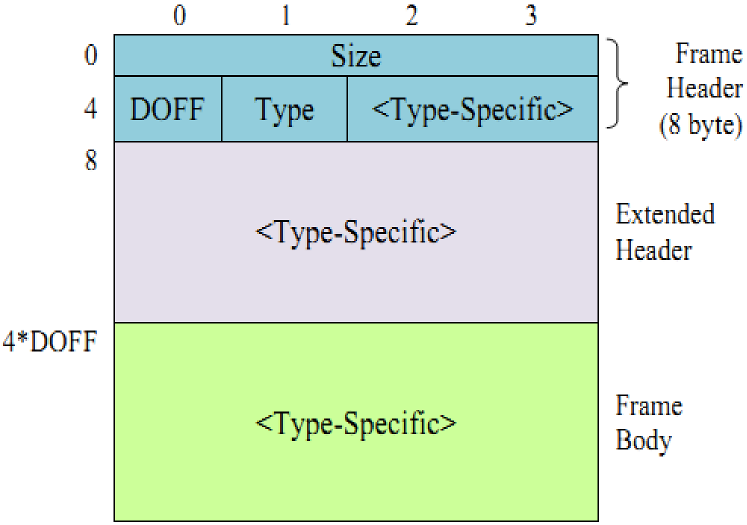


Fig. 4.10 – AMQP frame format

The first four bytes show the frame size. DOFF (Data Offset) gives the position of the body inside the frame. The Type field indicates the format and purpose of the frame. For example, 0x00 is used to show that the frame is an AMQP frame or type code 0x01 represents a SASL frame.

AMQP pros.

1. Store-and-forward feature in AMQP ensures reliability even after network disruptions.
2. This AMQP protocol is an open standard and interoperable messaging protocol.
3. AMQP provides reliable Quality of Service like at-most-once, at-least-once, exactly once.
4. AMQP is a secured protocol that is handled by SASL/TLS (authentication and security layer) in application layer.

AMQP cons.

1. AMQP is not reliable for lower bandwidths but can improve reliability with increase in bandwidth.
2. This protocol is not constrained and light-weighted protocol.
3. It does not support an automation discovery mechanism.

4.2.4 Comparison

Table 4.3 presents analysis of such data protocols as MQTT, AMQP, XMPP, DDS, CoAP and MQTT-SN [12].

Table 4.3. – Summary of data protocols in IoT

Protocol	Characteristics	Working	Advantages	Disadvant.	Applications
MQTT	Low power usage, M-M communication	Pub-Sub based protocol, main aim to collect data and transport to IT infrastructure	Save power and memory, Low power usage	Long-lived TCP connection, topic names are long strings	Home automation, Enterprise level applications
XMPP	Channel encryption and presence checking	Allows internet users to send instant messages	Secure, Service discovery, Very Robust, powerful	Data flow is more than XMPP server, lack world wide support	Instant Messaging, Group chat, Gaming, Vehicle Tracking

AMQP	Message queuing and interoperable	Designed to support wide variety of messaging and communication patterns	Highly reliable, Store & forward communication	Works at higher bandwidths only	Business Messaging, and in Banking Industry
DDS	Interoperable, data service with high performance	To connect one device to other device and also to share right data at the right place	Interoperable, saves bandwidth, flexible and reliable	Have no scalability	Medical Imaging, Military Systems, Hospital Integration, Farms
CoAP	Synchronous request response, 1-1 or M-M communication	Used in simple electronic devices that permits them to communicate interactively over a network	1-1 communication, M-M communication, Resource discovery	Less standard, not more mature and standard compared to MQTT	Smart homes, smart grid, Building automations
MQTT-SN	Light weight and Publish subscribe messaging protocol	Has been adapted for better function of devices where low power device usage is a primary concern	Open source, many-to-many communication protocol	Lacks support in Labeling messages which makes it difficult	Enterprise applications

As we can see, each protocol is better on its way depends upon its applications. Nevertheless it can be recommendable to determine further evaluations of performance metrics and appropriate qualitative interpretations for additional M2M protocols that can be applied in IoT.

Further, it is recommendable to determine further evaluations of performance metrics and appropriate qualitative interpretations for additional M2M protocols.

4.3 Cloud architecture for IoT

There are several aspects that apply to IoT systems that affect their architecture and implementation, thus to choose the cloud, as follows: scalability, big data, real time, highly distributed, heterogeneous systems, security, privacy and cloud computing [13]

The cloud components of IoT architecture are positioned within a three-tier architecture pattern comprising edge, platform and enterprise tiers [14].

1. The Edge-tier includes Proximity Networks and Public Networks where data is collected from devices and transmitted to devices. Data flows through the IoT gateway or optionally directly from/to the device then through edge services into the cloud provider via IoT transformation and connectivity.

2. The Platform tier is the provider cloud, which receives processes and analyzes data flows from the edge tier and provides API Management and Visualization. It provides the capability to initiate control commands from the enterprise network to the public network as well.

3. The Enterprise tier is represented by the Enterprise Network comprised of Enterprise Data, Enterprise User Directory, and Enterprise Applications. The data flow to and from the enterprise network takes place via a Transformation and Connectivity component. The data collected from structured and non-structured data sources, including real-time data from stream computing, can be stored in the enterprise data.

One of the features of IoT systems is the need for application logic and control logic in a hierarchy of locations, depending on the timescales involved and the datasets that need to be brought to bear on the decisions that need to be made.

Some code may execute directly in the devices at the very edge of the network, or alternatively in the IoT Gateways close to the devices. Other code executes centrally in the provider cloud services or in the enterprise network.

The term “edge computing” is sometimes applied to the case where code executes in the IoT Gateways or the devices. This is sometimes alternatively called “fog computing” to contrast with centralized "cloud computing", although fog computing can also contain one or more layers below the cloud that each could potentially provide capabilities for a variety of services like analytics. This design allows flexibility in how connectivity and services are designed for optimization and resiliency.

4.3.1 Traffic management principles

Methods of traffic management should take into account the features of the management of hierarchical systems and be based on the following principles of traffic distribution management [15]:

- the principle of decomposition;
- the principle of coordinating subnetworks operation;
- the principle of correlation the objectives of subnet management.

The traffic management includes:

- a set of interconnected network elements,
- the system of monitoring the network state,
- a set of means for managing the configuration as a response to the current state of the network, and enables taking actions that prevent unwanted future states using the prediction of the state and trends of traffic development.

In cloud computing, all IoT devices are directly connected to the cloud and computation totally depends on the cloud. However, all the above similar technologies do not exclusively depend on the cloud, but depend on some intermediate devices for computation; some of them do not even require a connection to the cloud [16].

Several computing paradigms exist, such as:

- Mobile Cloud Computing (MCC),
- Mobile-Edge Computing (MEC),
- Edge Computing (EC),
- Dew Computing (DC),
- Fog computing (FC),
- Fog-dew computing (FDC),

which use computing resources near underlying networks, located between the traditional cloud and edge devices, to provide better and faster application processing and services.

Fig. 4.11 shows the high-level architecture of these technologies.

4.3.2 Traffic management tasks and parameters

The central function of traffic management is efficient management of bandwidth due to the optimal assignment of traffic to switching nodes.

Currently, different methods of traffic management are used in information and communication networks. Most of them assume the possibility of external parameterization, that is, the transmission of traffic parameters directly to use control algorithms [17].

Some methods, such as, for example, the method of multiprotocol label switching of packets allow the modification or replacement of management algorithms that are a part of the management technology that is being implemented [18].

Two levels of data flow managing for control data transmission activity are available in existing communication networks.

The upper level controls access when a data transfer request is received.

The lower level of management involves using management algorithms where the estimates for the network traffic parameters obtained due to the methods that take into account the features of the data flows are transmitted.

To obtain the estimates of the transmission activity parameters at this level, the following methods can be used [19]:

- the method for estimating the size of the filtering buffers of the communication equipment;
- the method for synthesizing the stable estimation of the function of the traffic distribution density;
- the methods for managing the redistribution of virtual connection bandwidth taking into account priorities and competition among integral data streams.

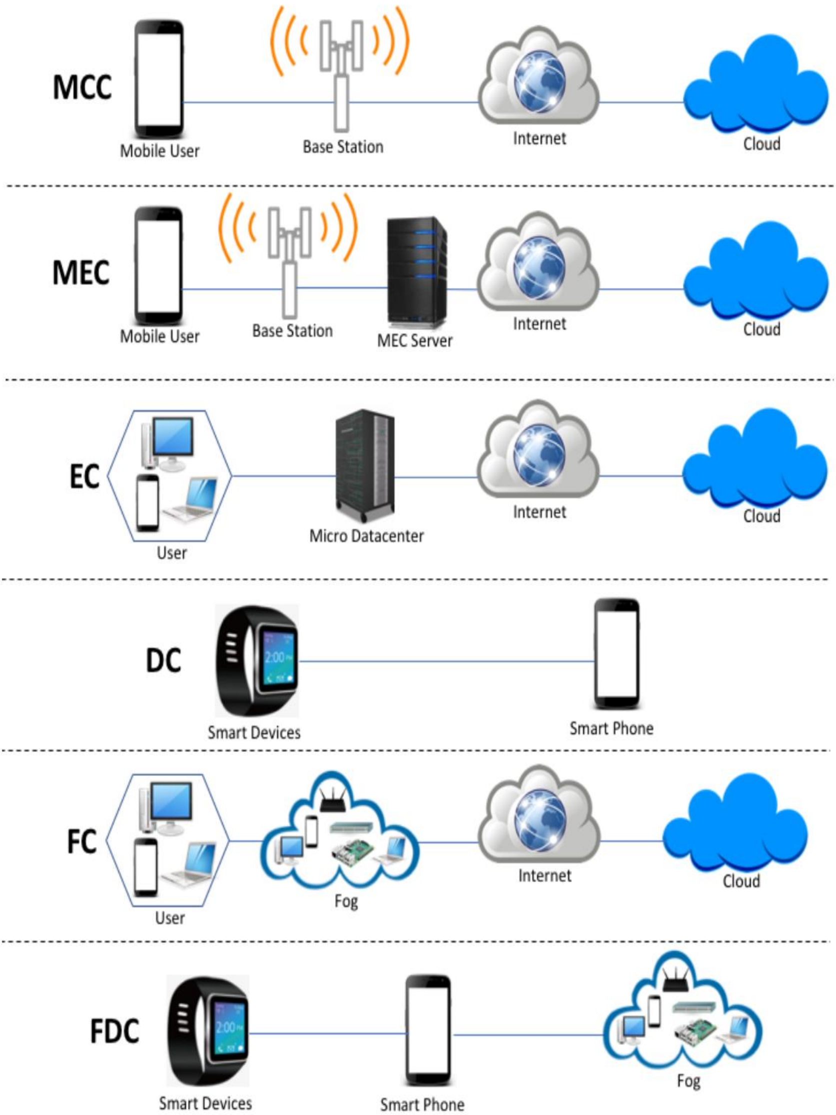


Fig. 4.11 – Architectures of MCC, MEC, EC, DC, FC and FDC.

Consistent application of the methods mentioned above, as well as similar methods, allows obtaining estimates of traffic control parameters.

4.3.3 Traffic types and related data services

Connection from the device to the IoT system is often a local network that connects the device with an IoT gateway – low power and low range in many cases to reduce the power demands on the device [13]. However, there are cases where the network connection is direct to the public network and no IoT gateway is required. In IoT systems, a wide range of alternative communication mechanisms are used which include local area networking using low-power, low-range methods, such as Bluetooth, BLE, and others. It may also include local area networking using Wi-Fi, to wide area networking using 2G, 3G, and 4G LTE.

Relationships for supporting IoT using cloud computing include such edge services as Load Balancers. These services provide distribution of network or application traffic across many resources (such as computers, processors, storage, or network links) to maximize throughput, minimize response time, increase capacity and increase reliability of applications. Load balancers can balance loads locally and globally. Load balancers should be highly available without a single point of failure. Load balancers are sometimes integrated as part of the provider cloud analytical system components like stream processing, data integration, and repositories.

4.3.4 Traffic management mechanisms

Such basic traffic management techniques for redistribution of the network resources can be highlighted [17]:

– method of statistical multiplexing;

– method of smoothing the data flow density;

– method of assessing the size of the filtering buffers of the communication equipment;

– method of synthesis of a stable estimation of the function of density of traffic distribution;

– methods for managing the redistribution of virtual connection bandwidth.

Methods of statistical multiplexing and method of smoothing the data flow density implement the technology and scientific principles of measurement, modelling, description and management of traffic to obtain the required characteristics. Possibility to smooth the profile of data flows traffic is an advantage of these approaches. Their drawback is that they do not take into account the properties of traffic, as when the peak values of data density occur, their commencement and short duration cannot be taken into account

Method of assessing the size of the filtering buffers of the communication equipment selects the optimal size of filter buffers for integral data flows that are served by a virtual channel. Thus it enables increasing the bandwidth of virtual channels.

Method of synthesis of a stable estimation of the function of density of traffic distribution analyzes the integral flow of fractal data. Its good feature so is to enable getting adequate assessment of control parameters.

Methods for managing the redistribution of virtual connection bandwidth is used while dynamic reserving the bandwidth. Its strong point is that he takes into account priorities and competition among integral data flows.

The three latter techniques have the same lack – they cannot be used for traffic management at the upper level of management (to govern the access when receiving a query for data transfer).

4.4 Effective data transmission speed in IoT networks

In the triad confidentiality, integrity and availability, most of the experts pay their attention to the informational confidentiality problem. This is because informational compromise leads to the most significant business damages. However, the impact of informational integrity on cybersecurity in business continuously increases.

Every year the number of devices which are connected to the Internet rapidly grows. The popularity of social networks and instant messaging technologies leads to growth of transmitted traffic amount. For the purposes of load optimization of communication channels most of the applications which transmit textual information use algorithms for data compression. When using those algorithms, it is essential to pay attention to maintaining the data integrity. The changes made to one single bit of information can lead to inability to restore the whole message.

Martin Ruubel attaches significant importance to the IoT data integrity in his article [40]: Strangely all the security focus seems to be on privacy, as if the public disclosure of the contents of your fridge is something to be feared. We argue that integrity is by far the more important component of the CIA (Confidentiality, Integrity, and Availability) security triad. Privacy might cost you some embarrassment but integrity (of your medical devices, of your car's braking system, of your flight's altimeter, of your power supply) can easily cost you your life.

Based on all of the above, the integrity problem must be considered as highly important. It must be secured, not controlled. Using simple integrity control, message signature jamming may lead to full message rejection that will increase data processing from IoT devices time. That delay can be critical enough for some of the systems (life-support system, secondary breaking system).

4.4.1 Effective speed evaluation

It can be used following definitions to evaluate effective data transmission speed in packet networks, provided all devices operate correctly [42]:

$$R_e = f(R_0, V_k, n_p, t_r, \varepsilon, P_e, z, K_p) \quad , \qquad (4.1)$$

where R_0 - data transmission speed that has been proved theoretically; V_k - code speed; n_p - data packet length; t_r - signal propagation speed in relation to time for packet analyzing and negative acknowledgement; P_e - unit data element error probability; z - negative acknowledgements count, K_p.- binary value for noise-immune codes use during message transmission fact.

Take the case where packet has length n_p and contains k of informational elements ($V_k = k/n_p$), and P is packet error probability.

Then average packet transmission time can be defined as below considering z possible negative acknowledgements:

$$t_l = T_p \sum_{i=1}^{z} P_i \quad , \qquad (4.2)$$

where T$_p$ – non-recurring packet transmission time.

p^z- transmission bus rejection due to jamming probability in relation to time for restoring T_v. Considering these comments, effective data transmission speed can be defined as:

$$R_e = V_k n_p \left[T_p \sum_{i=1}^{z-1} P_i + p^z (T_p + T_v) \right]^{-1} . \qquad (4.3)$$

The following expression is right if noise-immune codes (K$_p$=1) that can identify errors is used in transmission:

$$P \approx P(\geq 1, n_p) = P_e n_p^{\varepsilon} \quad , \qquad (4.4)$$

where $P(\geq 1, n_p)$ - is distortion of one and more elements in packet on length n_p possibility.

If we substitute expression (4.4) in expression (4.3) obtaining expression:

$$R_e = V_k R_o \frac{n_p (1 - P_e n_p^{\varepsilon})}{(R_o t_A + n_p) + R_o T_v (P_e n_p^{\varepsilon})^z} . \qquad (4.5)$$

where $1 - P_e n_p^{\varepsilon}$ - is rate of decreasing R_e due to noise effect.

Expression (4.5) demonstrates an effect of general factors, that effect on R_e decreasing. Multiplier $1 - P_e n_p^{\varepsilon}$ represents noise effect on R_e decreasing. Summand $R_o t_A + n_p$ in denominator represents loss amount of R_e due to time that receiver has spent on message analysis and time that sender was listened for confirmation of acceptance t_A. Second summand $R_o T_v (P_e n_p^{\varepsilon})^z$ specifies amount of losses R_e that are caused by possible noise effect and exceeding maximum permitted negative acknowledgements amount z.

Analysis of expression (4.5) shows that depending on packet length effective speed has maximum, which value depends on R_o, V_k,

t_A, z, P_e, ε and T_V. Based on service profile, the R_e value defines real network devices throughput and packet transmission time, as well as the effect of noise on network. ParameterR_e binds load stress parameter to service quality. The maximum effective transmission speed R_{emax} can be achieved at some value of optimal packet length n_{popt}, and is defined from expression $dR_e/dn_p = 0$ that doesn't have analytical solution relatively to n_p even if hardware had high reliability.

Effective information transmission speed R_e has (depending on n_p) maximum value R_{emax}, that reduces to n_{popt} values with the increase of R_o. Extremum of $R_e = f(n_p)$ dependence is the is sharper when R_o is greater. This reflects less criticality of n_p choosing when R_o decreases.

4.4.2 Errors grouping factor

Error grouping can be defined as error grouping rate or grouping factor [43] in communication. The factor is defined statistically for every type of data transmission networks. The grouping factor has value in range 0≤$\mathcal{E}$ ≤1. Value of $\mathcal{E}$ =0 corresponds to independent errors distribution. Errors are grouped in packets, when grouping rate increased, and when the rate is maximum ($\mathcal{E}$ =1), errors are grouped in one packet. According to errors grouping rate, there are channels with low, middle and high grouping rate defined, where $\mathcal{E}$ ≤0,3; 0,3<$\mathcal{E}$ ≤0.5; $\mathcal{E}$ >0.5.

4.4.3 Messages transmission quality parameters in international network

According to [43] there are defined such BER parameters in international network as normal – BER<10^{-6}; low – 10^{-6} ≤BER< 10^{-3} (accident-sensitive state); unallowable – BER ≥ 10^{-3} (accident state).

It must be said that international standards (except BER) define other channels quality parameters [43]: Errored Second (ES) и

Severely Errored Second (SES). Errored Second Ratio (ESR) and Severely Errored Second Ration (SESR) are also been used. ESR and SESR, are defined as ratio of number errored seconds and number of severely errored seconds to the total number of seconds in the measurement. ESR=0,02, and SESR=0,001 in modern networks [43]. Adaptive algorithm that was suggested in subsection 4.4.1 can greatly effect on the channels quality parameters that was specified in subsections 4.4.2 and 4.4.3.

4.5 Analysis of error model in IoT network

Just before the development of noise-immune codes it is essential to conduct the mathematical modeling of communication channels and to analyze the behavior of interference in the communication channels taking into account different input parameters and negative impact of cybercriminals.

Considerable amount of developed models uses clearly mathematical approach for describing the error flow. The models are: the Berger-Mandelbrot model [44], the Brusilovskiy model [44] and the Aksenov-Voronin model [45]. In all of these models the physical part of processes, which happen in the communication channels, is ignored. The mechanism of group errors generation is implicitly expressed.

In order to get more realistic results it is better to build the mathematical model of errors flow using the mathematical concepts, which are close to real world physical properties, which occurs in the communication channels. There are a lot of models, which consider physical concepts to some extent, whereas those concepts lead to transmitted data corruption. The best known models are: the Hilbert model [44], the Eliot-Hilbert model [44], the Frichman-Svobody model [45], the Freulich-Bennet model [44] and the Popov-Turin model [44].

Having analyzed the existing error models, the authors came to a conclusion about the most precise models which were built on multidimensional distribution of Freulich-Bennet. The authors conducted an experiment during which there was defined the dependency of amount of errors in the message on communication channel characteristics and on the message length. For this experiment authors programmatically implemented the described above error models with parameters, which correspond to the real characteristics of communication channels. During the experiment there was calculated the amount of blocks, in which there was found at least one error. There was also found the possibility of obtaining more than one error in the 4 and 8 bit blocks and the amount of such mini blocks in the transmitted message. This was presented in a previous work [46].

The results of modeling the communication channel [46] using the proposed error grouping models show the possibility to use the way of restoring information during data exchange in the telecommunication systems, described in [41]. The usage of mini blocks of 4 and 8 bits for noise-immune encoding simplifies the encoding and decoding algorithms, which has a major impact on energy consumption. It is better to use models of independent errors generation. Those are considered to be "tougher" in comparison to the proposed models listed above.

In the modern communication systems the usage of turbo codes (class of high-performance forward error correction codes) with algorithms of soft decoding has become more popular. It shall be taken into account that the studies conducted on turbo encoding principles with parallel work of coder devices unveiled a number of flaws of such systems. The most significant of them are: a large number of iterative transformations for potentially possible results, signal delays during the implementation of symbol intersection procedures, long length of code combinations and the difficulties of implementation of the parametrized adaptation. However, in most of the communication systems it is more rational to use other coder device constructions, the ones which implement short encoding. Nevertheless, those constructions are still required to provide the high level of authenticity of processed information.

4.6 Noise-immune codes speed comparison procedure

One of the most popular code classes is BCH codes (Bose-Chaudhuri-Hocquenghem codes) [47]. This class is a class of great variety of cyclic codes which are used for informational protection to avoid errors. Also, they are used as a first cascade of turbo codes. These codes can be built with predefined adjustment properties. Precisely, they can have predefined minimum code distance, which helps to detect the amount of errors and corrects them [47]. Meanwhile, the predefined code properties can lead to excessive code redundancy and bigger energy consumption for encoding/decoding devices, in such situation when the number of obstacles in the communication channel is less than theoretically calculated maximum. That is why this is not the best idea to use such codes in the energy saving systems. The authors propose to compare the speed of BCH codes with codes described in [41]. For these purposes there was developed the algorithm for comparison of encoding/decoding speed described in [47]. The results of this algorithm's work are shown in the Table 4.4.

Looking at Table 4.4, it is obvious that the least amount of time is needed for generation of sequence with check bits of informational code sequence. The average amount of time is needed to form checksum. Altogether, both of these procedures take 1 sec. BCH code matrices generation takes a lot more time – 3.6 secs on average.

Table 4.4. – Average time of work of algorithms with different probabilities

Probability (%)	Average time of checksum generation (s)	Average time of verification sequence generation (s)	Average time of BCH code generation (s)
100	0,783	0,157	3,604
95	0,78	0,145	3,6
90	0,777	0,134	3,23

When using the code tables described in [47] it is reasonable to ignore the time of checksum generation because it has already been taken into account in the code tables. It means that the time of noise-immune code generation takes only a fraction of a second, which is way faster than the time of BCH code generation. It is apparent, that the algorithm described in [9] is much more energy efficient, because it takes less time to be processed on CPU.

The main task is to develop the energy efficient noise-immune codes. The purpose of the study is to gain reasonable results from estimation of the energy efficiency properties of noise-immune codes in the different implementations and communication channels.

The authors conducted the theoretical research [47] on the topic of algorithm's ability to restore the message in the real world communication channels in order to prove its maintainability [41]. Obviously, the amount of received errors will be defined by channels quality, transmitting message length and type of errors occurring in the channel. In [46] it is shown, that using this method, there is a high probability to introduce all errors as single and independent.

To prove that this method can be used in [47] there was formulated and proved the following theorem.

Theorem 1. In order to restore the initial message which is being transmitted and has only one error it is sufficient to successively change 4 bits in the distorted tetrad.

Conclusion 1. It is enough to change 4n bits with maximum number of changes 4n for restoring the initial message with n errors.

However, the simultaneous distortion of 2 bits in the 4 bit sequence can lead to collisions. In the work [47] the authors proposed the mechanism for checksum generation and formulated the theorem which proves the efficiency of checksum usage.

Theorem 2. The modification of any two elements in the sequence of 4 bits which satisfies the paired relationship check does not lead to collisions [47].

In order to achieve the desired goal there were solved the following tasks like: conduction of analysis of existing methods of efficiency estimation of noise-immune codes; development of complex efficiency assessment factor of noise-immune codes which indicates the estimate of energy efficiency of codes usage in the different communication channels having different configurations of encoding and decoding tools.

Moreover, the task of comparative assessment can only be solved using the complex criteria. This criteria has to include the following: the minimization of power consumption, maximization of number of symbols, which can be rectified in the chosen code, minimization of amount of correcting symbols, minimization of time for noise-immune message generation and its refresh etc. [46]. Furthermore, in our opinion, it is worth of taking into account the system’s reaction in case of impossibility to restore the received message and the generation of repeated request for information transmission.

The generation of energy efficient noise-immune codes is an important task because sophistication of algorithms of noise-immune encoding leads not only to complex effectiveness evaluation of their usage (especially during the energy efficiency assessment of different implementations of the same code) but also to impossibility of their usage in such communication systems, where energy consumption is critical.

4.7 Research of noise-immune codes energy efficiency in IoT

The main objective of the research is getting sufficient results in estimating the energy efficiency of the noise-immune codes use in various implementations and communication channels.

To achieve the objective following tasks have been accomplished:

– existing estimating methods of noise-immune codes efficiency have been analyzed;

– complex index of estimating of noise-immune codes efficiency has been developed. The index includes energy efficiency estimating depending on use of codes in various communication channels and with various encoding and de-coding devices.

When the methods of estimating of noise-immune code use efficiency are developed, it’s necessary to use complex efficiency indexes that must consider the following statements:

– ability to design codes for any communication channel, not only behavior of codes in real system;

– ability to abstract from real hardware platform when using complex indexes;

– have ability to compare code speed both in software and hardware implementation;

– the energy efficiency index in complex index on every code estimating state.

Following complex index is suggested for modern noise-immune codes efficiency analysis:

$$K_{к} = K_{w1}\frac{K_1}{K_1+K_2} + K_{w2}\frac{F_{cpu}}{t_{\Sigma}c} + K_{w3}\frac{F_{cpu}}{t_{\Sigma}dc} + K_{w4}\frac{Ko_{sn}}{Ko_{tot}} + K_{w5}\frac{K_{er}}{K_{er}+K_{fer}} + \\ + K_{w6}\frac{K_{\text{мах}}}{K_{tot}} + K_{w7}\frac{P_{nr}}{P_{st}} + K_{w8}E_b(K_1+K_2) \quad (4.6)$$

where Кw1, Кw2, Кw3, Кw4, Кw5, Кw6, Кw7, Кw8 – are weighting factors that are defined by experts;

К1 – the number of informational symbols;

К2 – the number of checking symbols;

F_{cpu} - CPU frequency that is measured in cycles;

$t_{\Sigma}c$ - total time, that is needed to complete encoding operations of noise-immune encoding algorithm with equal data packets, that is measured in CPU cycles;

$t_{\Sigma}dc$ - total time, that is needed to complete decoding operations of noise-immune encoding algorithm with equal data packets and communication channels, that is measured in CPU cycles;

Кosn – the number of operations in the algorithm, that are completing simultaneously;

Кotot – total number of operations in the algorithm;

Кer – error number, that have occurred as a result of transmission through communication channel;

Кfer – the number of errors, that have been proved theoretically and fixed;

K_{tot}- total number of bits for transmission using the algorithm;

K_{max}– maximum possible number of error bits in the algorithm that could be corrected;

P_{nr}– transmitter's capacity during use of the noise-immune code;

P_{st}– standard transmitter's capacity;

E_b– noise-immune code energy efficiency per bit of transmitted information.

The complex efficiency index consists of eight indexes. Let us look more closely at indexes that are responsible for energy efficiency:

a) $\frac{P_{nr}}{P_{st}}$ – practical index, that is formed based on transmitter's capacity in regular mode and with use of chosen noise-immune codes. The index shows energy gain rate when used on specific communication channel;

b) $E_b(K_1 + K_2)$– practical index, that is formed based on energy cost of encoding and decoding of one bit of information per total number of transmitted data.

The novelty of obtained results is the following.

1. The methodology of use of noise-immune codes in IoT communication channels efficiency and energy efficiency estimation, that is based on use of complex efficiency indexes, that combine precise mathematical efficiency and energy efficiency in IoT communication channels estimation abilities and methods of expert estimation, that correlate efficiency of noise-immune systems embedding results in various communication channels. The most critical noise-immune codes' parameters in respect to energy efficiency were analyzed too;

2. The methodology of noise-immune encoding estimation according to results of mathematical errors modelling in communication channels gained further improvement;

Practical relevance of received results lies in the following:

– They let us raise estimation completeness of energy efficiency of various noise-immune codes embedding in IoT infrastructure;

– They are base for developing of informational security systems that have integrity control and can analyze the energy efficiency of its use.

4.8 Code tables use for data transmission in IoT infrastructure

The most important factor is encoding block size, when using encoding in IoT. In considering code tables, which are shown in [47], the most relevant encoding block size is 4 bits [41]. The main argument for choosing that size is concept of restoration of bits that where received by mistake – a large controlled block size results large variants must be scanned.

As a result of modelling [41] it was discovered, that the probability of existing controlled blocks with more than 1 mistake is not higher than 0.02, when transmitting 1 KB message. And the probability of mistake in one symbol being 10-3. At the same time the average errored controlled block count was a little more than 8 and the maximum count was 18.

Based on experiment results [47], it appears that grouped errors become single if controlled blocks are small enough. From this point of view 6-bit code tables use is optimal in channels with error probability $P_{ош}≈10^{-3}$.

From the point of view of data structuring in computers we must divide the introduced bit sequence on two parts for 3 bits in each and add one parity bit. Therefore, a pattern, that includes controlled blocks size in 4 bits, may be built. Also 8-bit sequence is suitable for further processing and transmission.

4.8.1 The procedure of setting up the code tables

The procedure of setting up the code tables provided that 6 bits are used per symbol encoding plus 2 bits for noise-immune encoding is shown in [47]. Let us analyze typical code table of provided method. Codes of translating to different alphabet are specified in Table 4.5. They are used in method specified in [41]. Codes of translating to different alphabet have the highest interest, because errored decoding that codes leads to irreversible losses and unavailability to decode whole message text. For example, when you send 111100 (translating to alphabet 4) the fourth bit would be errored and we receive 111000 (translating to alphabet 3). Because of it, all following transmitted messages would be decoded incorrectly and repeated symbols transmission or repeated decoding of previously received messages would be needed because of verification control. The mistake could be solved by 2 control bits, but if the mistake will occur in two mini-blocks 4 bits each simultaneously it would be impossible to correct the message.

Table 4.5. – Code symbols for different alphabet translation

Code table index	Binary view	Alphabet symbols
51	110011	Translation to alphabet 1
52	110100	Translation to alphabet 2
56	111000	Translation to alphabet 3
60	111100	Translation to alphabet 4
63	111111	Translation to alphabet 5

Method that is described in [41] has other sufficient disadvantages except listed above:

– alphabet translation symbols are duplicated in all code tables, that reduces symbols number, that could be selected for other symbols significantly;

– limited number of notes for translation symbols to another alphabet can be separated, that reduces simultaneously used alphabets number;

– minimal code distance between alphabet translation symbols is 1, which does not let to discover single error. If checksum is used the minimal code distance will be increased to 2, but that would not let us to discover pair errors.

4.8.2 The procedure of setting up the translation table

To eliminate defects listed above the authors have suggested method consisting in forming separate code table with symbols of translation to different alphabet. The method consisting in creating special table, that contains codes of all alphabets that are used in system, instead of putting code of translation to another alphabet in every code table. In every alphabet exist single combination for translation to code table.

The authors suggest using combination that are filled with zeros for translation code security increasing. Using one transition combination allows you to allocate additional space in tables that can be used for the most commonly used symbols or special commands of IoT devices. This also significantly reduces the probability of wrong decoding of the transition combination due to interference impact.

After receiving that combination decoder would know that following byte should be decoded from another code table. Code tables for 9 alphabets example is reviewed in Table 4.6.

Minimal code distance for every combination without control bits is 2, as shown in Table 4.6. That let us to discover single errors. In case of two control bits use, code distance increases to 4. That factor let us consider that using the suggested code table for translation to different alphabet symbols allows us to dis-cover double error. This fact let us increase translation to different alphabet symbols decoding validity and decrease probability of repeated transmission because of wrong symbols decoding.

It should also be noted that 9 alphabets, that are reviewed in Table 4.6, almost twice the number of alphabets that used in the method tables [41]. Using suggest-ed encoding method let us increase the number of simultaneously used alphabets or symbols significantly.

This let us use encoding procedure suggested in [41] in international data transmission systems and with some national alphabets more effectively.

Table 4.6. – Code table for translation to different alphabet symbols

Code table index	Binary view	Alphabet symbols
0	100001	Translation to alphabet 1
1	001100	Translation to alphabet 2
2	110000	Translation to alphabet 3
3	110101	Translation to alphabet 4
4	111111	Translation to alphabet 5
5	011110	Translation to alphabet. 6
6	101101	Translation to alphabet 7
7	010010	Translation to alphabet 8
8	000011	Translation to alphabet. 9

Another advantage of suggested procedure in comparison with the one stat-ed in [41] is less data redundancy in every code table as far as only one code combination, which is translation to alphabet table, is repeated in every table.

4.9 Work related analysis

Besides references mentioned in this section, such publications can be recommended concerning communications and protocols for IoT: [20-22].

Comprehensive information about technical characteristics of IoT network technologies such as frequency band, range, data rate, battery life, topology and others can be found in [23].

Scientific approach to problems concerning IoT area can be found in recent publications (including mentioned) of such Institutions as University of Newcastle upon Tyne (UK), Royal Institute of Technology (Sweden), University of Coimbra (Portugal), Leeds

Beckett University (UK), The Institute of Information Science and Technologies (Italy): [8-10, 16, 24-34].

Cloud computing technology is successfully used for IoT recently. Nevertheless, being far from end-users, oriented to cloud computing IoT systems face several challenges including high response time, heavy load on cloud servers and lack of global mobility. It is inefficient in some cases to send large amount of data from IoT devices to the cloud, due to the high cost of communication bandwidth, and due to the high redundancy of data. Instead of moving information to the cloud, it may be more efficient to move the applications, storage, and processing equipment closer to the data produced by sensors. Fog and edge computing technologies are well suited to address this issue by moving the mentioned services closer to area where data is produced [35].

Fog computing is an emerging concept that puts the cloud services closer to the end users. Inheriting main concepts from cloud computing, fog provides computation, storage, and networking services to end-users, anywhere along the thing-to-cloud continuum, according to OpenFog Consortium.

The idea is to serve the requests that demand real-time and low-latency services at the fog, and to send the requests that demand permanent storage or require extensive analysis to the cloud. Due to the countless benefits of fog, the research in this area has been gaining attention, and researchers have recently started to define visions, basic notions, and possible architectures of fog computing [36-38]. Actual and detailed survey on edge computing is given in [39].

Conclusions and questions

IoT is based on the networking of things, so communication between objects is the most important topic in this area. This chapter focuses on connectivity techniques and protocols of IoT. Such topics as architecture of IoT networks, delays in wireless sensor networks, new version of Bluetooth and the use for M2M such protocols as MQTT, AMQP and HTTP/HTTPS are discussed.

When applying cloud technology for Internet of Things such it is necessary such a consideration to take into account. Cloud computing being far from end-users, suffer from some issues including high response time, heavy load on cloud servers and lack of global mobility.

It is inefficient in some cases to send large amount of data from IoT devices to the remote cloud resources, because of high communication delays and redundancy of data. Instead of moving information to the cloud, it may be more efficient to move the applications, storage, and processing equipment closer to the data produced by sensors. Fog and edge computing technologies were developed recently to address this issue.

The presented complex efficiency index lets us to provide a comparative analysis of noise-immune codes that are used in different communication channels. The index considers energy efficiency of code use both in the communication channel and during coding and encoding procedures.

Based on error models' experimental studies and speed of noise-immune codes, considering received data and using developed efficiency index it should be pointed out that presented noise-immune code meets requirements of green technologies and uses energy component as effectively as possible in most cases. Optimal table constructing let us to decrease the redundancy of transferred messages. New translation table inclusion let us to increase resistance to translating to different alphabet by mistake. This is achieved through code distance between translation table elements increasing.

Using the proposed methods will significantly increase the speed and efficiency of the transmission of control signals in the IoT infrastructure and choose the most appropriate method for securing data integrity at the design stage.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What basic network topologies can be the networking standards categorized into?

2. What are the advantages and drawbacks of each network topology?

3. What are the main reasons of delays in wireless network?

4. How can the area of the Bluetooth 5.0 be represented among the IoT applications?

5. Which IoT protocols of application level for M2M communication there are?

6. What is the main problem to use the standards HTTP/HTTPS for IoT application?

7. What elements does the MQTT protocol specification provide?

8. Which scheme does MQTT protocol utilize?

9. What pros and cons of MQTT protocol can be defined?

10. Which types of messages does AMQP protocol define?

11. What pros and cons of MQTT protocol can be defined?

12. Which tiers does the IoT cloud architecture consist of?

13. What are the alternatives to cloud computing?

14. What principles of traffic distribution are the methods of traffic management based on?

15. What methods can be used to obtain the estimates of the transmission activity parameters?

16. Which basic traffic management methods can be highlighted for the network resources redistribution?

17. What does effective data transmission speed in IoT communication mean?

18. Which are features of complex index (4.7)? Which attributes/parameters does it take into account?

19. How are code tables for data transmission in IoT used?

20. What does noise-immune encoding mean?

References

1. W. Stallings, "The Internet of Things: Network and Security Architecture", *Internet Protocol Journal*, vol. 18, no. 4, pp. 2-24, 2015.

2. ITU-T, "Overview of the Internet of Things", *Recommendation Y.2060*, 2012.

3. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys and Tutorials,* Article vol. 17, no. 4, pp. 2347-2376, 2015.

4. M. Pacelle, "3 topologies driving IoT networking standards. The importance of network architecture". [Online], Apr. 4, 2013. Available: http://radar.oreilly.com/2014/04/3-topologies-driving-iot-networking-standards.html [Accessed: 07 - Feb. - 2019].

5. R. V. P. Yerra and P. Rajalakshmi, "Effect Of Relay Nodes On End-to-end Delay In Multi-hop Wireless Ad-hoc Networks", *Proceedings 2013 IEEE 27th International Conference on Advanced Information Networking and Applications,* pp. 343-348, 2013.

6. S. Böcker, C. Arendt, and C. Wietfeld, "On the suitability of bluetooth 5 for the IoT: Performance and scalability analysis", in *28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2017*, 2018, vol. 2017-October: Institute of Electrical and Electronics Engineers Inc., pp. 1-7.

7. "Bluetooth Special Interest Group", Bluetooth *Core Specification v 5.0*, Dec. 2016.

8. J. Wåhslén and T. Lindh, "Real-time performance management of assisted living services for Bluetooth low energy sensor communication", in *15th IFIP/IEEE International Symposium on Integrated Network and Service Management, IM 2017*, P. Chemouil *et al.*, Eds., 2017: IEEE Engineers Inc., pp. 1143-1148.

9. L. Novelli, L. Jorge, P. Melo, and A. Koscianski, "Application Protocols and Wireless Communication for IoT: A Simulation Case Study Proposal", in *11th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2018*, 2018: Institute of Electrical and Electronics Engineers Inc.

10. J. Yoneyama, C. Artho, Y. Tanabe, and M. Hagiya, "Model-based network fault injection for IoT protocols", in *14th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2019,* G. Spanoudakis, E. Damiani, and L. Maciaszek, Eds., 2019: SciTePress, pp. 201-209.

11. OASIS Standard, "OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0". [Online], 2012. Available: http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf [Accessed: 07 - Feb. - 2019].

12. M. Anusha, B. E. Suresh, M. R. L. Sai, K. A. Vamsi, and Bhagyasree B., "Performance analysis of data protocols of internet of things: a qualitative review", vol. 15, no. 6, pp. 37-46, 2017.

13. Cloud Customer Architecture for IoT, "Cloud Standards Customer Council". [Online], 2016. Available: https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-IoT.pdf [Accessed: 07 - Feb. - 2019].

14. "The Industrial Internet Consortium's Industrial Internet Reference Architecture IIRA paper" [Online]. Available: http://www.iiconsortium.org/IIRA.htm [Accessed: 07 - Feb. - 2019].

15. V. G. Olifer and N. A. Olifer, *Computer networks. Principles, technologies, protocols,* 4-th ed. St. Petersburg: Peter, 2010.

16. R. K. Naha *et al.*, "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions", *IEEE Access,* Article vol. 6, pp. 47980-48009, 2018.

17. V. Kosenko, E. Persiyanova, O. Belotskyy, and O. Malyeyeva, "Methods of managing traffic distribution in information and communication networks of critical infrastructure systems", vol. 2, no. 2, pp. 48-55, 2017.

18. K. N. Qureshi, A. H. Abdullah, A. N. Hassan, D. K. Sheet, and R. W. Anwar, "Mechanism of multiprotocol label switching for forwarding packets in virtual private network", *Middle - East Journal of Scientific Research*, vol. 20, no. 12, pp. 2117-2127, 2014.

19. K. Angrishi, "An end-to-end stochastic network calculus with effective bandwidth and effective capacity", *Computer Networks,* Article vol. 57, no. 1, pp. 78-84, Jan 2013.

20. S. C. Li, L. D. Xu, and S. S. Zhao, "The IoT: a survey", *Information Systems Frontiers,* vol. 17, no. 2, pp. 243-259, Apr 2015.

21. R. H. Aswathy and N. Malarvizhi, "Internet of things: A survey on protocols and security risks", *International Journal of Engineering and Technology (UAE),* vol. 7, no. 1, pp. 15-20, 2018.

22. W. Burakowski *et al.*, "Traffic management for cloud federation", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. LNCS 10768, Springer Verlag, pp. 269-312, 2018.

23. J. Mocnej, A. Pekar, W.K.G. Seah, I. Zolotova, "Network traffic characteristics of the IoT application use cases. Technical report ECSTR18-01". [Online]. School of engineering and computer science, Victoria university of Wellington, 6 Jan 2018. Available: https://ecs.victoria.ac.nz/foswiki/pub/Main/TechnicalReportSeries/IoT_network_technologies_embfonts.pdf [Accessed: 07 - Feb. - 2019].

24. S. Mohamed, M. Forshaw, and N. Thomas, "Automatic generation of distributed run-time infrastructure for internet of things",

in *2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017*, 2017: Institute of Electrical and Electronics Engineers Inc., pp. 100-107.

25. B. Negash, T. Westerlund, and H. Tenhunen, "Towards an interoperable Internet of Things through a web of virtual things at the Fog layer", *Future Generation Computer Systems-the International Journal of Escience,* Article vol. 91, pp. 96-107, Feb 2019.

26. X. V. Wang and L. H. Wang, "A cloud-based production system for information and service integration: an internet of things case study on waste electronics", *Enterprise Information Systems,* Article vol. 11, no. 7, pp. 952-968, 2017.

27. X. V. Wang and L. H. Wang, "A cloud-based production system for information and service integration: an internet of things case study on waste electronics", *Enterprise Information Systems,* Article vol. 11, no. 7, pp. 952-968, 2017.

28. G. Marques, C. R. Ferreira, and R. Pitarma, "A System Based on the Internet of Things for Real-Time Particle Monitoring in Buildings", *International Journal of Environmental Research and Public Health,* Article vol. 15, no. 4, p. 14, Apr 2018.

29. P. Diogo, L. P. Reis, and N. V. Lopes, "Internet of Things: A System's Architecture Proposal", in *9th Iberian Conference on Information Systems and Technologies (CISTI)*, Barcelona, SPAIN, Jun 18-21 2014, NEW YORK: IEEE, in Iberian Conference on Information Systems and Technologies, 2014.

30. J. S. Silva, A. Loureiro, A. Skarmeta, and F. Boavida, "Special issue on management of IoT", *International Journal of Network Management,* Editorial Material vol. 28, no. 5, p. 2, Sep-Oct 2018.

31. T. Baker, M. Asim, H. Tawfik, B. Aldawsari, and R. Buyya, "An energy-aware service composition algorithm for multiple cloud-based IoT applications", *Journal of Network and Computer Applications,* Article vol. 89, pp. 96-108, Jul 2017.

32. V. Chang, V. Kantere, and M. Ramachandran, "Emerging services for Internet of Things", *Journal of Network and Computer Applications,* Editorial Material vol. 89, pp. 1-2, Jul 2017.

33. G. S. Jamnal, X. D. Liu, L. Fan, and M. Ramachandran, "Cognitive Internet of Everything (CIoE): State of the Art and Approaches", in *Emerging Trends and Applications of the Internet of*

Things, (Advances in Wireless Technologies and Telecommunication (AWTT) Book Series. Hersey: Igi Global, 2017, pp. 277-309.

34. S. Bergamaschi *et al.*, "Big Data Research in Italy: A Perspective", *Engineering,* Article vol. 2, no. 2, pp. 163-170, Jun 2016.

35. A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On Reducing IoT Service Delay via Fog Offloading", *IEEE Internet of Things Journal,* Article vol. 5, no. 2, pp. 998-1010, Apr 2018.

36. M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities", *IEEE Internet of Things Journal,* Review vol. 3, no. 6, pp. 854-864, Dec 2016.

37. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things", in *1st ACM Mobile Cloud Computing Workshop, MCC 2012*, Helsinki, pp. 13-15, 2012.

38. K. Liang, L. Q. Zhao, X. L. Chu, and H. H. Chen, "An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing", *IEEE Network,* Article vol. 31, no. 1, pp. 80-87, Jan-Feb 2017.

39. H. Bangui, S. Rakrak, S. Raghay, and B. Buhnova, "Moving to the Edge-Cloud-of-Things: Recent Advances and Future Research Directions", *Electronics,* Review vol. 7, no. 11, p. 31, Nov 2018.

40. Ruubel Martin, "Privacy and Integrity on the IoTs", [Online]. Nov. 22, 2013. Available: https://guardtime.com/blog/privacy-and-integrity-on-the-internet-of-things-if-all-you-have-is-a-pki-hammer-dot-dot-dot [Accessed: 07 - Jan. - 2019].

41. V. Y. Pevnev, M. V. Tsuranov, M. F. Logvinenko, "Noise-immune codes evaluation methodics", *Radio-electronic and computer systems: technical science,* Article vol. 5, pp. 165-170, 2016

42. V. G. Morozov, L. P. Purtov, A. S. Zamriy, "Experimental data generalization bases on possibility and error grouping index", *Communication devices technics*, Article vol. 4(2), pp.53-60, 1981

43. ITU-T, "Performance limits for bringing-into-service and maintenance of inter-national multi-operator PDH paths and connections", *Recommendation M.2100*, 2003.

44. S. Haykin, "Digital communication systems". Hoboken, NJ: J. Wiley & Sons. 2014.

45. R. Fano, "Transmission of information". Cambridge, Massachusetts: The M.I.T. Press. 1961.

46. V. Pevnev, Y. Novakov, M. Tsuranov, V. Kharchenko, "The Method of Data Integrity Assurance for Increasing IoT Infrastructure Security". *Proceedings of the 31th International Conference on Information Technologies, Sofia, Bulgaria*, pp.27-36. 2017.

47. V. Pevnev, M. Tsuranov, A., "Noise-immune encoding: The aspects of cybersecurity assurance". *Proceedings of the 9th IEEE International Conference on Dependable Systems, Services and Technologies, Kyiv, Ukraine,* pp. 248-252, 2018.

PART II. DATA SCIENCE FOR IOT AND IOE

5. FOUNDATIONS of DATA SCIENCE for IoT and IOE

DrS, Prof I. S. Skarga-Bandurova,
Dr. T. O. Biloborodova (V. Dahl EUNU)

Contents

Abbreviations

AI – Artificial intelligence

ANN – Artificial Neural Network

ARIMA – Autoregressive Integrated Moving Average

DF – Data Fusion

DL – Deep Learning

DOF – Degrees of Freedom

ECG / EKG - electrocardiogram

ID – Identifier

IoE – Internet of Everything

IoRT Internet of Robotics

IoT – Internet of Things

IMU – Inertial Measurement Unit

INS – Inertial Navigation Systems

IT – Information Technology

JVM – Java Virtual Machine

ML – Machine learning

QoS – Quality of Service

SN – Sensor Network

WSN – Wireless Sensor Network

5.1 IoT and IoE ecosystem

In most general form, an IoT ecosystem consists of the following essential components (see Fig. 5.1): (1) Smart devices and hardware, (2) Connectivity, (3) Cloud IoT platforms, (4) Services and Applications.

If we look at cycle transferring data from the IoT device to the c we can find the following data types:

1) Smart device identifier (ID);
2) Smart device location;
3) Sensor values.

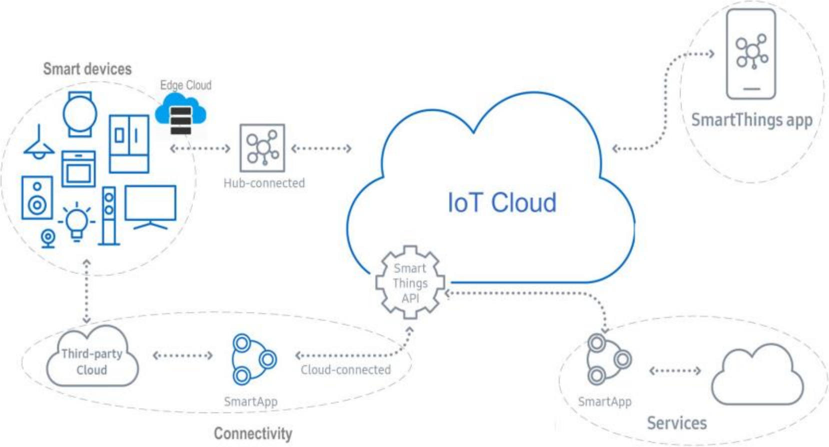


Fig. 5.1 – IoT and IoE ecosystem (adapted from [1])

This data is not only for the interaction between devices but also a data collected by all devices. An identifier enables to connect a device with further data, e.g. the ID of an NFC card establishes contact with data of the card holder and can be used for the access control. Location data provide tracking devices in a spatial context, thus enabling asset tracking. Finally, data produced by sensors make possible to gather information about the environment. The data from devices is collected and stored, e.g. in a log file, either on the device itself or on the controller. Collected data can be further processed, as detailed in Table 5.1.

Table 5.1 – Levels of data processing in an IoT scenario

Level	Description	Example
None	Data from sensors is just collected for further manual processing	Device Error Log
Group	Information from multiple similar devices is used together to detect spatial trends or outliers	Crop Monitoring
Temporal	Data from a device is collected over time which are not feasible to be done manually	Freight Tracking
Temporal Group	Historical information across multiple dimensions is collected for a group of devices	Smart Parking
Complex	Advanced aggregation supplements sensor input with additional data sources	Dashboard

Connectivity in IoT environments can be wired or wireless and includes different technologies like LoRa, WiFi , Bluetooth, Satellite, Cellular LTE-M, Sigfox etc.

Cloud IoT platforms can be deployed as the third-party clouds, edge platforms (which are also cloud hosted) or consolidated mega platforms. They are the places where data storage and processing are performed. Cloud IoT platforms should offer the next features:

– Extensiblity;
– Flexibility;
– Security;
– Data Intelligence.

The extensive usage of data in IoT ecosystem means that analytic models should be executed and delivered to any end point and in any environment – large or small.

According to [2], companies today need to write applications that respond to events such as:

– Device sensor sending in a data packet;
– Virtual Reality/Augmented Reality interface event;
– Alexa voice command;

- Inbound/outbound email or phone call from a customer;
- Chatbot response to a chat message from a customer;
- Social media events such as "like", "follow", or "share";
- Rest API generated event from an external system;
- Status changing on an account, customer record, etc.;
- Website post or get a request.

This means an extensive usage of data science best practices.

5.1.1. Data science for IoT vs. Traditional data science

There are at least ten differences data science for IoT from traditional data science, namely they are [3]:

- Working with hardware and radio layers;
- Edge processing;
- Utilizing specific analytics models in different IoT verticals;
- Preprocessing for IoT;
- Real-time processing;
- The need of synchronization and sensor fusion in IoT;
- Deep learning for IoT;
- Privacy, Insurance, and Blockchain for IoT;
- AI: Machines teaching each other (cloud robotics);
- IoT and AI layer for the Enterprise.

Working with the hardware and the radio layers means that IoT involves a range of devices and a variety of radio technologies. Each of the IoT layers has a specific set of smartdevices and radio technologies. For example, for wearable device we can easily use Bluetooth 4.0 while for Industrial IoT, we should choose the cellular technologies which guarantee Quality of Service (QoS).

Edge processing [4] means that we replicate processing and data storage closely to the data source. IoT devices gather vast amounts of data that need to be processed instantly, and then move these data to a centralized database. Edge processing will speed the time and quality of data analytics.

For example, a machine on a factory analyzes the quality of the parts of detail. If the part fails to meet the requirements, as determined by an optical scanner, then it should be automatically rejected.

Another example is the Airbus A350 that has about 6000 sensors and generates 2.5 Tb of data per day. All these sensors generate readings every couple of seconds on the operations and performance of a particular product.

If the jet engine has a place to house a Java Virtual Machine (JVM) and an analytic model (i.e., lightweight rules-based model), it means that the model can be executed right on the engine. In case when model can stream the data to a network, it can be run on a gateway or intermediate server. This means "execution" the pre-built modes at the edge, while actually build (test, refine, test, refine) the analytic models by bringing the detailed sensor data back to a central data and analytics environment (also known as Data Lake) [5].

Utilizing specific analytics models in different IoT verticals stem from the fact that different IoT layers operated different volume of data and real time implementations of the same models. It goes to show that the same models can be used for different purposes like sensor analytics, edge analytics and core analytics and along with this operates with various data volumes (see Fig. 5.2).

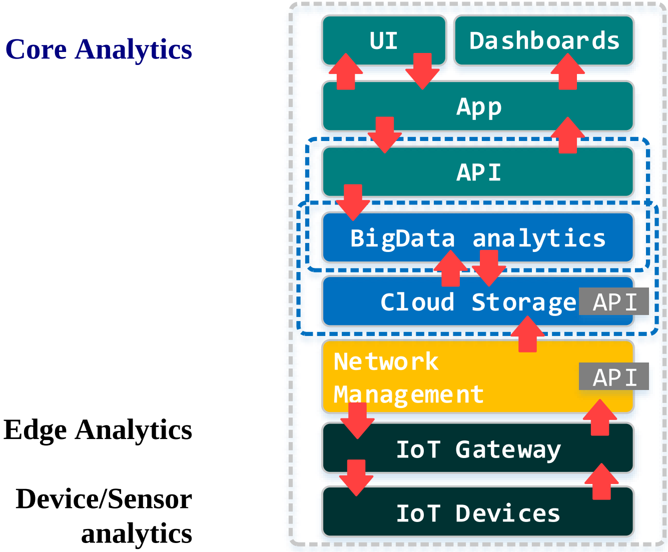


Figure 5.2 – Levels of analytics in IoT ecosystem

The IoT data lifecycle (see Fig. 5.3) starts from data production on smart devices then data get through aggregation, transfer, filtering and preprocessing, and finally to storage and archiving. Some operations like data collecting, filtering, fusion are performed online and considered as communication-intensive operations. Others, such as high-level preprocessing, long-term storage and archival, and in-depth processing/analysis are performed offline and considered as storage-intensive operations. The autonomous edges can provide real-time data to certain queries that why they are considered more communication-intensive than storage-intensive. [6].

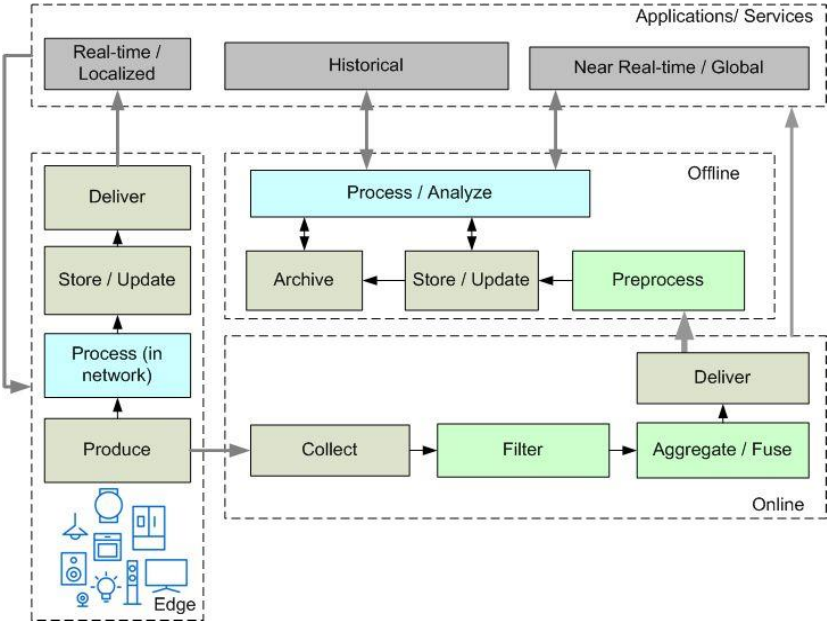
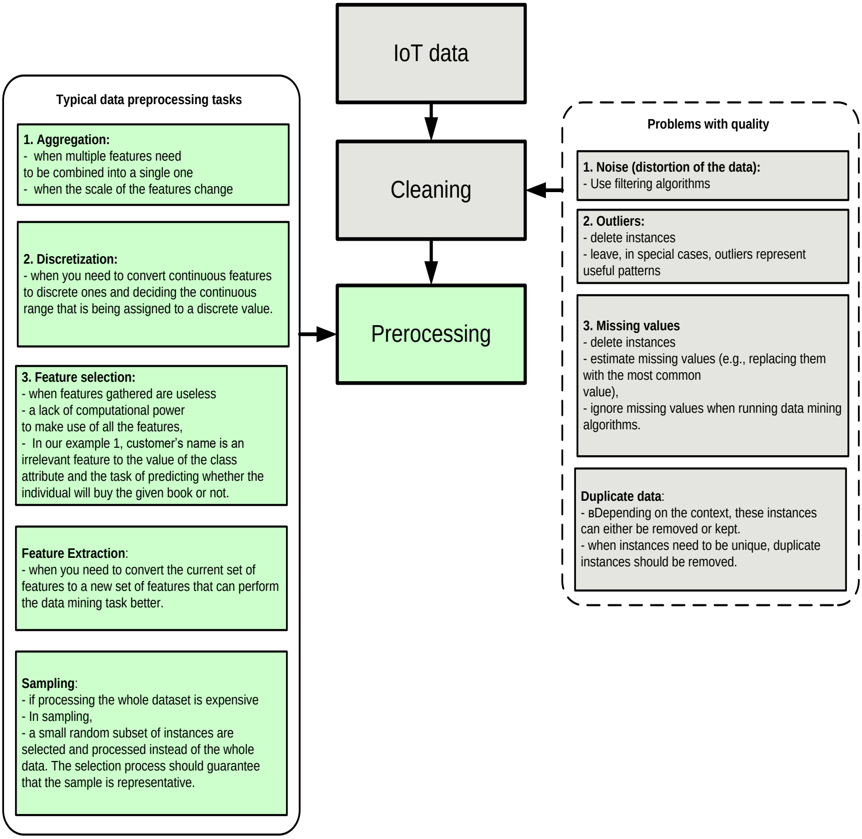


Figure 5.3 – IoT data lifecycle and data management (adapted from [6])

Preprocessing for IoT. Due to their nature, IoT data come from different sources and have varying formats and structures. That is why, in many cases, they need to be preprocessed. This procedure is necessary to handle missing data, remove redundancies, and integrate data from different sources into a unified format before being committed to storage. IoT datasets need a different form of preprocessing. Typical preprocessing operations are shown in Fig. 5.4. They include data cleaning (to filter out noisy data elements), interpolation (to cope with missing values), normalization (to deal with heterogeneous sources, temporal alignment), and formatting.



Typical data preprocessing tasks

1. Aggregation:
- when multiple features need to be combined into a single one
- when the scale of the features change

2. Discretization:
- when you need to convert continuous features to discrete ones and deciding the continuous range that is being assigned to a discrete value.

3. Feature selection:
- when features gathered are useless
- a lack of computational power to make use of all the features,
- In our example 1, customer's name is an irrelevant feature to the value of the class attribute and the task of predicting whether the individual will buy the given book or not.

Feature Extraction:
- when you need to convert the current set of features to a new set of features that can perform the data mining task better.

Sampling:
- if processing the whole dataset is expensive
- In sampling,
- a small random subset of instances are selected and processed instead of the whole data. The selection process should guarantee that the sample is representative.

Problems with quality

1. Noise (distortion of the data):
- Use filtering algorithms

2. Outliers:
- delete instances
- leave, in special cases, outliers represent useful patterns

3. Missing values
- delete instances
- estimate missing values (e.g., replacing them with the most common value),
- ignore missing values when running data mining algorithms.

Duplicate data:
- вDepending on the context, these instances can either be removed or kept.
- when instances need to be unique, duplicate instances should be removed.

Figure 5.4 – Typical preprocessing operations

Task-specific preprocessing also may be in demand to filter and clean data before meaningful operations take place.

The role of synchronization and sensory fusion in IoT. Sensor fusion involves combining data from different sensors and sources in such a way that the resulting information becomes less uncertain than in case if these sources were used separately. The term "less uncertain" in this case may mean more accurate, more complete, or more reliable. Sensor fusion has always played a key role in applications such as the aerospace industry:

In aerospace applications, accelerometers and gyroscopes are often combined into an inertial measurement unit (IMU) that measures orientation based on multiple sensor inputs, known as Degrees of Freedom (DOFs). Inertial Navigation Systems (INS) for spacecraft and aircraft can cost thousands of dollars due to the rigorous precision and drift tolerance as well as high reliability.

Recently, the merger of sensors is relevant in self-driving cars and drones, where inputs from multiple sensors can be combined to display more event information.

*Real Time processing and IoT***.** IoT performs many operations at real time involving both fast and big data. Hence, Real Time applications provide a natural synergy with IoT. Many IoT applications like Fleet management, Smart grid, Twitter stream processing etc. have unique analytics requirements based on both fast and large data streaming. These operations include [7]:

Real-time tagging: As unstructured data flows from various sources, the only way to extract signal from noise is to classify the data as it comes.

Real-time aggregation: Any time we aggregate and compute data along a sliding time window we are doing real-time aggregation.

Real-time temporal correlation needs for identifying emerging events based on location and time, real-time event association from largescale streaming social media data, etc.

Deep learning (DL) plays an important role in IoT analytics, namely fast and streaming data analytics, to support applications with high-speed data streams and requiring time-sensitive (real-time or near real-time) actions. DL models bring two important improvements for IoT analytics.

First, they reduce the need for hand crafted and engineered feature sets to be used for the training [8]. Consequently, some features that might not be apparent to a human view can be extracted easily by DL models. Second, DL models improve the accuracy of results of processing and analyzing timely big data applications.

*Privacy, Insurance and Blockchain for IoT***.** The blockchain is the perfect technology to make IoT principles widely used and to add a new property and application of datasets.

Blockchain logs the truth information; thus these datasets are entirely trustable [9]. It allows utilities to monitor every instance of every data structure created by an application and monitors all accesses, when the situation is freed, and when the memory in which it was stored was overwritten.

From the above reasoning, we assume that blockchain can improve the forensic investigation in smart environments and will enable to solve the different tasks of privacy, digital forensics and insurance.

AI: Machines teaching each other (cloud robotics)

Internet of robotics (IoRT) is a new emergency technology, mix of Cloud Computing, Artificial Intelligence (AI), Machine Learning and IoT.

According to ABI Research [10], IoRT is defined as, "Intelligent devices that can monitor events, fuse sensor data from a variety of sources, use local and distributed 'intelligence' to determine a best course of action." IoT enables to integrate multiple robots to perform one single task in highly cooperative and coordinated manner.

5.1.2 The IoT ecosystem and IoT challenges in data science

The deployment of IoT rapidly increases the amount of data at once offering the opportunity for the application and development of big data analytics.

The relationship between IoT and big data, which is shown in Fig. 5.5, can be divided into three steps to enable the management of IoT data [11].

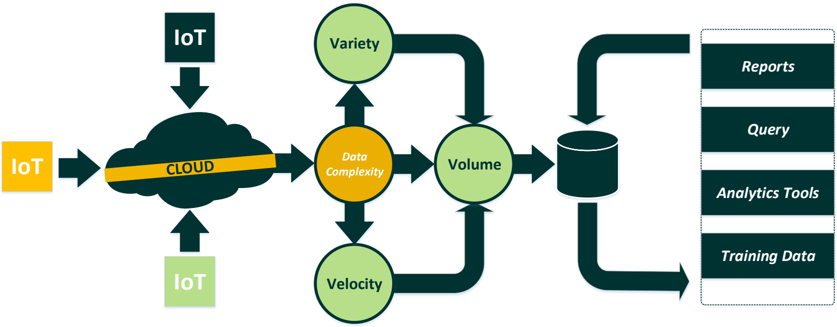


Fig. 5.5 – Relationship between IoT and big data analytics

The first step includes managing IoT data sources, where connected smart devices use applications to interact with each other. For example, the interaction of devices such as street web cameras, traffic lights, and smart home devices, generates large amounts of data with different formats. This data can be stored in low-cost commodity storage on the cloud. In the second step, the generated data become "big data" based on their volume, velocity, and variety. These vast amounts of data are stored in big data storages in shared distributed cloud databases. The last step applies analytics tools such as MapReduce, Spark, Splunk, etc. that analyze these big IoT data sets. The four levels of analytics start from training data, then move on to analytics tools, queries, and reports. The next paragraphs present some challenges in the field of IoT data analytics: (1) Privacy and security; (2) IoT data mining; (3) Visualization; (4) Integration.

5.1.2.1 IoT data security and Privacy

Privacy issues arise when the system risks receiving or retrieving personal information through comprehensive data analytics tools, even though the data is created by anonymous users. With the proliferation of big data analytics technologies used in big IoT data, the issue of privacy has become a significant issue in data sharing. Another security risk associated with IoT data is the heterogeneity of the types of devices used and the nature of the data generated, such as raw devices, data types, and communication protocols. These devices can have different sizes and shape offline and are designed to interact with typical applications.

Concerning IoT data, the following security issues may arise:

(a) timely updates - the complexity of updating systems,

(b) incident management - the detection of suspicious traffic patterns among legitimate cases and the possible non-registration of undetermined cases,

(c) interoperability - own procedures and specific trading procedures complicate the detection of hidden or zero-day attacks,

(d) protocol convergence - although IPv6 is currently compatible with the latest specifications, this protocol is not yet fully deployed. No answer can now resolve these issues and manage the security and privacy of the interconnected devices.

However, the following recommendations can overcome these difficulties.

(a) First, a truly open ecosystem with standard APIs is required to avoid compatibility and reliability issues.

(b) Second, the devices must be securely protected when communicating with peers. (c) Third, the devices must be rigorously encoded using the best security practices to protect against common security and privacy threats.

5.1.2.2 IoT data mining

The size and heterogeneity of data place new demands on data mining and a variety of data sources. Also, compared to small data sets, large data sets contain more deviations and ambiguities, which require additional preprocessing steps, such as cleaning, shrinking, and transferring.

Another issue is to extract accurate and knowledgeable information from a large amount of different data. Thus, obtaining precise information from complex data requires analyzing the properties of the data and finding the association between different data points. Moreover, the researchers chose existing data extraction algorithms in different ways.

(a) improve the discovery of knowledge from a single source,

(b) implement data mining techniques for multi-threaded platforms;

(c) study and analysis of dynamic data mining methods and streaming data.

Moreover, synchronization problems may arise in parallel computations, while information is exchanged through various data retrieval methods. This bottleneck of data transfer methods has become an open question in big IoT data analytics that needs to be addressed.

5.1.2.3 Visualization

Data visualization in the case of big and heterogeneous data (unstructured, structured and semi-structured) is a difficult task. Currently, most of the big data visualization tools used for IoT show poor performance results in terms of functionality, scalability and response time. To provide effective visualization with uncertainties during the visual analytics process, several important issues need to be addressed, such as

(a) visual noise - most of the objects in the dataset are closely related to each other, and thus, users can perceive different results from the same sensor;

(b) loss of information - applying reduction methods to visible data sets can lead to loss of information;

(c) large image observation — data visualization tools have inherent problems with respect to the aspect ratio, resolution design, and limits of physical perception;

(d) frequent image changes — users will not notice rapid changes in data during output; and

(e) high performance requirements — high-performance requirements are imposed because data is generated dynamically in an IoT environment. Besides, advanced analytics-enabled methods allow you to use interactive graphics on laptops, desktops, or mobile devices, such as smartphones and tablets.

5.1.2.4 Integration

Integration means having the same representation of different formats.

Data integration involves all processes involved in collecting data from different sources, as well as storing and providing data in a single view. At each point in time, various forms of data are constantly generated by social networks, IoT and other communication and telecommunication approaches. The data obtained can be divided into three groups:

(a) structured data, such as data stored in traditional database systems, including rows and columns;

(b) semi-structured such as HTML, XML, etc.;

(c) unstructured data such as video, audio, and images.

Overlapping the same data, improving productivity and scalability, and providing real-time access to data are some of the data integration challenges that need to be addressed in the future.

Another problem is to correct structures in semi-structured and unstructured data before integrating and analyzing these types of data. Information such as entities and relationships can be retrieved from textual data through text mining, machine learning, natural processing, and information retrieval.

However, new technologies should be developed to extract images, videos and other information from other non-text formats of unstructured data. It is expected that the text analysis will be carried out using several specialized extractors for the same text.

5.2 Scientific analytics models used in the IoT verticals

Processing of massive data has become a constant in research, which is revealing the hidden information to better understand the output data. Data processing is important in the study of IoT data process. IoT data often are routinely non-linear and non-stationary in nature. Existing approaches are in general low effectiveness to cope with the progressively larger data.

5.2.1 The causal influence estimation for IoT data

The main goal in IoT signal processing is to quantify the relationship between simultaneously observed time-series data and to reveal interactions between this data. If IoT signals are potentially non-stationary, we can use the technique to explore the causal effects based on Granger causality analysis.

The method, which is based on estimation of residuals of the time-varying autoregressive prediction parameters, can be described as follow. The residuals of the prediction process, which potentially carry information about the nonlinear causality by an Autoregressive Integrated Moving Average (ARIMA) model, are used.

Granger causality has been widely applied to detect the causal influence of system variables. The general idea of Granger causality as follows. If a time-invariant jointly regressive model of time series X_1 and X_2 much improves the prediction accuracy of the autoregressive model of time series X_1, then there is Granger causality from X_2 to X_1. Otherwise, there is no Granger causality from X_2 to X_1. Although Granger causality has received wide applications, some researchers have criticized Granger causality from different aspects. Hu et al. [12] provide evidence that Granger causality cannot reveal true causality.

The technique includes a causality model based on Granger causality method and used the autoregressive model residuals.

To describe the method, as the source data, we define two time-series X_1 and X_2, which allow the use of the following autoregression model (5.1) and the combined regression model (5.2).

$$\begin{cases} X_{1,t} = \sum_{j=1}^{m'} \mathrm{a}_{1,j} X_{1,t-j} + \epsilon_{1,t} \\ X_{2,t} = \sum_{j=1}^{m'} \mathrm{a}_{2,j} X_{1,t-j} + \epsilon_{2,t} \end{cases}, \tag{5.1}$$

$$\begin{cases} X_{1,t} = \sum_{j=1}^{m} a_{11,j} X_{1,t-j} + \sum_{j=1}^{m} a_{12,j} X_{2,t-j} + \eta_{1,t} \\ X_{2,t} = \sum_{j=1}^{m'} a_{21,j} X_{1,t-j} + \sum_{j=1}^{m} a_{22,j} X_{2,t-j} + \eta_{2,t} \end{cases}, \tag{5.2}$$

where a - the parameters of the model, $t = 0,1, \cdots, N$, ε_i and η_i - errors prediction are uncorrelated by time and are of zero means and variances for each time series.

The suggested estimation of true causality for time series calculating as follows. Following equality (2), X_1, t consists of three parts: $\sum_{j=1}^{m} a_{11,j} X_{1,t-j}$ (the last values of *X₁*), $\sum_{j=1}^{m} a_{12,j} X_{2,t-j}$ (the last values of X_2) and the residual η_1, *t*. Each part has a specific role in X_1, *t*.

If $\sum_{j=1}^{m} a_{12,j} X_{2,t-j}$ has a larger part among these three parts, then we can be seen that X_2 has a stronger causal effect on X_1 or conversely. This is used to assess the true causal influence of X_2 on X_1 in the time series, that is, the causality method should describe how much of X_2 is among these three parts.

Following this, the assessment of the true causal influence of X_2 on X_1 used further can be represented as follows (5.3).

$$n_{X_2 \to X_1} = \frac{\sum\limits_{t=m}^{m} (\sum\limits_{j=1}^{m} a_{12,j} X_{2,t-j})^2}{\sum\limits_{h=1}^{2} \sum\limits_{t=m}^{N} (\sum\limits_{j=1}^{m} a_{1h,j} X_{h,t-j})^2 + \sum\limits_{t=m}^{N} \eta_{1,t}^2} . \tag{5.3}$$

An assessment of the true causal influence of X_1 on X_2 for time series can be similarly represented as follows (5.4).

$$n_{X_1 \to X_2} = \frac{\sum\limits_{t=m}^{m} (\sum\limits_{j=1}^{m} a_{21,j} X_{1,t-j})^2}{\sum\limits_{h=1}^{2} \sum\limits_{t=m}^{N} (\sum\limits_{j=1}^{m} a_{2h,j} X_{h,t-j})^2 + \sum\limits_{t=m}^{N} \eta_{2,t}^2} \tag{5.4}$$

This approach enables to calculate the relationship between simultaneously observed time-series data and to reveal interactions between this data for the next analysis.

5.2.2 Supervised algorithms and unsupervised algorithms applicable to IoT datasets

Typically, after the processing, analytics models used to IoT data to extract important sensor / devices signals parameters, and then are interpreted and classified. Classification is used to distinguish normal parameters from some event or abnormality. For example, by ECG monitoring cardiologists detect which region of the myocardium experiences failure (see Fig. 5.6).

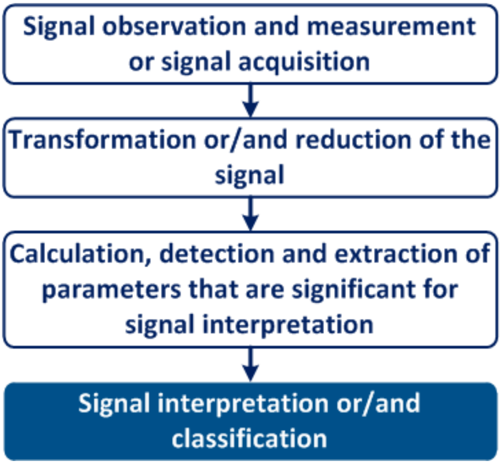

Signal observation and measurement or signal acquisition
Transformation or/and reduction of the signal
Calculation, detection and extraction of parameters that are significant for signal interpretation
Signal interpretation or/and classification

Fig. 5.6 – Processing of sensor / devices signals

Classification (supervised learning) - a technique used to predict the label for data instances and is a process. The classification process aims to identify unknown data, on the basis of a training set of data containing observations whose classes are known. For example, classification in biomedicine faces several difficulties: researchers spend a long time to accumulate enough knowledge to distinguish different related cases, as normal and abnormal. Labeled data can be presented as the follow matrix, where *x* is value of input attribute, and *y* is output attribute or label.

$$\begin{bmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{bmatrix} = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix}$$

Clustering (unsupervised learning) is the process of finding structural information in data without labeling.

Clustering mechanisms separates and organizes unlabeled data into different groups whose members are similar to each other in some metric. The clustering quality depends on both the similarity measure used by the method and its implementation. Unlabeled data can be presented as the follow matrix, where *x* is an attribute value.

$$\begin{bmatrix} x_{11} & \dots & x_{1m} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}$$

Usually, cluster analysis includes follow steps (see Fig. 5.7).

1. *Feature selection or extraction.* Feature selection chooses distinguishing features from a set of candidates and feature extraction uses data transformations to generate useful and novel features.

2. *Clustering algorithm design or selection.* This step is usually combined with the selection of a corresponding proximity measure. Patterns are grouped according to whether they resemble each other and the proximity measure directly affects the formation of the resulting clusters.

3. *Cluster validation.* Different approaches generally lead to different clusters and even for the same algorithm, parameter identification or the sequence of input patterns may affect the final results. Consequently, effective evaluation standards and criteria are important to provide the researcher with a degree of confidence for the results derived from the used algorithms.

4. *Results interpretation.* The ultimate goal of clustering is to provide meaningful insights from the original data.

Unsupervised learning from IoT-based system includes:

b. Automated detection of abnormal events.

c. Automated detection of anomalies time series data.

d. Models for data prediction.

Anomaly detection includes:

- anomalies detection by looking for any values beyond a certain threshold;
- anomalies detection by the structure of the waveform;

- more subtle errors - a change in the shape of a periodic waveform, for examples.

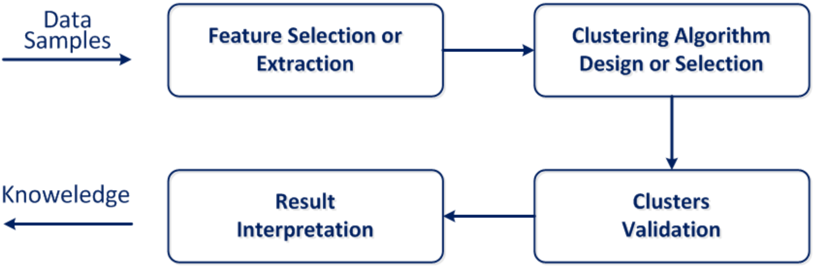

Data Samples
Feature Selection or Extraction
Clustering Algorithm Design or Selection
Clusters Validation
Result Interpretation
Knoweledge

Fig. 5.7 – The stages of cluster analysis (unsupervised learning)

Cluster analysis is used to define an anomaly as being some pattern in sensor / devices signals, for example, in the ECG waveform. Created a library of normal waveform shapes is used to try and reconstruct a waveform to be tested. If the reconstruction is poor, then the waveform is likely to contain something abnormal and is, therefore, an anomaly.

The particular method we'll be using is called k-means clustering.

The k-means algorithm clusters data by trying to separate samples in *n* groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to a large number of samples and has been used across a large range of application areas in many different fields.

The k-means algorithm divides a set of ***N*** samples *X* into *K* disjoint clusters *C*, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from *X*, although they live in the same space. The k-means algorithm aims to choose centroids that minimize the *inertia*, or within-cluster sum of squared criterion:

$$\sum_{i=0}^{n} \min_{\mu_i \in C} \left(\left\| x_j - \mu_i \right\|^2 \right) \tag{5.5}$$

Inertia, or the within-cluster sum of squares criterion, can be recognized as a measure of how internally coherent clusters are.

The properties of the k-means cluster algorithm are presented in Table 5.2.

Table 5.2 – The properties of the k-means cluster algorithm

k-means algorithm	**Properties**
Parameters	Number of clusters
Scalability	Very large N samples, medium K clusters
Usecase	General-purpose, even cluster size, flat geometry, not too many clusters
Metric used	Distances between points (centroids)

Process of anomalies detection, using ECG signal, include follow steps.

- Split signal waveform into segments of n samples.
- Space formation in n dimensions, with each segment representing one point.
- Clustering determination of segment points, and determination the centroids of the clusters.
- Cluster centroids provide a library of normal waveform shapes.
- signal waveform reconstruct for tested using cluster centroids learned during training.
- Reconstruction error on any segment indicates the anomalous shape.

The main step in real-time analysis is using of sliding windows. Split the waveform into overlapping segments, with the section of the original data sampled sliding along by two samples each time. This approach is used to get instances of each waveform shape with a variety of horizontal translations. The approach is to apply a window function to the data, which leads the start and end of the signal to be zero. The clustering consists of the following steps:

- Signal segmentation when we split data into overlapping segments.

- For the selected segment, find the centroid of the cluster that best matches this segment.
- Segment reconstruction when the cluster centroid is used for this segment.
- Data signal reconstruction.

Anomaly has produced a shape in the waveform that had not been seen before. The waveform around that point could not be reconstructed using the learned shape library. This gives a reconstruction error that could be easily detected (see Fig. 5.8).

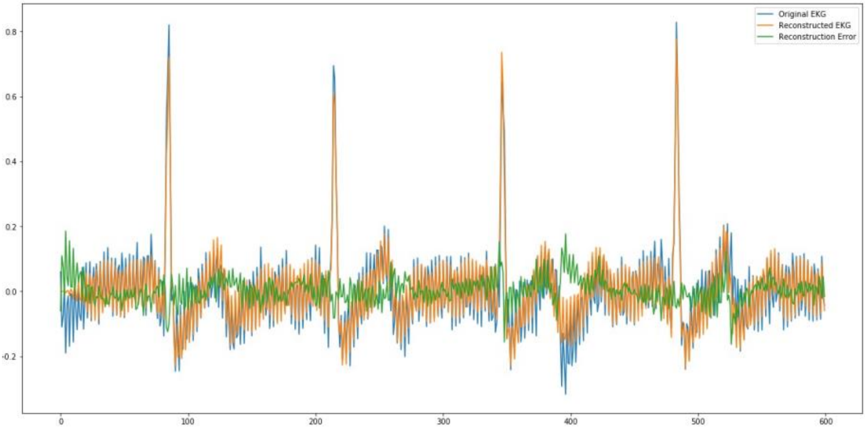
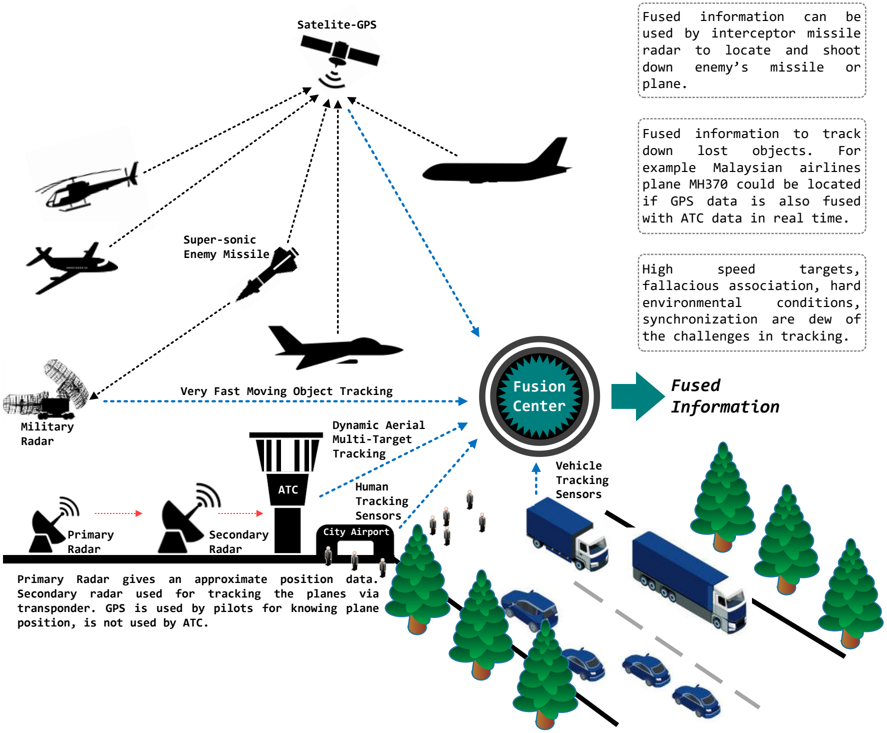


Fig. 5.8 – Original signal (blue line), reconstructed signal (yellow line) and error reconstruction (green line) using clustering

5.3 Data fusion and time series data processing from IoT devices

In IoT, environment information fusion can be used in various areas to enhance the IoT ubiquitous aspect. These areas are environmental monitoring, healthcare, crisis management, monitoring, controlling, tracking, intelligence gathering, and many others.

Data fusion (DF) in IoT can take place at four stages: decision level, feature level, pixel level and signal level. With respect to Fig. 5.9, IoT data fusion can also be seen with two different perspectives. First, it can be viewed as a single hop where every sensor transmits data to the data fusion center directly. Second, it can occur by a multi-hop process, where data passes across adjacent sensors.



Fused information can be used by interceptor missile radar to locate and shoot down enemy's missile or plane.

Fused information to track down lost objects. For example Malaysian airlines plane MH370 could be located if GPS data is also fused with ATC data in real time.

High speed targets, fallacious association, hard environmental conditions, synchronization are dew of the challenges in tracking.

Primary Radar gives an approximate position data. Secondary radar used for tracking the planes via transponder. GPS is used by pilots for knowing plane position, is not used by ATC.

Figure 5.1. Wireless sensor network architecture for data fusion in IoT

DF in the Internet of Things (IoT) paradigm can play a major role in IoT due to the following reasons:

- DF makes information more intelligent, decisive, sensible and precise which is coming from multiple sensors and sources. The information from each sensor per se may not make much sense.
- A statistical benefit of fusion is obtained by computing the N independent observations; one can anticipate that the data are amalgamated in an optimal manner.

- In IoT, a big challenge is making very low power sensors which do not need battery replacements over their lifetimes; this popularizes the demand for energy efficient sensors. It has been an established fact that the sensors with high accuracy can result in the consumption of a high amount of power. To handle this issue, a set of very low power consumption sensors can be used with low accuracy. By using data fusion, highly accurate information will be created.
- DF can be helpful in handling the big data is-sues of IoT because we are fusing data from many sensors into more precise and accurate information.
- Another critical advantage of DF is that it helps to hide the critical information or the semantics which are responsible for the fused results. Examples of this are in military applications, some critical medical areas and in intelligence buildings.

5.3.1 Data Fusion Challenges

DF has multiple challenges ahead, which are explored in various literature. Some of them are listed below:

- Data Imperfection: Sensor data is imprecise at times; it can be inaccurate and uncertain. This behavior is not infamous in wireless sensor networks. The imperfection must be dealt with effectively with the use of data fusion algorithms.
- Ambiguities and Inconsistencies: Impreciseness is not the only factor responsible for data inconsistencies; the environment in which a sensor is operating is largely responsible as well. Outlier detection, replacement and data imputation are vital in IoT environment.
- Conflicting Nature: The conflicting nature of data can give rise to counter-intuitive results. The problem of conflicting data is visible more in evidential belief reasoning and Dempster’s rule of combination. The data fusion algorithm must take critical care while treating conflicting data.
- Data Correlation and Alignment: This problem is more common in wireless sensor networks (WSNs) and can result in over or under confidence in a data fusion algorithm. An alignment problem which is also known as a sensor registration problem occurs when

sensor data is transformed from every sensor's local frame to a common frame prior to fusion.

- Trivial Features: In IoT environment, applications may consist of several hundreds and thousands of sensors sensing different parameters. These sensed values in large setups such as smart cities and industrial plants consist of trivial and non trivial data. Processing of trivial data may affect the data fusion accuracy. Thus most relevant features need to be selected before data fusion.
- Dynamically Iterative Process: Data fusion is not a static process in nature; however, dynamically iterative needs regular refinement of the estimates in a fusion environment. No Magical Algorithm: With time researches in data fusion area has advances and high performance algorithms are there now. However, it is still difficult to say that a perfect data fusion algorithm exists.

5.3.2 Mathematical methods of data fusion

Data fusion techniques can be classified based on the mathematical methods into three broader categories:

- Probability-based methods including Bayesian analysis, statistics, and recursive operators (see Fig. 5.10).
- AI-based techniques including classical machine learning, fuzzy logic, Artificial Neural Networks (ANN) and genetic evaluation.
- Theory of Evidence based Data Fusion methods.

In respect to current challenges, a significant issue in IoT environment is that suddenly a number of sensors can awake, adding several nodes in WSNs.

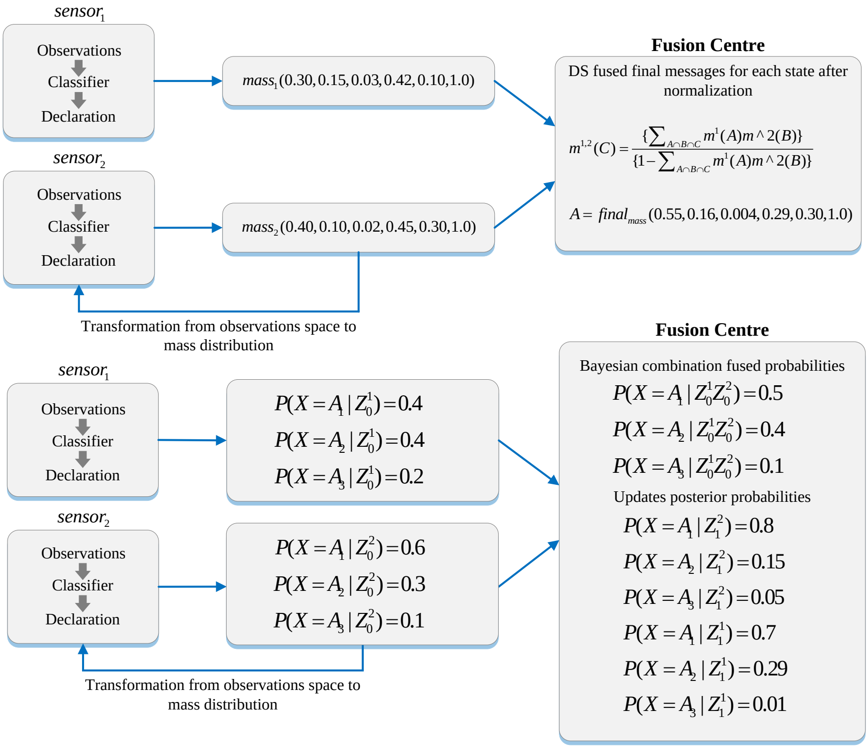
The data fusion algorithm must be efficient to deal with these kinds of situations. To address this issue, there are several sub-optimal algorithms. These algorithms include the following:

Naïve fusion

It is one of the simplest data fusion techniques. It is anticipated that the dependency between the density functions is minimal; however, the technique is unreliable. Due to the lack of past information, over-confidence can occur. The naïve fusion equation can be written as:

$$p(x) = \frac{p_1(x)p_2(x)}{\int p_1(x)p_2(x)dx}, \tag{5.6}$$

where $p_1(x)$ and $p_2(x)$ are the density function fusion probabilities.



$sensor_1$
Observations
Classifier
Declaration
$mass_1(0.30, 0.15, 0.03, 0.42, 0.10, 1.0)$
$sensor_2$
Observations
Classifier
Declaration
$mass_2(0.40, 0.10, 0.02, 0.45, 0.30, 1.0)$
Transformation from observations space to mass distribution
Fusion Centre
DS fused final messages for each state after normalization
$m^{1,2}(C) = \frac{\{\sum_{A\cap B\cap C} m^1(A)m \wedge 2(B)\}}{\{1-\sum_{A\cap B\cap C} m^1(A)m \wedge 2(B)\}}$
$A = final_{mass}(0.55, 0.16, 0.004, 0.29, 0.30, 1.0)$
$sensor_1$
Observations
Classifier
Declaration
$P(X=A_1 | Z_0^1) = 0.4$
$P(X=A_2 | Z_0^1) = 0.4$
$P(X=A_3 | Z_0^1) = 0.2$
$sensor_2$
Observations
Classifier
Declaration
$P(X=A_1 | Z_0^2) = 0.6$
$P(X=A_2 | Z_0^2) = 0.3$
$P(X=A_3 | Z_0^2) = 0.1$
Transformation from observations space to mass distribution
Fusion Centre
Bayesian combination fused probabilities
$P(X=A_1 | Z_0^1 Z_0^2) = 0.5$
$P(X=A_2 | Z_0^1 Z_0^2) = 0.4$
$P(X=A_3 | Z_0^1 Z_0^2) = 0.1$
Updates posterior probabilities
$P(X=A_1 | Z_1^2) = 0.8$
$P(X=A_2 | Z_1^2) = 0.15$
$P(X=A_3 | Z_1^2) = 0.05$
$P(X=A_1 | Z_1^1) = 0.7$
$P(X=A_2 | Z_1^1) = 0.29$
$P(X=A_3 | Z_1^1) = 0.01$

Fig. 5.10 – DF with Theory of Evidence approach (DS combination) and DF with Probability-based method (Bayesian combination)

Channel filter

It is a simple data fusion approach. Only the first ordered redundant data is taken into account. Every channel has a pair of agents, a transmitting agent and a receiving agent. Redundant information is removed by the transmitting agent. However, ad-hoc WSNs transmitting data sometimes does not reach the other end. Therefore, the receiving agent can do the task of transmitting agent in the dynamic ad hoc WSNs. Channel Filter fusion equation is given as:

$$p(x) = \frac{p_1(x)p_2(x)/\bar{p}(x)}{\int p_1(x)p_2(x)/\bar{p}(x)dx}, \tag{5.7}$$

where $\bar{p}(x)$ is the previous density function received. The advantage of this algorithm is that there is no need to maintain a high volume of history of past activities. Though one disadvantage is that during the filter, dependent information is removed. However, this effect can be minimized if the time between current processing and when redundancy occurred is too long.

Chernoff fusion

In unknown dependency distribution, the Chernoff technique can be used. Theoretically, two arbitrary density functions can be combined using Chernoff fusion in a log linear fashion. However, fused density may be distracted. Another disadvantage is that extensive computation is required. Chernoff equation is given as:

$$p(x) = \frac{p_1^w(x)p_2^{1-w}(x)}{\int p_1(x)p_2(x)/\bar{p}(x)dx}, \tag{5.8}$$

where $w \in [0,1]$.

SVM based data fusion

SVM based data fusion algorithm is proposed for fusing multispectral and panchromatic data gather for the purpose of remote sensing of Shaoxing City, China. Ageneralize mathematical formulation of [71] is given below.

The training of SVM is given as:

$$(w\square x) - b = 0. \tag{5.9}$$

$$(w \times x) - b \le -1, \text{ if } y_i = -1, \tag{5.10}$$

$$(w \times x) - b \ge 1, \text{ if } y_i = +1, \tag{5.11}$$

where sample number is denoted by m, input data dimensions are denoted by n, "•" is the dot product and w is the normal direction of the hyper plane is the Euclidean function.

5.4 Work related analysis

Data Science for IoT-based systems is a hot topic has been proposed for several schools. Thus, A. Jaokar and J. Bernard [13] developed an online course on Data Science for Internet of Things that cover unique aspects of Data Science for IoT including Deep Learning, Complex event processing/sensor fusion and Streaming/Real time analytics, designing autonomic systems and investigating their ability to support IoT challenges. Our partners from EU universities are actively involved in research and development for IoT-based systems, applying new knowledge in business and the educational process. For example, the University of Coimbra suggests several courses in this topic, namely, "Experimental methods for Data Science", "Intelligent Sensors", "Emergent Internet services". The objectives of course on Emergent Internet services are the knowledge about telematics applications fundamentals, Internet applications and emerging Internet services. Course "Experimental methods for Data Science" covers several topics of computer science, such as communication networks, algorithms and artificial intelligence.

School of Engineering at Newcastle University focuses on a broad range of communications, sensors and signal processing. One of their courses CSC8621 "Computing Foundations of data Science" introduces the fundamental computing concepts and techniques underpinning contemporary data science. The module aims to provide students with grounding in program design and implementation, programming environments. Furthermore, it explores how to apply and devise algorithms for a particular problem. Another one is CSC8632

"Data Science in the Wild" covers the principles of research and professional skills that are required to practice Data Science in the real-world. The taught content is complemented by practical experience where the students embed themselves in a suitable institute or immerse themselves in a research area to gain domain knowledge. Insights gained from this experience are then used to propose a Data Science project within that company, institute or area of research and evaluate the merits of the proposal based on each of taught topics listed.

Conclusions and questions

IoT is one of the forerunners in data generation that led challenges in Data Science and requires an academic foundation in statistics and computer science combined with domain knowledge, practical resourcefulness and research skills. There are at least ten differences that distinguish data science for IoT from traditional data science that should be taken into account while dealing with this topic. Since data is generated from various sources, it is also important to characterize them and choose and apply the appropriate algorithms. Besides, to understand algorithm application, it is also important to understand IoT applications, data characteristics, and visions how to combine these notations. Also, we discussed some effective research approaches to resolve an essential challenge in nowadays research on the IoT as well as discuss scientific analytics models used in IoT.

In order to better understand and assimilate the course content that is presented in this section, we encourage you to answer the following questions.

1. What kinds of data can be obtained from IoT device?
2. Levels of data processing in an IoT scenario
3. What features should offer a cloud IoT platform?
4. What are the main differences between data science for IoT from traditional data science?
5. For what purposes data preprocessing is used?
6. Typical preprocessing operations.
7. Why real-time processing is important for IoT?
8. What security issues may arise in IoT applications?
9. How to overcome security issues in IoT?

10. What challenges exists in IoT data analytics?
11. Why data visualization is tricky for IoT?
12. What challenges arise with data integration?
13. What is a main goal in IoT signal processing?
14. For what purpose causal influence estimation is used?
15. What difference between supervised and unsupervised algorithms applicable to IoT datasets?
16. When cluster analysis is applicable?
17. For what purpose data fusion is used?

References

1. "The SmartThings Ecosystem" [Online]. Available: https://smartthings.developer.samsung.com/docs/index.html [Accessed: 22- June- 2019].

2. "IoT - Internet of Things", *San Diego Consulting Group* [Online]. Available: https://www.sandiegoconsultinggroup.com/ioe-m2m-iot-paas [Accessed: 22- June- 2019].

3. A. Jaokar, "Data Science for Internet of Things (IoT): Ten Differences From Traditional Data Science", *KDnuggets*, 2016. [Online]. Available: http://www.kdnuggets.com/2016/09/data-science-iot-10-differences.html [Accessed: 22- June- 2019].

4. B. Schmarzo, "The Internet of Things and Analytics at the Edge" [Online]. Available: https://infocus.emc.com/william_schmarzo/internet-of-things-analytics-edge/ [Accessed: 22- June- 2019].

5. "The Internet of Things (IoT) and Analytics at The Edge – InFocus Blog: Dell Technologies Services", *InFocus Blog | Dell Technologies Services*, 29-Nov-2017. [Online]. Available: https://infocus.dellemc.com/william_schmarzo/internet-of-things-analytics-edge/. [Accessed: 30-Jul-2019].

6. M. Abu-Elkheir, M. Hayajneh, N.A. Ali, "Data management for the internet of things: design primitives and solution", *Sensors* (Basel, Switzerland) vol. 13,11 15582-612. 14 Nov. 2013, doi:10.3390/s131115582

7. V. Granville, "Data Science for IoT vs Classic Data Science: 10 Differences", Data Science Central. [Online]. Available:

https://www.datasciencecentral.com/profiles/blogs/data-science-for-iot-vs-classic-data-science-10-differences. [Accessed: 30-Jul-2019].

8. Y. Le Cun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436, 2015.

9. F. Hawlitschek, B. Notheisen, and T. Teubner, "The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy", *Electronic Commerce Research and Applications*. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1567422318300292 [Accessed: 30-Jul-2019].

10. "Internet of Robotic Things" *ABI Research* [Online]. Available: https://www.abiresearch.com/market-research/product/1019712- the-internet-of-robotic-things/ [Accessed: 30-Feb-2019].

11.R. Rout, R.K. Mohanayak, "Big Data with reference to IoT: Architecture, Opportunities and Challenges", *IOSR Journal of Engineering (IOSRJEN)*, vol. 1(10), pp. 2045-2053, 2018.

12.Hu, X., Hu, S., Zhang, J., Kong, W. and Cao, Y.,. A fatal drawback of the widely used Granger causality in neuroscience. *IEEE International Conference on Information Science and Technology (ICIST)*, pp. 61-65, May 2016.

13.A. Jaokar, J.-J. Bernard, "Data Science for Internet of Things", *CreateSpace Independent Publishing Platform*, 64 p. 2015.

PART II. DATA SCINCE FOR IOT AND IOE

6. DATA MINING AND PROCESSING FOR THE IOT

DrS., Prof. I. S. Skarga-Bandurova,

Dr. T.O. Biloborodova (V. Dahl EUNU)

Contents

Abbreviations

AI – Artificial intelligence
BD – Big data
CI/CD – Continuous integration and continuous delivery
CPLD – Complex programmable logic device
CRISP-DM – Cross industry standard process for data mining
DM – Data mining
GPS – Global Positioning System
HDFS – Hadoop Distributed File System
IoT – Internet of things
KDD – Knowledge discovery in databases
ML – Machine learning
MOBB – Maximally Overlapped Binpacking driven Bursting
R&D – Research and development
RDF – Resource Description Framework
RFID – Radio Frequency Identification
SPA – Scalable and yet customizable data PArtitioning framework
SPARQL – SPARQL Protocol and RDF Query Language
SEJITS – Selective Embedded Just-InTime Specialization
WSN – Wireless Sensor Network
WSAN – Wireless Sensor and Actuator Network

6.1 Data mining for IoT

The massive data generated by the Internet of Things (IoT) are considered of high value in the different area, and data mining techniques can be applied to IoT to extract hidden information from data. This Chapter present material to review data mining in knowledge view, technique view, including classification, clustering, association analysis, time series analysis, and massive data analysis. Data mining involves discovering novel, interesting, and potentially useful patterns from large data sets and applying algorithms to the extraction of hidden information.

The data mining technique is actual in the Internet of Things concept and arises from the need to manage and analyze big sensors data. In order to make wise decisions both for people and for the things in IoT, data mining technologies are integrated with IoT technologies for decision making support and systems optimization. Data mining includes discovering new, interestingness, and potentially helpful patterns from data and hidden information extraction using algorithms.

This chapter covers main aspects related to data mining and processing for IoT-based solutions. We give a systematic review of data mining models and techniques for IoT: classification, clustering, frequent pattern mining for IoT, association analysis, stream mining and data mining of massive datasets. Data mining involves discovering novel, interesting, and potentially useful patterns from large data sets and applying algorithms to the extraction of knowledge from information obtained from IoT devices.

IoT applications generate more than 2.5 quintillion data bytes daily [1]. To convert this data into knowledge, data mining systems are necessary. Data mining is one of the most valuable technologies enable to identify unknown patterns and making IoT smarter. Data mining enables to find and discover novel, interesting, and useful patterns from large data sets and generate new knowledge from information obtained from IoT devices. However, basic data mining algorithms and technologies are not sufficient for IoT framework. So, it becomes a great challenge to collect, analyze and manage IoT data as well as to generate and update data mining algorithms for IoT purposes. In this chapter we discuss some data mining approaches applicable for IoT. Figure 6.1 shows an overall level for transformation of data and depicts a level of services where big data mining for IoT is applicable.

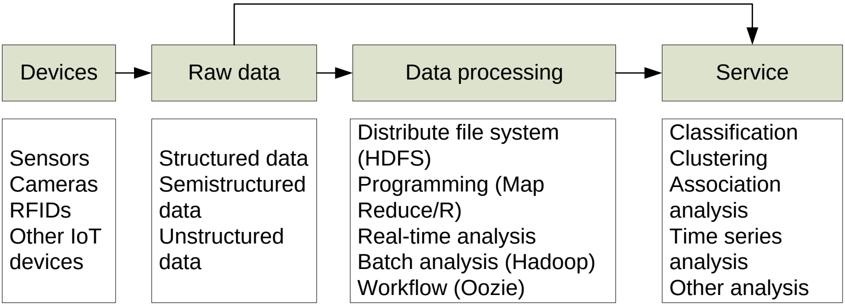


Fig. 6.1 – Big data mining based on IoT (Adapted from [2])

The general purpose of any data mining process is to build a best predictive or descriptive model of a large amount of data that not only fits or explains it but is also able to generalize to new data [3, 4]. It is assumed that the concept of data mining for IoT will stimulate business models for IoT. Based on a broad understanding of data mining functionality, data mining is the process of finding interesting knowledge from large amounts of data stored in any database, data repositories, or other data repositories. Internet of Things (IoT) and related technologies are easily integrated with existing methods, technologies, data mining algorithms. An important aspect of data mining of the IoT-based system is the effective structure of the system, which should take into account security, data privacy, data sharing mechanisms, scalability, etc. Such a data mining system for IoT includes data acquisition devices, raw data properties, extraction levels, processing, data analysis, it is necessary to take into account the properties of the IoT devices when planning data mining for IoT [5]. Technically, every IoT thing can create data, but technical issues and challenges on how to handle this data and how to obtain useful information have still emerged.

IoT devices and sensors have two general limitations that must be considered when designing and planning the operation of IoT data mining systems:

- Limited energy resource device.
- Limited device memory.

The instability of the network connection and availability of the thing due to the unpredictable mobility of devices, different battery discharge rates, equipment failures and lack of a priori knowledge of the hardware and software characteristics of devices [6-8].

The power source must match the data, i.e. be sufficient for IoT calculations. Storing information leads to the expenditure of battery energy. The solution to this problem is provided by storing and performing computational operations using remote IoT computing resources, such as a server and / or cloud.

IoT systems create a huge amount of dynamic data. Analysis and extracting useful information from this data with data mining can facilitate the automation of intelligent decision making.

IoT data can be:

- multimodal and heterogeneous;
- noisy and incomplete;
- unbalanced and biased;
- dependent on time and location;
- dynamic, different data quality;
- almost always require real-time analysis.

Since IoT data is the basis for extracting knowledge, it is important to have high quality information. This condition can directly affect the accuracy of knowledge extraction.

Traditionally, the data mining tasks can be classified generally into two types: prediction and description. Successful use of classical approaches of data mining is focused on classification, clustering, frequently sequences, associative analysis, time series analysis, and outlier analysis (see fig. 6.2).

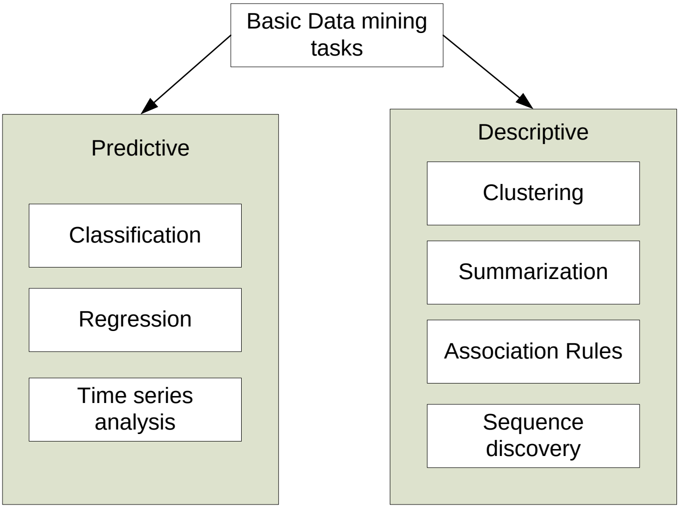


Fig. 6.2 – IoT data characteristics

1. Classification is the finding models or functions that describe and recognize classes of data, or concepts for the task of objects class prediction whose class label is unknown.

2. Clustering use for input data without a known output data.

3. Association analysis is the pattern discovery of association rules conjuncts that frequently occur together in studied data.

4. Time series analysis involves methods and techniques for time series data analysis to extract meaningful data properties and useful data characteristics.

5. Outlier analysis describes and modeling regularities or trends for objects whose characteristics change in time.

6.1.1 Basic idea of using data mining for IoT

One of the most important questions that knowledge discovery in databases (KDD) and data mining (DM) technology can solve is how to transform the data generated or captured by IoT into knowledge that serve to the environment and people.

The main characteristics of the source data of IoT-based system are the following [2]:

1. They are really big data.

2. Heterogeneity of the sources being combined and the types of data: the data of the IoT-based system may include several data sources, for example, data from sensors, historical data, which may also have different formats: numerical, categorical, textual, binary, etc.

3. The complexity of recoverable knowledge: due to heterogeneity and a large amount of data when extracting knowledge, it is necessary to analyze their properties and the interrelation of various data sources. These characteristics require special attention in the process of data mining of the IoT-based system for obtaining an effective and high-quality result. In the process of extracting useful knowledge, there are the following problems.

1. Data extraction: data can be combined from various sources, they are diverse and heterogeneous, and noisy.

2. Uncertainty and incompleteness of data: compliance with data security and confidentiality causes uncertainty and incompleteness of data in the extraction of useful knowledge.

To solve these problems, approaches and methodologies are being developed that try to minimize their consequences. Tracking and detection of data errors, preprocessing filtering, data reduction mechanisms are used. To combine data from several sources, parallel programming models are used, for which classical approaches to data mining are adapted.

The use of models depends on the area of IoT in which they are applied. For example, in ecology: pollution prediction, anomaly detection, prediction and interpolation of missing events are common. In medicine, traditional models used to predict a patient's conditions can include his history, clinical data as input along with real-time status monitoring data.

It is also important to consider that IoT includes temporary and massive data.

The merging of data or data fusion is associated with combining data from different sources so that the information obtained has less uncertainty than would be possible when these sources were used individually. The term "reducing uncertainty" in this case may mean more accurate, more complete or more reliable, or refer to the result of an emerging presentation based on combined information.

The data mining for IoT are similar to the base ones, but there are some significant differences. The process of extracting useful patterns from raw data is known as Knowledge discovery in databases (KDD). It is illustrated in Figure 6.3.

The Knowledge Discovery KDD process takes raw data as input and provides statistically significant in Databases (KDD) patterns found in the data (i.e., knowledge) as output. From the raw data, a subset is selected for processing and is denoted as target data. Target data is preprocessed to make it ready for analysis using data mining algorithm. Data mining is then performed on the preprocessed (and transformed) data to extract interesting patterns. The patterns are evaluated to ensure their validity and soundness and interpreted to provide insights into the data.

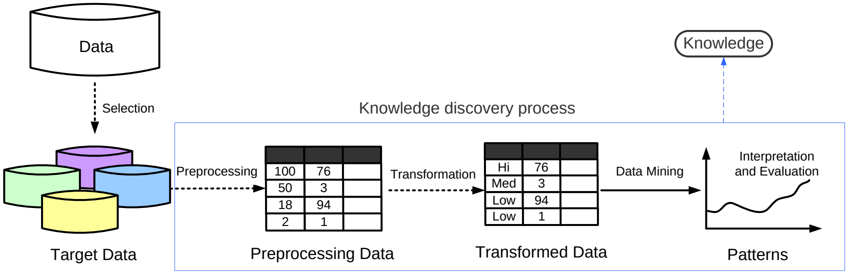
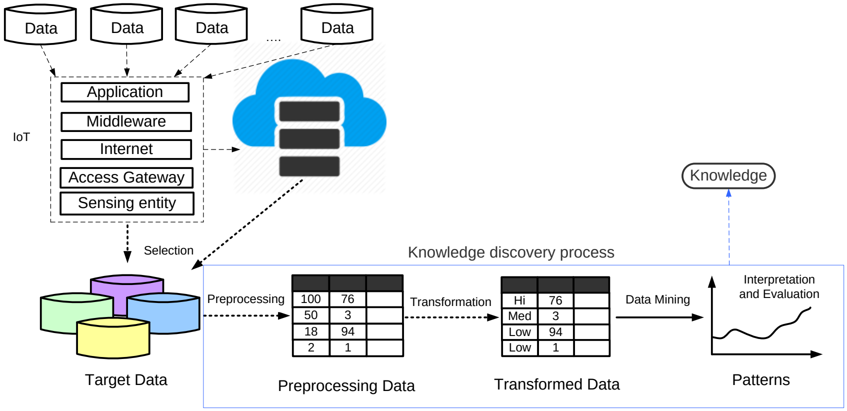


Fig. 6.3 – Traditional Knowledge Discovery in Databases (KDD) process

Based on the data mining and IoT overview, the data mining in IoT process is as follows (Figure 6.4): Data Mining for IoT begins with the first step of capturing data generated from IoT devices which includes: Sensor networks, Actuators, WSN (Wireless Sensor Network), WSAN (Wireless Sensor and Actuator Network), RFID (Radio Frequency Identification) Tags, Cameras, GPS etc.

To store and analyze such large amount of data, data warehouses are used where data preprocessing (cleaning the data (removing noisy, inconsistent and incomplete data), vectorization the data), data transforming which includes converting the data into the forms appropriate for data analyzing, and data reducing are performed.

Next step is selecting an appropriate data mining methodology for converting the preprocessed data into knowledge. KDD, when applied to IoT, will convert the data collected by IoT into useful information that can then be converted into knowledge.



100	76	
50	3	
18	94	
2	1	

Hi	76	
Med	3	
Low	94	
Low	1	

Fig. 6.4 – KDD process for IoT data

Currently only small scale data from IoT systems can be mined. So, it is a big challenge to implement existing DM techniques to a large scale IoT distributed systems [9].

To determine which algorithm to use for a particular task, we need to first define the task and aim of analysis. Some of tasks include finding unusual data points, predicting values or categories, structures discovery, feature extraction and more.

Table 6.1 shows some examples of using data mining algorithms for IoT data [10, 11].

Table 6.1 – Using data mining algorithms for IoT data

Mining algorithm	Goal	Data Source
Clustering	Network performance enhancement	Wireless sensor
	Inhabitant action prediction	X10 lamp and home application
	Provisioning of the needed services	Raw location tracking data
	Housekeeping	Vacuum sensor
	Managing the plant zones	GPS and sensor for agriculture
	Relationship in a social network	RFID, smart phone, PDA, and so on
Classification	Device recognition	RFID
	Traffic event detection	GPS, smart phone, and vehicle sensor
	Parking lot management	Passive infrared sensor
	Inhabitant action prediction	RFID, sensor, video camera, microphone, wearable kinematic sensor, and so on
	Inhabitant action prediction	Video camera
	Inhabitant action prediction	Microphone
	Physiology signal analysis	Wireless ECG sensor
Frequent Pattern	RFID tag management	RFID
	Spatial colocation pattern analysis	GPS and sensor
	Purchase behavior analysis	RFID and sensor
	Inhabitant action prediction	RFID and sensor
Anomaly Detection	Smart Traffic	GPS, smart phone, and vehicle sensor
	Smart Environment	Wireless sensor, smart phone
	Traffic Prediction	GPS, smart phone, and vehicle sensor
	Finding Anomalies in Power Dataset	RFID, wireless sensor
Hybrid	Inhabitant action prediction	RFID and sensor

6.1.2 Classification for IoT

Classification is an important technique in data mining that assigns items in a collection of target categories or classes. Classifying the data sets into different categories would make it easier to understand the data. This is called supervised learning technique. In this technique, a training data is given by the use of which a classifier model is build and based on this model the future pattern is predicted.

Supervised and unsupervised learning are widely used in IoT data mining [12]. The goal of supervised learning is to predict the corresponding output vector for a given input vector. Tasks in which the output label value is discrete are known as classification problems. Classification assumes some prior knowledge to guide the partitioning process to construct a set of classifiers to represent the possible distribution of patterns.

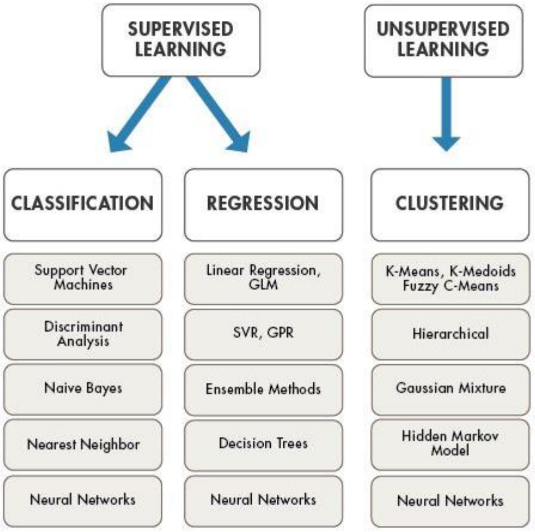


Fig. 6.5 – Some supervised and unsupervised learning methods

Generally, the classification task can be defined as follows: for given a set of labeled data *L* and a set of unlabeled data *NL*, we need to find a classifier or set of classifiers (i.e., the hyperline or prediction function) for *NL* using the set of labeled data *L*.

The use of classification methods is a solution to the problem of uncertainty and incompleteness of IoT data. In this context, the use of data mining always includes solving two related tasks: defining regular links between data elements and using these patterns to solve classification problems: predicting the values f some elements from known values of other elements. There are a lot of classification methods, they also includes decision tree learning, naive Bayes classifier, k-nearest neighbor classifier, and classification with network information and regression methods such as linear regression and logistic regression

6.1.3 Clustering for IoT

One of the main goals of unsupervised learning is the process of identifying similar cluster patterns in the input data, called clustering [12]. In addition, the goal may be to open a useful internal representation for the input data by preprocessing the original input variable, in order to transfer it to a new space of variables. This preprocessing stage can significantly improve the result of subsequent data analysis and is called the extraction function [13].

Clustering is widely used to handle streaming data [14]. Many clustering algorithms have been developed for their analysis, which can be grouped into the following main categories.

Partitioning-based clustering algorithm — it tries to find the best partitioning into data points where the intracluster similarity is maximum and the intercluster similarity is minimal.

Hierarchical clustering algorithms work by decomposing data objects into a cluster tree.

Grid-based clustering algorithms do not depend on the distribution of data objects. In fact, it breaks the data space into several cells that form grids. Clustering using algorithms in this category has a fast processing time, since it does not depend on the number of data objects.

Density-based clustering algorithms are designed to detect arbitrary clusters. If the two points are close enough and the area around them is dense, then these two data points merge and contribute to the construction of the cluster.

Characteristics of data streams do not allow the use of traditional density-based clustering. Recently, many density based clustering algorithms have been extended for data streams. The main idea of these algorithms is to use the density-based method in the clustering process and at the same time overcome the limitations that are determined by the nature of the data flow. Density-based clustering algorithms are categorized into two broad groups called density microclustering and density grid-based clustering algorithms.

In cluster analysis, an important parameter that needs to be determined is a measure of similarity (or dissimilarity) between individual objects that are clustered [15]. One of the most popular measures of measuring the similarity between two vectors in *d*-dimensional space is the Euclidean distance.

$$\begin{gathered} d(x_1, x_2) = \|x_1 - x_2\| = \sqrt{\sum_{r=1}^{d} (x_{1r} - x_{2r})^2}, \\ d(p, q) = \sum_{k-1}^{n} (p_k - q_k). \end{gathered} \tag{6.1}$$

Clustering helps to solve the following IoT data analysis tasks.

- Processing of data of high dimension. Often complex concepts of the real world are accompanied by a large number of functions. This strengthens the assessment tool (for example, a classifier) to deal with a large number of functions for learning and in order to be able to generalize afterwards. Inside these functions it is often either redundant or irrelevant, and their use usually affects the complexity and the need for computational resources.
- Cluster heterogeneity. Distance-based clustering algorithms tend to find spherical clusters with the same size and density. Clustering algorithms that can detect clusters of arbitrary shape, size, density, and data coverage help to gain a deeper understanding of the various correlations between functions, which, in turn, can greatly facilitate the decision-making process.

- Interpretable results. The high dimension of the data space is cumbersome for rendering methods.

6.1.4 Frequent Pattern Mining for IoT

IoT-based systems generate large amounts of data. Recently, much attention has been paid to promising methods for extracting interesting knowledge from data from the IoT-based system. Data mining algorithms that have low computational complexity are being developed [16-18]. The processes of forming frequently occurring patterns, creating association rules are computationally simple in this respect and are often used as methods for finding interesting knowledge.

The disadvantage of this analysis is the detection of patterns, rules that do not contain meaningful information.

Since IoT-based systems generate large amounts of data, it is necessary to use appropriate measures of significance, which have a strong correlation between the data, to search for frequently occurring patterns, associations.

The basic prerequisites of the model for the effective detection of commonly occurring patterns in IoT data are [19]:

1. Determination of the relevant significance parameters for the detection of patterns that meet the downward closure property, when all subsets of the frequent set of features are frequent, to reduce the search space.

2. Compactness of the structure of the model, obtained by using the distributed and parallel methods of data mining.

3. Adaptability of the model structure for effective analysis of the latest relevant information and extraction of relevant patterns in the data. To fulfill this condition, the optimal size of the data window is determined, which helps to avoid the rapid obsolescence of information.

The dimension of the rule space depends on the minimum threshold of parameters defining the significance of the rules. If the minimum threshold is set high, then we can extract valuable knowledge. On the other hand, at a low minimum threshold of the rule significance parameter, an extremely large number of association rules are generated, most of which are non-informative. In this case, the actual correlation in the data is hidden among a huge number of insignificant rules [20].

Frequent pattern mining in some domain often involve real challenges arise from their nature and the field of application. Frequent pattern and associations rules involve many items that hard to interpret and generate a lot of outcomes. Worth-while, noting that despite all the algorithms improvements, the obtained association rules can either be too obvious, or contradict a priori knowledge, or contain redundant information. The task of frequent pattern mining and mining association rules is to generate minimal set of rules providing complete coverage of outcomes with objective parameters, such as support and confidence greater or equal than some pre-specified thresholds of minimum support and minimum confidence, respectively.

Frequent pattern mining process include several steps.

Data is often of different types: numeric, categorical. For frequent pattern mining and further associative analysis, numerical attributes transformation into the nominal scale.

The transformation process realizes different goals depending on the approach. In a normative-oriented approach, the reduction of indicators makes it possible to determine the value of a variable with respect to certain generally accepted norms, or to compare the results, giving a definition of the value of a variable with respect to the other values.

When the criterion-oriented approach is given, the value shows the percentage of compliance with the value of the variable to a specific criterion.

Frequent pattern mining to get associative rules, sort the rules according to the class, reduce the number of sorted rules by the elevator parameter as follows.

Step 1: set formation L_1 of one-item sets c_1, that often meet and determine their support;

Step 2: set formation L_k k-item sets, that often meet. Each member of the set has a set of ordered *($i_j < i_v$, if $j<v$)* itemes *F* and he support value of the set $supp_F > supp_{min}$:

$$L_k = \left\{ \left(F_1, supp_1\right), \left(F_2, supp_2\right), ..., \left(F_q, supp_q\right), \right\}, \tag{6.2}$$

where $F_j = \left\{ i_1, i_2, ..., i_k \right\}$.

The definition from the set L_k of *k*-item sets, corresponding to a certain minimum threshold value of support.

Step 3: based on the specific sets of element sets from step 2, the formation of the set C_k rules *k*-item sets is potentially often encountered. Each member of the set has a set of ordered *($i_j < i_v$, if $j<v$)* itemes *F* and a support value of the set of *supp*.

Formation of a set *k*-item sets into frequent sets. According to this, the integration into *k*-item rules of *(k-1)*-item sets, s carried out, often encountered. Each rule $R \in C_k$ is formed by adding v to an *(k-1)*-item set *v*, that frequently occurring item with another *(k-1)*-item set *q*, that frequently occurring.

Step 4: reduction of all uninteresting rules using measures of determining the interestingness of rules.

6.1.5 Association analysis

The results of data mining methodologies are certain patterns and trends whereby we have to find out the interesting patterns best suit to our needs. However, after applying the data mining methodologies for IoT environments, a large number of patterns are evaluated. All of these patterns are not interesting. So, we have to find only interesting ones. Patterns become interesting when they are unknown till yet and not expected. With this purpose, associative analysis can be used.

The goal of associative data analysis is to identify associations between input and output data, identify the most specific factors for the qualitative separation of variables into classes, and quantitatively describe the relationship between these events. When defining associations in the data, a large number of rules are usually obtained. To determine their information value, it is necessary to use methods to reduce their number and determine from them potentially interesting ones.

In general case, association analysis algorithms generate a huge number of items and can produce up to hundreds of association rules. An association rule is an implication expression

$$\mathrm{R} : \mathrm{X} \rightarrow \mathrm{Y},$$

where X denoted antecedent and Y denotes consequent $\mathrm{X} \bigcap \mathrm{Y} = \varnothing$. Both X and Y are considered as a set of conjuncts of the form $\mathrm{c}_1,\mathrm{c}_2$ ..., c_k . The strength of the association rule is measured in terms of its support (s), confidence and interestingness.

For pair of rule-candidates, binary variables R_1 and R_2 the lift is equivalent to interest factor, which is defined as follows:

$$\mathrm{I}(\mathrm{R}_1, \mathrm{R}_2) = \frac{\mathrm{s}(\mathrm{R}_1, \mathrm{R}_2)}{\mathrm{s}(\mathrm{R}_1) \cdot \mathrm{s}(\mathrm{R}_2)}. \tag{6.3}$$

The measure of interestingness in this case can be interpreted as follows:

$$I^{I}(R_1, R_2)\begin{cases} = 1, & \text{if } R_1 \text{ and } R_2 \text{ are independent,} \\ > 1, & \text{if } R_1 \text{ and } R_2 \text{ are positively correlated,} \\ < 1, & \text{if } R_1 \text{ and } R_2 \text{ are negatively correlated.} \end{cases} \tag{6.4}$$

In order to increase the information importance of rules, it is necessary to reduce their number and focus on potentially interesting ones. Further exploration of interestingness leads us to discovering different subjective and probabilistic measures of interestingness. To determine the interestingness of the rule, various probabilistic measures are used: support, confidence, Goodman-Kraskal, Pyatetsky-Shapiro, Laplace, etc. In the present study, for reducing number of rules we applied three level technique proposed in [21] beginning with detection of deviations in data, then testing of differences among adjusted attributes and finally, quantifying the interestingness of association rules.

1. Detecting deviations in data is performed as follows.

The every conjunct cj from association rule set is represented in the form <A = V>, where A is an item name (attribute), Dom (A) is the domain of A, and I (value) $\in$ Dom (A). Degree of deviation is defined as deviation between two conjuncts $\Delta(c_i, c_j)$ and is calculated on the basis of the comparison between the items of the two conjuncts. For conjuncts c_i, c_j deviation of c_i with respect to c_j is defined as a Boolean function as follows:

$$\Delta(c_i, c_j) = \begin{cases} 0, & \text{if } A_i = A_j \text{ and } V_i = V_j, \\ 1, & \text{if } A_i = A_j \text{ and } V_i \neq V_j. \end{cases} \tag{6.5}$$

2. The differences among adjusted attributes can be calculated using the following formula:

$$\bar{d}(R_1, R_2) = \begin{cases} 0, & \text{if } |R_1| = |R_2| \forall c_i \in R_1, \exists\ c_j \in R_2, \text{ that } \Delta(c_i, c_j) = 0, \\ 1 & \forall c_i \in R_1, \neg\exists\ c_j \in R_2, \text{ that } \Delta(c_i, c_j) = 1, \\ \dfrac{\sum\limits_{c_i \in R_1, c_j \in R_2} \min \Delta(c_i, c_j)}{|R_1|}, & \text{othervise,} \end{cases} \tag{6.6}$$

where R_1 and R_2 are considered as two sets of conjuncts c_i and c_j.

Parameter value $\bar{d}$ = 0 indicates that R_1 and R_2 are identical, $\bar{d}$ = 1 indicates the maximum deviation between rule sets, and the other $\bar{d}$ values between 0 and 1 are defined as a transient deviation.

3. Quantifying the interestingness of association rules

Let $R_1 : X_1 \rightarrow Y_1$ and $R_2 : X_2 \rightarrow Y_2$ be two association rules, then interestingness of a rule R_1 with respect to the rule R_2 is calculated as follows:

$$I^{II}(R_1, R_2) = \begin{cases} 0, & \text{if } \bar{d}(X_1, X_2) = 0 \text{ and } \bar{d}(Y_1, Y_2) = 0, \\ \left(\min\limits_{S\in R} \bar{d}(X_1, X_2) + \bar{d}(Y_1, Y_2)\right)/2, & \text{if } \bar{d}(X_1, X_2) \ge \bar{d}(Y_1, Y_2), \\ \left(\bar{d}(X_1, X_2) + \min\limits_{S\in R} \bar{d}(Y_1, Y_2)\right)/2, & \text{if } \bar{d}(X_1, X_2) < \bar{d}(Y_1, Y_2), \\ 1, & \text{if } \bar{d}(X_1, X_2) = 1 \text{ and } \bar{d}(Y_1, Y_2) = 1. \end{cases} \qquad (6.7)$$

According to formula (6.7), $I^{II} = 0$ indicates that R_1 and R_2 are identical, $I^{II} = 1$ denotes maximum deviation between R_1 and R_2. Other cases indicate different deviations in the interestingness of association rules. To select interesting rules the user should specify the threshold of their interestingness. The anti-monotone property based on the threshold of the measure of interest can be applied to reduce the dimension of the resulting rule set. The anti-monotone property is that the measure of the interest of any set of elements should not exceed the minimal measure of interest of any of its subsets. This property greatly facilitates the mining rules.

6.2 Mining of Massive Datasets

IoT systems include multiple heterogeneous networked embedded devices that generate massive amounts of data. Massive Data IoT leads to problems of processing and data mining [22]. Figure 6.6 presents the main challenges associated with processing and mining massive data sets.

The large amount of data, the high transfer rate and the variety of properties of large IoT data necessitate a new requirement for intelligent analysis of such data and the diversity in data sources is also a problem [23].

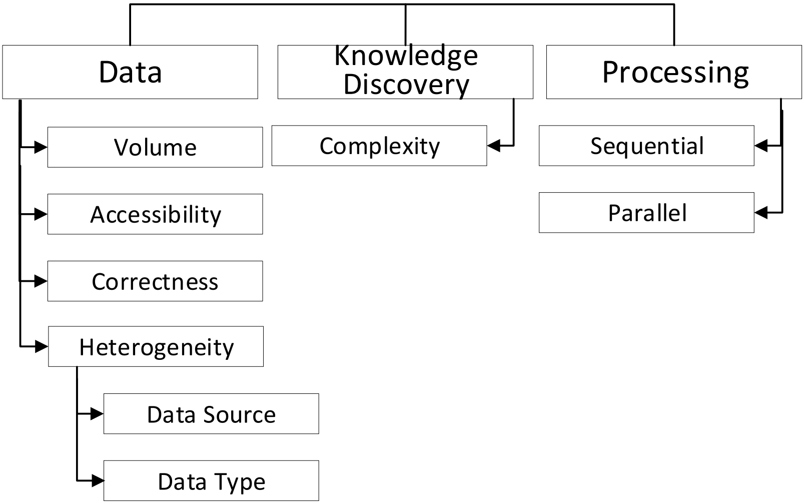


Fig. 6.6 – Data mining issue in IoT

In addition, compared to small data sets, massive data sets contain more anomalies and ambiguities that require additional preprocessing steps [24]. Another problem is to extract accurate and useful information from large volumes of diverse data.

In accordance to [25], the massive data are generally collected from different heterogeneous sources (e.g., video cameras, sensors, RFID, other IoT devices, people, etc.) providing heterogeneous sensing data (e.g., text, video, sound). In this context, heterogeneous data processing (e.g., fusion, classification) brings new challenges and open new possibilities for systems. Obviously, these random variables from heterogeneous sensors have different probability distributions.

Define z_n as the data from the n-th sensor and $\mathrm{Z} := \left\{z_n\right\}_{n=1}^{N}$ as the heterogeneous data set, the margins $\left\{z_n\right\}_{n=1}^{N}$ are generally differently or heterogeneously distributed.

In many IoT applications, problems are often modeled as multi-sensor data fusion, distribution estimation or distributed detection. For detection, this tasks joint probability density function $f(\mathrm{Z})$ of the heterogeneous data set Z is needed to get from the marginal probability density function $\left\{f(z_\partial)\right\}_{n=1}^{N}$.

In these cases, one often uses simple models such as the product model or multivariate Gaussian model, which lead to suboptimal solutions [26]. Other approaches are based on copula theory, to tackle heterogeneous data processing in IoT. In copula theory, it is the copulas function that couples' multivariate joint distributions to their marginal distribution functions, mainly thanks to the Sklar theorem.

Sklar' theorem can be present as follow. Let F be an N-dimensional cumulative distribution function with continuous marginal probability density function F_1, F_2, ..., F_N. Then there is a unique copulas function C such that for all z_1, z_2, ..., z_N in [−∞, +∞]

$$F(z_1, z_2, ..., z_N) = C(F_1(z_1), F_2(z_2), ..., F_N(z_N)) \tag{6.8}$$

Next, the probability density function can be obtained by the N-order derivative of (6.8)

$$\begin{aligned} f(z_1, z_2, ..., z_N) &= \frac{\partial^N}{\partial_{z_1}, \partial_{z_2}, ..., \partial_{z_N}} C(F_1(z_1), F_2(z_2), ..., F_N(z_N)) \\ &= f_p(z_1, z_2, ..., z_N) c(F_1(z_1), F_2(z_2), ..., F_N(z_N)) \end{aligned} \tag{6.9}$$

where $f(z_1, z_2, ..., z_N)$ is the product of the marginal probability density function $\left\{ f(z_\partial) \right\}_{n=1}^{N}$ and c(·) is the copula density weights the product distribution appropriately to incorporate dependence between the random variables. The technique on the selection of proper copula functions is presented in [27].

6.2.1 CRISP-DM data mining process methodology for IoT domain

CRISP-DM (Cross Industry Standard Process for Data Mining) is an interdisciplinary data mining standard [28]. CRISP-DM uses six steps for data mining.

1. Understanding the business - involves understanding how the goals and requirements of the project are related to the business goals, formulating the problem of data mining based on this understanding.

2. Understanding data - includes the initial stages of collecting and analyzing data to obtain initial information about their properties, determining the quality of data, identifying preliminary patterns and forming hypotheses.

3. Data preparation - includes the definition and execution of all actions that convert the raw data into the final data set. The stage includes the selection of tables, observations, variables, as well as conversion and data cleansing that are compatible with the modeling methods used.

4. Modeling - selection, application, optimization of modeling methods.

5. Evaluation - the constructed models are tested and, using selected criteria, their effectiveness is evaluated.

6. Deployment - includes the organization and presentation of knowledge generated by the model in an easily interpretable form for the end user.

Consider the steps of CRISP-DM in the context of the Internet of Things in accordance with Data Science for Internet of Things - The Problem Solving Methodology. The methodology includes: problem definition, preparation, modeling.

The basis of most research and development is the need to identify possible problems, tasks and develop optimal ways to solve them. In the context of IoT, solving a problem means solving the original problem and providing incremental feedback.

The preparation stage, unlike the standard process, must take into account the diversity of data sources and the architecture of IoT systems.

Proceeding from this, at this stage the following processes are distinguished, which should be carried out iteratively:

- definition of data requirements;
- IoT architecture design;
- collection, cleaning, intelligence data analysis;
- continuous improvement.

The determination of the necessary data is carried out taking into account the scope of IoT usage. This could be IoT for health and healthcare, smart homes and cities, smart transportation, industrial, energy systems, etc.

When designing architecture, it is necessary to take into account the technology of IoT systems, which includes sensors, networks and analytical tools.

There are several factors that influence the choice of components when solving a specific problem. The choice is determined by the accuracy and reliability, availability and security, data transfer speed, energy efficiency.

The selected constituent elements will determine some characteristics of the data and, accordingly, analytical tools.

This is followed by the collection, purification and intelligence analysis of available data. This process is typical of CRISP-DM technology and allows you to isolate additional information based on the available data, for example, incorrect, inappropriate operation of one of the network devices, problems with receiving, transmitting data. This information is used to refine the IoT architecture until an optimal solution is reached.

The modeling stage is the stage of building a model, evaluating its effectiveness and the quality of solving the problem. The stage includes the following processes:

- model design to solve a specific problem;
- model evaluation;
- model and architecture deployment.

As in the preparation stage, these processes should be carried out iteratively, until the optimal parameters of the model are reached.

Evaluation of the model is carried out using classical statistical methods and parameters. Also, it should be evaluated in terms of solving the problem. If the model does not improve the basic state of the problem, then iteration of all stages is necessary, starting from the preparation stage. since some assumptions about the data could be erroneous.

After obtaining an optimal assessment of the model, the architecture and model are deployed.

All the above steps in the context of IoT require continuous improvement. This is due to the rapidly evolving nature of IoT technology, Data Mining methods. Also, this is due to the problem being solved, which can also change and, therefore, the models used are changed to correct it.

The continuous improvement phase includes the following iterative processes:

- feedback and understanding;
- specification of the IoT architecture;
- refinement model;
- deployment of a new model and architecture.

Also, depending on the context, there are several elements that can stimulate the improvement of IoT, for example, such as:

- change in performance of the current architecture and model;
- additional user needs;
- the emergence of new tools, methods, algorithms that can help solve the problem in a cheaper or faster way;
- the emergence of new data streams.

IoT systems produce a large amount of data. Often, they are presented in the form of time series, analyzed in real time, which determines the methods of intellectual analysis at the modeling stage. The main task of time series analysis is the detection of anomalies. For this, the classical classification models used to detect anomalies are most often used. However, more and more often new approaches are being researched and developed, which show good results when analyzing streaming data, when combining data from several sources (data fusion).

6.2.2 Map reduce

In IoT applications, mass data processing such as MapReduce is constructed for parallel and distributed data processing [29]. Querying and reasoning for data can be adapted to large data is a more flexible approach.

One of the most popular parallel processing methods in cloud platform is MapReduce [30] and its open source implementation Hadoop for cloud-based parallel or distributed data processing. For the parallelization, scalability, load balancing, and fault-tolerance is MapReduce is widely used in cloud platforms for query processing for data analysis.

The MapReduce disadvantage does not directly support more complex operations such as fusion. More research on high-level, declarative management of complex data such as RDF is required for massively parallel processing of IoT data in the cloud.

1) Parallel processing methods for complex operations: a processing framework is used for massive data processing, incremental calculation, and iterative processing. The framework is implicitly used to synchronize the parallel programs execution without any user specification for events and trigger reactions to process the data. The Selective Embedded Just-InTime Specialization (SEJITS) [31] executes complex analytic queries on massive semantic graphs in big-data analytics.

2) Parallel processing methods for semi-structural data: for the RDF data processing task, effectiveness and tunable data partitioning framework SPA [32], that use at distributing processing of big RDF data, is presented to fast processing support of different size as well as complexity. A MapReduce framework is designed to carry out SPARQL query processing. Thus, RDFS reasoning can be involved in deductive databases and thus recursive query processing techniques are implemented.

3) Parallel processing methods for data stream: the stream data that push up to cloud storage and the processing algorithm is tasked with data without explicitly storing it.

The disadvantage of parallel frameworks in the cloud such as MapReduce and its variations is an unable to support complex parallel processing effectiveness. Basical algorithms of the sequential pattern may raise the scalability challenge when dealing with large data.

For problems decision of optimizing parallel data mining, a heuristic cloud bursting algorithm, Maximally Overlapped Bin packing driven Bursting (MOBB), is developed. It considers the time overlap to improve data mining parallelization. The authors [33] present Ripple, a middleware that is built on iterated MapReduce for distributed data analytics with the support of different styles of analytics in the same platform and on the same data.

Mainly, distributed processing in cloud environment based on MapReduce. It's can be carry out, after the expansion of different type (structured, semi-structured and unstructured) data. However, on consideration of some MapReduce disadvantages, such as high communication cost, unneeded processing and lack of interaction ability in real-time processing, the methods of high-performance distributed data processing without MapReduce are required in some application related to complex processing.

New pattern for parallel and dynamic data processing in large IoT data environment, data can be defined by types, state and analysis tasks. Parallel and particle data processing framework is needed to enable the execution MapReduce pattern in dynamic cloud infrastructures, in contrast with centralized master server implementations. These re-build and execution data mining algorithm are not applicable for big data analysis system. Despite its evident merits such as scalability, faulttolerance, ease programming, and flexibility, MapReduce has limitation in interactive or real-time processing on handling IoT data processing. MapReduce is not perfect for every large-scale analytical task. Its high communication cost and redundant processing is an IoT application problems.

6.2.3 Finding Similar Items

Often the data looks like a collection of data sets, and the goal is to find pairs of sets that have a relatively large proportion of common elements.

The time step data described by a certain number of features in the feature vector can be used to extract useful information as well as to construct these feature vectors.

The main definitions of finding similar items as follows [34].

Dissimilarity measure is measuring the amount of divergence between two items in relation to their feature vector. Defined the dissimilarity measure between items I_u and I_n based on the feature vector of each item follows $D_m(I_u, I_n)$. $D_m(I_u, I_n) < \delta \Rightarrow I_u \square I_n$ (I_u is similar to I_n), δ is a user denote threshold value.

Affinity group is the item-set similar to item I_u in relation to *p*-th attribute A_p of the feature vector and it is called affinity group of I_u and defined by $C_{A_p}(I_u)$.

$$\begin{aligned} C_{A_p}(I_u) &= \left\{ I_n \in D_n \,\middle|\, (I_u \square I_n) \wedge (A = A_p) \right\} \\ &= \left\{ I_n \in D_n \,\middle|\, D_m(I_u, I_n) < \delta \right\} \end{aligned} \qquad (6.10)$$

K-Similar item group: defined D_{ϖ} as the real items dataset and D_{ϖ} its locally concealed version. D_{ϖ} satisfies the property of k-similar item group (where K is defined value) provided for every item $I_u \in D_{\varpi}$. There is at least k-1 other distinct fake items $I_{n_1}, \ldots, I_{n_{(k-1)}} \in D_n$ forming affinity group as follows

$$FV(I_{n_1}) \square FV(I_u), \forall 1 \le i \le k-1 . \qquad (6.11)$$

6.3 Stream mining

Stream mining refers to a wide class of methods that can be used in IoT systems to analyze continuous data streams from multiple sources in order to detect various relationships and turn data into manageable information [35]. The results of the analysis are used to take informed action.

The stream data has a time stamp, a data source identifier and a data value read at a given time stamp. When solving certain tasks, it is possible to combine several data streams or to combine streaming data with static data.

Real-time data streaming includes various applications. Some of them are E-commerce, Network monitoring, Risk management, Fraud detection, Pricing, and analytics, etc. Information acquires from such analysis gets companies to determine many aspects of their business and customer activity such as – service usage, server and website activity, and devices positioning, people, and physical goods –and enables them to respond promptly to emerging situations.

Streaming analytic and stream processing solutions enable to perform mathematical or statistical analysis on the fly continuously. They handle high volumes of data in real time with a scalable, highly available and fault-tolerant architecture and enable data analysis in motion.

6.3.1 Stream Processing and Streaming Analytics: introduction and motivation

Stream mining can be viewed as a system for processing and analyzing streaming data from a set of source data sources for extracting valuable information in real time.

There are three main approaches for processing streaming data.

Atomic: each incoming data item is processed separately.

Micro batching: data packets are processed.

Windowing: represents the combination of the two approaches above. The method allows supports the processing of each data element, but creates pseudo-packages (windows) to make it more efficient. Also allows the use of more complex interpretations, such as a sliding window.

The choice depends on the specific problem being solved and the amount of data flow.

The figure 6.7 shows a diagram describing the stream mining process for healthcare data, supplemented by historical data, taking into account the simultaneous processing of data by various logical operations, which may include calculations, filtering data (leaving only useful, meaningful information), transformation, merging, search for patterns, etc.

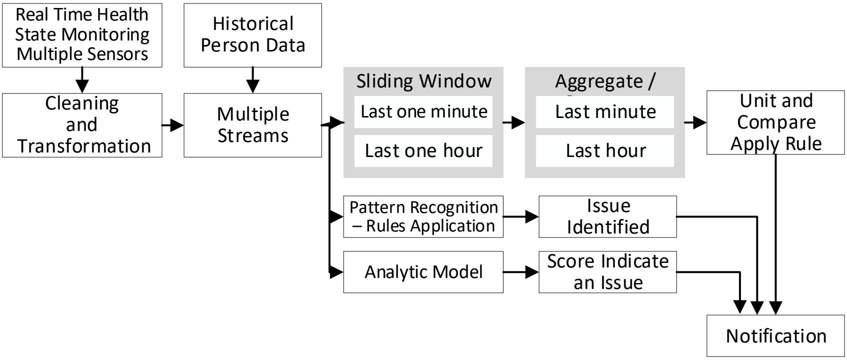


Fig. 6.7 – Stream mining process, supplemented by historical data

Continuous streaming data can be obtained from several sources, and the time step of their transmission may differ. Presents the possible use of static data in conjunction with stream data. Data received from sensors, devices, may be noisy, with missing values, etc. Therefore, the following is the stage of cleaning and transformation of data. At this stage, if necessary, data normalization can be carried out, combining several data streams for analysis.

Next, the size of the sliding windows is determined, which may differ. It helps to compare what is happening now with what has happened recently. These windows can be quite large, up to the size of the window in weeks, months, and will save data for these periods. Aggregation and summing up allow us to obtain statistical parameters, such as, for example, minimum, maximum, average, standard deviation, etc.

Next is the combination and comparison, which includes the rules obtained when creating models.

The diagram represents an analytic step that performs pattern matching. In this context, pattern matching is similar to business rules. For example, if the variable X is above a certain value, and the variable Y is below a certain value, and the variable Z is outside a certain range, then send a warning.

We combine, compare data, apply the received rules and get the value of the output event.

6.3.2 Real-Time Data-Stream Analysis

The overall goal in the development of streaming data processing algorithms is to calculate functions on vector *A* at different points during the entire lifetime of the stream. The main problem in the flow model for calculating queries is that the size of the flow vector n is usually huge, which makes it impractical (or even impracticable) to be stored. The workspace for streaming data processing algorithms is limited to a small amount of data received over a small amount of time.

6.3.3 Data Streaming Models & Basic Mathematical Tools

Stream data is represented by a massive, dynamic one-dimensional vector *A*[*1 ... n*] and involves the use of standard presentation methods (for example, rows or columns) [36].

Dynamic vector *A* is represented as a continuous stream of updated data, where the *j*-th update is presented as $\langle k, c[j] \rangle$ and changes the *k*-th record of *A* as following $A[k] \leftarrow A[k] + c[j]$.

Time-series model is one of the types of streaming data. In this model, the *j*-th update is presented as $\langle j, A[j] \rangle$ and updates come in ascending order *j*, simulating the data series of time series.

The conventional streaming model assumes that streaming starts with a well-defined starting point t_0 and at any time t takes into account all streaming updates between t_0 and t. However, with the dynamic changing nature of the data, such models quickly become obsolete, become irrelevant and may give incorrect results.

Sliding window model [37] is one of the most noticeable and intuitive models, which essentially considers only the window of the latest updates observed in the stream so far - updates outside the window are automatically deleted. The determination of the window size can either be based on time (for example, updates observed in the last W time units), or based on counting observations (for example, the last W updates). The main limiting condition in this model is the size of the window W.

A window can be represented as $\{W_j: j\geq 1\}$ of length n, where $W_j=\{e_j, e_{j+1}, ..., e_{j+n-1}\}$ is a window of consecutive data elements. To create a window, various algorithms are used, the main difference of which lies in the balance between the amount of required memory and the degree of dependence between successive patterns of elements.

6.4 Work related analysis

Data mining techniques for IoT based applications has been present in literature for areas tasks decision, such as supervised and unsupervised learning for IoT applications [5, 14, 32], frequent pattern recognition and association analysis [4, 19, 21], massive IoT data mining [22, 24], stream data mining [36]. he detailed surveys on the approaches, tools and techniques employed in existing for IoT data mining can be found in [2, 23].

Practical skills in mass data processing for IoT applications are presents in [31, 32, 33] for parallel and distributed data processing.

The using of data mining fundamental techniques for IoT are directs to map reduce, finding similar items. Its help the scientists and researchers to develop, control and monitoring the IoT-based application in different area.

Our partners from EU universities are actively involved in research and development for IoT data mining technique, using new knowledge to theoretical and applied sciences and, also, in education. For example, the University of Newcastle upon Tyne focuses on a wireless networking for computers, embedded devices or sensors and suggests a postgraduate course in Internet of Things and Sensor Networks. In this course, it is expected that most students are received the practical experience of wireless networking for computers, embedded devices or sensors and demonstrated knowledge of techniques and theory in this area [38].

The University of Coimbra is proposing a master course - Management of Internet Infrastructures and Services. It's focuses on the management of IT infrastructures, including the planning and management of the network services, planning, monitoring, fault management, service level management, the related network and service management protocols, and the administration of IT services and infrastructures based on virtualized environments and/or installed according to cloud-based paradigms [39].

The Knowledge Discovery and Data Mining course is suggesting in KTH [40]. The course is involved some KDD aspects, for example, such as classification and clustering, Bayesian networks and graphical models, prediction and sequence mining etc. As a learning result student will know the fundamental approaches to knowledge discovery and data mining, the main theoretical foundations, as well as its code of practice.

Conclusions and questions

Section 6, the materials for module MC 2.2 of master's course "Technologies of data mining and processing" are presented. They can be used for preparation for lectures and self-learning. The aim of the module is to give master students knowledge of principles and aspects of the IoT-based technologies of data mining and processing: teach to the basic idea of using data mining for IoT, data mining models, and techniques for IoT, stream mining and mining of massive datasets.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What are the IoT data properties?
2. What are the main IoT data characteristics?
3. The difference between classification and clustering analyzes.
4. What problems arise from the data analysis?
5. The general limitations of IoT devices and sensors that must be considered.
6. The difference between missing completely at random (MCAR) data and not missing at random data (NMAR).
7. What are the general methods for handling missing data?
8. What are the estimates of efficiency of methods for handle missing data?
9. What challenges decides from the clustering data?
10. How to estimate classification error rate (CER)?
11. What the goal of frequent pattern mining and mining association rules?
12. What the general measures are used to determine the interestingness of the rule?

13. What are the CRISP-DM steps?
14. MapReduce advantages and disadvantages.
15. What the main approaches for processing streaming data?
16. How sliding windows can be represented?

References

17. A. Jaokar, "Data Science for Internet of Things (IoT): Ten Differences from Traditional Data Science" *Kdnuggets. com*, 2019. [Online]. Available: https://www.kdnuggets.com/2016/09/data-science-iot-10-differences.html [Accessed: 22 December 2018].

18. F. Chen, P. Deng, J. Wan, D. Zhang, A. Vasilakos and X. Rong, "Data Mining for the Internet of Things: Literature Review and Challenges", *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 431047, 2015. DOI: 10.1155/2015/431047.

19. A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: part I", *IEEE Transactions on Evolutionary Computation*, vol.18, no.1, pp. 4–19, 2014. DOI: 10.1109/TEVC.2013.2290086.

20. Bhuiyan, M.Z.A. and Wu, J., "Event detection through differential pattern mining in Internet of Things", *In 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, October 2016, pp. 109-117. DOI: 10.1109/MASS.2016.024.

21. D. Lee, H. Lee, "IoT service classification and clustering for integration of IoT service platforms", *The Journal of Supercomputing*, vol.74, no.12, pp.1-17. DOI: 10.1007/s11227-018-2288-7.

22. Viswanathan, H. et al (2015) "Uncertainty-aware autonomic resource provisioning for mobile cloud computing". *IEEE Trans Parallel Distribution Systems*, vol.26, no.8, pp. 2363–2372. DOI: 10.1109/TPDS.2014.2345057.

23. H. Chen, et al, "Uncertainty-aware real-time workflow scheduling in the cloud". *In 2016 IEEE 9th; International Conference on Cloud Computing (CLOUD),* June 2016, pp. 577–584. DOI: 10.1109/CLOUD.2016.0082.

24. P. Jamshidi, C. Pahl, NC Mendonça, "Managing uncertainty in autonomic cloud elasticity controllers". *IEEE Cloud Computing*, vol.3, no.3, pp. 50–60. DOI: 10.1109/MCC.2016.66.

25. P. Gupta, R. Gupta "Data Mining Framework for IoT Applications", *International Journal of Computer Applications,* vol.174, no.2, pp. 4-7. DOI: 10.5120/ijca2017915316.

26. C.-W. Tsai, C-F. Lai, M.-C. Chiang, L.T. Yang, "Data Mining for Internet of Things: A Survey". *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77-97. DOI: 10.1109/SURV.2013.103013.00206.

27. M.S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A.P. Sheth, "Machine learning for Internet of Things data analysis: A survey". *Digital Communications and Networks,* vol.4, no.3, pp.161-175. DOI: 10.1016/j.dcan.2017.10.002.

28. C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006. DOI: 10.1117/1.2819119.

29. A. Amini, H. Saboohi, T. Ying Wah, T. Herawan, "A fast density-based clustering algorithm for real-time internet of things stream". *The Scientific World Journal*, 2014. http://dx.doi.org/10.1155/2014/926020.

30. I. Yankine, "Unsupervised clustering of IoT signals through feature extraction and self organizing maps", *Padova Digital University Archive* 2017. [Online]. Available: http://tesi.cab.unipd.it/54590/ [Accessed: 23- Feb- 2019].

31. Boukerche, R.W. Pazzi, and R.B. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications", *In Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems* October 2004, pp. *157*-164. DOI: 10.1145/1023663.1023692.

32. A. Boukerche, S. Samarah, "Novel Algorithm for Mining Association Rules in Wireless Ad-hoc Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems,* vol.19, no.7, pp. 865-877. DOI: 10.1109/TPDS.2007.70789.

33. S.K. Tanbeer, C.F. Ahmed and B.S. Jeong, "An Efficient SinglePass Algorithm for Mining Association Rules from Wireless Sensor Networks", *IETE Technical Review*, vol. 26, no.4, pp. 280-289. DOI: 10.4103/0256-4602.52997.

34. M. Rashid, I. Gondal, and J. Kamruzzaman, "Mining associated patterns from wireless sensor networks", *IEEE Transaction on Computers*, vol. 64, no. 7, pp. 1998–2011, 2014. DOI: 10.1109/TC.2014.2349515.

35. Y.K. Lee, W.Y. Kim, Y.D. Cai and J. Han, "CoMine: Efficient Mining of Correlated Patterns", *Proc. on ICDM*, November 2003, pp. 581-584. DOI: 10.1109/ICDM.2003.1250982.

36. H. Kaur, S.K. Wasan, A.S. Al-Hegami, V. Bhatnagar, "A unified approach for discovery of interesting association rules in medical databases". *In: Perner, P. (ed.) ICDM 2006. LNCS*, July 2016 vol. 4065, pp. 53–63. doi:10.1007/11790853_5.

37. T. Hu, H. Chen, L. Huang, and X. Zhu, "A survey of mass data mining based on cloud-computing", *In Proc. Anti-Counterfeiting, Secur. Identificat.*, August 2012, pp. 1–4. DOI: 10.1109/ICASID.2012.632535.

38. M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I.A.T. Hashem, A. Siddiqa, I. Yaqoob, "Big IoT data analytics: architecture, opportunities, and

open research challenges". *IEEE Access*, vol.5, pp.5247-5261. DOI 10.1109/ACCESS.2017.2689040.

39. A. Gani, "A survey on indexing techniques for big data: Taxonomy and performance evaluation", *Knowl. Inf. Syst.*, vol. 46, no. 2, pp. 241–284. DOI 10.1007/s10115-015-0830-y.

40. G. Ding, L.Wang, Q. Wu, "Big data analytics in future internet of things", *Arxiv.org*, 2019. [Online]. Available: https://arxiv.org/abs/1311.4112 [Accessed: 23- Feb- 2019].

41. D. Mari, S. Kotz, "Correlation and Dependence". London, U.K.: Imperial College Press, 2001.

42. S. G. Iyengar, P. K. Varshney, and T. Damarla, "A parametric copula-based framework for hypothesis testing using heterogeneous data", *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2308-2319. DOI: 10.1109/TSP.2011.2105483.

43. A.Shi-Nash, D.R. Hardoon, "Data analytics and predictive analytics in the era of big data", *Internet of Things and Data Analytics Handbook*, pp.329-345. DOI: 10.1002/9781119173601.ch19.

44. H. Cai, B. Xu, L. Jiang, A. V. Vasilakos, "IoT-based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges", *IEEE Internet of Things Journal*, vol.1, no.4, pp. 1–1. doi:10.1109/jiot.2016.2619369

45. S. Blanas, J. M. Patel, V. Ercegovac, J. Rao, E. J. Shekita, and Y. Tian, "A comparison of join algorithms for log processing in mapreduce", *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM*, June 2010, pp. 975–986. DOI: 10.1145/1807167.1807273.

46. K. Lu, M. Sun, C. Li, H. Zhuang, J. Zhou, and X. Zhou, "Wave: Trigger based synchronous data process system", *In Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on. IEEE*, May 2014, pp. 540–541. DOI: 10.1109/CCGrid.2014.124.

47. A. Lugowski, S. Kamil, A. Buluc¸, S. Williams, E. Duriakova, L. Oliker, A. Fox, and J. R. Gilbert, "Parallel processing of filtered queries in attributed semantic graphs", *Journal of Parallel and Distributed Computing*, vol. 79, pp. 115–131. doi:10.1016/j.jpdc.2014.08.010.

48. K. Lee, L. Liu, Y. Tang, Q. Zhang, and Y. Zhou, "Efficient and customizable data partitioning framework for distributed big rdf data processing in the cloud", *In IEEE CLOUD*, June 2013, pp. 327–334. DOI: 10.1109/CLOUD.2013.63.

49. A. M. Elmisery, M. Sertovic, B. B. Gupta, "Cognitive Privacy Middleware for Deep Learning Mashup in Environmental IoT", *IEEE Access*, vol.6, pp. 8029–8041. doi:10.1109/access.2017.2787422.

50. L. Canzian, M. V. Der Schaar, "Real-time stream mining: online knowledge extraction using classifier networks", *IEEE Network*, vol.29, no.5, pp. 10–16. doi:10.1109/mnet.2015.7293299.

51. M.J. Carey, S. Ceri, P. Bernstein, U. Dayal, C. Faloutsos, J.C. Freytag, G. Gardarin, W. Jonker, V. Krishnamurthy, M.A. Neimat, P. Valduriez, "Data-Centric Systems and Applications" 2008.

52. M. Garofalakis, J. Gehrke, R. Rastogi (eds.) "Data Stream Management Processing High-Speed Data Streams". Springer-Verlag Berlin Heidelberg, 2016.

53. M. Datar, A. Gionis, P. Indyk, R. Motwani, "Maintaining stream statistics over sliding windows", *SIAM J. Comput.*, 2002, vol.31, no.6, pp. 1794–1813. DOI: 10.1137/S0097539701398363.

54. "The Internet of Things - Postgraduate - Newcastle University", *Ncl.ac.uk*, 2019. [Online]. Available:. https://www.ncl.ac.uk/postgraduate/modules/EEE8092/ [Accessed: 20- Feb-2019].

55. "Management of Internet Infrastructures and Services - Master course", *Apps.us.pt*, 2018. [Online]. Available: https://apps.uc.pt/courses/EN/unit/79262/15101/2017-2018?type=ram&id=5041. [Accessed: 20- Feb- 2019].

56. KTH | IK3342", *Kth.se*, 2019. [Online]. Available: https://www.kth.se/student/kurser/kurs/DD3342?l=en. [Accessed: 20- Feb-2019].

7. DEEP LEARNING FOR IoT

Assoc. Prof., Dr. V. S. Koval, Asst. Prof., Dr. M. P. Komar (TNEU)

Contents

Abbreviations

ADAS – Advanced Driver Assistance Systems

AI – Artificial Intelligence

ANPR – Automatic Number Plate Recognition

CCTV – Closed-Circuit Television

CNN – Convolutional Neural Networks

CUDA – Compute Unified Device Architecture

DDoS – Distributed Denial-of-Service

DNN – Deep Neural Networks

DRM – Direct Path Ratio Mask

ES – Embedded System

IoT – Internet of Things

IT – Information Technology

NECA – North East Combined Authority

NT – Network Technology

RBM – Restricted Boltzmann Machine

ReLU – Rectified Activation Function

SGD – Stochastic Gradient Descent Method

7.1 Basics of machine learning and neural networks

`

7.1.1 Introduction to machine learning and artificial intelligence

The most important features of IoT include artificial intelligence (AI), connectivity, sensors, active engagement, and small device use. One of the possible things that combine IoT and AI is Smart Things [1]. This concept is the combination of three technologies: the Internet of Things, Artificial Intelligence and Semantic Web (Fig. 7.1)

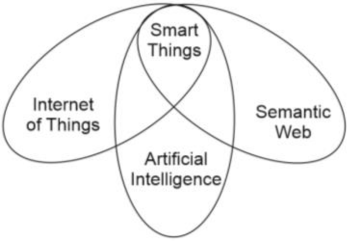


Fig. 7.1 – Concepts of Smart Things

Artificial intelligence is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry. Research associated with artificial intelligence is highly technical and specialized.

Machine learning is a class of artificial intelligence methods. The key feature of it is providing learning during generating a solution for the problem. In order to design such methods, mathematical statistics, numerical methods, optimization methods, probability theory, graph theory, and various other techniques for working with data are used [2, 3]. Therefore, machine learning is a part of artificial intelligence in the field of computer science. The relationship between the components of artificial intelligence is shown on Fig. 7.2.

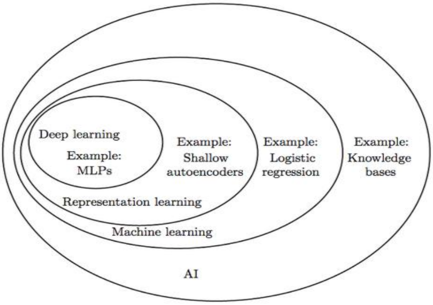


Fig. 7.2 – Artificial Intelligent Structure [4]

The development of artificial intelligent systems that combine the advantages of biological creatures and modern computer technology creates the potential prerequisites for the transition to a qualitatively new stage of evolution in computing. One of the power techniques in artificial intelligence is artificial neural networks. This technique is very important for applications in IoT.

Neural networks have different periods of their development [5]:

- 1943, W. MCculloch and W. Pitts - have shown that any logical functions can be realized using of the threshold neural elements;

- 1949, Donald O. Hebb proposed a teaching law for artificial neural networks;

- 1959, Gerald F Rosenblatt proposed a neural network model, called as Perceptron;

- 1959, W. Widrow and M. Hoff proposed a training procedure, called as "Delta rule";

- 1969, M. Minsky and S. Papert showed limitations of the perceptron;

- 1986, the authors Rumelhart, Hinton, Wiliams proposed a back propagation algorithm, which became an effective means for training of the multilayer neural networks;

- 2006, there was a new breakthrough in the development of multilayer perceptron known as the deep neural networks (DNN), introduced by G. Hinton.

Deep neural networks in the general case represent the further development of multilayer perceptron. They allow processing and analysis of a large amount of data, as well as modeling cognitive processes in various fields due to the multilayered architecture. At present, most high-tech companies use DNN to design a variety of intelligent systems. According to the opinion of scientists, the deep neural networks are capable in the near future to significantly change the everyday life of most people on our planet.

7.1.2 The model of neuron

The main element of the artificial neural network is the artificial neuron. It performs a nonlinear transformation of the sum of input signals, multiplied by weighting coefficients:

$$y = F(\sum_{i=1}^{n} \omega_i x_i) = F(WX),$$

where $X = (x_1, x_2, \ldots, x_n)^T$ − input signal; $W = (\omega_1, \omega_2, \ldots, \omega_n)$ − weight vector; F − nonlinear transformation function.

The nonlinear transformation function is called the activation function of the neuron. The scheme of the neuron is shown in Fig. 7.3 and consists of an adder and a unit of non-linear transformation *F*. Each *i*-th input of the neuron corresponds to a weighting factor (synapse), which characterizes the strength of the synaptic connection by analogy with the biological neuron (Fig. 7.3) [6].

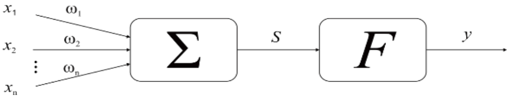


Fig. 3 – Model of artificial neuron

The sum of the products of weighting coefficients is called a weighted sum. It is a scalar product of the vector of weight coefficients on the input vector:

$$S' = \sum_{i=1}^{n} \omega_i x_i = (W, X) = |W| \cdot |X| \cdot \cos\alpha,$$

where $|W|, |X|$ − the length of the vectors W and X; α – angle between the vectors W i X.

The lengths of *W* and *X* vectors are determined basing on their coordinates:

$$|W| = \sqrt{\omega_1{}^2 + \omega_2{}^2 + \ldots + \omega_n{}^2},$$

$$|X| = \sqrt{x_1{}^2 + x_2{}^2 + \ldots + x_n{}^2}.$$

Since the length of a weight vector after training is $|W| = \mathrm{const}$ for a neuron element, the magnitude of the weighted sum is determined by the projection of the input on the weight vector:

$$S' = |W| \cdot |X| \cdot \cos\alpha = |W| \cdot X_W,$$

where X_W – the projection of the input vector X on the weight vector W .

If the input vectors are normalized $|X| = \text{const}$, then the magnitude of the weighted sum depends only on the angle between the vectors X and W . Then, for the different input signals, the weighted sum will be varied according to the cosine-law. It reaches its maximum value with the collinearity of the input and weight vectors.

If the communication power ω_i is negative, then it is called as braking. Otherwise, the synaptic connection is intensifying. The vector of the input signal is called as the pattern of the input activity, and the vector of the output signal is the pattern of the output activity of the neural network.

The considered model of artificial neuron is biologically inspired and is used to construct artificial neural networks.

7.1.3 The classification of artificial neural networks

The artificial neural networks can be classified according to the different criteria [5, 6, 7, 8]:

By the nature of the training *process:*

1) Supervised (with the teacher), when the output space of the neural network outputs is known. It is assumed that there are input signals and reference reactions to them. In the process of training there is a purposeful modification of the synaptic connections of the neural network in order to achieve the best correspondence between the real output values of the network *Y* and their reference values *e* (Fig. 7.4).

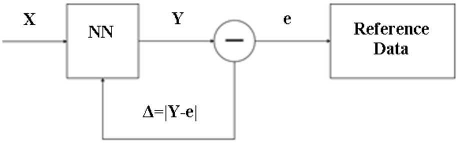


Fig. 7.4 – Supervised training

2) Unsupervised (without a teacher). In this case, the neural network forms the output space of decisions basing only on the incoming actions. Such networks are called self-organizing (Fig. 7.5).

3) Reinforcement learning. It occurs based on the reinforcement signal *r* from the external environment (Fig. 7.6).

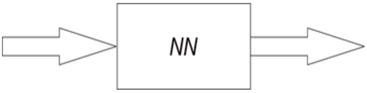


Fig. 7.5 – Unsupervised training

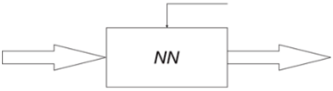


Fig. 7.6 – Reinforcement learning

By the nature *of the synapse setting:*

1) Fixed-line networks. In this case, the weighting coefficients of the artificial neural network are selected immediately, based on the condition of the problem, with:

$$\frac{dW}{dt} = 0 ,$$

where W – characterizes the weights of the network.

2) Networks with the dynamic connections. For them, in the learning process, there is a setup procedure of synaptic connections, with:

$$\frac{dW}{dt} \neq 0 .$$

According to architecture and training procedures the following neural networks classifies:

I. Perceptron neural networks. They are usually supervised one. Their architecture are based on a multilayer perceptron with gradient descent method of training. These type of artificial neural networks include:

1.1. Multilayer perceptron.

1.2. Recurrent neural networks in which there is a feedback between input and output. The initial value at the same time is determined depending on both the incoming and the previous output values of the neural network (Fig. 7.7).

1.3. Recycling neural networks (autoencoder, auto-associative, replication networks), which are characterized by both direct $y = f(x)$ and reverse $x = f^{-1}(y)$ transformation of information (Fig. 7.8).

1.4. Convolutional neural networks that represent the further development of the perceptron.

1.5. Deep neural networks that carry out a deep nonlinear hierarchical transformation of information.

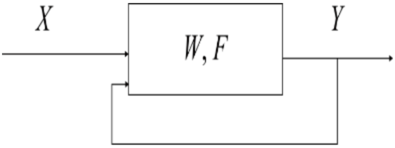


Fig. 7.7 – Recurrent neural network

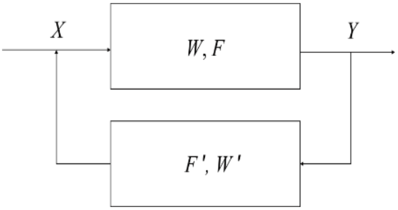


Fig. 7.8 – Recycling neural network

II Self-organized neural networks. They are characterized by learning without a teacher (unsupervised). Such training is based only on signals from the environment. These types of artificial neural networks include:

2.1. Kohonen Neural Network.

2.2. Neural networks of adaptive resonance.

III Relaxation neural networks in which the circulation of information takes place until the initial values stop to change (equilibrium state):

3.1. Hopfield Neural Networks.

3.2. Hamming's Neural Network.

3.3. Bidirectional associative memory.

IV. Hybrid neural networks. Characterized by using of two approaches to train – supervised and unsupervised:

4.1. Counter propagation networks.

4.2. Neural networks with radial-basic activation function (RBF networks).

4.3. Fuzzy neural networks that are characterized by using combination of fuzzy logic and neural networks.

V. Neural immune networks that characterized by using both of artificial immune systems and neural networks.

7.2 Deep learning neural networks

These types of neural networks have been successfully used to solve various problems in artificial intelligence, such as processing and speech recognition, natural language processing, pattern recognition, visualization, etc. All of the mentioned solutions are parts of Internet of Things applications. In general, deep neural networks perform deep hierarchical transformation and represent a neural network with many layers of neural cells.

7.2.1 The specifics of deep learning neural networks architecture

The following deep neural network (DNN) architectures are using worldwide:

- deep belief neural networks;
- deep perceptron;
- deep convolutional neural networks of different types: R - CNN, Fast - CNN, Faster - CNN, SSD, ResNet and so on;
- deep recurrent neural networks;
- deep autoencoder;
- deep recurrent convolutional neural network (deep RCNN).

Historically, the first neural networks were deep belief network and deeper perceptron, which generally are multilayer perceptron with more than two hidden layers. The main difference between the deep belief neural networks from deep perceptron consist in a fact that the first one is not a feed forward network. By 2006, the scientific community believed that there are no sense to apply perceptron with more than two hidden layers for efficient nonlinear conversion of input data space into the output. This paradigm was based on the theorem, that perceptron with one hidden layer - universal approximator. Another aspect of the problem is that attempts to use backpropagation algorithm for training perceptron with three or more hidden layers don't lead to improvement in various tasks. Using of the generalized delta rule for training perceptron with many hidden layers leads to attenuation gradient in the propagation of the signal from the last to the first layer. In 2006, James Hinton suggested "greedy" algorithm of layered learning (greedy layer-wise algorithm), which became effective learning tool for deep neural networks. It was shown that the deep neural network has high efficiency nonlinear transformation and presentation of data compared to traditional perceptron.

The first proposed convolutional neural network model was Yann LeCun - LeNet, with architecture that is shown in Fig. 7.9. It was the first model contained convolutional layers that alternate twice and sub-sampling layers and three fully-connected layers. This architecture with the parameters is considered a classic [9]. This architecture without major changes still used in medical applications for diagnostic by recognizing images and video.

Another significant convolutional neural network is a network AlexNet, proposed Alex Krizhevsky [9]. AlexNet convolutional neural network designed to recognize the objects with any complexity in large image dimensions. In 2012, it won the competition ImageNet, significantly ahead of its competitors.

This network was very similar to the network LeNet. The main difference between LeNet and AlexNet consists in more massive and complex architecture [10]. It also has only one convolutional layer, several pooling layers and fully-connected layers. The architecture of the AlexNet neural network is shown on Fig. 7.10.

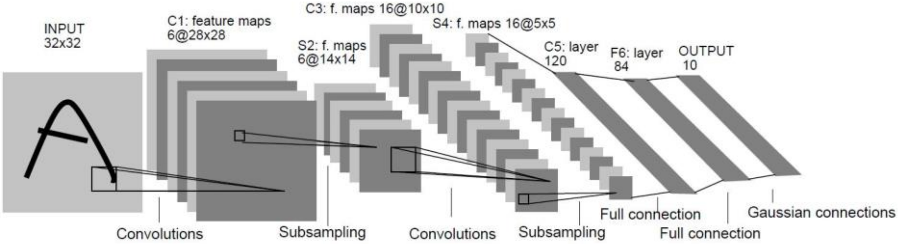


Fig. 7.9 – Architecture of convolutional neural network LeNet

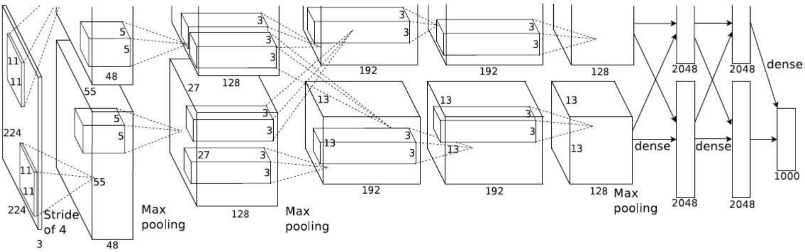


Fig. 7.10 – Network architecture AlexNet

The researches devoted to the development of the network shows the content of about 96 convolutional filters used in the design of neural networks (Fig. 7.11).

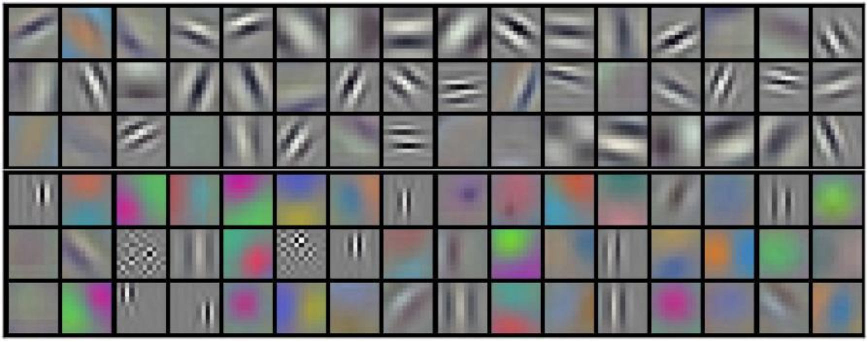


Fig. 7.11 – Filters AlexNet neural network

Another significant contribution to building of a highly efficient neural network architectures became ZF Net, established Matthew Zeiler and Rob Fergus in 2013. This network is a modification of AlexNet network. The main features of which are the better set of network parameters, increasing the size of convolutional layers and reducing the size of the displacement and size filters in the first convolutional layer. ZF Net network architecture is shown on Fig. 7.12. This type of neural network is very efficient for working with images.

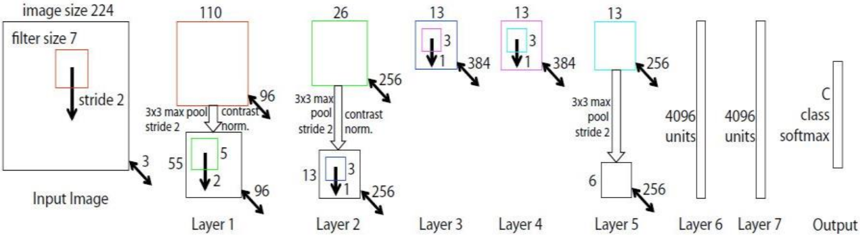


Fig. 7.12 – ZF Net Network architecture

In 2014, Google introduced GoogLeNet architecture of the convolutional neural networks [11]. This convolutional network was winner Large Scale Visual Recognition Challenge 2014 (ILSVRC). The architecture of GoogLeNet is very deep - up to 22 layers. Despite this, it has 10 times fewer parameters than AlexNet network, which positively affects to the performance and memory usage. Also, it uses small size filters, and pooling layer is implemented by choosing the mean value. The architecture of the network shown in Fig. 7.13.

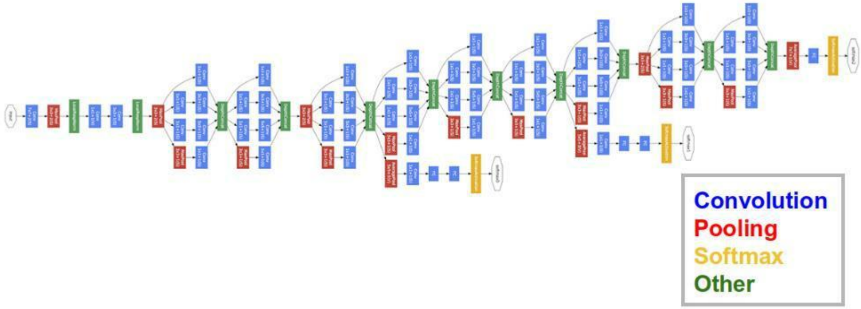


Fig. 7.13 – GoogLeNet architecture of convolutional neural network

The next significant contribution to the development of neural network architectures made VGGNet network, created Karen Simonyan and Andrew Zisserman, as well as GoogLeNet, participated in the competition ILSVRC 2014. This convolutional neural network, as well as GoogLeNet, is very deep (up to 16 layers) and consists of many convolutional and pooling layers with small dimensions (3x3 - size of convolutional layer, 2x2 - size of pooling layers). The disadvantage of this network consists in storing up to 140 million network parameters, making it huge and low productivity [12].

One of the most modern convolutional neural networks today is ResNet. It won the competition ILSVRC 2015. The architecture of the network involves a large number of convolutional layers containing a large number of filters (512) with a small size (3x3). The depth of the network can reach up to 152 layers. This type of network is one of the most effective convolutional neural networks today.

7.2.2 The training methods of deep learning neural network

Training of a neural network – it is a process of determining the weights of connections between neurons. There are three approaches to train of the artificial neural networks: supervised training, unsupervised learning training and reinforcement learning that were described above in section 7.1.3.

In practice, there are two basic methods for training of the DNN.

1) The method of prior training, which consists in two phases:

– pre-training of neural network by sequentially training each layer step by step starting from the first one. This training is unsupervised and is based on a restricted Boltzmann machine (RBM);

– fine-tuning of the synaptic connections of the entire network using the backpropagation algorithm or "wake-sleep algorithm".

2) Stochastic gradient descent method (SGD) with rectified activation function (ReLU).

Currently there are adopted the following paradigms for studying of deep neural networks. If the dimension of the training set is much higher than the number of configurable network settings, then it is necessary to use SGD training method for the neural network. If the dimension of the training set is comparable with the number of adjustable network parameters, then it is necessary to use pre-training method for the neural network based on RBM and backpropagation algorithm.

An important step in the training methods is pre-training of deep layers. There are two basic approaches: autoencoder and RBM (Fig. 7.14).

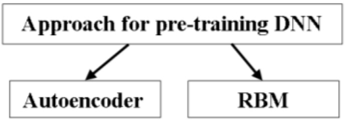


Fig. 7.14 – Approaches for pre-training of deep neural networks

The first approach is called autoencoder and is based on the notion of each layer as auto associative neural network. The second approach is based on the representation of each layer of the neural network in the form of restricted Boltzmann machine. Let consider the autoencoder training approach in more details.

Autoencoder Training approach

This approach is based on the principle of presenting each network layer as autoassociative neural network. In this case, training process star from training of a first layer as autoassociative neural network to minimize the total square error, then the second and so on. In this case each network layer trained by using backpropagation algorithm. After that, the procedure of fine-tuning for synaptic connections is applied to train the all DNN using backpropagation algorithm.

In order to see the procedure in practice, let consider a perceptron with three hidden layers (Fig. 7.15). Then, according to autoencoder method primarily taken the two layers of the neural network (1 and 2). Thereafter PCA-neural network is constructed with three layers as 1-2-1 (Fig. 7.16a). The backpropagation algorithm is applied to train the network with minimizing the square error. The duration is usually no more than 100 epochs.

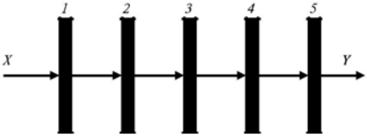


Fig. 7.15 – Perceptron with three hidden layers

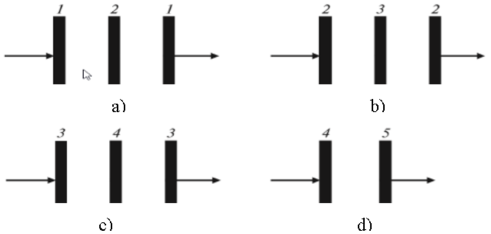


Fig. 7.16 – Autoencoder teaching method:
a) - 1st and 2nd layer; b) - 2nd and 3rd layer;
c) - 3rd and 4th layer; d) - 4th and 5th layer

Then, it is rejected the restoring layer (last layer) and fixing the weights of hidden layer and construct autoassociative network with these two layers neural network 2 - 3 - 2 (Fig. 7.16b) that learns on the data coming from the previous (2nd layer). The process continues until the last (Fig. 7.16d). As a result of layered training it is obtained the pre-trained self-organized neural network. Further fine-tuning is provided by using supervised backpropagation algorithm.

There are a lot of deep neural network architectures and training methods. Currently, there are two dominant architectures: convolutional neural networks that have been successfully used for computer vision problems, and recurrent networks, widely used for natural language processing tasks.

The first convolutional neural network trained by a combination of supervised learning using autoencoder networks and deep-belief methods. Modern techniques use only supervised training methods.

Another important direction in the development of neural networks is transfer learning. This approach consists in using of a neural network that trained on one type of data for other types of problems. As a result, it is obtained less time for training. Perspective is also sharing convolutional and recurrent neural networks, reinforcement learning.

7.3 Deep learning neural network applications for IoT

7.3.1 Attack detection scheme using deep learning approach for IoT

Internet of things is at the beginning of its evolution and currently have serious necessity in the information security issues. IoT security has become one of the most important aspects of new technologies. They are not perfect and could be attacked by DDoS (Distributed Denial-of-Service) attacks. For example, the DDoS-attack with a malicious Mirage code was occur in September 2016 as a significant event. In it, the hundreds of thousands cameras and other devices from video surveillance systems were involved as a means of attack. The results of researches from Raconteur company claims that 25% of all cyberattacks by 2020 will be implemented in the IoT area [13].

DDoS is a network attack directed to the occurrence of a situation in which service in a network stop functioning. These attacks are characterized by the generation of a large amount of traffic simultaneously with a large number of IP addresses, which leads to overload and blocking the server. Deep neural network can be used to detect and classify DDoS attacks. There exist special NSL-KDD data set at the Internet that could be used to research DDoS attack [14]. Each sample consists of 41 parameters. Each record has a "normal" or "attack type" label and distinguish six types of DDoS attacks: back, ground, neptune, pod, swamp, tear.

Currently, there are such popular frameworks as Caffe, Theano, Tensorflow and Torch that are used for deep neural network implementation.

Fig. 7.17 shows an example of a deep neural network architecture that was selected experimentally using the Caffe framework [15].

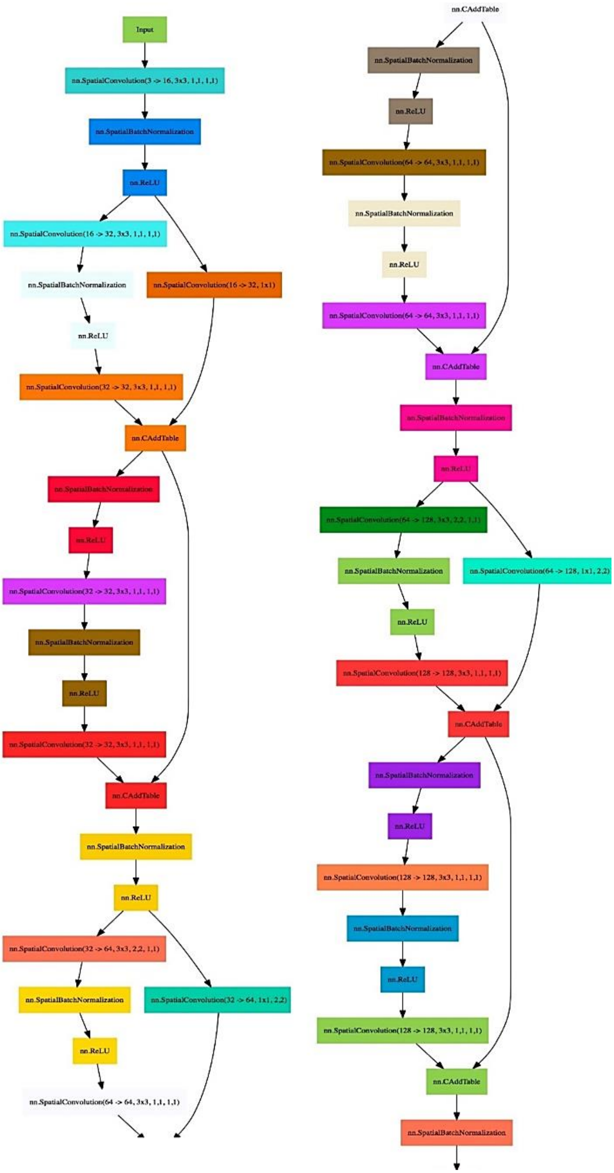

Input
nn.SpatialConvolution(3 -> 16, 3x3, 1,1, 1,1)
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(16 -> 32, 3x3, 1,1, 1,1)
nn.SpatialBatchNormalization
nn.SpatialConvolution(16 -> 32, 1x1)
nn.ReLU
nn.SpatialConvolution(32 -> 32, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(32 -> 32, 3x3, 1,1, 1,1)
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(32 -> 32, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(32 -> 64, 3x3, 2,2, 1,1)
nn.SpatialBatchNormalization
nn.SpatialConvolution(32 -> 64, 1x1, 2,2)
nn.ReLU
nn.SpatialConvolution(64 -> 64, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(64 -> 64, 3x3, 1,1, 1,1)
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(64 -> 64, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(64 -> 128, 3x3, 2,2, 1,1)
nn.SpatialBatchNormalization
nn.SpatialConvolution(64 -> 128, 1x1, 2,2)
nn.ReLU
nn.SpatialConvolution(128 -> 128, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(128 -> 128, 3x3, 1,1, 1,1)
nn.SpatialBatchNormalization
nn.ReLU
nn.SpatialConvolution(128 -> 128, 3x3, 1,1, 1,1)
nn.CAddTable
nn.SpatialBatchNormalization

Fig. 7.17 – The DNN architecture for DDoS attack detection

On the Fig. 7.18 shows matrices with experimental results for the training sample, the test sample, the sample for verification and the total performance (from left to right). Correct values for the matching of the network output concentrate in two categories: True Positive (TP) and False positive (FP).

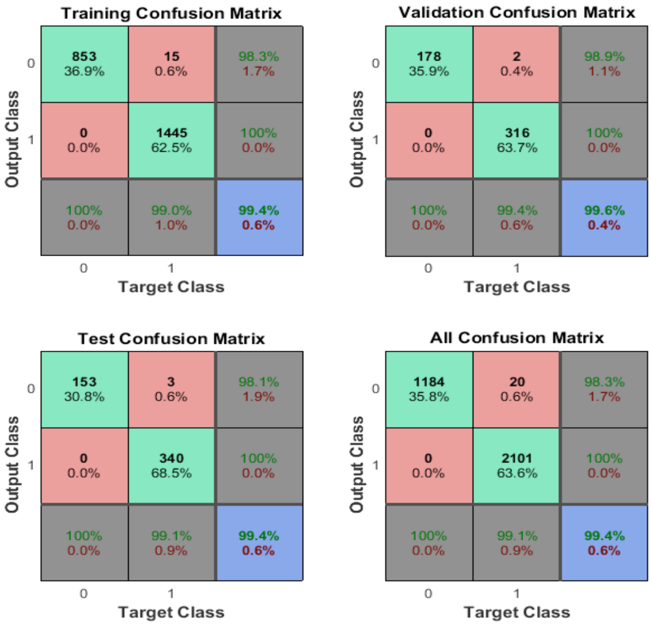


Fig. 7.18 – Experimental results

As can be seen from the presented research results, the model of the deep neural network is characterized by a general classification accuracy in 99.4%. Thus, presented DNN architecture allows successfully detect and classify DDoS attacks.

7.3.2 Deep learning for the real-time embedded systems for IoT

Embedded system is a specialized computer system or computing device designed to perform a limited number of functions. Typically, embedded systems are as a part of a device, including hardware and mechanical components, and are present in many modern devices [16].

Considering the practical applications for the Internet of things, there is a question of the place for embedded systems. In the simplest form, the relationship between the terms: the Internet of things and the embedded system, can be expressed as:

$$IoT = ES + NT + IT.$$

Depending on this, the Internet of Things is a combination of embedded system (ES), network technology (NT) and information technology (IT).

The intelligent data analysis and deep neural networks stay more popular at the embedded systems in the world. It is based on the resources of the device itself, as well as cloud computing. The basic principles of the deep neural networks usage in the embedded systems can be represented by the following sequence of information: sensory data acquisition, deep neural network training, and the applications of the trained neural networks (Fig. 7.19) [17].

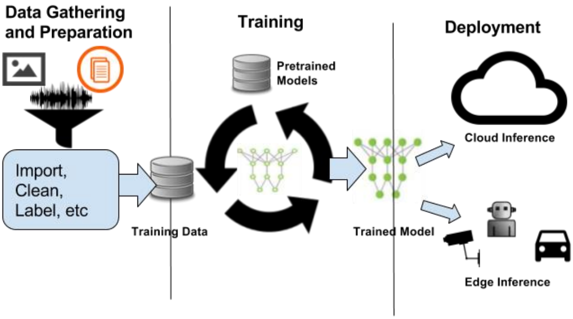


Fig. 7.19 – Basic deep learning workflow for embedded systems

Since the use of deep neural networks requires large computing power, companies are moving towards the hardware implementation, by creating processors.

One of the possible platforms for improving performance is the use of GPU-focused systems. Due to the use of hundreds of cores in graphics processors, it is possible to implement high-floating-point computing. Thus, Nvidia's Compute Unified Device Architecture (CUDA) graphics architecture has been created. The latest Nvidia implementations provide CUDA support for most of the deep neural network operations. In particular, there is support for CUDA-supported frameworks such as Caffe, Torch and Tensorflow.

An alternative to CUDA architectures is parallel computing. In 2009, Khronos offered OpenCL, that is an open standard for parallel computing on a wide range of software, such as GPU, DSP or FPGA. This direction allows the processors other than NVIDIA companies to enter to the application market with deep neural networks.

The market for cloud interfaces in the field of embedded systems is tremendous growth, by supporting of the Internet giants such as Google, Facebook, Baidu or Alibaba. For example, Google Cloud and Microsoft Azure. As a result, favorable conditions for the development of embedded systems are created due to the developed APIs for image classification, natural language processing and face recognition that developers can easily integrate into their regional applications.

Another area of rapid growth at the embedded systems – mobile processor units. Such processors due to the use of deep neural networks provide new features to smartphones. One example is the integration of Apple's neuron engines into the A11 Bionic chip, which allows it to add high-precision locking to the iPhone X. The Chinese microprocessor HiSilicon also released its Kirin 970 processor, which has a Neural Processing Unit. Some of the latest Huawei smartphones have been developed with new DNN processors. The table 7.1 provide a list of some modern processor chips for DNN.

In general the applications of deep neural networks in the embedded systems can be classified into the following groups [18]:

1. Automotive. Deep neural networks provide the sophisticated image processing that advanced driver assistance systems (ADAS) need to recognize signs, pedestrians, and vehicles.

2. Security and Surveillance. Embedded systems that offer face recognition based on neural networks are increasingly employed in camera-based surveillance. Coupled with audio sensors, neural networks can identify sounds, such as breaking glass or dogs barking, and trigger a planned response.

3. Augmented Reality. Real-time augmented reality applications on battery-powered mobile devices rely on deep learning and energy-efficient operation.

4. Smart Home. Sophisticated interpretation and response to voice commands and audio inputs by smart appliances and personal assistants depend on deep learning.

5. Retail Automation Facial age and gender profiling enabling retail kiosks to match offers to customers, and natural language processing allowing them to interact, both require deep learning processing.

6. Healthcare. Deep learning supports audio keyword detection and natural language processing in patient diagnostic systems.

Table 7.1 – The list of modern chips for DNN [17]

Company	Chip	Remarks
Nvidia	Tegra	Jetson TX1,TX2
	Xavier	DL operations. Drive PX Xavier, PX Pegasus
Intel	Movidius	Vision Processing Unit (VPU) targeting computer
	Myriad	vision for drone, robotics, etc.
	MobileEye	MobileEye EyeQ is specifically built for autonomous driving market
Qualcomm	Snapdragon 600/800	Neural Network Engine SDK uses Hexagon DSP + Adrenu GPU for building efficient DL inference for edge devices
Samsung	Exynos 9 Series 9810	Target smartphones: e.g. Galaxy S9
HiSilicon/Huawei	Kirin 970	Target smartphones: e.g. Huawei Mate and Honor
Rockchip	RK 3399Pro	Target security monitoring, drones, etc.
Mediatek	Helio P and X series	Target smartphones: e.g. Oppo and Meizu.

Thus, the development of deep neural networks is an important factor in the development of embedded systems that will allow creation of the new applications within the IoT that previously were impossible.

7.3.3 Pattern recognition for IoT

Pattern recognition is one of the most fundamental problems in the theory of intelligent systems. On the other hand, the problem of image recognition has great practical importance [19].

Let consider the basic principles of deep neural networks design for pattern recognition. For the face recognition problem, the use of convolutional neural networks is quite effective. The relatively stable results for studding of the face recognition problem was obtained starting up to 2010 [4]. By this time, geometric approaches have been used to solve problems of this type. Now, this problem solves by using the procedures that divide the image into the pixels (or segments: 2x2, 3x3, 5x5, 11x11 pixels), and they are the input layer of the neural network. Signals from this input layer are transmitted to the next layers of the neural network using synapses, taking into account weighting factors until the original value is obtained in output layer. In a case of inconsistency of the calculated output with the desired one, an iterative correction of DNN weights is performed.

Around this task, the ImageNet international competition provides by academies for the selection of neural network architectures that better recognize images. The ImageNet database in 2010 contained 15 million images divided into 22 thousand categories. The first neural network that won the ImageNet international competition in 2012 was the convolutional neural networks AlexNet (Fig. 7.10). Despite the fact that this neural network is rather small (just 7 hidden layers), it contains 650 thousand neurons with 60 million parameters.

In general, the architecture of the neural network can be presented by the following elements: inputs, outputs and convolutional layers (Fig. 7.20). There is a plurality of definitions about content of each convolutional layer. One of the accepted definitions represents this layer by the three components: the convolution, the detector and the pooling.

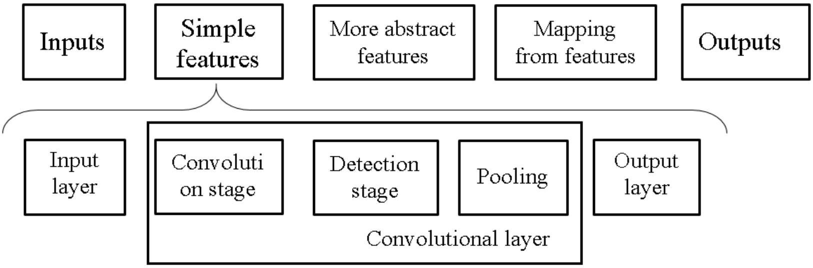


Fig. 7.20 – DNN architecture for pattern recognition

Thus, the input image is processed by the layers (filters with different sizes) of deep neural network. These filters crate a set of attributes that come later into the classifier. Inside the filters, a set of signs will be represented by a variety of "inclined sticks" with shades of color, then by a part of the person, and then the entire person will be recognized by one neuron of the output layer (Fig. 7.21).

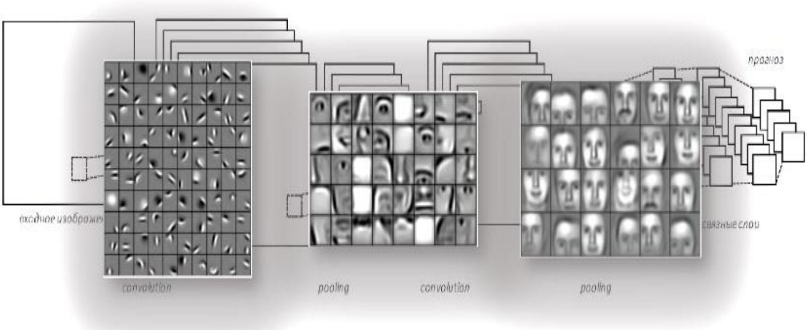


Fig. 7.21 – The general approach for face recognition by DNN

The following classical tasks can be distinguished for convolutional deep neural networks at the IoT (Fig. 7.22) [20]:

– the definition of boundaries is the very low-level task for which the convolutional neural networks are used;

– determining the vector to the normal allows to reconstruct a three-dimensional image from a two-dimensional one;

– saliency, the definition of objects of attention - is what the attention of a person when considering this picture.

– semantic segmentation provides the separation of the image into objects classes without having the information about the objects themselves, that is, before the stage of their recognition;

– allocation of boundaries - is the allocation of segments broken down into classes;

– the allocation of parts of the human body involves the allocation of segments of the parts of the human body;

– object recognition is the most high-level task.

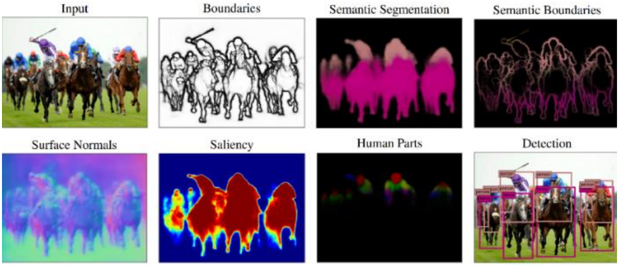


Fig. 7.22 – Classical tasks for convolutional DNN at IoT

7.4 Work related analysis

There are a wide range of applications for deep neural networks today. In particular, information about the latest areas of deep neural network implementations are provided by magazines and Internet publications [4, 21], as well as the Wikipedia [22]. Let consider the most popular and cited researches of deep neural networks.

1. Medicine and health

Artificial intelligence encompasses biological sciences, medicine and health as an industry [22]. High results in deep neural network applications reached by scientists from KTH Royal Institute of Technology, Stockholm, Sweden for analyzing orthopedic trauma radiographs (Fig. 7.23) [23]. It was extracted 256,000 wrist, hand, and ankle radiographs from Danderyd’s Hospital. As a result, all networks exhibited an accuracy of at least 90% when identifying laterality, body part, and exam view. So far it has never been applied in an orthopedic setting, and this study determine the feasibility of using deep learning for diagnosing on skeletal radiographs.

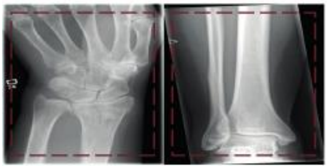


Fig. 7.23 – Images from the dataset.

The area within the red box is the section presented to the network in order to classify the image. The left image is of a wrist fracture while the right image is without any apparent fracture [23].

2. Image manipulation

Ideas based on application of DNN for image processing. Image segmentation has been explored for many years and still remains a crucial vision problem. Some efficient or accurate segmentation algorithms have been widely used in many vision applications. However, it is difficult to design a both efficient and accurate image segmenter. The scientists from the KTH Royal Institute of Technology, Stockholm, Sweden propose a novel method called "Deep embedding learning", which can efficiently transform pixels into image segmentation [24]. With the deep similarities, it is possible to merge the pixels into large segments (Fig. 7.24). The evaluation results demonstrate achievement of a good tradeoff between efficiency and effectiveness.

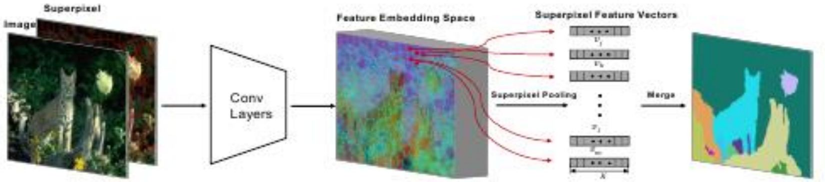


Fig. 7.24 – The pipeline of image segmentation algorithm [24]

3. Robotics

A deep neural network is also widely used in robotics. This area is represented by researches from KTH Royal Institute of Technology, Stockholm, Sweden that presents techniques that use DNN for human-robot collaboration [25]. At the intelligent manufacturing the visual observation of human workers' motion provides informative clues about the specific tasks to be performed, thus can be explored for establishing accurate and reliable context awareness. Towards this goal, the deep learning investigates as a data driven technique for continuous human motion analysis, leading to improved robot planning and control in accomplishing a shared task. Authors use AlexNet deep neural network for human motion recognition during assembling of car engine (Fig. 7.25 demonstrate training process).

An experimental case study on car engine assembly demonstrated recognition accuracy of over 96%.

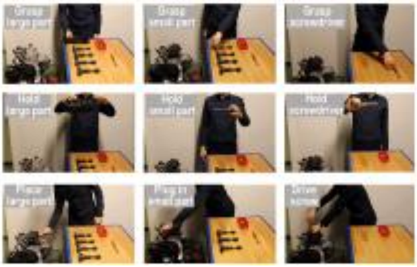


Fig. 7.25 – Examples of human action images for training the AlexNet [25]

4. Handwritten Signature Recognition

Reliable identification and verification of off-line handwritten signatures from images is a difficult problem with many practical applications. This task is a difficult vision problem within the field of biometrics because a signature may change depending on psychological factors of the individual. Motivated by advances in brain science which describe how objects are represented in the visual cortex, advanced research on deep neural networks has been shown by scientist from the University of Coimbra, Portugal to work reliably on large image data sets [26]. It was proposed deep learning model for offline handwritten signature recognition, which is able to extract high-level representations (Fig. 7.26). Also, it was propose a two-step hybrid model for signature identification and verification improving the misclassification rate in the well-known GPDS database.

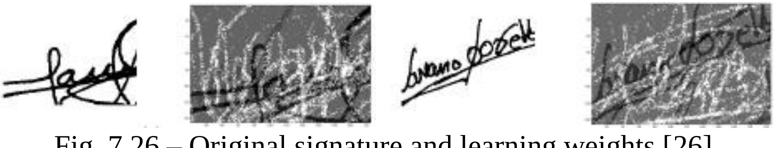


Fig. 7.26 – Original signature and learning weights [26]

5. Analytics

Manufacturing analytics is of paramount importance in many plants today, and its relevance increases in the current big data context of Industry. The fields of statistics, chemometrics, and machine learning are expected to provide tools that effectively handle many of the characteristics of industrial data. In this paper, the task of image-based product classification is considered. This is a supervised learning problem where the input is an image and the output is a unique label attributed to the image from a finite set of labels corresponding to the available product classes. This is a prevalent and highly relevant industrial challenge and recent developments in deep learning have proven to be successful in increasing the image classification accuracy, providing state-of-the-art results. Thus, researchers from the University of Coimbra, Portugal leverage deep neural networks' ability to automatically learn features from images and test their performance in a real industrial context for predicting the pellet shape (Fig. 7.27) [27].

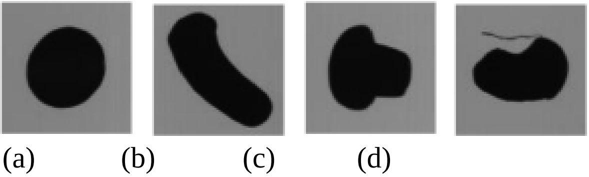
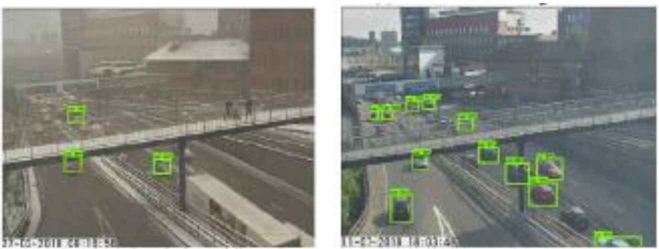


Fig. 7.27 – Pellet examples: (a) good pellet with a round shape, (b) and (c) represent pellets with significant deviations from a good pellet; (d) is an example of a pellet containing a tail [27]

6. Traffic flow analysis

Traffic flow analysis is fundamental for urban planning and management of road traffic infrastructure. Automatic number plate recognition (ANPR) systems are conventional methods for vehicle detection and travel times estimation. However, such systems are specifically focused on car plates, providing a limited extent of road users. The advance of open-source deep learning convolutional neural networks (CNN) in combination with freely-available closed-circuit television (CCTV) datasets have offered the opportunities for detection and classification of various road users. The researchers from the Newcastle University, UK, presented CNN models to analyze traffic flow in various weather conditions and seasons of the year [28]. Such imagery is collected from the North East Combined Authority (NECA) Travel and Transport Data, Newcastle upon Tyne, UK. Results show that the fine-tuned MobileNet model with 98.2% precision, 58.5%

recall and 73.4% harmonic mean could potentially be used for a real time traffic monitoring application with big data, due to its fast performance. Compared to MobileNet, the fine-tuned Faster region proposal R-CNN model, providing a better harmonic mean (80.4%), recall (68.8%) and more accurate estimations of car units, could be used for traffic analysis applications that demand higher accuracy than speed (Fig. 7.28). This research ultimately exploits machine learning algorithms for a wider understanding of traffic congestion and disruption under social events and extreme weather conditions.



(a) Car detection during a blizzard (b) Car detection during rush hour on a summer day

Fig. 7.28 – Car detection example with fine-tuned SSD MobileNet from a NECA CCTV location (NECA, 2018a) [28]

7. Computer vision

Computer vision-based assistive technology solutions can revolutionize the quality of care for people with sensorimotor disorders. The goal is to enable trans-radial amputees to use a simple, yet efficient, computer vision system to grasp and move common household objects with a two-channel myoelectric prosthetic hand. The scientists from the Newcastle University, UK developed a deep learning-based artificial vision system to augment the grasp functionality of a commercial prosthesis [29]. The conceptual novelty consists in classifying objects with regards to the grasp pattern without explicitly identifying them or measuring their dimensions. A convolutional neural network (CNN) structure was trained with images of over 500 graspable objects. For each object, 72 images, at 5^{o} intervals, were available. Objects were categorized into four grasp classes, namely: pinch, tripod, palmar wrist neutral and palmar wrist pronated. The CNN setting was first tuned and tested offline and then in realtime with objects or object views that were not included in the training set. The general structure if control system is shown on Fig. 7.29.

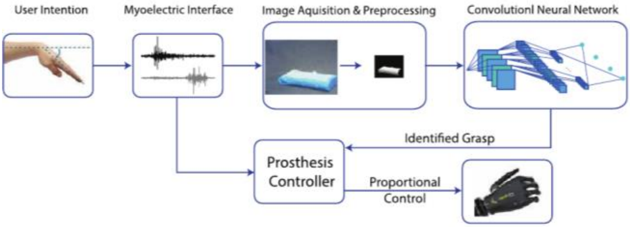


Fig. 7.29 – Overall control structure [29]

The classification accuracy in the offline tests reached 85% for the seen and 75% for the novel objects; reflecting the generalizability of grasp classification. The proposed design constitutes a substantial conceptual improvement for the control of multi-functional prosthetic hands. Authors show for the first time that deep-learning based computer vision systems can enhance the grip functionality of myoelectric hands considerably.

8. Malware detection

The increasing amount of malware variants seen in the wild is causing problems for Antivirus Software vendors, unable to keep up by creating signatures for each. The methods used to develop a signature, static and dynamic analysis, have various limitations. Machine learning has been used by Antivirus vendors to detect malware based on the information gathered from the analysis process. However, adversarial examples can cause machine learning algorithms to miss-classify new data. The authors from the Leeds Beckett University, Leeds, UK use a method for malware analysis by converting malware binaries to images (Fig. 7.30) and then preparing those images for training within a Generative Adversarial Network [30]. These unsupervised deep neural networks are not susceptible to adversarial examples. The conversion to images from malware binaries should be faster than using dynamic analysis and it would still be possible to link malware families together. Using the Generative Adversarial Network, malware detection could be much more effective and reliable.

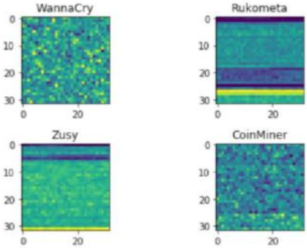


Fig. 7.30 – Image arrays of four malware images [30]

9. Speech recognition

The performance of deep neural network based monaural speech separation methods is limited in reverberant and noisy room environments. In [31] the authors from the Newcastle University, UK propose a new DNN training target, which incorporates geometric information describing the target speaker and microphone to improve the performance in reverberant and noisy room environments. As a results it were exploited the geometric information to provide the position of the target speaker and microphone to estimate the direct path impulse response, which is used to calculate the direct path speech (Fig. 7.31). Based on the direct path speech, we calculated the direct path ratio mask (DRM) that is a new training target.

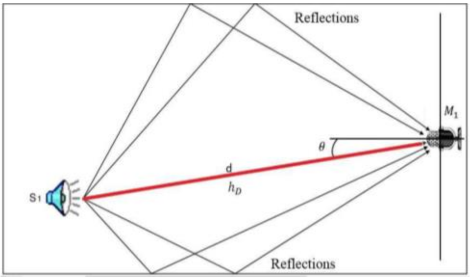


Fig. 7.31 – Monaural speech separation setup within a reverberant room environment, the distance and angle between the target speaker and sensor are shown [31]

10. Smartphone Authentication

Continuous authentication is receiving increased attention from providers of on-line services, particularly due to the ability of mobile apps to collect user-specific sensor data. However, the approaches proposed so far are either not accurate enough to provide a high-quality user experience or restricted by engineering challenges to capture data continuously. Scientists from the Newcastle University, UK propose an approach based on a deep learning autoencoder, which achieves an equal error rate as low as 2.2% in tested real-world scenarios. The suggested system only relies on accelerometer data and does not require a high number of features, therefore reducing the computational burden. As a result, it was proposed a cloud-based continuous authentication biometric system for the smartphone, which can detect fraudulent access by exploiting the user's specific motion patterns [32].

Conclusions and questions

In this section, the materials for module MC2.3 "Deep learning for IoT" of MSc course "Data science for IoT and IoE" are presented and can be used for preparation to lectures and self-learning. The section outlines the development of machine learning and artificial intelligence, as well as the principles of the deep neural networks operations and analysis of their applications for the Internet of Things. This area developing rapidly and is one of possible way to look into the future of information technologies.

The practical important of the presented above materials of this section consist in creating possibility to use of well-known frameworks like Matlab, Cafe, TensorFlow or other for quick researching of deep neural network technologies in IoT applications.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What is machine learning and its place in IoT?
2. What does the artificial intelligence term mean?
3. What are the types of deep neural networks?
4. What are the criteria for neural network classification?

5. What was the problem with perceptron?
6. Why the gradient-based training of deep neural networks from random initialization is often unsuccessful?
7. What approaches are used for training of artificial neural network?
8. What the specifics of deep neural networks comparing with non-deep architectures?
9. What the know architectures of deep neural network you know?
10. How provides training of the deep neural network?
11. What does the embedded system mean?
12. How does the embedded system work?
13. What is pattern recognition?
14. What the content of the convolutional layer of deep neural network?
15. How to use deep neural network applied for DDoS attack recognition?
16. What the deep neural network frameworks you know?
17. What are the applications of deep neural network for IoT?

References

1. The Internet of Things. (2019). Internet of Things and the Prelude to Artificial Intelligence. [online] Available at: http://www.infiniteinformationtechnology.com/the-internet-of-things-prelude-to-artificial-intelligence [Accessed 28 Jul. 2019]

2. En.wikipedia.org. (2019). Machine learning. [online] Available at: https://en.wikipedia.org/wiki/Machine_learning [Accessed 28 Jul. 2019].

3. Alpaydin, E. Introduction to machine learning / Ethem Alpaydin - 3rd ed. Cumberland: MIT Press, The. 2014. 613 pp.

4. Goodfellow, I., Bengio, Y. and Courville, A. (n.d.). Deep Learning / MIT Press book. 787 pp.

5. Golovko V.A. Neural network data processing technologies / V. A. Golovko, V. V. Krasnoproshin. Minsk: BGU, 2017. 263 pp.

6. Golovko V.A. Neural Networks: training, models and applications. / V Golovko, Moscow: IPRZHR 256. 2001. 256pp.

7. Golovko V. Neurointelligence: theory and application. Book1 / V.Golovko. BPI, Brest. 1999. 264 pp.

8. Golovko V. Neurointelligence: theory and application. Book2 / V.Golovko. BPI, Brest. 2001. 228 pp.

9. A. Krizhevsky, I. Sutskever, GE Hinton. ImageNet Classification with Deep Convolutional Neural Networks. // Proceedings of Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 1097-1105.

10. Elter M., Horsch A., CADx of mammographic masses and clustered microcalcifications: a review. 2009. Med. Phys., 36, pp. 2052-2068.

11. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going Deeper with Convolutions. Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR 2015). June 7-12, 2015, pp. 1-12.

12. K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. Proceedings of 3rd International Conference on Learning Representations (ICLR2015), Hilton San Diego Resort & Spa, May 7-9, 2015, pp. 1-14.

13. Uk.wikipedia.org. (2019). *Security of Internet of Things*. [online] Available at: https://uk.wikipedia.org/wiki/Безпека_інтернету_речей [Accessed 28 Jul. 2019].

14. Kdd.ics.uci.edu. (2019). KDD Cup 1999 Data. [online] Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed 28 Jul. 2019].

15. Caffe.berkeleyvision.org. (2019). Caffe | Deep Learning Framework. [online] Available at: http://caffe.berkeleyvision.org [Accessed 28 Jul. 2019].

16. En.wikipedia.org. (2019). Embedded system. [online] Available at: https://en.wikipedia.org/wiki/Embedded_system [Accessed 28 Jul. 2019].

17. Medium. (2019). Using Deep Learning Processors For Intelligent IoT Devices. [online] Available at:

https://medium.com/iotforall/using-deep-learning-processors-for-intelligent-iot-devices-1a7ed9d2226d [Accessed 28 Jul. 2019].

18. CEVA. (2019). Deep Learning for the Real-Time Embedded World | CEVA. [online] Available at: https://www.ceva-dsp.com/app/deep-learning/ [Accessed 28 Jul. 2019].

19. Alibaba Cloud Community. (2019). Deep Learning vs. Machine Learning vs. Pattern Recognition. [online] Available at: https://www.alibabacloud.com/blog/deep-learning-vs-machine-learning-vs-pattern-recognition_207110 [Accessed 28 Jul. 2019].

20. Habr.com. (2019). Neural networks: practical applications [online] Available at: https://habr.com/post/322392/ [Accessed 28 Jul. 2019].

21. Medium. (2019). Top 15 Deep Learning applications that will rule the world in 2018 and beyond. [online] Available at: https://medium.com/@vratulmittal/top-15-deep-learning-applications-that-will-rule-the-world-in-2018-and-beyond-7c6130c43b01 [Accessed 28 Jul. 2019].

22. En.wikipedia.org. (2019). Deep learning. [online] Available at: https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks [Accessed 28 Jul. 2019].

23. Jakub Olczak, Niklas Fahlberg, Atsuto Maki, Ali Sharif Razavian, Anthony Jilert, André Stark, Olof Sköldenberg and Max Gordon. Artificial Intelligence for Analyzing Orthopedic Trauma Radiographs // Acta Orthopaedica journal, Volume 88, Issue 6, 2 November 2017, Pages 581-586.

24. Liu, Y.,Jiang, P.-T.,Petrosyan, V.,Li, S.-J.,Bian,J.,Zhang, L.,Cheng, M.-M. DEL: Deep Embedding Learning for Efficient Image Segmentation // 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm; Sweden; 13-19 July 2018, Pages 864-870.

25. Peng Wang, Hongyi Liu, Lihui Wang, Robert X. Gao. Deep Learning Based Human Motion Recognition for Predictive Context-Aware Human Robot Collaboration // CIRP Annals - Manufacturing Technology, Elsevier USA, Volume 67, Issue 1, 1 January 2018, Pages 17-20.

26. Bernardete Ribeiro, Ivo Gonçalves, Sérgio Santos and Alexander Kovacec, Deep Learning Networks for Off-Line

Handwritten Signature Recognition // Proceedings of CIARP 2011: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Chile, November 15-18, 2011, pp 523-532.

27. Rendall, R., Castillo, I.,Lu, B., Colegrove, B., Broadway, M., Chiang, L.H., Reis, M.S. Image-Based Manufacturing Analytics: Improving the Accuracy of an Industrial Pellet Classification System Using Deep Neural Networks // Elsevier Journal: Chemometrics and Intelligent Laboratory Systems, Volume 180, 15 September 2018, pp. 26-35.

28. Peppa, M. V., Bell, D., Komar, T., and Xiao, W. Urban Traffic Flow Analysis Based On Deep Learning Car Detection From CCTV Image Series // Conference Proceedings of SPRS TC IV Mid-term Symposium “3D Spatial Information Science – The Engine of Change”, vol. XLII-4, 2018, pp. 499-506.

29. Ghazaei G, Alameer A, Degenaar P, Morgan G, Nazarpour K. Deep Learning-Based Artificial Vision for Grasp Classification in Myoelectric Hands, Journal of Neural Engineering, Volume 14, Issue 3, 2017, 18 pp.

30. Joakim Kargaard,Tom Drange, Ah-Lian Kor, Hissam Twafik, Emlyn Butterfield, Defending IT Systems Against Intelligent Malware // The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT’2018, 24-27 May, 2018, Kyiv, Ukraine, pp. 411-417.

31. Xian Y., Sun Y., Chambers J.A., Naqvi, S.M. Geometric Information Based Monaural Speech Separation Using Deep Neural Network // Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary Telus Convention Center, Canada 15-20 April, 2018, pp. 4454-4458.

32. Centeno M.P., Van Moorsel A., Castruccio S., Smartphone Continuous Authentication Using Deep Learning Autoencoders // Proceedings of 15th Annual Conference on Privacy, Security and Trust, PST 2017, Calgary, Canada, 27-29 August 2017, pp. 147-155.

8. BIG DATA FOR IoT BASED SYSTEMS

Dr. Ass.Prof. O. Tarasyuk, DrSc. Prof. A. Gorbenko (KhAI)

Contents

Abbreviations

ACID – Atomicity, Consistency, Isolation and Durability

BASE – Basically Available, Soft state, Eventually consistent

CAP – Consistency, Availability, Partition tolerance

CQL – Cassandra Query Language

IoT – Internet of Things

NoSQL – Not (Not Only) SQL

PDF – Probability Distribution Function

RDBMS – Relational Data Base Management System

SQL – Structured Query Language

8.1 Big data and NoSQL databases

8.1.1 A concept of Big data

The term *big data* started to show up sparingly in the early 1990s, and its prevalence and importance increased exponentially as years passed. Nowadays big data is often seen as integral to a company's data strategy. Big data is an application domain that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional (e.g. relational) data base management systems.

Big data has specific characteristics and properties that can help to understand both the challenges and advantages of big data initiatives. Big data was originally associated with three key concepts: *volume*, *variety*, and *velocity* [1]. Other concepts later attributed to big data are *veracity* (i.e. how much noise is in the data) [2] and *value* [3]. In [4] authors present their 10V's big data model and discuss five additional important characteristics describing big data:

1. Volume. Volume is probably the best known characteristic of big data; this is no surprise, considering more than 90 percent of all today's data was created in the past couple of years. The current amount of data can actually be quite staggering. Here are some examples: (i) 300 hours of video are uploaded to YouTube every minute; (ii) an estimated 1.1 trillion photos were taken in 2016, and that number is projected to rise by 9 percent in 2017. As the same photo usually has multiple instances stored across different devices, photo or document sharing services as well as social media services, the total number of photos stored is also expected to grow from 3.9 trillion in 2016 to 4.7 trillion in 2017; (iii) in 2016 estimated global mobile traffic amounted for 6.2 exabytes per month. That's 6.2 billion gigabytes.

2. Velocity. Velocity refers to the speed at which data is being generated, produced, created, or refreshed. Sure, it sounds impressive that Facebook's data warehouse stores upwards of 300 petabytes of data, but the velocity at which new data is created should be taken into account. Facebook claims 600 terabytes of incoming data per day. Google alone processes on average more than 40000 search queries every second, which roughly translates to more than 3.5 billion searches per day.

3. Variety. When it comes to big data, we don't only have to handle structured data but also semistructured and mostly unstructured data as well. As you can deduce from the above examples, most big data seems to be unstructured, but besides audio, image, video files, social media updates, and other text formats there are also log files, click data, machine and sensor data, etc.

4. Variability. Variability in big data's context refers to a few different things. One is the number of inconsistencies in the data. These need to be found by anomaly and outlier detection methods in order for any meaningful analytics to occur. Big data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources. Variability can also refer to the inconsistent speed at which big data is loaded into your database.

5. Veracity. This is one of the unfortunate characteristics of big data. As any or all of the above properties increase, the veracity (confidence or trust in the data) drops. This is similar to, but not the same as, validity or volatility (see below). Veracity refers more to the provenance or reliability of the data source, its context, and how meaningful it is to the analysis based on it. Knowledge of the data’s veracity in turn helps us better understand the risks associated with analysis and business decisions based on this particular data set.

6. Validity. Similar to veracity, validity refers to how accurate and correct the data is for its intended use. According to Forbes, an estimated 60 percent of a data scientist's time is spent cleansing their data before being able to do any analysis. The benefit from big data analytics is only as good as its underlying data, so you need to adopt good data governance practices to ensure consistent data quality, common definitions, and metadata.

7. Vulnerability. Big data brings new security concerns. After all, a data breach with big data is a big breach. Collecting and storing big data often leads to data leakages as computer systems are prone to have vulnerabilities which can be exploited by hackers. One of the example is AshleyMadison hack in 2015 [5] caused leakage of more than 25 gigabytes of company data, including user details. Another example, as reported by CRN in May 2016 a hacker called Peace posted data on the dark web to sell, which allegedly included information on 167 million LinkedIn accounts and 360 million emails and passwords for MySpace users.

8. Volatility. Volatility refers to deciding how old does data need to be before it is considered irrelevant, historic, or not useful any longer and how long does data need to be kept for.

Before big data, organizations tended to store data indefinitely – a few terabytes of data might not create high storage expenses; it could even be kept in the live database without causing performance issues. In a classical data setting, there not might even be data archival policies in place. However, due to the velocity and volume of big data, however, its volatility needs to be carefully considered including rapid retrieval of information when required. With big data the costs and complexity of a storage and retrieval process are magnified.

9. Visualization. Another characteristic of big data is how challenging it is to visualize. Current big data visualization tools face technical challenges due to limitations of in-memory technology and poor scalability, functionality, and response time. You can’t rely on traditional graphs when trying to plot a billion data points, so you need different ways of representing data such as data clustering or using tree maps, sunbursts, parallel coordinates, circular network diagrams, or cone trees.

10. Value. Last, but arguably the most important of all, is value. The other characteristics of big data are meaningless if you don’t derive business value from the data. Substantial value can be found in big data, including understanding your customers better, targeting them accordingly, optimizing processes, and improving machine or business performance. You need to understand the potential, along with the more challenging characteristics, before embarking on a big data strategy.

8.1.2 NoSQL databases

NoSQL (Non SQL or Not Only SQL) databases have become the standard data plat-form and a major industrial technology for dealing with enormous data growth. They are now widely used in different market niches, including social networks and other large-scale Internet applications, critical infrastructures, business-critical systems, IoT and industrial applications. NoSQL databases designed to provide horizontal scalability are often offered as a service by Cloud providers.

A concept of NoSQL databases [6] has been proposed to effectively store and provide fast access to the Big Data sets whose volume, velocity and variability are difficult to deal with by using the traditional Relational Database Management Systems. Most NoSQL stores sacrifice the ACID (atomicity, consistency, isolation and durability) guarantees in favour of the BASE (basically available, soft state, eventually consistent) properties [7], which is the price to pay for distributed data handling and horizontal scalability.

NoSQL databases use different data structures (e.g. key-value, wide-column, document, graph, etc.) compared to tabular relational databases. These databases are schema-free, support easy replication, have simple API, eventually consistent, and can handle huge amounts of data. It makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it must solve. The primary objective of a NoSQL database is to have: (i) simplicity of design; (ii) horizontal scaling, and (iii) finer control over availability.

Trade-offs between consistency, availability and latency, which is in the very nature of NoSQL databases. Although these relations have been identified by the CAP theorem in qualitative terms [8, 9], it is still necessary to quantify how different consistency settings affect system latency. Understanding this trade-off is key for the effective usage of NoSQL solutions.

While there are many NoSQL databases on the market, various industry trends suggest that Apache Cassandra is one of the top three in use today together with MongoDB and HBase [10].

Apache Cassandra is an open source, distributed and decentralized/distributed storage system for managing very large amounts of structured data spread out across multiple datacentres. It provides highly available service with no single point of failure. Cassandra is a *wide-column-oriented database* (a *wide-column* store uses tables, rows, and columns, but unlike a relational database, the names and format of the columns can vary from row to row in the same table. A wide column store can be interpreted as a two-dimensional key-value store.). It was created at Facebook and implements a Dynamo-style replication model with no single point of failure, but adds a more powerful “column family” data model. Cassandra is being widely used by Facebook, Twitter, Cisco, Rackspace, eBay, Twitter, Netflix, and other Internet companies.

Apache HBase is an open source, non-relational, distributed database modeled after Google’s BigTable and is written in Java. It is developed as a part of Apache Hadoop project and runs on top of HDFS, providing BigTable-like capabilities for Hadoop.

MongoDB is a cross-platform *document-oriented database* system that avoids using the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas making the integration of data in certain types of applications easier and faster.

There have been a number of studies, e.g. [11, 12, 13, 14, 15], evaluating and comparing the performance of different NoSQL databases. Most of them use general competitive benchmarks of usual-and-customary application workloads (e.g. Yahoo! Cloud Serving Benchmark, YCSB). Reported results show that depending on the use case scenario, deployment conditions, current workload and database settings any NoSQL database can outperform the others. Other recent related works, such as [16, 17, 18], have investigated measurement-based performance prediction of NoSQL data stores. However, the studies, mentioned above, do not investigate an interdependency between consistency and performance that is in the very nature of such distributed database systems and do not study how consistency settings affect database latency.

In this work we put a special focus on quantitative evaluation of one of the fundamental Big Data trade-offs between data consistency and performance using the Cassandra database as a typical example of distributed data storages. Apache Cassandra offers a set of unique features (e.g. tuneable consistency, extremely fast writes, ability to work across geographically distributed data centres, etc.) and provides high availability with no single point of failure which makes it one of the most flexible and popular NoSQL solutions. Moreover, we would like to equip the developers of distributed systems that use Cassandra as the distributed data storage with the practical guidance allowing them to predict the Cassandra latency taking into account the required consistency level and to coordinate consistency settings of read and write requests in an optimal manner.

8.1.3 Big data trade-offs between consistency, availability and latency

The CAP conjecture [8], which first appeared in 1998-1999, defines a trade-off between system availability, consistency and partition tolerance, stating that only two of the three properties can be preserved in distributed replicated systems at the same time. Gilbert and Lynch [9] view the CAP theorem as a particular case of a more general trade-off between consistency and availability in unreliable distributed systems which assume that updates are eventually propagated. System partitioning, availability and latency are tightly connected. A replicated fault-tolerant system becomes partitioned when one of its parts does not respond due to arbitrary message loss, delay or replica failure, resulting in a timeout. System availability can be interpreted as a probability that each client request eventually receives a response.

Failure to receive responses from some of the replicas within the specified timeout causes partitioning of the replicated system. Thus, partitioning can be considered as a bound on the replica’s response time [19]. A slow network connection, a slow-responding replica or the wrong timeout settings can lead to an erroneous decision that the system has become partitioned. When the system detects a partition, it has to decide whether to return a possibly inconsistent response to a client or to send an exception message in reply, which undermines system availability.

The designers of the distributed fault-tolerant systems cannot prevent partitions which happen due to network failures, message losses, hacker attacks and components crashes and, hence, have to choose between availability and consistency. One of these two properties has to be sacrificed. If system developers decide to forfeit consistency they can also improve the system response time by returning the fastest response to the client without waiting for other replica responses until the timeout, though this would increase the probability of providing inconsistent results. For example, modern distributed database systems, e.g. Apache Cassandra [20], can provide a discrete set of different consistency levels for each particular read or write request. The response time can theoretically vary between zero and infinity, although in practice it ranges between a minimal affordable time higher than zero and the application timeout. Availability varies between 0% and 100% as usual.

The architects of modern distributed database management systems and large-scale web applications such as Facebook, Twitter, etc. often decide to relax consistency requirements by introducing asynchronous data updates in order to achieve higher system availability and allow a quick response. Yet the most promising approach is to balance these properties. For instance, the Cassandra NoSQL database introduces a tuneable replication factor and an adjustable consistency model so that a customer can choose a particular level of consistency to fit with the desired system latency.

The CAP theorem helps the developers to understand the system trade-offs between consistency and availability/latency [21, 22]. Yet even though this theorem strongly suggests that better consistency undermines system availability and latency, developers do not have quantitative models helping them to estimate the system response time for the chosen consistency level and to achieve a precise trade-off between them. Our interpretation of the CAP theorem and the trade-offs resulting from the CAP is depicted in Fig. 8.1.

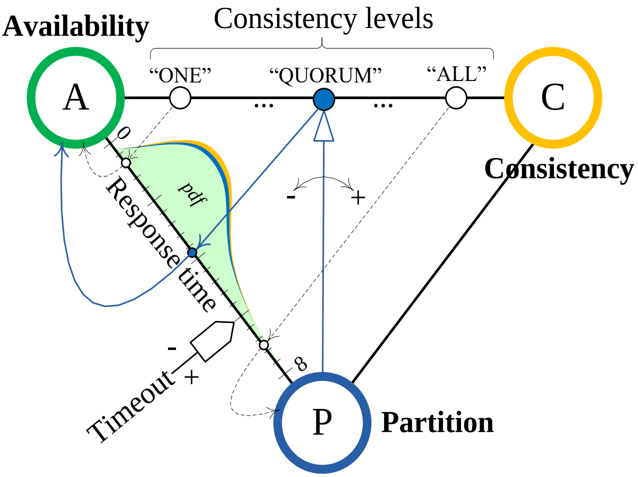


Fig. 8.1 – The CAP trade-offs model

The application timeout can be considered as a bound between system availability and performance (in term of latency or response time) [23]. Thus, system designers should be able to set up timeouts according to the desired system response time, also keeping in mind the choice between consistency and availability. The response time itself is a random variable which can be described by a probability distribution function (*pdf*). This function, in turn, can be determined experimentally during operation or by benchmarking [24].

Besides, benchmarking is essential for analysing the quantitative effect that consistency settings exert on performance of distributed systems and replicated data storages.

8.2 Big data modelling using Cassandra NoSQL data storage

8.2.1 The Cassandra NoSQL database

Cassandra offers robust support for clusters spanning multiple datacenters with asynchronous masterless replication allowing low latency operations (especially for writes/updates).

The key Cassandra features are:

– ***elastic scalability:*** Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement;

– ***always on architecture:*** Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure;

– ***fast linear-scale performance:*** Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster; therefore it maintains a quick response time;

– ***flexible data storage:*** Cassandra accommodates all possible data formats including structured, semi-structured, and unstructured. It can dynamically accommodate changes to data structures according to user needs;

– ***easy data distribution:*** Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers;

– ***fast writes:*** Cassandra was designed to run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency;

– ***tunable and eventual consistency:*** Cassandra is typically classified as an AP system, meaning that availability and partition tolerance are generally considered to be more important than consistency; though, managing multiple replicas Cassandra offers a tunable level of consistency for writes and reads, all the way from “writes never fail” to “block for all replicas to be readable”, with the quorum level in the middle; Cassandra also manages eventual consistency of reads, upserts and deletes

8.2.2 Cassandra consistency model

The Cassandra NoSQL database extends the concepts of *strong* [25] and *eventual* [26] consistency by offering *tuneable* [27] consistency. Consistency in Cassandra can be configured to trade-off availability and latency versus data accuracy.

Consistency level among replicated nodes can be controlled on a per-operation basis. Thus, for any given read or write operation, a client can specify how consistent the requested data must be. The *read consistency level* specifies how many replica nodes must respond to a read request before returning data to the client application. In turn, the *write consistency level* determines the number of replicas on which the write must succeed before returning an acknowledgment to the client.

It is worth noting that Cassandra supports two types of write operations with the tiny difference between them: insert and update. Cassandra treats both insert or update operations as upserts (update-or-insert) [28]. It adds each new row to the database without really checking on whether a duplicate record exists. This makes it possible that many versions of the same row may exist in the database. Periodically, the rows stored in memory (in a structure called memtable) are streamed to disk into structures called SSTables. At certain intervals, Cassandra compacts smaller SSTables into larger SSTables. If Cassandra encounters two or more versions of the same row during this process, it only writes the most recent version to the new SSTable and drops the original SSTables, deleting the outdated rows.

All Cassandra read and write requests support the following basic consistency settings [29]:

– ONE: data must be written to the commit log and memtable of at least one replica node before acknowledging the write operations to a client; when reading data, Cassandra queries and returns a response from a single replica (the nearest replica with the least network latency);

– TWO: data must be written to at least two replica nodes before being acknowledged; read operations will return the most recent record from two of the closest replicas (the most recent data is determined by comparing timestamps of records returned by those two replica);

– THREE: similar to TWO but for three replicas;

– QUORUM: a quorum of nodes needs to acknowledge the write or to return a response for a read request; a quorum is calculated by rounding down to a whole number the following estimate: $replication\_factors/2+1$;

– ALL: data must be written to all replica nodes in a cluster before being acknowledged; read requests return the most recent record after all replicas have responded. The read operation will fail even if a single replica does not respond.

If Cassandra runs across multiple data centres, a few additional consistency levels become available: EACH_QUORUM, LOCAL_QUORUM, LOCAL_ONE.

The sum of nodes written and read being greater than the replication factor always ensures strong data consistency [29]. Thus, if data consistency is of a top priority, one can ensure that a read always reflects the most recent updates by using the following:

$$(nodes\_written + nodes\_read) > replication\_factor \qquad (8.1)$$

Otherwise, the eventual consistency occurs. For example, if Cassandra uses a replication factor of 3, the strong consistency is ensured if, either:

– the QUORUM consistency level is set for both write and read requests;

– the ONE consistency level is set for writes and ALL for reads;

– the ALL consistency level is set for writes and ONE for reads.

The weaker consistency level, the faster Cassandra should perform read and write requests. Balancing between *nodes_written* and *nodes_read* in (8.1), Cassandra users can give the priority to read or write performance still guaranteeing the strong data consistence.

8.2.3 Cassandra data model

Cassandra uses the following concepts to store data.

1. Cluster. Cluster is a collection of nodes or data centers arranged in a ring architecture. Cassandra database is distributed over several machines that operate together. The outermost container is known as the cluster. For failure handling, every node contains a replica, and in case of a failure, the replica takes charge. Cassandra arranges the nodes in a cluster, in a ring format, and assigns data to them.

2. Keyspace. Keyspace is the outermost container (i.e. a database) for data in Cassandra. The basic attributes of a Keyspace in Cassandra are:

– *replication factor* − it is the number of machines in the cluster that will receive copies of the same data;

– *replica placement strategy* − it is the strategy to place replicas in the ring (e.g. simple, rack-aware or data center-shared strategies);

– *column families;* keyspace is a container for a list of one or more column families; a column family, in turn, is a container of a collection of rows; each row contains ordered columns; column families represent the structure of your data; each keyspace has at least one and often many column families.

3. Column family. A column family is a container for an ordered collection of rows (row is a unit of replication in Cassandra). Each row, in turn, is an ordered collection of columns. Column Families in Cassandra are like tables in Relational Databases. Each Column Family contains a collection of rows which are represented by a `Map<RowKey, SortedMap<ColumnKey, ColumnValue>>`. The key gives the ability to access related data together; unlike relational tables column family's schema is not fixed and Cassandra does not force individual rows to have all the columns. User can freely add any column to any column family at any time (see Fig. 8.2).

A Cassandra column family has the following attributes:

– *keys_cached* − it represents the number of locations to keep cached per SSTable;

– *rows_cached* − it represents the number of rows whose entire contents will be cached in memory;

– *preload_row_cache* − it specifies whether you want to pre-populate the row cache.

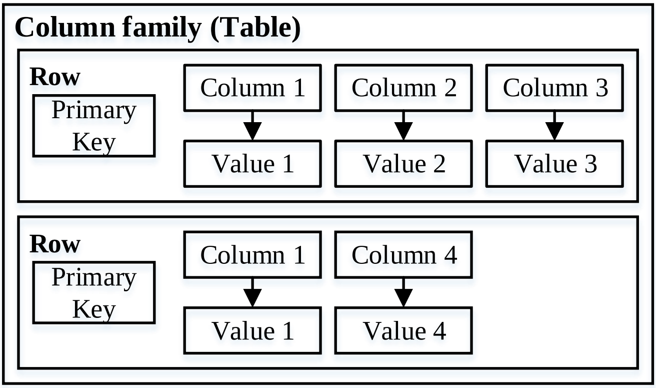


Fig. 8.2 – Cassandra column family (table)

4. Column. A column is the basic data structure of Cassandra with three values, namely key or column name, value, and a time stamp; it is a unit of storage in Cassandra. The columns and the number of columns in each row may vary in contrast with a relational database where data are well structured.

5. Super column. A super column is a special column, therefore, it is also a key-value pair. But a super column stores a map of sub-columns Generally column families are stored on disk in individual files. Therefore, to optimize performance, it is important to keep columns that you are likely to query together in the same column family, and a super column can be helpful here.

6. Collections. Unlike the concepts of foreign keys and joins used by relational databases, relationships in Cassandra are represented using collections. For example, in a relational database a grouping such as a user’s multiple email addresses is related with a many-to-one joined relationship between a user table and an email table. Cassandra avoids joins between two tables by storing the user’s email addresses in a collection column in the user table. Each collection specifies the data type of the data held and can be one of the following three: set, list, map. However, a collection is only appropriate if the data for collection storage is limited. If the data has unbounded growth potential, like messages sent or sensor events registered every second, using collections is not suitable. Instead, a table with a compound primary key where data is stored in the clustering columns should be used.

Cassandra introduced the Cassandra Query Language (CQL) as a simple interface for accessing Cassandra, which is alternative to the traditional Structured Query Language (SQL) used by RDBMS. CQL adds an abstraction layer that hides implementation details of this structure and provides native syntaxes for collections and other common encodings. Language drivers are available for Java (JDBC), Python (DBAPI2), Node.JS (Helenus), C++, etc.

8.2.4 Basic rules of Cassandra data modeling

The key differences in doing data modeling for Cassandra versus a relational database include the following [30].

1. No joins. You cannot perform joins in Cassandra. If you have designed a data model and find that you need something like a join, you'll have to either do the work on the client side, or create a denormalized second table that represents the join results for you. This latter option is preferred in Cassandra data modeling. Performing joins on the client should be a very rare case; you really want to duplicate (denormalize) the data instead.

2. No referential integrity. Although Cassandra supports features such as lightweight transactions and batches, Cassandra itself has no concept of referential integrity across tables. In a relational database, you could specify foreign keys in a table to reference the primary key of a record in another table. But Cassandra does not enforce this. It is still a common design requirement to store IDs related to other entities in your tables, but operations such as cascading deletes are not available.

3. Denormalization. In relational database design, we are often taught the importance of normalization. This is not an advantage when working with Cassandra because it performs best when the data model is denormalized. It is often the case that companies end up denormalizing data in relational databases as well. There are two common reasons for this. One is performance. Companies simply can't get the performance they need when they have to do so many joins on years' worth of data, so they denormalize along the lines of known queries. This ends up working, but goes against the grain of how relational databases are intended to be designed, and ultimately makes one question whether using a relational database is the best approach in these circumstances.

A second reason that relational databases get denormalized on purpose is a business document structure that requires retention. That is, you have an enclosing table that refers to a lot of external tables whose data could change over time, but you need to preserve the enclosing document as a snapshot in history. The common example here is with invoices. You already have customer and product tables, and you'd think that you could just make an invoice that refers to those tables. But this should never be done in practice. Customer or price information could change, and then you would lose the integrity of the invoice document as it was on the invoice date, which could violate audits, reports, or laws, and cause other problems.

In the relational world, denormalization violates Codd's normal forms, and we try to avoid it. But in Cassandra, denormalization is, well, perfectly normal. It's not required if your data model is simple. But don't be afraid of it.

4. Query-first design. Relational modeling, in simple terms, means that you start from the conceptual domain and then represent the nouns in the domain in tables. You then assign primary keys and foreign keys to model relationships. When you have a many-to-many relationship, you create the join tables that represent just those keys. The join tables don’t exist in the real world, and are a necessary side effect of the way relational models work. After you have all your tables laid out, you can start writing queries that pull together disparate data using the relationships defined by the keys. The queries in the relational world are very much secondary. It is assumed that you can always get the data you want as long as you have your tables modeled properly. Even if you have to use several complex subqueries or join statements, this is usually true.

By contrast, in Cassandra you don’t start with the data model; you start with the query model. Instead of modeling the data first and then writing queries, with Cassandra you model the queries and let the data be organized around them. Think of the most common query paths your application will use, and then create the tables that you need to support them. Detractors have suggested that designing the queries first is overly constraining on application design, not to mention database modeling. But it is perfectly reasonable to expect that you should think hard about the queries in your application, just as you would, presumably, think hard about your relational domain. You may get it wrong, and then you’ll have problems in either world. Or your query needs might change over time, and then you’ll have to work to update your data set. But this is no different from defining the wrong tables, or needing additional tables, in an RDBMS.

5. Designing for optimal storage. In a relational database, it is frequently transparent to the user how tables are stored on disk, and it is rare to hear of recommendations about data modeling based on how the RDBMS might store tables on disk. However, that is an important consideration in Cassandra. Because Cassandra tables are each stored in separate files on disk, it’s important to keep related columns defined together in the same table.

A key goal that we will see as we begin creating data models in Cassandra is to minimize the number of partitions that must be searched in order to satisfy a given query. Because the partition is a unit of storage that does not get divided across nodes, a query that searches a single partition will typically yield the best performance.

6. Sorting is a design decision. In an RDBMS, you can easily change the order in which records are returned to you by using ORDER BY in your query. The default sort order is not configurable; by default, records are returned in the order in which they are written. If you want to change the order, you just modify your query, and you can sort by any list of columns. In Cassandra, however, sorting is treated differently; it is a design decision. The sort order available on queries is fixed, and is determined entirely by the selection of clustering columns you supply in the CREATE TABLE command. The CQL SELECT statement does support ORDER BY semantics, but only in the order specified by the clustering columns.

8.2.5 An example of Cassandra database design for IoT system

Requirement specifications. IoT system includes 10 sensors. Each of sensors sends data 100 times per second. Thus, the key requirements can be specified as following:

1) the system should continue recording if a single node stops working;

2) the system should record 1000 new records per second despite possible node/network failures;

3) the system should report all measures collected by any sensor for the specified day during few milliseconds;

4) the system should report all measures collected by any sensor during the specified period of time as quick as possible.

Database design. Cassandra executes writes much faster than other SQL and NoSQL databases, thus it can be a perfect choice to deal with thousands of write request per second. Besides, to address reliability requirements developers can chose a multi-replicated

Cassandra cluster (e.g. replication factor = 3) architecture where one of nodes is deployed in Clouds. To address the third requirements one can decide to store all measures collected within the same day in a single raw. With this purpose we need to create a composite primary key which consists of an event time used as a clustering key and also a composite partition key including the current date and a sensor id:

```
CREATE TABLE temperature_events_by_day (
   day text,
   sensor_id uuid,
   event_time timestamp,
   temperature double,
   PRIMARY KEY ((day, sensor_id), event_time)
)
WITH CLUSTERING ORDER BY event_time DESC;
```

where `day` – is the text of the following format: 'YYYY-MM-DD'; `(day,sensor_id)` – is a composite partition key; `event_time` – is a clustering key.

A *partition key* is used to identify the partition or node in the cluster that stores that row. When data is read or written from the cluster, a function called *Partitioner* is used to compute the hash value of the partition key. This hash value is used to determine the node/partition which contains that row.

The purpose of the *clustering key* is to store row data in a sorted order. The sorting of data is based on columns, which are included in the clustering key. This arrangement makes it efficient to retrieve data using the clustering key.

Thanks to the reverse sorting inside the row (`WITH CLUSTERING ORDER BY event_time DESC`), we will always get the most important (the latest) data at the fingertips.

The search for the partition key which is the day/sensor_id is a very fast operation in Cassandra. Thus, the proposed data model meets the third requirement.

To implement the forth requirement it is quite possible to use the above table. It will require searching for the certain day(s) at the first stage and comparing time stamps within each day at the second one. However, this can take a while if the database includes data collected during quite a few days. Taking into account the fact that the system has only 10 sensors one can consider creating the second key space where each row corresponds to the certain sensor ignoring days:

```
CREATE TABLE temperature_events_by_day (
   sensor_id uuid,
   event_time timestamp,
   temperature double,
   PRIMARY KEY (sensor_id, event_time)
)
WITH CLUSTERING ORDER BY event_time DESC;
```

where `sensor_id` – is a partition key; `event_time` – is a clustering key.

When inserting the data into that table we should limit the lifetime of each cell in order not to increase 2 billion columns (Cassandra limitation). Considering that each sensor gives no more than 100 readings per second:

$$2^{31} / (24\ \textit{hours} \cdot 60\ \textit{minutes} \cdot 60\ \textit{seconds} \cdot 100\ \textit{events per second}) =$$
$$= 2147483648 / (24 \cdot 60 \cdot 60 \cdot 100) = 248.55\ \textit{days}.$$

Thus, when inserting data into that table it is necessary to ensure that after 248 days the oldest data will be deleted from the table:

```
INSERT INTO temperature_events (
   sensor_id, event_time, temperature)
VALUES (
   '12341234-1234-1234', toTimestamp(now()), 36.6)
TTL 21427200;
```

where `TTL 21427200` – is the time to live in seconds equivalent to 248 days.

Querying the `temperature_events` table to retrieve measures collected by any sensor during the specified period of time is much faster. Though, in the application code, one will also need to put a condition that if the requested data goes beyond the last 248 days, then the `temperature_events_by_day` table should be used.

Duplicating the same value several times is the normal practice for NoSQL databases. In our scenario writing data to `temperature_events` table is much faster than to `temperature_events_by_day`. This is because in the first case Cassandra does not need to look for the node(s) a new value should be written to. It is know in advance from the sensor id which is used as a partition key. Reading data is also very fast.

8.3 Cassandra performance benchmarking

8.3.1 Experimental setup and benchmarking scenario

Cassandra deployment setup. As an object of our experiments we have deployed the 3-replicated Cassandra 2.1 cluster in the Amazon EC2 cloud (Fig. 8.3). Replication factor equal to 3 is the most typical setup for many modern distributed computing systems and Internet services, including Amazon S3, Amazon EMR, Facebook Haystack, DynamoDB, etc.

The cluster was deployed in the AWS US-West-2 (Oregon) region on c3.xlarge instances (vCPUs – 4, RAM – 7.5 GB, SSD – 2x40 GB, OS – Ubuntu Server 16.04 LTS).

Benchmark. Our work uses the YCSB (Yahoo! Cloud Serving Benchmark) framework which is considered to be a de-facto standard benchmark to evaluate performance of various NoSQL databases like Cassandra, Mongo, Redis, HBase and others [11]. YCSB is an open-source Java project.

The YCSB framework includes 6 out-of-the-box workloads [11], each testing different common use case scenarios with a certain mix of reads and writes (50/50, 95/5, read-only, read-latest, read-modify-write, etc.). In our experiments we used the read-only Workload C, and the Workload A, slightly modified to run write-only operations.

All the rest Cassandra and YCSB parameters (e.g. request distribution, testbed database, etc.) were set to their default values.

The YCSB Client is a Java program that generates data to be loaded to the database, and run the workloads. It was deployed in the same Amazon EC2 cloud data centre on a separate virtual machine.

Benchmarking scenario. A general methodology on benchmarking Cassandra and other NoSQL databases with YCSB can be found in [10, 31]. However, unlike this and other works (e.g. [11, 12, 13, 14, 15]) studying and comparing the maximal databases throughput we put the focus on analysing the dynamic aspects of Cassandra performance under different consistency settings. In particular, we analyse how database latency and throughput depend on a current workload (i.e. number of concurrent requests/threads).

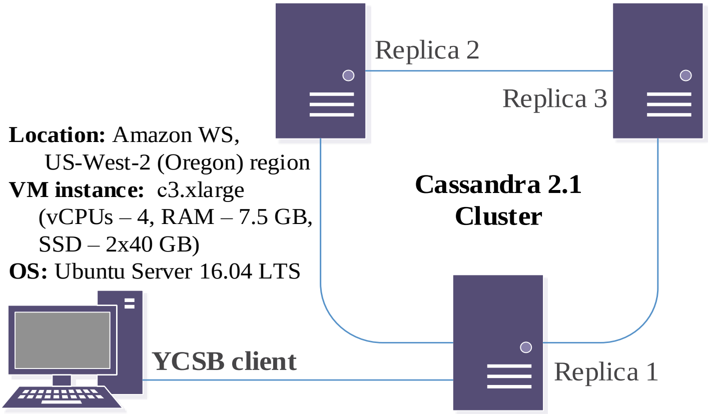

Replica 2
Replica 3
Location: Amazon WS, US-West-2 (Oregon) region
VM instance: c3.xlarge (vCPUs – 4, RAM – 7.5 GB, SSD – 2x40 GB)
OS: Ubuntu Server 16.04 LTS
Cassandra 2.1 Cluster
YCSB client
Replica 1

Fig. 8.3 – Experimental setup: Cassandra cluster

To achieve this we run a series of YSCB read and write performance tests on Apache Cassandra with a number of threads varied from 10 to 1000. The operation count within each thread was set to 1000.

The same scenario was run for read and write workloads on the 3-replicated Cassandra cluster with the three different consistency settings: ONE, QUORUM and ALL.

8.3.2 Raw data analysis and the cold start phenomenon

YCSB supports different measurement types including 'histogram', 'timeseries' and 'raw'. In our experiments we set it to 'raw' when all measurements are output as raw datapoints in the following csv format: operation (READ|WRITE), timestamp of the measurement (ms), latency (us). As a result we can plot response delay graphs depicted, for example, in Fig. 8.4 and Fig. 8.5.

The 'Raw' measurement type, used in our experiments, requires further manual analysis of the benchmarked data. Though, it also provides a great flexibility for a posterior analysis and allows us to get important insights rarely discussed by other researchers.

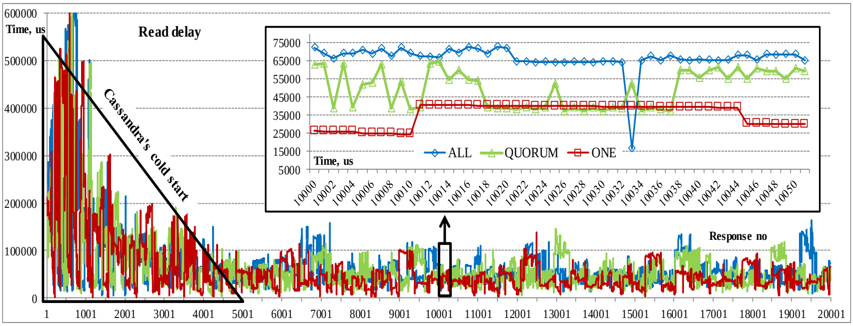


Fig. 8.4 – A fragment of the READ delay graph, 500 threads

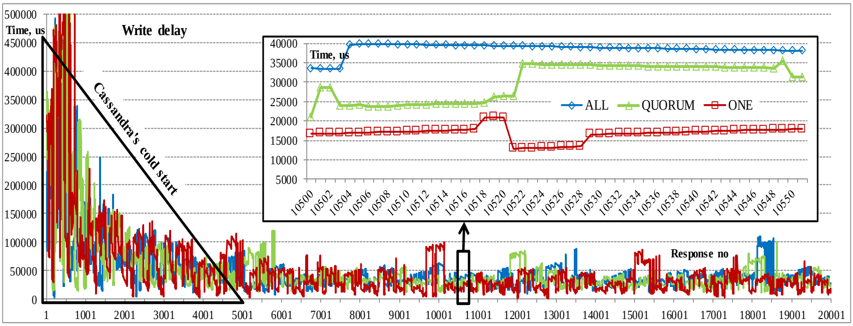


Fig. 8.5 – A fragment of the WRITE delay graph, 500 threads

In particular, we noticed that the *cold start* phenomenon can have a significant effect on the results of the Cassandra performance analysis.

This situation exhibits itself through the initial period of low performance observed in the beginning of each read and write tests (see Figs. 8.4-5). In general, its duration depends on the database size, available RAM, intensity of read and write requests and their distribution and other factors.

In all our experiments this period lasted approximately 800-1000 milliseconds. This phenomenon is explained by the fact that Cassandra uses three layers of data store:

- *memtable* (stored in RAM and periodically flushed to disk),
- *commit log* and
- *SSTable* (both are stored on disk).

If the requested row is not in *memtable*, a read needs to look-up in all the *SSTable* files on disk to load data to *memtable*.

In addition, Cassandra also supports integrated cacheing and distributes cache data around the cluster. Thus, during the cold start period Cassandra reads data from *SSTables* to *memtables* and warms up cache.

It is clear that researchers should take this phenomenon into account and to ignore the period of cold start in further statistical analysis. Otherwise, the average performance estimates would be significantly biased.

For instance, in our experiments the delays measured during the cold start were on average 5-8 times longer than the ones measured during the rest of time.

8.3.3 Read/write latency and throughput statistics

Tables 8.1 and 8.2 summarise the results of Cassandra performance benchmarking. They show that the average delay for both read and write requests increases almost linearly as the number of threads increases.

As we expected, when Cassandra is configured to provide consistency level ONE, the latency of both read and write operations is lower (by 9% and 26% correspondingly) than the average response time of the ALL consistency setting.

The QUORUM setting demonstrates a rational balance between delays and data consistency. Besides, our results confirm the claim that Cassandra has very high write speed. Indeed, write operations are almost 25% faster on average than read requests independently of consistency settings.

Table 8.1 shows that Cassandra read throughput is saturating after 500 threads reaching its maximal level at around 10000 requests per second for the ONE consistency setting.

The maximal throughput for QUORUM and ALL consistency settings volatiles around 9000 and 8300 requests per second correspondently.

Cassandra writes executed under the ONE consistency level reach the maximal throughput of 13552 requests per second. For the QUORUM and ALL consistency settings it fluctuates around 12500 requests per second (see Table 8.2).

Table 8.1 shows that Cassandra read throughput is saturating after 500 threads reaching its maximal level at around 10000 requests per second for the ONE consistency setting. The maximal throughput for QUORUM and ALL consistency settings volatiles around 9000 and 8300 requests per second correspondently.

Cassandra writes executed under the ONE consistency level reach the maximal throughput of 13552 requests per second. For the QUORUM and ALL consistency settings it fluctuates around 12500 requests per second (see Table 8.2).

A combination of average delay and average throughput columns of Tables 8.1 and 8.2 allow us to analyse how the average read and write delays depend on the current workload. When the workload reaches the maximal Cassandra throughput, delays increase in exponential progression (see Fig. 8.6-8.7).

These figures clearly show performance benefits offered by weaker consistency settings. It is also shown that delays become highly volatile when Cassandra operates under the heavy workload close to its maximal throughput.

Table 8.1 – Cassandra READ performance statistics

Threads	ONE		QUORUM				ALL			
			Delay		Throughput		Delay		Throughput	
	Average delay, us	Average throughput, ops./s	average, us	% of ONE	average, ops./s	% of ONE	average, us	% of ONE	average, ops./s	% of ONE
10	8120	1136	8150	100%	1023	90%	9111	112%	959	84%
50	14207	3425	14277	100%	3195	93%	14472	102%	3181	93%
100	14268	6323	18139	127%	5189	82%	17428	122%	5380	85%
200	20153	9259	26350	131%	7022	76%	29218	145%	6471	70%
300	31038	9325	35996	116%	7815	84%	41326	133%	7011	75%
400	38928	9661	48054	123%	7998	83%	52921	136%	7313	76%
500	49931	9723	59799	120%	8173	84%	65569	131%	7438	76%
600	56433	10221	72983	129%	8016	78%	77215	137%	7586	74%
700	69527	9799	79919	115%	8567	87%	84427	121%	8156	83%
800	74766	10487	87445	117%	9041	86%	92092	123%	8522	81%
900	89479	9848	98086	110%	9006	91%	107238	120%	8281	84%
1000	91854	10708	110762	121%	8872	83%	117367	128%	8398	78%
Average:				117%		85%		126%		80%

Table 8.2 – Cassandra WRITE Performance Statistics

Threads	ONE		QUORUM				ALL			
			Delay		Throughput		Delay		Throughput	
	Average delay, us	Average throughput, ops./s	average, us	% of ONE	average, ops./s	% of ONE	average, us	% of ONE	average, ops./s	% of ONE
10	8941	921	9101	102%	1066	116%	9106	102%	1053	114%
50	13098	3720	11737	90%	4043	109%	12591	96%	3861	104%
100	14550	6228	14747	101%	6284	101%	15535	107%	5937	95%
200	20625	8361	22040	107%	8274	99%	20464	99%	8803	105%
300	24119	10679	26078	108%	10434	98%	26858	111%	10271	96%
400	28944	12741	35338	122%	10380	81%	33319	115%	11294	89%
500	36831	12486	37784	103%	12400	99%	41364	112%	11530	92%
600	41412	13444	43940	106%	12548	93%	45963	111%	12354	92%
700	48256	13533	53276	110%	12514	92%	53192	110%	12587	93%
800	55629	13369	61712	111%	12372	93%	64856	117%	11813	88%
900	61410	13552	67329	110%	12568	93%	67615	110%	13051	96%
1000	69554	12985	71726	103%	13328	103%	78333	113%	11841	91%
Average:				106%		98%		109%		96%

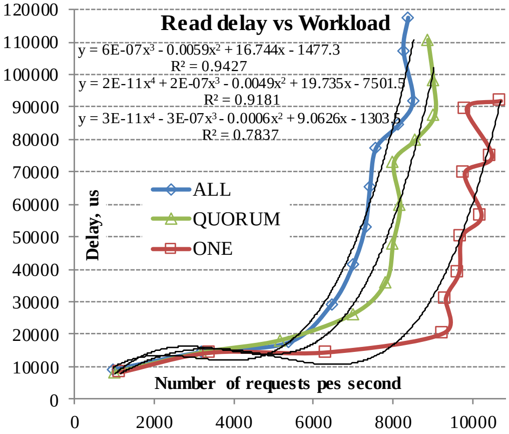


Fig. 8.6 – Average Cassandra READ delay depending on the current workload

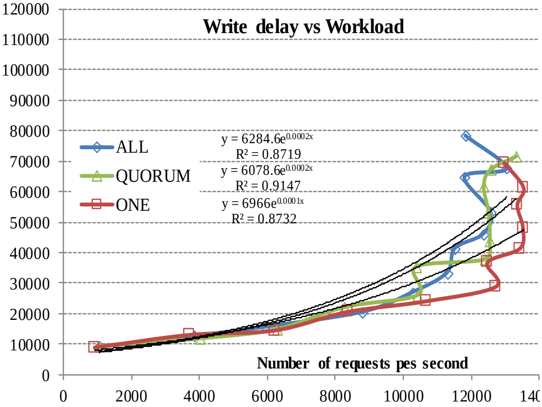


Fig. 8.7 – Average Cassandra WRITE delay depending on the current workload

8.3.4 Theoretical regressions of the Cassandra performance

Graphs presented in Fig. 8.8 depict the discrete set of measured values. They do not allow system developers to precisely estimate database latency throughout the whole range of possible workloads.

A regression function estimated from the experimental data will effectively solve this problem. As far as measured read/write delays have clear non-linear dependency on the workload we tried exponential and polynomial regression functions. As a measure of extrapolation accuracy we use the R-squared value (often referred to as the goodness-of-fit [32]) estimated by the Microsoft Excel. Our analysis (see Table 8.3) shows that the polynomial regression (8.2-4) best fits the read experimental statistics while write statistics can be better interpolated by the exponential function (8.5-7).

$$y_{\mathrm{ALL}}^{Read}(x)=6\cdot 10^{-7}x^{3}-0.0059x^{2}+16.744x-1477.3 \tag{8.2}$$

$$y_{\mathrm{QUORUM}}^{Read}(x)=2\cdot 10^{-11}x^{4}+2\cdot 10^{-7}x^{3}-0.0049x^{2}+19.735x-7501.5 \tag{8.3}$$

$$y_{\mathrm{ONE}}^{Read}(\mathrm{x})=3\cdot 10^{-11}x^{4}-3\cdot 10^{-7}x^{3}-0.0006x^{2}+9.0626x-1303.5 \tag{8.4}$$

$$y_{\mathrm{ALL}}^{Write}(x)=6284.6e^{0.0002x} \tag{8.5}$$

$$y_{\mathrm{QUORUM}}^{Write}(x)=6078.6e^{0.0002x} \tag{8.6}$$

$$y_{\mathrm{ONE}}^{Write}(x)=6966.0e^{0.0001x} \tag{8.7}$$

where $y_{\mathbf{ALL}}^{Read}$, $y_{\mathbf{QUORUM}}^{Read}$, $y_{\mathbf{ONE}}^{Read}$, $y_{\mathbf{ALL}}^{Write}$, $y_{\mathbf{QUORUM}}^{Write}$, $y_{\mathbf{ONE}}^{Write}$ – Cassandra read/update response time for different consistency settings [us]; *x* – the current workload [ops/sec].

Table 8.3 – Goodness-of-fit for READ/WRITE delay regressions

		Polynomial regression			Exponential regression
		order=2	order=3	order=4	
Read statistics	ALL	0.91	**0.94**	0.94	0.89
	QUORUM	0.86	0.90	**0.91**	0.89
	ONE	0.70	0.77	**0.78**	0.76

Write statistics	ALL	0.76	0.77	0.78	**0.87**
	QUORUM	0.85	0.88	0.88	**0.91**
	ONE	0.75	0.78	0.78	**0.87**

Using regression functions helps to predict Cassandra delays for different consistency levels under different workload (see Table 8.4).

Table 8.4 – Cassandra READ/WRITE delay regressions

Workload, requests/s	Average read delay*, us			Average write delay**, us		
	ONE	QUORUM	ALL	ONE	QUORUM	ALL
1000	6882	7615	9970	8028	7196	7454
2000	12586	14750	13418	9251	8520	8840
3000	15275	16187	12757	10661	10087	10484
4000	15156	14737	11878	12286	11942	12434
5000	13174	13742	14671	14158	14138	14747
6000	11013	17076	25027	16315	16738	17490
7000	11099	29141	46836	18802	19817	20743
8000	16595	54871	83989	21667	23461	24601
9000	31404	99730	140376	24969	27776	29177
10000	60171	169712	219888	28775	32884	34604

*polynomial regression; **exponential regression

8.4 Methodology of optimal consistency setup

8.4.1 Finding the optimal consistency settings

As it was analysed in previous section Cassandra can guarantee the strong data consistency model if a sum of replicas written and read is higher than the replication factor. It means that for a three-replicated system there are 6 possible read/write consistency settings guaranteeing the strong data consistency:

1) ‘Read ONE – Write ALL’ (1R-3W);
2) ‘Read QUORUM – Write QUORUM’ (2R-2W);
3) ‘Read ALL – Write ONE’ (3R-1W);
4) ‘Read QUORUM – Write ALL’ (2R-3W);
5) ‘Read ALL– Write QUORUM’ (3R-2W);
6) ‘Read ALL – Write ALL’ (3R-3W).

Besides, two additional settings ‘Read ONE – Write QUORUM’ (1R-2W) and ‘Read QUORUM – Write ONE’ (2R-1W) provide 66.6% consistency confidence.

Finally, ‘Read ONE – Write ONE’ (1R-1W) setting can guarantee the only 33.3% consistency confidence. If a system developer wants to ensure that a read operation always reflects the most recent update it can opt for one of the first six settings. However, our experiments clearly show that the fewer replicas are invoked the faster Cassandra performs read/write operations. Thus, in practice one should choose between the only three settings: 1R-3W, 2R-2W and 3R-1W.

As all three settings guarantee the strong consistency a system developer could be interested in choosing the combination providing the minimal response delay on average. In turn, the response delay and Cassandra throughput depend on the current workload and the ratio between read/write requests. Using regression functions (8.2-7) we can predict the average Cassandra latency under the mixed workload:

$$y_{1R\text{-}3W}(x) = P_{Read} \cdot y_{ONE}^{Read}(x) + P_{Write} \cdot y_{ALL}^{Write}(x) \tag{8.8}$$

$$y_{2R\text{-}2W}(x) = P_{Read} \cdot y_{QUORUM}^{Read}(x) + P_{Write} \cdot y_{QUORUM}^{Write}(x) \tag{8.9}$$

$$y_{3R\text{-}1W}(x) = P_{Read} \cdot y_{ALL}^{Read}(x) + P_{Write} \cdot y_{ONE}^{Write}(x) \tag{8.10}$$

where P_{Read}, P_{Write} – probabilities of read/update requests, $P_{Read} + P_{Write} = 1$.

Table 8.5 provides some estimates of Cassandra performance for different settings guaranteeing the strong consistency. It is shown that none of the consistency settings provides the lowest response time in all possible situations (the minimal values are underlined).

For instance, if the percentage of write requests significantly prevails read re-quests (up to 10/90%) the 2R-2W consistency setting provides the lowest delay for workloads less than 6000 requests per second. However, with the increase of the percentage of write requests the 1R-3W setting becomes more preferable.

Increasing the number of requests per second and the percentage of read requests make the 2R-2W and especially 3R-1W setups very inefficient demonstrating the exponential grow of Cassandra latency.

However, these setups still provide the lowest delay in heavy 'write mostly' workloads when the percentage of read requests is less than 20%.

Table 8.5 – Cassandra delay under different workloads depending on READS/WRITES proportion

Workload, requests/s	Read/Write = 1/99%			Read/Write = 10/90%			Read/Write = 50/50%			Read/Write = 90/10%		
	1R-3W	2R-2W	3R-1W	1R-3W	2R-2W	3R-1W	1R-3W	2R-WU	3R-1W	1R-3W	2R-2W	3R-1W
1000	7448	7201	8047	7396	7238	8222	7168	7406	8999	6939	7573	9776
2000	8877	8582	9293	9215	9143	9668	10713	11635	11334	12211	14127	13001
3000	10532	10148	10682	10963	10697	10870	12880	13137	11709	14796	15577	12548
4000	12461	11970	12281	12706	12221	12245	13795	13339	12082	14884	14457	11919
5000	14731	14134	14163	14590	14099	14209	13961	13940	14414	13331	13782	14620
6000	17425	16742	16403	16842	16772	17187	14252	16907	20671	11661	17042	24156
7000	20647	19910	19082	19779	20749	21605	15921	24479	32819	12063	28209	44033
8000	24521	23775	22290	23801	26602	27899	20598	39166	52828	17395	51730	77757
9000	29200	28495	26123	29400	34971	36510	30291	63753	82673	31182	92535	128836
10000	33640	33008	29709	37161	46567	47886	47388	101298	124331	57614	156029	200777

The 3D surface plots in Fig. 8.8 demonstrate the domains in the input workload where the particular consistency setting provides the best result.

This information can be extremely useful for system developers allowing them to dynamically change consistency settings of read and write requests in an optimal way still guaranteeing the strong data consistency. Our results demonstrate the general dependencies of Cassandra delays and allow us to formulate the basic principles for choosing certain consistency settings.

However, the presented data remain unique for our experimental setup and might not exactly match other installations. Obviously, the exact delays depend on the size and structure of the column family, used hardware, number of nodes and their geographical distribution.

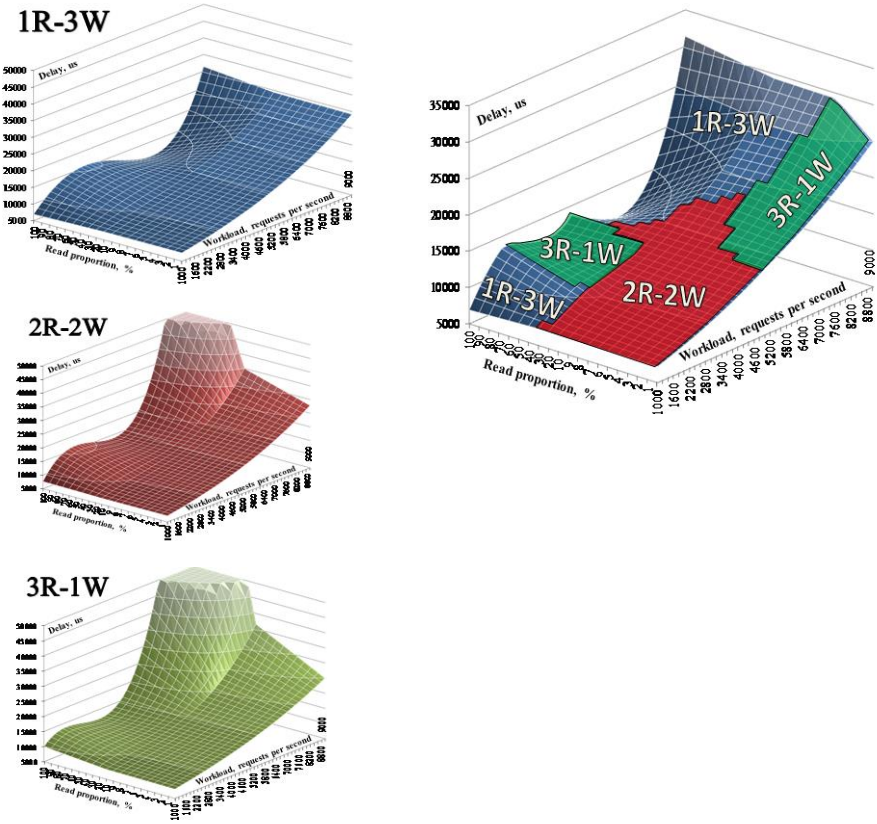


Fig. 8.8 – Workload domains with the optimal consistency settings

8.4.2 Experimental-based methodology for optimal coordination of consistency settings

In this section we generalize the experimental data reported in the paper by proposing a universal methodology to be used by system engineers for predicting Cassandra delays and coordinating consistency settings for read and write requests in an optimal manner. The methodology employs a benchmarking approach to quantify Cassandra latency and throughput and consists of the following five steps:

1. Deploying and running a Cassandra database in a real production environment.
2. Modifying the YCSB workloads to execute application-specific read and write queries. This will help to evaluate Cassandra performance in the realistic application scenarios.
3. Benchmarking the Cassandra database under different workloads (threads per second) with different consistency settings following the benchmarking scenario described in Section 4.1.
4. Finding regression functions that accurately interpolate average read/write delays, collected experimentally, depending on the workload and read/write consistency settings described in Section 5.2.
5. During the real operation, (7)–(9) should be used to choose the optimal consistency settings providing the minimal average response time under the certain workload taking into account the actual ratio of read and write requests.

The proposed methodology enables a run-time optimization of consistency settings to achieve the maximal Cassandra performance and still guarantee the strong data consistency.

Conclusions and questions

Our work discuss big-data solutions introduced to support rapid data grow and experimentally investigates the interplay between consistency and performance of the Cassandra NoSQL database. This is an important part of the fundamental trade-off between Consistency, Availability and Partition tolerance which is in the very nature of globally-distributed systems and large-scale replicated data storages.

Our results show that a particular consistency setting can significantly affect Cassandra response time and throughput that have to be accounted during system design and operation. The strong data consistency can increase database latency by 25% and degrade its throughput by 20% on average.

The Cassandra database offers developers a unique opportunity to tune consistency setting for each read or write requests. Besides, it is possible to guarantee the strong data consistency by coordinating consistency settings for read and write requests to ensure that the sum of nodes written and read is greater than the replication factor.

One of our major findings is the fact that the optimal consistency settings maximizing Cassandra performance significantly depend on the current workload and a ratio of read and write requests. To this end, we propose a benchmarking-based approach to optimal coordination of consistency settings at run-time to minimise the Cassandra response time and still guarantee the strong data consistency.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What is *Big Data*? What purpose this term is used for?.
2. What are properties of Big Data? Explain 10Vs big data model.
3. What are core differences between SQL and NoSQL databases?
4. What is a difference between ACID and BASE properties?
5. What is consistency in term of NoSQL databases? What is eventual consistency?
6. Describe examples of NoSQL databases and their data models?
7. What does the CAP theorem postulate?
8. How are consistency and performance (i.e. latency) of replicated data storages interconnected?
9. What are basic rules of Cassandra data modeling?
10. What is YCSB? What purpose it is used for?

References

1. D. Laney, "3D data management: Controlling data volume, velocity and variety," *META Group Research Note*, vol. 6, no. 70, 2001.
2. P. Goes, "Design science research in top information systems journals," *MIS Quarterly: Management Information Systems,* vol. 28, no. 1, 2014.
3. B. Marr, "Big Data: The 5 Vs Everyone Must Know," 6 March 2014. [Online]. Available: https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know/.
4. G. Firican, "The 10 Vs of Big Data," 8 February 2017. [Online]. Available: https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx.
5. N. Lord, "A Timeline of the Ashley Madison Hack," 27 July 2017. [Online]. Available: https://digitalguardian.com/blog/timeline-ashley-madison-hack.
6. E. Evans, "NoSQL 2009," 12 May 2009. [Online]. Available: http://blog.sym-link.com/2009/05/12/nosql_2009.html.

7. D. Pritchett, "Base: An Acid Alternative," *ACM Queue*, vol. 6, no. 3, pp. 48-55, 2008.

8. E. Brewer, "Towards Robust Distributed Systems," in *19th Annual ACM Symposium on Principles of Distributed Computing*, Portland, USA, 2000.

9. S. Gilbert and N. Lynch, "Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51-59, 2002.

10. Github, "Benchmarking Cassandra and other NoSQL databases with YCSB," [Online]. Available: https://github.com/cloudius-systems/osv/wiki/Benchmarking-Cassandra-and-other-NoSQL-databases-with-YCSB.

11. B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," in *1st ACM Symposium on Cloud Computing*, Indianapolis, Indiana, USA, 2010.

12. R. Hecht and S. Jablonski, "NoSQL Evaluation. A Use Case Oriented Survey," in *IEEE International Conference on Cloud and Service Computing*, Washington, USA, 2011.

13. V. Abramova, J. Bernardino and P. Furtado, "Testing Cloud Benchmark Scalability with Cassandra," in *IEEE 10th World Congress on Services*, Anchorage, USA, 2014.

14. J. Klein, I. Gorton, N. Ernst, P. Donohoe, K. Pham and C. Matser, "Performance Evaluation of NoSQL Databases: A Case Study," in *1st ACM/SPEC International Workshop on Performance Analysis of Big Data Systems*, Austin, USA, 2015.

15. G. Haughian, R. Osman and W. Knottenbelt, "Benchmarking Replication in Cassandra and MongoDB NoSQL Datastores," in *27th International Conference on Database and Expert Systems Applications*, Porto, Portugal, 2016.

16. V. A. Farias, F. R. Sousa, J. G. R. Maia, J. P. P. Gomes and J. C. Machado, "Regression based performance modeling and provisioning for NoSQL cloud databases," *Future Generation Computer Systems*, vol. 79, p. 72–81, 2018.

17. F. Karniavoura and K. Magoutis, "A measurement-based approach to performance prediction in NoSQL systems," in *25th IEEE Int. Symposium on the Modeling, Analysis, and Simulation of Computer and Telecom. Systems (MASCOTS’07),* Banff, Canada, 2017.

18. F. Cruz, F. Maia, M. Matos, R. Oliveira, J. Paulo, J. Pereira and R. Vilaca, "Resource usage prediction in distributed key-value datastores," in

IFIP Distributed Applications and Interoperable Systems (DAIS'2017), Heraklion, Crete, 2017.

19. E. Brewer, "CAP twelve years later: How the "rules" have changed," *Computer*, vol. 45, no. 2, pp. 23-29, 2012.

20. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35-40, 2010.

21. D. Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design," *IEEE Computer*, vol. 45, no. 2, pp. 37-42, 2012.

22. R. Guerraoui, M. Pavlovic and D. Seredinschi, "Trade-offs in replicated systems," *IEEE Bulletin of the Technical Committee on Data Engineering*, vol. 39, no. 1, pp. 14-26, 2016.

23. Gorbenko and A. Romanovsky, "Time-outing Internet Services," *IEEE Security & Privacy*, vol. 11, no. 2, pp. 68-71, 2013.

24. O. Tarasyuk, A. Gorbenko, A. Romanovsky, V. Kharchenko and V. Ruban, "The Impact of Consistency on System Latency in Fault Tolerant Internet Computing," in *Distributed Applications and Interoperable Systems. Lecture Notes in Computer Science*, vol. 9038, A. Bessani and S. Bouchenak, Eds., Berlin, Springer-Verlag, 2015, pp. 179-192.

25. Fekete and K. Ramamritham, "Consistency Models for Replicated Data," in *Replication*, vol. LNCS 5959, B. Charron-Bost, F. Pedone and A. Schiper, Eds., Berlin, Springer-Verlag, 2010, pp. 1-17.

26. S. Burckhardt, "Principles of Eventual Consistency," *Foundations and Trends Programming Languages*, vol. 1, no. 1-2, pp. 1-150, 2014.

27. DataStax, Inc., "Apache Cassandra 2.1 for DSE. About data consistency," 14 February 2018. [Online]. Available: https://docs.datastax.com/en/cassandra/2.1/cassandra/dml/dmlAboutDataConsistency.html.

28. N. Neeraj, Mastering Apache Cassandra, Birmingham: Packt Publishing Ltd. , 2013.

29. DataStax, Inc., "Apache Cassandra 2.1. Configuring data consistency," 14 February 2018. [Online]. Available: https://docs.datastax.com/en/cassandra/2.1/cassandra/dml/dml_config_consistency_c.html.

30. J. Carpenter and E. Hewitt, Cassandra: The Definitive Guide, O'Reilly Media, 2016, p. 92.

31. Cooper, "Running a Workload," 2 Jul 2013. [Online]. Available: https://github.com/brianfrankcooper/YCSB/wiki/Running-a-Workload.

32. J. L. Devore, Probability and Statistics for Engineering and the Sciences, 8th Edition, Boston (USA): Cengage Learning, 2011.

PART III. MOBILE AND HYBRID IOT-BASED COMPUTING

9 MOBILE AND NETWORKING FOR IOT

Dr. V.O. Butenko, D.A. Butenko (KhAI),
Dr. E.B. Odarushchenko (PSAA)

Contents

Abbreviations

AIOTI - Alliance for Internet of Things Innovation
API - Application Programming Interface
AR - Augmented Reality
BLE - Bluetooth Low Energy
BPR - Boot Progress Register
ESTI - European Telecommunication Standards Institute
HLA - High-Level Architecture
HUD - Heads-Up Display
ID - Identifier
IoT - Internet of Things
IT - Information technology
MR - Mixed Reality
MVC - Model-View-Controller
OHMD - Optical Head-Mounted Display
OS - Operating System
OT - Operational Technology
PAC - Pointer Authentication Code
QoS - Quality of Service
SSL - Secure Sockets Layer
SSOs - Standards Setting Organizations
STF - Special Task Force
UI - User Interface
UID - Unique User Identifier
VR - Virtual Reality

9.1 Evolution of mobile and IoT standards and development

9.1.1 Brief view on IoT standards classification

Internet of Things (IoT) is being one of the most emerging technologies of the past few years. The wide IoT society is bringing to live ambitious concepts to deploy the large and complex information technology (IT) systems. However, the newest IoT ideas can be reached only under strict system of standards that regard to various IoT domains.

There are already a number of existing documents that allow to address many of the requirements of IoT systems in a large spectrum of solutions (ranging from consumer to industrial) for a large number of domains, as various as cities, e-health, e-house, transportation, etc. IoT community have started to work for years now to adapt existing general scope standards for IoT needs and develop new one [1]. However, there is always a risk of duplication, fragmentation, competition between standards organisations. In its 2016 communication on “ICT Standardisation Priorities for the Digital Single Market” [2], the European Commission stated that: “However, the IoT landscape is currently fragmented because there are so many proprietary or semi-closed solutions alongside a plethora of existing standards. This can limit innovations that span several application areas”. The European Commission also outlines an essential way-forward [2]: “Largescale implementation and validation of cross-cutting solutions and standards is now the key to interoperability, reliability and security in the EU and globally”.

The IoT standartisation community has taken two dimensions into account [3]:

1. Expansion of the reach of “horizontal layers” standards versus “vertical-domains”- specific standards. The IoT landscape developed by the Alliance for Internet of Things Innovation (AIOTI) Work Group 3 on Standardisation has used the distinction between the horizontal and vertical domains for the classification of the organisations that are active in IoT standardization. “Vertical” domain represents 8 sectors where IoT systems are developed and deployed and “horizontal” layer that group standards across all domains, for example telecommunications. The European Telecommunication Standards Institute (ESTI) Special Task Force (STF) 505 report on the IoT Landscape [4] has identifies 329 standards that apply to IoT systems, which gives importance of “vertical” versus “horizontal” standards scheme.

2. Specialization of general purpose standards for application to more complex and demanding domains. This is the case with the convergence of IT and OT (Operational Technology) in the industrial domain. The AIOTI has developed the High-Level Architecture (HLA) that defines three layers and provides even more complete way to classify the standards:

- The Application layer contains the communications and interface methods used in process-to-process communications;
- The IoT layer groups IoT specific functions, such as data storage and sharing, and exposes those to the application layer via Application Programming Interfaces (APIs). The IoT Layer makes use of the Network layer’s services;
- The Network layer services can be grouped into data plane services, providing short and long range connectivity and data forwarding between entities, and control plane services such as location, device triggering, quality of service (QoS) or determinism.

9.1.2 Mobile wireless industry standards

During last decade mobile technologies have changes the modern society. High speed mobile wireless communications made it possible to use mobile devices not just for calls, but for internet browsing, email, social media and various different applications. The series of technological revolutions in wireless technology standards have changed the way we live and work. For mobile wireless, this started with the second generation (“2G”) digital cellular systems and through third, fourth and fifth generations (“3G”, “4G”, “5G”) to the and future of six (“6G”) systems. Each system builds upon a long series of technology standards. Higher data transmission speeds and efficient communications enabled by these technologies have unleashed a range of new mobile data services (e.g., applications and streaming videos) and complex products (e.g., smartphones and tablets). Various firms are developing common technology standards to meet the rising user needs while insuring devices interoperability.

Being simultaneously collaborative and competitive today's diverse mobile wireless industry heavily relies at technology standards. Without common standards, users would not experience the worldwide interoperability and interconnectivity across mobile devices at the core of wireless's business and consumer appeal.

Here we provide brief description of various mobile wireless industry standards application during the basic stages of new technology development.

The industry evolves mainly in three basic stages: development of standardized technology, development of standards-compliant product by device and infrastructure makers, deployment of network connectivity and offer devices to customers through retail shop (Figure 9.1) [5].

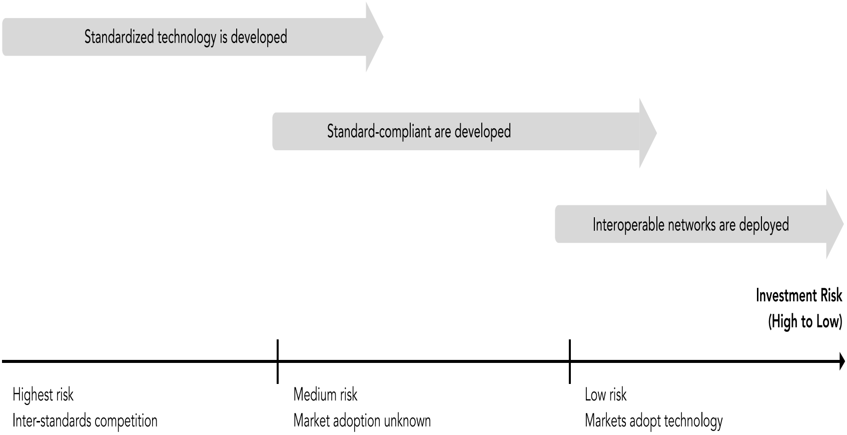

Standardized technology is developed
Standard-compliant are developed
Interoperable networks are deployed
Investment Risk
(High to Low)
Highest risk
Inter-standards competition
Medium risk
Market adoption unknown
Low risk
Markets adopt technology

Figure 9.1 – Stages of new technology development

1. Stage One: *Development of Technology Standards* often begins with standards setting organizations (SSOs), which provide a platform for industry scientists and engineers to consider new features. Any participant can propose a new feature and in case if proposal receives approval by every participant, the SSO start to generate

common technical solutions to enable those features. As a result new technology standards that aim to solve complex technology problems are developed. For example, developing standards for fast and efficient data transmission, seamless connection transitions as users move at fast speeds, and video streaming all required years of innovation. Contributing firms face two main risks while developing standards - strong interstandard competition and weak market adoption. Competition between several standards that address the same technological problem through very different solutions is a common case, for example, VHS defeated its rival BetaMax in the video standard wars, the wireless standards GSM and IS-95 were developed for 2G wireless communications based on different underlying technologies, and wireless cellular standard LTE prevailed over WiMax (IEEE 802.16e) in 4G wireless communications [5].

2. Stage Two: *Development of Products Firms* can only develop standards-compliant products once a standard is almost complete and commercialization often continues long standard definition. For instance, 3GPP published the first release of the 3G standard in 2000 and issued significant releases through 2007. On the product side, only in 2008 the Apple Inc. started supplying iPhones incorporating the 3G. At the product development stage, the risk of interstandard competition is significantly mitigated. Yet, market adoption risks still remain until products are rolled out in the marketplace [5].

3. Stage Three: *Deployment of Networks.* Network operators play an important role in the third stage of the mobile wireless value chain. Network operators make large capital investments in blocks of primary physical assets required for wireless communication. Operators deploy and maintain the infrastructure (i.e., the base stations and the servers) to provide the mobile wireless services. During network deployment, both interstandard competition and market adoption risks present at stage one and stage two of the technology evolution process are significantly mitigated. Operators have the benefit of rolling out and scaling up their networks based on gauging the consumer demand for a given technology [5].

9.2 Developing applications for Android and iOS

9.2.1 Developing applications for Android

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. Android is designed primarily for touchscreen mobile devices.

It’s source code is released by Google under open source licenses, thus being popular among various companies which require a ready-made, low-cost and customizable operating system for high-tech devices.

Android's open nature has encouraged a large IT community to use the open-source code as a foundation for community-driven projects [6].

The Figure 9.2 presents a generalized Android system architecture.

Linux with approximately 100 patches is in the base of all Android layers. This provides basic system functionality like process and memory management, various devices (camera, keypad, display etc.) management. Also, the kernel handles networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

Android Runtime [6] section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Android Runtime also provides a set of core libraries which enable developers to write applications using Java.

The Application Framework [6] layer provides many higher-level services to applications as Java classes. Developers widely use these services in their applications.

Android applications run in a sandbox [7], a system isolated area that does not have access to the rest of the system's resources, unless access permissions are explicitly granted by the user when the application is installed.

As we have briefly viewed a general Android system architecture the next step is to introduce the Google official recommendations to the basic app architecture (Figure 9.3).

Those recommendations can be charges as a good starting point for most situations and workflows. The common architectural principles are known to be as follows:

1. Separation of concerns.

This principle is about of keeping the UI-based classes Activity and Fragment as lean as possible to avoid various lifecycle-related problems. These classes should only contain logic that handles UI and operating system interaction.

2. Drive UI from model. Model components that are responsible for handling the data for an app. They are independent from View objects and app components, so they're unaffected by the app's lifecycle. This may help users not to lose data if the OS destroys app to free up resources and an app will continue to work even with bad network connection.

Application of this principle can make app more testable and consistent.

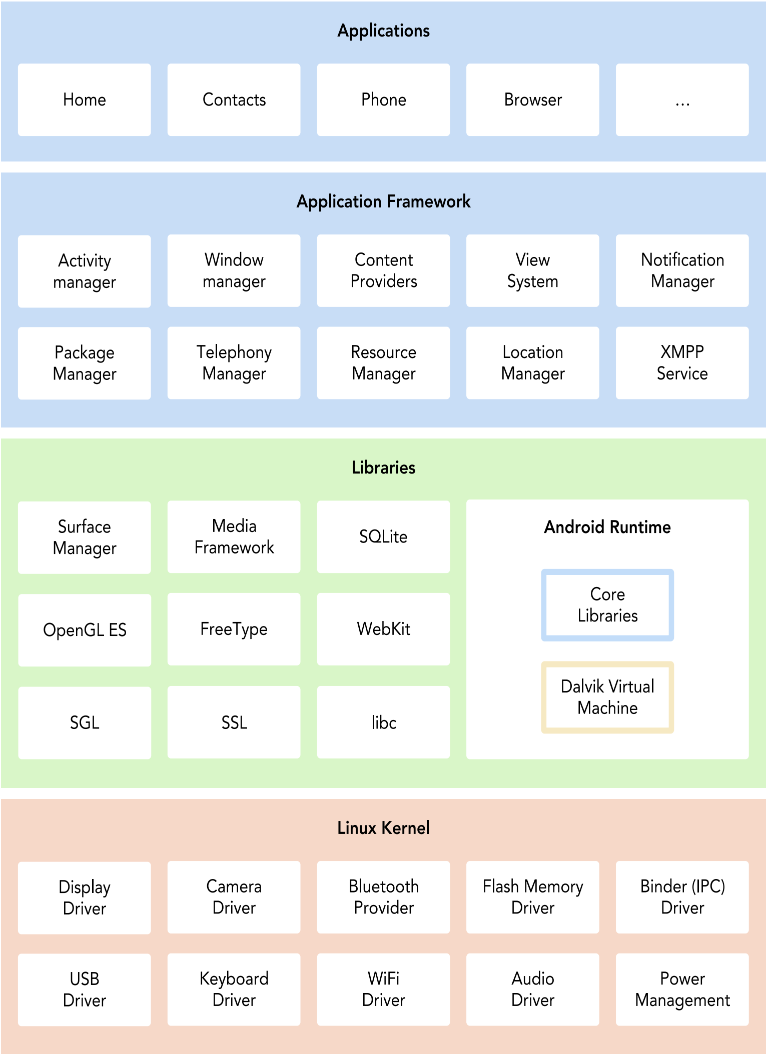


Figure 9.2 - Android system architecture

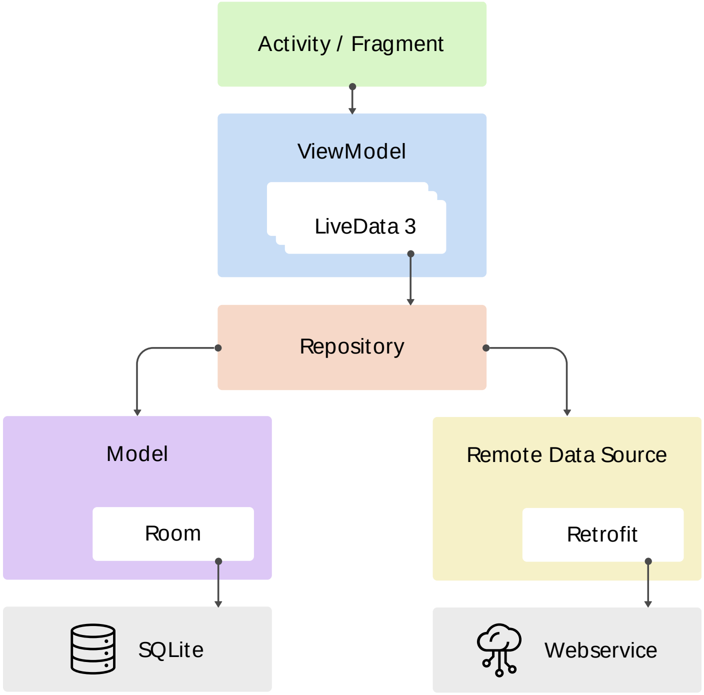


Figure 9.3 – Recommended Android app architecture

It should be noted that each component depends only on the one component level below it, for instance activities and fragments can only depend on ViewModels. The repository is the only class that have multiple dependencies on components below it – Model and Remote Data Source. This design aims to create the better user experience, for example, if the user comes back to app several minutes after they've closed it, they instantly see a user's information that the app persists locally.

Google has also stated a few basic recommendations [8] which aim to bring the best practices into live of upcoming apps. While being non mandatory they can help to make the code more robust, testable and maintainable in the long run:

1. Avoid such entry points as activities, services and broadcast receivers as the source of data.

2. Create well-defined boundaries of responsibility between various modules of an app.
3. Expose as little as possible from each module.
4. Consider how to make each module testable in isolation.
5. Focus on the unique core of your app so it stands out from other apps.
6. Persist as much relevant and fresh data as possible.
7. Assign one data source to be the single source of truth.

9.2.2 Developing applications for iOS

iOS is the operating system that runs on iPhone, iPod touch, and iPad devices. It manages the device hardware and provides the technologies for native apps implementation. The OS also ships with various system apps, that provide basic system services to the user. Native apps are built using the iOS system frameworks and Objective-C or/and Swift languages and run directly on iOS.

The iOS Architecture is Layered at the highest level, iOS acts as an intermediary between the underlying hardware and the apps that appear on the device. Apps communicate with the hardware through a set of well-defined system interfaces that protect app from hardware changes.

The implementation of iOS architecture can be viewed as set of four levels: Cocoa Touch, Media, Core Services and Core OS (Figure 9.4). Lower level represents fundamental services and techs on which app rely and high-level contains more sophisticated services and techs [9].

The Cocoa Touch layer gives the key frameworks for building iOS applications. It defines the basic application infrastructure and support for main technologies such as multitasking, touch-based input, push notifications, and many high-level system services.

The Media layer contains the graphics, audio, and video technologies for creating the best multimedia experience available on a mobile device.

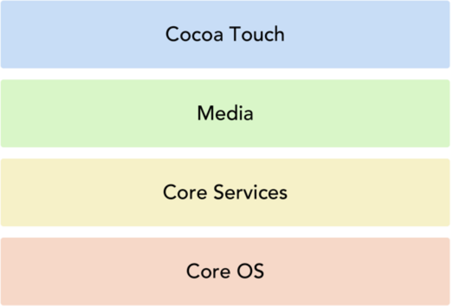


Figure 9.4 – iOS architecture layers

The Core Services layer contains the fundamental system services that all applications use. Even if they are not used directly in app, many parts of the system are built on their base.

The Core OS layer contains the low-level features that most other technologies are built upon. This framework is applied in situations where the app needs to explicitly deal with security or communicating with an external hardware accessory.

The official Apple recommendations states to use a model-view-controller (MVC) for iOS apps. This pattern clearly separates the data and logic from the visual presentation (Figure 9.5).

The role of objects presented on figure 6 is as follows [9]:

1. UIApplicationObject - manages the event loop and other high-level app behaviors, reports key app transitions and some special events to its delegate.

2. App delegate object – working in tandem with UIApplicationObject to handle app initialization, state transition and other high-level events.

3. Documents and data model objects – stores an app’s content and are specific for every app.

4. View controller objects – manages the presentation of an app content on screen, being a single view that collect subviews.

5. UIWindowObject – coordinated the presentation of one or more views on a screen.

6. View objects, control objects and layer objects – views and controls provide visual representation of an app content; layer object are data objects that represent visual content.

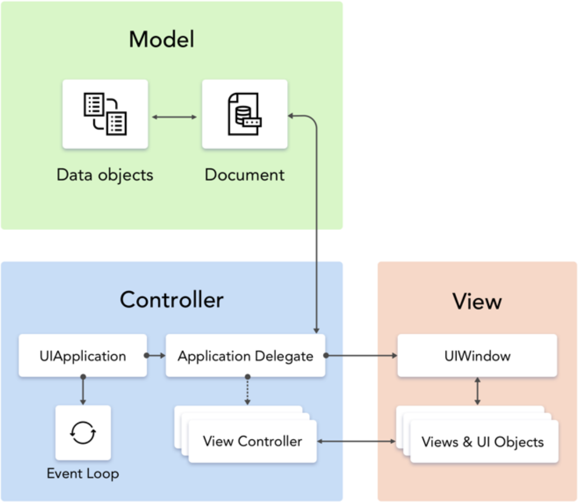


Figure 9.5 – Key objects of an iOS application

The Apple official documentation also states few tips to improve the app performance and thus bring better user experience. Those recommendations are non-mandatory [9]:

- Avoid doing work that requires polling, which prevents the CPU from going to sleep;
- Avoid accessing the disk too frequently.
- Do not draw to the screen faster than is needed as this is an expensive operation when it comes to power.
- Connect to external network servers only when needed.
- When connecting to the network, transmit the smallest amount of data needed to do the job.
- Provide network connection using the Wi-Fi radios whenever possible as Wi-Fi uses less power and is preferred over cellular radios.
- Eliminate memory leaks.

- Make resource files as small as possible.
- Use Core Data or SQLite for large data sets.
- For protocols define data formats to be as compact as possible.
- Avoid using chatty protocols.
- Limit the type of work on the main thread of an app.

9.2.3 Basic user-based iteration types of IoT with Android and iOS applications

There are four basic types of iteration between IoT and mobile devices that aim to inform user on various situations.

The first one is touch gestures. Most IoT devices that can be paired to mobile phones and participate in active everyday user life have small sizes. Thus the developer need to handle the display size limits as well as constrained amount of recognizable gestures. For example, the Apple Watch does not support multi-touch gestures, but introduce the force touch and digital crown. A force touch is triggered by forcefully pressing the screen which will open a context menu and the digital crown lets the user scroll the screen or its elements without obstructing it.

The second is notifications. For both Android and iOS applications notifications mainly used to communicate with user while application is not in the foreground and to provide entry to the app. Scheduled notifications are provided with a specific set of content and are triggered by a time or location value. Notifications should offer lightweight experiences, such as replying to a message, opening a location on a map, or playing a song. Notification templates are available for instant messaging, music playback, and calendar events.

The third type of interaction are compilations - small elements which are mainly presented on wearable IoT devices, for example smartwatches. Compilations provide quick access to information provided by the corresponding application. At minimum, complications show the application icon, but they can also display data generated by the application. A calendar application could show the current date, and an activity tracker the status of the day’s goal. Tapping the complication launches the application. Even where it possible the applications are not required to provide complications, but they can show several benefits for having the application's complication on the user's device. First, the application can conveniently show useful data to the user without starting the application. Second - the application will be kept suspended in memory which lets the system rapidly wake it up when the complication is tapped, and it also gives the application a bigger budget for background tasks.

The last type of existing interactions have voice commands. This can enable hands-free interactions with Android Wear and Handoff in case of using for iOS.

9.3 Usability, security and privacy concepts for Android and iOS applications

9.3.1 Usability, security and privacy concepts for Android

Basic usability concepts. Building assessable products that will refer to need of various people is a key concept for nowadays success applications. The material design presented by Google is built upon three main principles: clearness, robustness and specifity. The *clear* principle stands for helping users navigate in app by designing clear layouts with distinct calls on actions. The *robust* means to design app that accommodates a variety of users and *specific* states for supporting assistive technologies, which are specific for the platform [10].

Main assistive technologies as screen readers, magnification devices, wheelchairs, hearing aids or memory aids are recommended for use while increasing the application usability.

An app should make it easy to navigate in UI giving users confidence in knowing where they are in app and what is important. The app should give user feedback, so the visual feedback and touch feedback is strongly recommended for app. The navigation should have clear tasks flows with minimal amount of steps, the controls have to be easily allocated and clearly written.

Every added button, image or line of the text increases the UI complexity. The Google states that UI can be significantly simplifies by adding clear visible elements, sufficient contrast and size, hierarchy of importance and by presenting key information that is discernable at a glance.

The design colors have to support usability, thus means that all users should understand the content and even users with low vision can see basic elements and navigate freely in app. For users who are colorblind or cannot see differences in colors the design should include elements in addition to color that will ensure they receive the needed amount of information.

There are special requirements for touch targets sizes that have to be at least 48 x 48 dp or vary 7 - 10 mm in physical size, regardless of screen size. In most cases touch targets should be separated by 8 dp of space or more if possible. The pointer targets, that apply to the use of motion-tracking pointer devices should be at least 44 x 44 dp.

The set of official recommendations [10] contains information on visible and non-visible text presentation, selection of names for design elements, application of hint speech that may provide extra information for non-clear actions, application of touch exploration, vise motions usage and etc.

The final design should be tested with platform accessibility settings turned on both during and after implementation.

Basic privacy and security concepts. Users should be always in control of their data and make respected decisions about which information is collected and further used. According to android official developer website [11], the security architecture of android OS divides the permission systems into several level:

1. *Normal* - requested by the app permissions which are posing no threat to privacy. For instance, setting time zone. When a normal permission is requested by an app it is automatically granted by the Android device.

2. *Dangerous* - permissions which can pose a threat on privacy. For example, access to the device storage. User see the dialog box which is prompting the user to allow or deny the permission. But if a user agrees to give access, then app can access other dangerous permission without the consent of the user.

3. *Signature level* - all application packages (.apk) files must be signed with a certificate whose private key is held only by than App developer. This certificate helps the android OS to identify the app author.

4. *User ID and File Access* - when android OS is being installed into a new phone, Linux gives a unique user ID to each package. Any data stored by an App is given the unique user ID and cannot be accessed by other packages.

According to Android developers all the dangerous permissions are placed into a special group called permission group (Table 9.1). The user is only notified when the app requests dangerous permissions.

Table 9.1 - Dangerous Permission Groups

Permission Group	Permission
Calendar	Read, write
Camera	Camera
Contacts	Read, write, get account
Location	Access fine location, access coarse location
Microphone	Record audio
Phone	Read phone state, call, read call log, write call log, add voice mail, process outgoing calls
SMS	Send, receive, read, receive wap push, receive MMS
Storage	Read, write
Sensors	Body sensors

Linux kernel is a foundation of Android platform that provide several security features including a user-based permission model, process isolation, extensible mechanism for secure IPC, ability to remove unnecessary and potentially insecure parts of the kernel.

The fundamental security objective is to isolate user resources from one another. By doing this Android prevents user A from reading files of user B, ensure that user A does not exhaust user B’s memory, resources and devices. The Android platform identifies and isolates app resources [12], which means an isolating apps from each other and protecting them and system from malicious apps. For this purpose the unique user ID (UID) is assigned to each application. The UID is further applied to set a kernel-level Application Sandbox.

Android provides a set of cryptographic APIs, which include implementations of standard and commonly used cryptographic primitives such as AES, RSA, DSA and SHA. APIs are also provided for higher level protocols such as SSL and HTTPS [12].

The ability to modify an Android device the users own is an important for developers, thus on many Android devices users have the ability to unlock the bootloader to allow installation of an alternate OS. These systems may allow to gain root access for debugging applications and system components.

Android 7.0 and later supports file-based encryption that allows different files to be encrypted with different keys that can be unlocked independently. Devices that support file-based encryption can also support Direct Boot, which allows encrypted devices to boot straight to the lock screen [12].

Android can be configured to verify a user-supplied password prior to providing access to a device. In addition to preventing unauthorized use of the device, this password protects the cryptographic key for full file system encryption [12].

9.3.2 Usability, security and privacy concepts for iOS

Basic usability concepts. The official human interface guidelines [13] contain wide amount of recommendations about every bar, view and control design to meet the best practices of user experience for each application.

As for accessibility concept iOS offers a variety of special features for users with vision or hearing loss and other disabilities. The official recommendations states that design should contain alternative text labels for images, icons and interface elements that can be audibly described with VoiceOver. If the application uses UIKit all interface elements can be automatically adapted to certain accessibility preferences. Thus, in case of implementing custom elements they should attempt to match the accessibility behavior. As in Android case Apple provides recommendations to use sufficient color contrast ratio. Every app should be tested with accessibility features.

The application should be as much interactive as possible, which means that even while loading some new element user should see an activity of an app.

The Apple recommend to use modality when it is critical to get user attention and prevent people from doing other things until they complete a task or dismiss a message or view. While being simple, short and narrowly focused a modal view can occupy the entire screen or a portion of the screen [13].

An app navigation should support the structure and purpose of an app. In iOS there are three main style navigations: hierarchical, flat and content-driven. The first one gives user ability to make one choice per screen until the destination is reached. The flat navigation style let user switch between multiple content categories. The last style let user move freely through content. The Apple recommends to use standard navigation components, touch gestures to create fluidity, tab bar to present peer categories of content, navigation bar to traverse a hierarchy of data etc [13].

Users want to be able to use applications on different devices, thus apps should be adaptable for various screen sizes [13]. To do so the Apple officially recommends to use Auto Layout which was developed for construction of adaptive interfaces. It can help to adapt: different device sizes and resolutions; different device orientations; split view; multitasking; dynamic type text-size changes; system feature availability and more.

For more information of recommendations on user interaction, system capabilities, visual design, icons and images, bars, views, controls and extensions please refer to official documentation [13].

Basic security and privacy concepts. iOS system security is designed so the software and hardware are secured across all core components of every device, which includes the boot-up process, software updates and Secure Enclave. The brief description for each component that build the general architecture of iOS security system are given further.

The bootloaders, kernel and it’s extensions, baseband firmware, which participate in startup process are cryptographically signed by Apple. This chain helps to ensure that even the lowest software levels are not tampered with. At the beginning the device application processor executes the immutable code, that was laid down during chip fabrication, from read-only memory (Boot ROM). The Boot ROM contains the Apple Root CA public key that is used to verify the iBoot bootloader. The iBoot verifies and runs the iOS kernel as soon as it finishes its tasks [14].

The Boot Progress Register (BPR) is used by Secure Enclave to limit access to user data for various modes.

Secure Enclave is a coprocessor that uses encrypted memory and includes a hardware random number generator. It provides all cryptographic operations for Data Protection key management and maintains the integrity of Data Protection. Communication between the Secure Enclave and the application processor is isolated. The Secure Enclave includes a dedicated Secure Enclave Boot ROM, which is also immutable [14].

The exploitation of memory corruption bugs is hold by pointer authentication codes (PACs). System software and built-in apps use PAC to prevent modification of function pointers and return addresses (code pointers). Doing so increases the difficulty of many attacks [14].

Security of user authentication process are granted by Touch ID, Face ID and passwords use. The first one is the fingerprint sensing system, that reads data from any angle and learns more about user fingerprint over time. Face ID uses TrueDepth camera system that maps geometry of user face to unlock the device. It uses neural networks for determining attention, matching, anti-spoofing and automatically adapts to changes in user appearance, and safeguards the privacy and security of biometric data. It should be noticed that Touch ID and Face ID don’t replace passcode but provide easy access to unlock device within thoughtful boundaries and time constraints. User can enter passcode anytime instead of Touch ID or Face ID, but the following operations always require a passcode instead of a biometric:

- updating software;
- erasing device;
- viewing or changing passcode settings.
- installing iOS configuration profiles.
- the passcode is also required if the device is in the following states: the device has just been turned on or restarted; the device hasn’t

been unlocked for more than 48 hours; the passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and a biometric hasn't unlocked the device in the last 4 hours; the device has received a remote lock command; after five unsuccessful biometric match attempts; after initiating power off/Emergency SOS [14].

Additionally, to the hardware encryption features Apple uses a Data Protection to further protect data stored in flash memory on the device. It allows the device to respond to common events such as incoming phone calls, but also enables a high level of encryption for user data. Data Protection is implemented by constructing and managing a hierarchy of keys, and builds on the hardware encryption technologies built into each iOS device.

9.4 IoT wearable systems

Wearable devices have managed to garner a position of significance in the consumer electronics market in a short time, and are considered a new means of addressing the needs of many industries.

There are six basic types of wearable devices:

- Smartwatches are computerized devices intended to be worn on the wrist, and have expanded functionality that is often related to communication. Most current smartwatch models are based on a mobile operating system. Manufacturers continue to develop their products and add waterproof frames, global positioning system (GPS) navigation systems, and fitness/health tracking features etc. With the addition of reliable, sensitive inertial sensors on them, smartwatches can now be used to capture and analyze hand gestures, such as smoking or other activities.
- smart eyewear are used for various purposes in optical head-mounted displays (OHMDs), heads-up displays (HUDs), Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), and smart contact lenses.
- fitness tracker are designed to monitor and track outdoor sport activities and measure fitness-related metrics, such as the speed and distance of running, exhalation, pulse rate, and sleeping habits.
- smart clothing can monitor the physical condition of the wearer, they include a broad list of wearables, ranging from sportswear and consumer sports apparel (smart shirts and body suits) to chest

straps, medical apparel, work wear monitoring apparel, military apparel, and e-textiles.

- wearable camera is mobile and flexible type of cameras, which can be splitted into two major groups: small cameras that can be attached to either the body or clothes, or can even be worn in the ear, and larger cameras with mounting attachments to affix to caps or helmets.
- wearable medical device typically consists of one or more biosensors used to monitor a variety of physiological data to prevent disease, provide early diagnoses, and facilitate treatment and home rehabilitation.

The properties, capabilities and applications area are presented in Table 9.2.

Table 9.2 – Wearable Devices Types, Properties, Capabilities and Applications

Type	**Properties**	**Capabilities**	**Applications**
Smartwatch	low operating power; user-friendly interface with both touch and voice commands	Displays specific information; payment; fitness/activity tracking; communication; navigation	Business, administration; marketing, insurance; professional sport and training; education
Smart eyewear	Controlled by touching the screen, head movement. Voice command and hand shake	Visualization; language interoperation; communication; task coordination	Surgery; aerospace and defense; logistics; education
Fitness tracker	High accuracy; light weighted; wireless communication	Physiological wellness; navigation; fitness/activity	Fitness; healthcare; professional sport;

		tracking; heart rate monitor	outdoor/indoor sport
Smart clothing	No visual interaction with user via display or screen; data are obtained by body sensors and actuators	Heart rate, daily activities, temperature, body position tracking; automatically heating or cooling the body	Professional; sport-fitness; medicine; military; logistics
Wearable camera	Making first-person capture attachable on clothes or body; smaller dimension; night vision	Captures real-time first-person photos and videos; live streaming; fitness/activity tracking	Defense; fitness; industry; education
Wearable medical device	Pain management; psychological tracking; glucose monitoring; sleep monitoring; brain activity monitoring	Cardiovascular diseases; psychological disorders; chronic diseases; diabetes; surgery; neuroscience; rehabilitation	Fitness; cardiovascular medicine; psychiatry; surgery; oncology; dermatology

9.4.1 Basics of wearable development for Android

Wear OS is a version of Google’s Android operation system designed for smartwatches and other wearable devices. It integrates Google Assistant technology and mobile notifications into smartwatches. Wear OS support Bluetooth, Wi-Fi, 3G and LTE connectivity, as well as a range of features and applications. The hardware manufacturing partners include Asus, Broadcom, Fossil, HTC, Intel, LG, Imagination Technologies, Motorola, New Balance, Qualcomm, Samsung, Huawei, Polar and TAG Heuer [15].

When it comes to creating an application developer need to know that wearable apps are similar to the apps that use the Android SDK, but differ in design and functionality. The basic functionality differences are: watch apps use watch-specific APIs, where applicable (for circular layout, wearable drawers, ambient mode, etc.); watch apps can access many standard Android APIs, but don't android.webkit, android.print, android.app.backup, android.appwidget and android.hardware.usb. The hasSystemFeature() can be applied to detect if the needed feature is supported by watch.

To meet good user experience results Google has presented few recommendations for design of applications that run on Wear OS:

1. The Wear OS is designed around two main actions – suggest and demand. The first one involves making useful suggestions to user and demand stands for taking commands from user.

2. The interfaces must be easy to read so to present needed information the clear information hierarchy is important.

3. All tap targets have to be well-spaces and easy to tap. Avoid relying on a large amount of user input to use the app.

4. All tasks have to be accomplished quickly, thus all user flows must be clear and simple.

Developer can face several limitations, while creating design for smartwatches, such as smaller screen size, less information density and limited battery life. Thus, all applied use cases need to make sense for the watch environment. The Wear OS apps should be designed to support only basic app core functionality. The unnecessary features and actions are not appropriate for the smartwatches apps. Each Wear OS app have to be tested for various device sized, for this purpose the preview tool can be applied.

As it was previously mentioned Wear OS works by communicating wirelessly over Bluetooth between the wearable and a device running Android. When the Android device has been paired with the wearable one, the Wear OS start sending a notification messages automatically to the watch, along with any wearable-specific rich notification parameters. When a connection has been created between the Android device and the wearable, notification messages can be then exchanged between the handheld device and the wearable to trigger appropriate actions on each device. The Wear OS runs a Bluetooth Low Energy (BLE) device, thus all applications have to run efficiently so they will make no bad impact on device's battery. The interaction between handled device and wearable is presented on Figure 9.6 [16].

Once the connection is created, the synching data can then start looking at sending between the two devices. While devices are paired each node can handle any given number of various functions. For instance, one node handles the camera on the mobile, while another one keep tracking a user's GPS coordinates on the wearable.

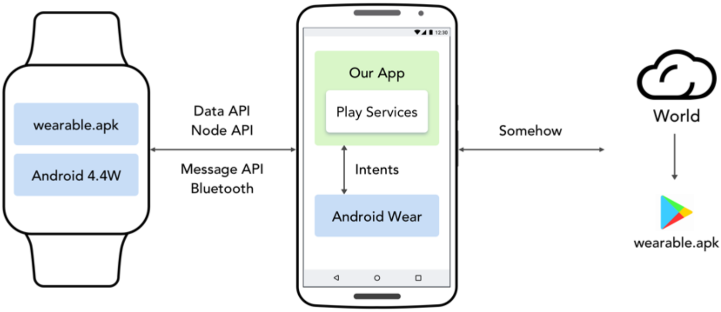


Figure 9.6 – Interaction between handled and wearable devices

The Figure 9.6 presents following APIs:

1. The NodeApi class is responsible for tracking all connected or disconnected nodes that have been established within the wearable network. The NodeListener interface method is applied for this purpose. When a node created a connection between the handheld and the wearable, MessageApi begins to send a message from the wearable device to the handheld device that it is paired with. This sends a notification to the NodeListener method that then start to get data about each node.

2. The MessageApi class is responsible for sending across short messages to each of the connected network nodes between the

wearable and the handheld device. Once a message has been received, MessageListener will be called so that it can get the message. The MessageListener is a background listener service on the receiving side.

3. The DataApi class is responsible for synching data between the connected wearable and the handheld device. It provides the synching mechanism on both sides. In addition to synching data, the DataApi automatically transfer data to the paired smartphone in case if connection was lost and then restored.

Basic steps for setting up environment and starting, developing and testing Wear OS project can be found in official documentation [15].

9.4.2 Basics of wearable development for iOS

Let us describe the main concepts of iOS wearables development based on the creating applications for Apple Watch, as this device known to be one of the most sailed wearable Apple product.

The Apple gives official recommendations on basic principles that have to be applied while creating Apple Watch app [13]:

1. The app’s interface has to be highly glanceable, so users can quickly and easily find necessary information.

2. Keeping the app’s snapshot up to date. As the app’s snapshot is displayed in the Dock, thus it should contain the most recent, relevant and actionable information. The alerts or error states must not be shown in snapshots.

3. The watch app should present only important information and facilitate interactions with that information.

4. The advantages of Handoff should be used to forward tasks back to iOS and macOS devices as needed [13].

5. The app should support as many complication families as possible. The complications display timely and relevant information about the app on the watch face. There are ten types of complications: circular; modular small and large; utility small and large; extra-large; graphic corner, circular, bezel and rectangular [13].

6. The modal sheets have to be used only to facilitate some critical tasks. A custom modal sheet is a full-screen view that slides over the app’s current screen. The modal sheets should never be used

to navigate the app's content. The modal sheets must not replace alerts and action sheets.

7. Avoid creating hierarchies deeper that 2-3 levels for navigation in the app.

A Watch application cannot be installed on the Apple Watch on its own; it always has to be paired via Wi-Fi or Bluetooth to the corresponding iPhone application. The communication and connection sessions are established with the Watch Connectivity framework. iOS application along with Watch application and the WatchKit Extension are building the Watch architecture (Figure 9.7).

The iOS application installed on the user's iPhone is called the parent application. iOS application installed on the iPhone that has a Watch application, will automatically be installed on the Apple Watch as well. Various Watch app processes are often delegated over to the parent app. The iPhone app shares data with the Watch application through the WatchKit Extension. By creating a WCSession utilizing the Watch Connectivity framework. The apps can create a connected for transferring data as user settings and application states.

The Watch App is a bundle of resources that resides on the parent application and are installed on the Apple Watch. These resources include the application storyboard, image and audio files and any localization files used for the Watch application. Within the Watch App bundle resides the WatchKit Extension.

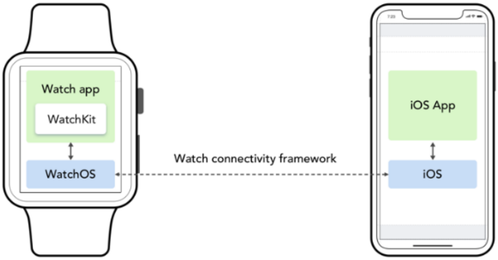


Figure 9.7 – The Watch application architecture

The WatchKit Extension is the area of the application hierarchy that connects the Watch and the parent applications together. The extension contains the code written for the Watch application and uses the WatchKit framework to manipulate the interface of a Watch application. It uses the classes of the framework to configure the Watch App's elements and create responses to the user interactions.

WatchKit is the framework for creating applications for the watchOS. WatchKit is watchOS's equivalent to the iOS's UIKit framework, as they both are used to create interfaces and interaction for the application [17]. A Watch application includes one or more interfaces which contain buttons, sliders, tables and many other visual elements. WatchKit uses the classes of the framework to configure these elements and to establish connection to the user interactions. The WatchKit framework provides a WKExtensionDelegate protocol which is a collection of methods that can be implemented to manage app-level behavior of the WatchKit Extension. The protocol mainly handles application states, snapshots and background tasks [17].

WatchKit notifies changes in the application's execution state to the extension delegate object. The state changes are triggered by major events in the lifetime of the application: not running, inactive, active, background, suspended.

The WKExtensionDelegate can respond to state transition events by implementing the delegate's application lifecycle methods. These methods can, for example, be used to preload resources, configure initial user interfaces and save application states and user data [17].

The WatchKit calls the applicationDidFinishLaunching(_:) method after finishing to launch the application, before the app's interface becomes visible. The method can be used to setup configurations and to initialize which view controller will be shown to the user at launch. The applicationDidBecomeActive() method tells the delegate that the Watch app is now visible and the processes can be continued, after the applicationWillResignActive() will signal the opposite. These methods are for pausing and resuming tasks. The applicationWillEnterBackground() method will tell the delegate that the application is about to enter the background. The method can be used to store the application state and information to ensure that the right state is loaded after the application re-enters the foreground. If the app is entering foreground after being backgrounded the applicationWillEnterForeground() method is called. The method is used to reload the state which the applicationWillEnterBackgroud() has stored [17].

Background tasks give the application a small amount of time to run in the background. Several different handler types are provided for the background tasks, each for a specific type of activity. Some of these tasks can be manually scheduled by the application, while some are automatically scheduled by the system [17]. Each background task is processed by the delegate's handle(_:) method, which determines the type of the task.

Snapshots are glimpses of the application state that are displayed in the dock when the watch's side button is pressed. The system keeps the most recently used apps in the dock so that they can be resumed quickly. These apps receive priority for background tasks. The dock can hold up to 10 applications at the same time [17]. Pressing the watch's side button behaves similarly as double-pressing the home button on an iPhone.

Watch Connectivity is a framework for establishing a two-way communication between the apps. The communication is used for data sharing. The framework is an essential part for making a Watch app and can also be used to update the watch's complications. To initiate communication between the applications, both the Watch and the iOS apps have to establish communication with WCSession class. Both apps have to setup their own sessions at some point of their execution. To configure the session, a delegate has to be assigned to the default WCSession object, after which the session's activate method has to be called to make the connection. [17].

Before transferring data, it is recommended to check whether the data can reach its destination. For this purpose the is Paired method is applied to check whether the Apple Watch is paired to an iPhone and vice versa. On the iPhone, the is WatchAppInstalled method can be called to check whether the Watch App has been installed on the paired device. For instant messaging, the session's is Reachable method must return true, or else the messages cannot reach the counterpart device. More detailed information is presented on the official Apple documentation page [17].

9.5 Publication of applications to the App Store and Play Market

The Apple's App Store is an ecosystem for millions of developers and mare than a billion of users. Once the application was developed it have to be submitted to the App Store. This process can be a challenging one as according to Apple 62% of common app rejections occur due to non-compliance with the main Apple guideline.

There are 8 main steps required to successfully publish, your iOS app in Apple's App Store:

1. Check on the compliance to Apple's App Store Guidelines. Apple wants user to feel confident that it's safe to install new app from App store. One of the biggest reasons for rejection is objectionable content, which means that apps should not include content that is offensive, insensitive, upsetting, intended to disgust, or in exceptionally poor taste. Any kind of racist, sexist or homophobic references, comments inciting religious intolerance, erroneous and false information can be treated as objectionable content.

2. Testing app to ensure there are no bugs or crashes. There are number of iOS devices – iPhones, iPads and iPods all come in different sizes, thus the more devices were tested with an app, the better the chance of App Store approval. It is also important to take care of user experience, because the negative opinions may cause app to fail on App Store.

3. Registering the account for an Apple developer program. This program allows to use additional Apple tools, see apps analytics, perform beta testing, and more. Registration in Apple's program is not free, and the payment is based on a yearly subscription. Private individuals or legal entities with one employee can enroll for $99 per year. For development teams the membership will cost $299 per year.

4. Creating an iTunes Connect app record. iTunes Connect is a suite of web-based tools for managing apps sold on the App Store for iPhone, iPad, Mac, Apple Watch, Apple TV, and iMessage. It's also used to manage content on the iTunes Store. As a member of the Apple Developer Program, developer can use iTunes Connect to submit and manage apps. The record has to be created in iTunes Connect before uploading app for potential distribution through the App Store. This

record includes all the information that is needed to manage the app through the distribution process and that appears in the App Store.

5. Configuring an app for distribution with the appropriate information. Before an app can be published, developer needs a collection of information to complete the process: an icon; a screenshot/app preview; metadata. Apple provides a set of Human Interface Guidelines that can help to size an icon properly. The screenshots and app previews are applied for visual communication to the user experience. The previews can either be images or a short video captured from an app that will be displayed on App Store product page. An app should have a detailed description with all necessary metadata. The metadata should include the following: the app name, the version number, the app category, a detailed description, any additional keywords. The final step is to create an app archive. The archive allows to build app and store it, along with critical debugging information, in a bundle that’s managed by the upload platform. Before the uploading an app to iTunes Connect, the standard iTunes Connect validation checks needed to be run on the archive to determine whether it meets minimum App Store requirements and ensure that it passes.

6. Uploading app. Once all of the necessary app details have been entered in iTunes Connect, developer can upload a build of an app using a platform such as Xcode or Application Loader. After the creation of an iTunes Connect record for the app, the upload is displayed on the Activity section of “My Apps.” The app status should be set to “Prepare for Submission.” Selecting the app on this page allows to view and edit app information. To upload an app to iTunes Connect, in the Archives organizer developer need to select the archive fro upload and click “Upload to App Store.” Provisioning profiles, or digital entities that uniquely tie developers and devices to an authorized iPhone Development Team and enables a device to be used for testing, are packaged with iOS apps so user devices can install them. If everything was done according to specifications, the app code will be sent to Apple’s servers. The app then will be verified for validation; if something goes wrong, a submission failure error will occur. Before releasing the app on the App Store, the developer may use Apple’s TestFlight to distribute the beta version (or app updates) to testers, who will provide valuable feedback.

7. Submitting last version for official review. Every app that is submitted to the store has to be reviewed by the Apple team before release. The app needs to comply with all specified guidelines in order to be approved. After initial verification, developer will see that the status changed to Waiting for Review. App review can take 1–3 days. If, however, the reviewer rejects an app, developer can communicate with Apple and resolve issues in the Resolution Center. A communication from Apple contains vital information about the reasons for an app's rejection, such as if the app is out of compliance with App Store Review Guidelines. Developer can correspond with Apple through the Resolution Center until resubmission the build to App Review.

8. Release. Once an app has been reviewed and approved, developer can request either Manual or Automatic release of the approved app. The automatic release is typically a phased release, sending iOS app in stages. This option is available for the submit an iOS version update and app has one of the following statuses: prepare for submission, waiting for review, in review, waiting for export compliance, pending developer release, developer rejected, rejected, metadata rejected. For this option app version update will be released over a 7-day period to a percentage of app users (selected at random by their Apple ID) on iOS with automatic updates turned on. Users are not notified that that they are in a phased release of an app. If the manual release was selected, the option "Release your app" will be applied when an app is ready for distribution to the potential users. The manual release may take from 1 hour to 1 day after release.

Let's discuss the process of app publication through the Google Play - one of the largest platforms for distributing, promoting, and selling Android apps. There are eight basic stages that have to be passed to publish new application on the Google Play:

1. Create a developer account. The account can be created even using an existing Google Account. The one-time registration fee is $25, and payment can be made after accepting the Developer Distribution Agreement. The registration can take up to 48 hours to fully proceed.

2. Linking the merchant account. If the developer is planning to publish a paid app or with in-app purchased the merchant account has to be created. This can be made through Play Console. Once the

account is created, it will be automatically linked to the basic developer account. A merchant account will let developer to manage an app sales and monthly payouts, as well as analyze sales reports right in Play Console.

3. Adding new application. This can be also made through Play Console. As soon as the application was added, developer will be taken to the store entry page to fill out all the details for app's store listing.

4. Prepare store listing. Before the application can be published the developer needs to prepare its store listing. These are all the details that will show up to customers on an app's listing on Google Play. The information required for store listing is divided into several categories: product details (title, short description, full description), graphic assets (screenshots, images, videos, promotional graphics and icons), language and translations, categorization (type and category of an app), contact details (email, website, phone number), privacy policy (URL linking to the privacy policy in store listing and within an app)

5. Upload APK to an app release. The Android Package Kit (APK) is the file format used by the Android OS to distribute and install apps. Google offers multiple ways to upload and release the APK. The app release has to be created before the file upload.

6. Providing an appropriate content rating. The 'unrated' apps may get removed from Google Play. To rate the app, the content rating questionnaire has to be filled out. The presented information has to be accurate, because in case of misrepresentation of the app's content can lead to suspension or removal from the Play Store. An appropriate content rating will also help to get to the right audience, which will improve the engagement rates.

7. Set up pricing and distribution. This stage requires the determination of an app monetization strategy. It is important to note that app can be always changed from paid to free, but it cannot be changes from free to paid.

8. Rollout release and app publication. Before reviewing and rolling out the release, the store listing, content rating, and pricing and distribution sections of an app each have a green check mark next to them.

9.6 Work related analysis

The role of standards in such emerging technologies as mobile and IoT is well recognized as one of the key enablers to deploy reliable solutions to ambitious ideas. However, due to continuous growth of this area and increasing amount of ways IoT can be applied to meet different user needs there are exists a problem of duplication, fragmentation and competition between standards organization. Thus unique solution is hardly can be reached in nowadays situation. This question was stated in work [2] and widely examined by Emmanuel Darmois, Laura Daniele and other authors in Chapter 6 "IoT Standards Landscape – State of the Art Analysis and Evolution" in [1]. The new trends in IoT standardization that address the emerging topics such as identification and addressing, semantic interoperability and concepts of security and privacy are analyzed in [1, 4].

The wireless mobile industry shows a constant growth through last decades which poses a permanent standards development. Higher data transmission speeds and efficient communications enabled by mobile technologies have unleashed a range of new mobile data services, for instance applications and streaming videos, and complex products, such as smartphones and tablets. While recognizing benefits of standardization a growing number of regulators are expressing concerns about the role of intellectual property rights in facilitation the commercialization of standardized technologies. This question is deeply analyzed in [5]. This work also provides wide overview on application of various standards in the mobile wireless industry during different stages of new technology development.

This section provides the basic of developing applications for two main mobile operation systems – Android and iOS. Taking into account the constant development of these OS and ongoing changes in programming languages for these systems, Section 9 covers main topics on recommendations for creating an app design, building app architecture and securing user data. The [6] gives a good introduction to core concepts of Android development and can help in mastering new skills. The official Google documentation [10] can help both newbie to get first practices in applying the material design ideas and experienced developer to follow hot design topics. The [7, 8, 11, 12] give the basic knowledge on existing Android application sandbox and recommended architecture, permissions and security concepts overview. Apple provides the detailed human interface guidelines [13] which covers topics of interface essentials, application architecture [9], user interaction, system capabilities, visual design, icons and images, bars, views, controls and elements. The latest updates in iOS 12.1 security were published by Apple on November 2018 and can be found on [14].

Wearables are one of the most widely used IoT devices in everyday life and smartwatches is a bright example of those devices. Both Google and Apple have presented their concepts for designing and developing applications that run on small screens of smartwatches. The application store shows a fast growth of apps for active life tracking, healthcare etc. In 2014 Google has presented the Wear OS [15, 16] for developing applications on smartwatches and the latest release in of this system was published in November 2018. On April 2015 Apple has also presented OS for Apple Watches – watchOS. The latest release of 2018 was presented in December [17].

The following courses and programs have been analyzed to develop lecture material for this module:

- Developing Android Apps with App Inventor (The Hong Kong University of Science and Technology) [18];

- Toward the Future of iOS Development with Swift (University of California, Irvine) [19];

- Mobile App Development with Swift (Curtin University, Australia) [20].

The following MSc and PhD courses and programs of ALIOT project EU universities have been taken into account as well:

– Coimbra University, Portugal: MSc program in Electrical and Computer Engineering [21], Programmable Electronic Devices (PhD Course) [22];

– KTH University, Sweden: MSc programs in Systems, Control and Robotics [23], Hybrid and Embedded Control Systems [24];

– Newcastle University, United Kingdom: MSc program Embedded Systems and Internet of Things (ES-IoT) [25].

Conclusions and questions

Study material presented in this section is covering the basic topics of application of main standards in IoT and wireless mobile industry areas, development of applications for Android and iOS operation systems as well as for wearables that uses Wear OS and watchOS. The strong knowledge of main principles of creating easy understandable and accessible design, which is one of the hot topics in nowadays apps design, can help a developer to meet needs of wider user audience and ensure the application success. Construction of clean app architecture, providing high security and privacy levels for mobile and wearable apps are key concepts in creation of reliable product.

The following questions can help to strengthen and assimilate the educational material that is covered in this section:

1. Define tree levels of standards classification that was developed by AIOTI.
2. Describe the first stage of new technology development – development of technology standards.
3. Describe the second stage of new technology development – development of product firms.
4. Describe the last stage of new technology development-deployment of networks.
5. Describe the main Android architectural principles.
6. Why it is important to drive UI from the model?
7. What type of frameworks gives the Cocoa Touch layer?
8. Describe the content that is presented by Media layer.
9. What type of app architecture Apple officially recommends to use?
10. Define key objects of iOS application.
11. Describe basic user-based interaction types of IoT to mobile applications.
12. Which three basic design concepts were stated by google as core of material design idea?
13. Describe the levels of Android permission system.
14. What type of permissions are dangerous?
15. Describe the main accessibility concepts that were recommended by Apple to use in iOS applications.

16. Define the boot-up process, software updates and Secure Enclave – three core components of iOS security system.
17. In which situations iOS requires to enter password instead of using TouchID or FaceID?
18. Define six types of wearable devices.
19. Describe the tasks of Node API, Message API and Data API.
20. What is WatchKit?
21. Describe the Watch Connectivity framework main functions.
22. Define the steps of publication of application to the App Store.
23. Describe the steps of publication of application to the Play Market.

References

1. Cognitive Hyperconnected Digital Transformation: Internet of Things Intelligence Evolution. – Editor: O. Vermesan, J. Bacquet. – River Publishers, Denmark, 2017. – p. 167 – 189.
2. European Commission communication on "ICT Standardisation Priorities for the Digital Single Market", COM(2016) 176 final. – Brussels, 19.4.2016.
3. AIOTI WG03 Report . IoT LSP Standard Framework Concepts Release 2.7. – 2017. – [electronic source: https://docbox.etsi.org/SmartM2M/Open/AIOTI/].
4. STF 505 TR 103 375 . SmartM2M IoT Standards landscape and future evolution. – 2016 . – [electronic source: https://docbox.etsi.org/SmartM2M/Open/AIOTI/STF505].
5. Gupta K. Technology Standards and Competition in the Mobile Wireless Industry. – Geo. Mason L. Rev. 2015. – p. 865 – 1021.
6. Meier R., Lake I. Professional Android. – John Wiley and Sons, Inc. – 2018, 928 p.
7. Application sandbox. – [electronic source: https://source.android.com/security/app-sandbox].
8. Guide to app architecture. – [electronic source: https://developer.android.com/jetpack/docs/guide].
9. Developer documentation archive. – [electronic source: https://developer.apple.com/library/archive/].

10. Design for Android. – [electronic source: https://developer. android.com/design/].

11. Permissions overview. – [electronic source: https://developer. android.com/guide/topics/permissions/overview/].

12. Security. - [electronic source: https://source.android.com/ security/].

13. Apple human interface guidelines. – [electronic source: https://developer.apple.com/design/human-interface-guidelines/ios/].

14. iOS security guide. – [electronic source: https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf].

15. Wear OS by Google. – [electronic source: https://developer. android.com/wear/].

16. Daniel, S F. Android Wearable Programming. – Packt Publishing. – 2015. – 201p.

17. WatchKit documentation. – [electronic source: https://developer.apple.com/documentation/watchkit].

18. Developing Android Apps with App Inventor. – [electronic source: https://www.coursera.org/learn/app-invenor- android?action=enroll].

19. Toward the Future of iOS Development with Swift.-[electronic source: https://www.coursera.org/learn/iosswift];

20. Mobile App Development with Swift.- [electronic source: https://www.edx.org/professional-certificate/curtinx-mobile-app-development-with-swift].

21. The Master program in Electrical and Computer Engineering [https://www.uc.pt/en/fctuc/deec/courses/mieec]

22. Programmable Electronic Devices [https://www.uc.pt/fctuc/ deec/ PhD_courses/Programmable_Electronic_Devices]

23. Master's programme in Systems, Control and Robotics [https://www.kth.se/en/studies/master/systems-control-robotics]

24. Hybrid and Embedded Control Systems [https://www.kth.se/student/ kurser/kurs/EL2450?l=en]

25. Embedded Systems and Internet of Things (ES-IoT) MSc [https://www.ncl.ac.uk/postgraduate/courses/degrees/embedded-systems-internet-of-things-msc/#profile].

10 CLOUD COMPUTING AND IOT

Dr. V.O. Butenko, D.A. Butenko (KhAI),
Dr. E.B. Odarushchenko (PSAA)

Contents

Abbreviations

API - Application Programming Interface
AWS - Amazon Web Services
DaaS - Data as a Service
DB - Database
DBaaS - Database as a Service
FCM - Firebase Cloud Messaging
FIFO - First In First Out
GAE - Google App Engine
HTTP - Hyper Text Transfer Protocol
HTTPS - Hyper Text Transfer Protocol Secure
IaaS - Infrastructure as a Service
IoT - Internet of Things
NaaS - Network as a Service
NIST - National Institute of Standard and Technologies
NoSQL - Not Only Structed Query Language
OSS - Operations Support System
PaaS - Platform as a Service
SAaaS - Sensing and Actuation as a Service
SaaS - Software as a Service
SDK - Software Development Kit
SenaaS - Sensor as a Service
SQL - Structured Query Language
UI - User Interface

10.1 Introduction to the IoT Cloud Computing

The Cloud computing is a breakthrough technology which made it possible to use the powerful IT capabilities from anywhere if the user has connection to the Internet. It provides range of services to IT capabilities such as software applications, storage, various networks, interfaces, infrastructures etc. According to the definition provided by National Institute of Standard and Technologies (NIST) [1]: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

Appeared in 2006 over the next few years the cloud computing made a huge impact on IT industry. Virtual unlimited storages, new processing capabilities under the fair prices enabled the realization of a new computing model in which the virtual resources are provided as general utilities. The range of new services that are delivered through the Internet by such large companies as Amazon, Google, Facebook gain an economical and technical benefits for both sides – customers and providers.

There are different types of clouds that were identified in [2]:

1. Private Cloud – type that is exclusively used, owned, managed and operated by a single organization for their internal use. This delivery model is appealing is compliance and security are highly important. The main advantage of private cloud is that business takes control of all it services, such as set pricing, policies, control access etc. It should be noted that this model requires additional expenses such as procuring hardware and software for cloud, employ an administrator to manage the services etc. In addition, multiple data centers which implement distributed storages and compute infrastructure can bring the greatest benefits from cloud along with greater capital expenditure.

2. Public Cloud – type with open use by any customer. One of the key advantages of using a public cloud is absence of high capital expenditure. This model uses "pay as you go" monetization scheme, so the customer can purchase computing and storage services as needed. One of the main disadvantages is that business customers are mostly

dependent on viability and reliability of the cloud provider and in case of service outage all services can be inaccessible.

3. Hybrid Cloud – composition of two or more distinct Cloud infrastructures, but mostly it combines public and private types. The business that uses private cloud can use a public one as an extension to the main cloud. There are two ways how this combination can be achieved – as separately managed service platforms and by enabling access to the public cloud from the internal service management platform.

However, there are several technical and business-related questions which are not solved. Those issues are related to security (e.g., data security and integrity, network security), privacy (e.g., data confidentiality) and service-level agreements. While outsourcing infrastructure to a Cloud provider, public Cloud customers are facing the price increases and range of reliability problems [3].

Internet of Things (IoT) is a novel concept in IT and communication sectors. It assists on the creation of relationships between different objects via Internet and keeps the constant communication so all user can get more efficient and intelligent experience. As every new technology the IoT is passing the extremely complex stage of standards creation. Due to various areas in which IoT can be applied and dozens of standards development institutions this stage seems to be one of the hardest, especially when it comes to security and safety issues.

Nevertheless, there can be stated at least seven benefits of merging Cloud with IoT [4]:

1. Affordable. As the Cloud provide the large infrastructure and equipment under the fair prices the customers can decrease their expense in the long run. The payment of Cloud is mainly ‘pay as you go’ plan so the users can avoid to make unnecessary of extra payments.

2. Secure. The data is stored remotely, thus the customer can be safe in case of the different devices loss. The information can be accessed through any device using the keys that are hold by unique client.

3. Efficient growth. The Cloud resources prove to be beneficial. The efficiency of the business increases along with efficiency of employees because they can work not only from office but from anywhere.

4. Elastic and scalable. The bandwidth, storage or any other recourses and be scaled under the present business needs.

5. Avoiding Downtime or Delay. Cloud is a network of connected servers, thus if any node is in the down state its load is distributed by other working nodes.

6. Environmental compatibility. Cloud conserves energy and provides efficient technical solutions to the business by emitting less carbon percentage than efficient use of resources resulting in Green Computing.

7. Incentives to new experiments. For developers, testers, different IT engineers it gives an opportunity to test experiments easily with the cloud. This can be very economical comparing to the testing in real environment.

10.1.1 Cloud computing architecture

The core concept that stands behind Cloud computing idea is to provide everything as a service. Garter defines [10] it as a style of computing in which massively scalable IT-related capabilities are provided "as a service" using Internet technologies to multiple external customers. Cloud computing consists of three main layers or models, "Infrastructure as a Service" (IaaS), "Platform as a Service" (PaaS), and "Software as a Service" (SaaS). There can be also defined two additional levels – Client and Hardware.

The Figure 10.1 shows the different layers of Cloud computing architecture [2].

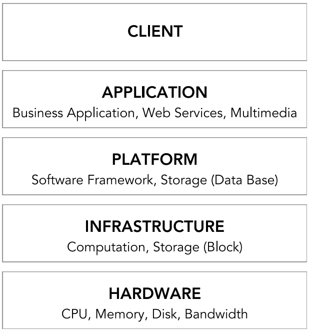

CLIENT
APPLICATION
Business Application, Web Services, Multimedia
PLATFORM
Software Framework, Storage (Data Base)
INFRASTRUCTURE
Computation, Storage (Block)
HARDWARE
CPU, Memory, Disk, Bandwidth

Figure 10.1 – Cloud Computing Architecture

The Cloud client level is a top one. It consists of computer hardware and software that relies on Cloud computing for different applications delivery or initially designed to deliver Cloud service.

The main characteristic of Application level is a network-based access to the available software that is managed from centralized locations and enable customers to access the needed applications remotely through the Internet. This level services “Software as a Service (SaaS)”. The key providers are SalesForce.com, NetSuite, Oracle, IBM and Microsoft.

The Platform level provides a computing platform using the Cloud infrastructure that gives all application typically required by the customer. Developers can get the set of required software for all stages of product creation – development, testing, deploying and finally hosting of web applications. The key examples are Google App Engine (GAE), Microsoft's Azure. This level presents the "Plaform as a Service (PaaS)".

The Infrastructure level provides the required infrastructure as a service. It gives a range of benefits for customers as they don’t have to purchase the required servers, data center or other the network resources. The clients can achieve a mush faster service delivery with low cost as they pay only for the time duration they use service. Examples are GoGrid, Flexiscale, Layered Technologies, Joyent and Mosso/Rackspace. This level presents the "Infrastructure as a Service (IaaS)" idea.

The lowest Server level consist of computer hardware and software which are required for delivery of the mentioned services.

As the integration of IoT concept into Cloud computing ideas is a nowadays reality it gives a birth to Things as a Service paradigm [3]. There is no particular standard which gives a strong definition to the proposed paradigms, thus here we give the original acronyms that were presented in initial papers. Generally, all these models are called XaaS, which means ‘X’ can be virtually anything:

1. *“Sensor-as-a-Service”* (SenaaS) encapsulates both physical and virtual sensors in to services according to Service Oriented Architecture (SOA). This type of service focuses on providing sensor management as a service rather than providing sensor data (collection and dissemination) as a service.

2. *“Sensing-and-Actuation-as-a-Service”* (SAaaS) provides indexing and querying services on sensing and actuation resources, that can be therefore aggregated, i.e. according to predefined thing-like labeling, and provided to final users, developers, providers, etc., as a service.

3. *“Database-as-a-Service”* (DBaaS). Using this service, the higher consolidation and better performance is achieved by the workload-aware approach to multi-tenancy that identifies the workloads that can be co-located on a database server. It uses the graph-based data partitioning scheme to deal with complex transactional workloads. The adjustable security scheme enables SQL queries to run over encrypted data.

4. *“Data-as-a-Service”* (DaaS) can be defined as the sourcing, management, and provision of data delivered in an immediately consumable format to customers as a service.

5. *“Network-as-a-Service”* (NaaS) provides tenants with access to additional computing resources collocated with switches and routers. Tenants can use this model to implement custom forwarding decisions based on application needs, process packets on-path, possibly modifying the payload, create new packets on the fly. Various efficient in-network services can be enabled with NaaS - data aggregation, stream processing, caching and redundancy elimination protocols.

10.1.2 Dynamic interactions

Development of new systems through selection, composition and coordination off-the-shelf components of services is common stages in software production, which are highly illustrated by the development of Cloud-based services. Thus, picking the appropriate communication model (e.g. synchronous or asynchronous, multicast or point to point) has a great impact on the final system properties.

While synchronous communications require some type of explicit or implicit authentication between tasks that are sharing data, the asynchronous type allows tasks to communicate data independently from the work they are doing. In other words, asynchronous type is decoupling send and receive events, thus providing the non-blocking communication.

The asynchronous communication between processes in the cloud is performed via messaging queues. Messages are helpful for constructing workflows, implementing distributed transactions as well as adaptation to the individual components failure within a distributed system. For instance, lets consider a Web interface that accepts user requests is is running unique server. In a tight coupled design this interface passes requests to the backend server and waits for the respond. In the backend server fails the interface will get an error, thus deliver a low user experience. In a loosely coupled design the interface would submit request to the queue and any server can read it, respond and finally remove it from the queue. If a single instance is not responding, the request will be processes by the running servers. If during the processing the backend instance crashes another instance still can read the request and respond to it because the message was not deleted from the queue [2]. This example briefly presents the way messaging queue enables more robust applications design (Figure 10.2).

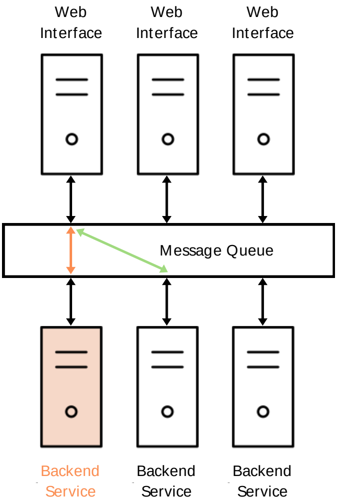


Figure 10.2 – Failure of One Component Accommodated by Other Components

The paper [5] gives a formalization and unified description of seven asynchronous communication models:

1. *Realizable with Synchronous Communication*. The execution can be realized with synchronous communication if every sent event is immediately followed by a received one. If the couple “sent-receive” events is viewed automatically this way can be treated as synchronous communication execution.

2. *First-in-first-out (FIFO) n-n Communication*. In this type of communication all messages are globally ordered and delivered in their emission order. As this model is based on single shared object – unique queue, its implementation on high loads is inefficient and unrealistic. Mostly, this model is treated as first step away from synchronous communication type.

3. *FIFO n-1 Communication*. In this type each peer has a unique input queue and a send event consist of adding message, without blocking it, at the end of the destination peer queue. The message will be removed from the queue according to its insertion order. This model is mainly used as an abstraction of asynchronous communication and its implementation requires a shares real-time clock or a global agreement of event order.

4. *FIFO 1-n Communication*. This model stands for processing messages from the peer in their send order. Being the dual to FIFO n-1 this type is less intuitive as it includes a separate global order on the receivers, one for each sender. Each peer has a unique queue where sent messages are put and destination peers fetch messages from this queue and notify about their reception.

5. *Causally Ordered Communication*. This model delivers messages according to the causality of their emissions, which means that, if message A is causally sent before message B then a peer cannot get B before A. Implementation of this model requires use of causal histories, share of causal relations or application of logical vector/matrix clocks.

6. *FIFO 1-1 Communication.* Messages between a couple of peers are delivered in their send order and messages from (to) different peers are delivered independently. If peer sends a message A and later message B, and if those two messages are consumed by the same peer, the message B cannot be received before A.

7. *Fully Asynchronous Communication.* The is no order on message delivery, thus messages can overtake others of be delayed.

The cloud architectures give an access to the large number of servers, thus bringing an ability to speed up the time for complex operations with parallel, distributed processing. A *map-reducing paradigm* is a suitable for cloud method that implement parallel application.

The map-reduce was originally created be Google for application development on data-centers with thousands of nodes. Nowadays it is widely used to solve a range of issues, such as:

1. Large-scale machine learning problems;
2. Clustering problems;
3. Extracting data to produce reports of popular queries;
4. Extracting properties of web-pages;
5. Processing a satellite data;
6. Language model processing;
7. Large-scale graph computations.

The basic idea that stands behind map-reduce mechanism is that some problems are inherently parallel – some steps in the computations can be done independently of other steps and result of individual computations can be further combined into one final result. The general problem can be disaggregated into number of steps followed by an aggregation process.

The map-reduce programming model can be summarized as follows (Figure 10.3). The computation takes a *{key_in, value_in}* pairs as an input set and produces *{key_out, value_out}* pairs on output. The use of map-reduce library expresses a sequential use of two functions *map* and *reduce*. The *map* function takes {*key_in, value_in*} pairs and produces a set of intermediate {*key_int, value_int*} pairs. Afterwards, the map-reduce library groups all {*key_int, value_int*} pairs associated with intermediate key and passes them to *reduce* function. The reduce merges the tuples with a same key trying to form as small output data set as possible – {*key_out, value_out*}.

Take for example, the click-stream data analysis problem. The click-stream data from Web site contains data on customer's interests, reviews on what that have read, navigation path, points where they have added a product to the card, etc. As one customer activity is independent from the others – this example is a good case for parallel processing analysis [2].

A map-reduce approach can be defined as follows:

1. Splitting the set of all clicks data be customer session;

2. Partition the customer session across *n* instances of the analytics program;

3. Scanning the click stream of each customer for the number of times the m-page sequence occurred;

4. Simplifying the pattern by searching through page types. This is a final step of the *map* function phase.

5. Combining results of each *map* phase to produce the aggregated umber of times each pattern occurred. This is a *reduce* function phase.

Using this approach, the large volumes of click stream data can be analyzed much faster in parallel than in a sequential manner, which creates a base for analyzing greater amount of data and forming in-depth analysis of customer interaction behavior.

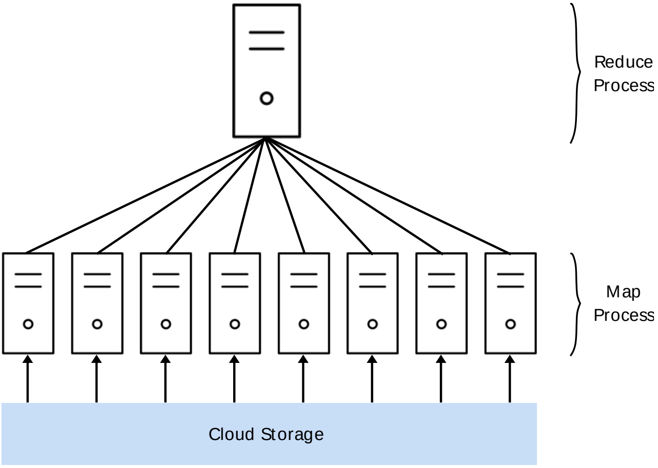


Figure 10.3 – “Map-reduce” Process

10.2 Economics of Cloud Computing

Cloud is the architecture and a set of services that enable access resources on demand. It is a dynamically scalable, on-demand, multi-tenant and often virtualized resources which are provided as a self-service over the Internet/Intranet; Public, Private and Hybrid Models. Cloud gives you the flexibility to handle fast paced customer requirements and also provide a reliable solution for your applications, which can have an option to scale incrementally without having a downtime. However, one needs to have a clear understanding on what specific outcomes are desired before considering the cloud platform.

Cloud computing enables a shift away from computing as a bundled hardware and software product that is bought through fixed capital investments to computing as a location independent and highly scalable service which can be bought on-demand over broadband networks from large-scale computing centers or "clouds" on a pay-per-use basis with little or flexible capital investment. The cloud approach leads to cost reductions and efficiency gains through economies of scale, distribution of costs over large pools of users, centralization of infrastructures in areas with lower costs, and improved resource utilization rates. These efficiency improvements allow large savings in operational costs significant reductions in the upfront capital costs required for new tech startups. As a result, many observers have characterized cloud computing as a disruptive general purpose technology with potential for enormous impacts on the economy as a whole.

Although relatively new, cloud computing is already a very significant part of the technology sector. A recent report by IT research and advisory firm Gartner reports The worldwide public cloud services market is projected to grow 17.3 percent in 2019 to total $206.2 billion, up from $175.8 billion in 2018, according to Gartner, Inc. In 2018, Gartner forecasts that the market will grow 21 percent, up from $145.3 billion in 2017 (Table 10.1).

The fastest-growing segment of the market is cloud system infrastructure services (*IaaS*), which is forecast to grow 27.6 percent in 2019 to reach $39.5 billion, up from $31 billion in 2018

By 2022, Gartner expects that 90 percent of organizations purchasing public cloud IaaS will do so from an integrated IaaS and platform as a service (*PaaS*) provider, and will use both the IaaS and PaaS capabilities from that provider.

Sid Nag, research director of Gartner mentioned that demand for integrated IaaS and PaaS offerings is driving the next wave of cloud infrastructure adoption. And we can expect that IaaS-only cloud providers will continue to exist in the future, but only as niche players, as organizations will demand offerings with more breadth and depth for their hybrid environments. Already, strategic initiatives such as digital transformation projects resulting in the adoption of multicloud and hybrid cloud fuel the growth of the IaaS market.

Table 10. 1 - Worldwide Public Cloud Service Revenue Forecast (Billions of U.S. Dollars)

	2017	**2018**	**2019**	**2020**	**2021**
Cloud Business Process Services (BPaaS)	42.2	46.6	50.3	54.1	58.1
Cloud Application Infrastructure Services (PaaS)	11.9	15.2	18.8	23.0	27.7
Cloud Application Services (SaaS)	58.8	72.2	85.1	98.9	113.1
Cloud Management and Security Services	8.7	10.7	12.5	14.4	16.3
Cloud System Infrastructure Services (IaaS)	23.6	31.0	39.5	49.9	63.0
Total Market	**145.3**	**175.8**	**206.2**	**240.3**	**278.3**

In enterprise computing, pay for use has increasingly gained acceptance, as IT strives to lower costs across infrastructures, applications, and services and pushes back the IT costs to the consumer. Nowadays, with cloud computing, pay for use has become both necessary in a shared environment and easier to implement.

Price wars have gotten a lot of press, with providers lowering prices dozens of times over the last few years, due to reductions in cost structure, but also in an attempt to gain share, receive publicity, and mark their place in the market to show it to competitors [6]. However, although cloud prices are of interest, so are cloud pricing models, such as spot instances, reserved instances and sustained-use pricing. It would be a mistake to consider pricing models after as they are a means of competitive differentiation as well as for creating value for both customers and providers [7].

There are a whole lot of reasons for opting for using cloud computing. The initial model for cloud pricing was pay per use, in which the price was proportional to the product or the quantity of resources and time allocated, for example, 2 medium instances * $0.10 per hour/instance * 3 hours. Since then, various providers have introduced other pricing models. ProfitBricks introduced per-minute billing. Google introduced sustained-use pricing, whereby customers with more consistent use pay a lower unit cost than those with spiky demand, with such determination made after the fact. Amazon Web Services (AWS) introduced reserved instances, which sound like reservations as for a hotel room, but are more akin to a discount for a volume commitment within a given time period, without implying pre-reservation of timeslots [7].

Dynamic pricing is a logical implication of clouds' perishable capacity. In 2009, AWS also introduced spot instances, which, among other things, brought dynamic pricing to the cloud. Such dynamic pricing, where the offering price varies over time, is well known in many industries, such as commodities, hotel rooms, airline tickets, and e-commerce. Amazon.com (the retailer, not the cloud) reportedly changes prices millions of times each day. Firms do such things for many reasons, including yield management of perishable resources such as airline seats, hotel rooms, and computing resources; response to competitor pricing moves; and A/B demand testing. Yield management might be disguised so that price shifts do not match actual momentary capacity, while still promoting higher overall utilisation. For example, Orna Agmon Ben-Yehuda and his colleagues claim that

AWS prices reflect a “random reserve price that is not driven by supply and demand.” Because AWS spot instances can be terminated at any time, the dynamic pricing at AWS is not exactly like it is at Amazon.com, where delivery is still assured if the purchase is made at the agreed-upon price. Moreover, there is no predefined purchase price, but rather a bid representing an agreed-upon price limit, thus providing attributes of an auction as well: customers willing to pay more are more likely to have their workloads run. Perhaps counterintuitively, both providers and customers benefit from this multiplexing of different classes of service. Customers can save money by running their workloads at off-peak times, and providers can manage yields of perishable cloud resources by lowering the price to promote demand and drive better utilisation, reducing idle capital. In the future, there is no doubt some enterprising cloud providers will introduce additional innovations to the industry, such as transparency into prices for future services, as with plane tickets. Moreover, prices are also likely to vary by location, and the price at each location will be dynamic.

Different locations serviced by the same or different providers might have different prices at any given time due to a variety of reasons: higher value due to proximity to a given location, such as a stock exchange; different capacity utilisation, due to aggregate customer behaviour such as “follow the sun” cycles or statistical variation; or differences in power (and cooling) costs or in how power is priced (for example, breakered power, draw power, actual power, or bundled). Power is particularly of interest, because it is a large fraction of compute costs, and such power costs can fluctuate dramatically in the short term, even today. As electric grids become smart and increasingly use intermittent energy sources such as wind turbines, and electric demand response systems use pricing actions and Internet of Things connections to regulate demand, such location-dependent fluctuations will likely increase, barring breakthroughs in energy storage capabilities, distributed power generation, or transmission costs. In such a world, cloud providers compete not only against competitors, substitutes (such as do-it-yourself in your own private cloud or a colocation facility), and between locations, but against their future selves if customers decide to defer purchase, anticipating lower future prices to run deferrable workloads.

10.2.1 Service models

As was shortly mentioned above, there are many vendors who provide Cloud Services in the market today; the big ones in the league being Amazon, Google, Rackspace and Microsoft. Services are offered in any of these models, IaaS, PaaS and SaaS (Figure 10.4).

There are some companies who also provide other services like DaaS (Desktop as a Service), Backup as a Service etc. but the first three are considered traditional and the most common ones.

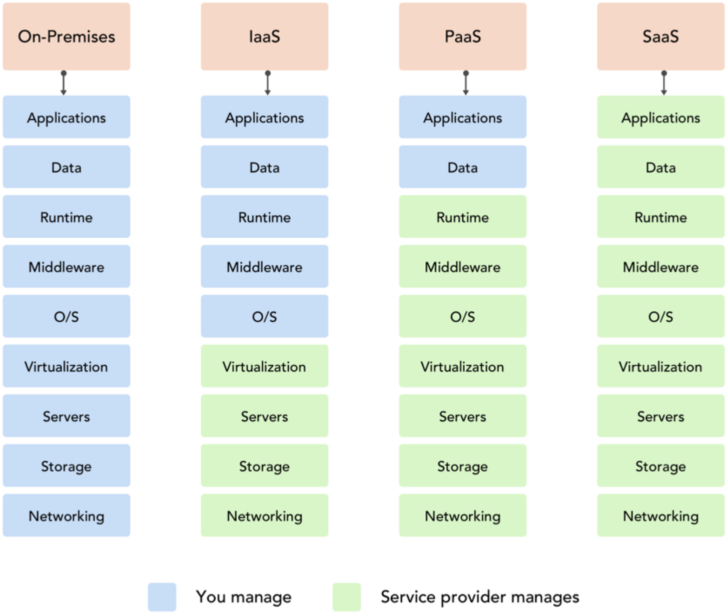


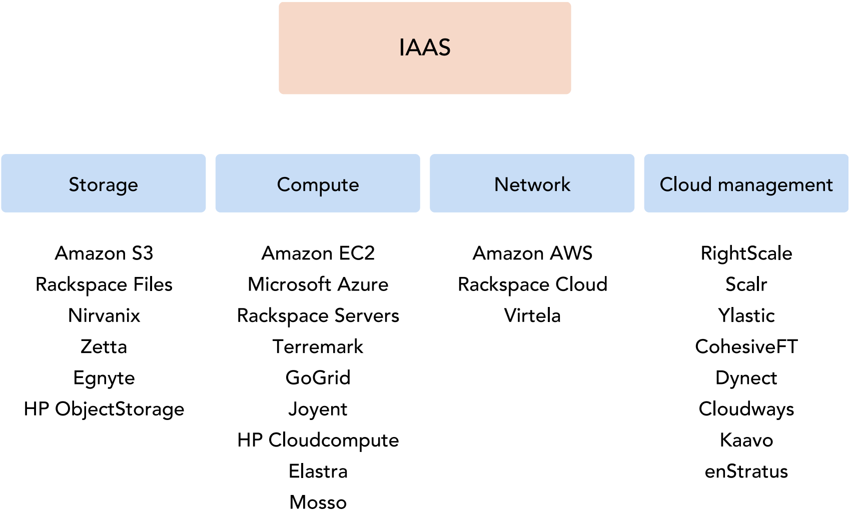
Figure 10.4 – Services provided with IaaS, PaaS and SaaS

Infrastructure as a service (IaaS) – Typically, this is case of platform virtualization environment which is offered as a service to you, along with raw storage and networking. Rather than purchasing servers, software, data-center space or network equipment, in your environment, you can instead buy those resources as a fully outsourced service. Suppliers typically bill such services on a utility computing basis; the amount of resources consumed (and therefore the cost) will typically reflect the level of your activity.

Top players of the market for IaaS are presented on Figure 10.5.

The key features of IaaS:

1. Instead of purchasing hardware outright, users pay for IaaS on demand.
2. Infrastructure is scalable depending on processing and storage needs, costs are scalable as well.
3. Saves enterprises the costs of buying and maintaining their own hardware.
4. Because data is on the cloud, there can be no single point of failure.
5. Enables the virtualization of administrative tasks, freeing up time for other work.



IAAS

Storage	Compute	Network	Cloud management
Amazon S3	Amazon EC2	Amazon AWS	RightScale
Rackspace Files	Microsoft Azure	Rackspace Cloud	Scalr
Nirvanix	Rackspace Servers	Virtela	Ylastic
Zetta	Terremark		CohesiveFT
Egnyte	GoGrid		Dynect
HP ObjectStorage	Joyent		Cloudways
	HP Cloudcompute		Kaavo
	Elastra		enStratus
	Mosso		

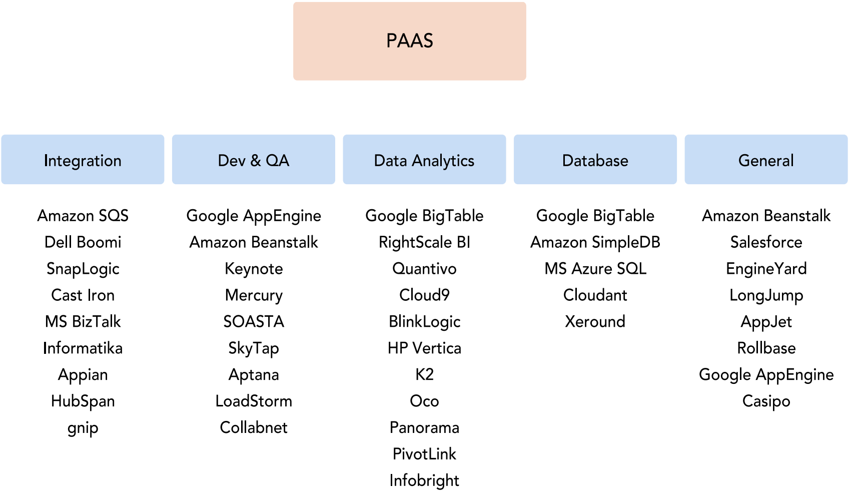
Figure 10.5 – IaaS Top Providers

Platform as a service (PaaS) – This delivers a computing platform and/or solution stack as a service. This often consumes the cloud infrastructure and sustaining cloud applications. This model facilitates you to deploy applications without the cost and complexity of buying and managing the underlying hardware and software layers. As the OS is managed by the Supplier, the underlying patches and updates are taken care by the vendor itself. You don’t need to worry about it.

Top players of the market for PaaS are shown on Figure 10.6.

Key PaaS features:

1. Provides a platform with tools to test, develop and host applications in the same environment.
2. Enables organizations to focus on development without having to worry about underlying infrastructure.
3. Providers manage security, operating systems, server software and backups.
4. Facilitates collaborative work even if teams work remotely.



PAAS

Integration	Dev & QA	Data Analytics	Database	General
Amazon SQS	Google AppEngine	Google BigTable	Google BigTable	Amazon Beanstalk
Dell Boomi	Amazon Beanstalk	RightScale BI	Amazon SimpleDB	Salesforce
SnapLogic	Keynote	Quantivo	MS Azure SQL	EngineYard
Cast Iron	Mercury	Cloud9	Cloudant	LongJump
MS BizTalk	SOASTA	BlinkLogic	Xeround	AppJet
Informatika	SkyTap	HP Vertica		Rollbase
Appian	Aptana	K2		Google AppEngine
HubSpan	LoadStorm	Oco		Casipo
gnip	Collabnet	Panorama		
		PivotLink		
		Infobright		

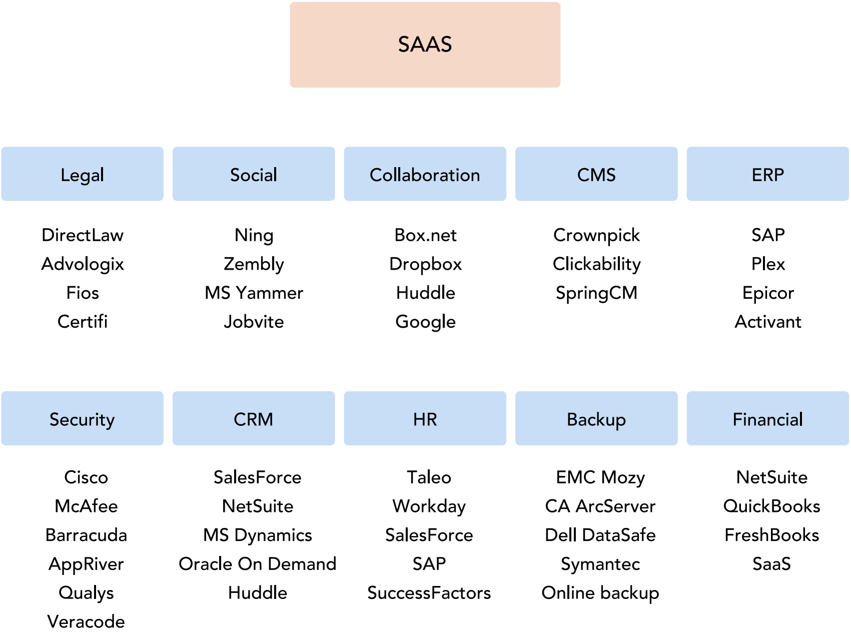
Figure 10.6 – PaaS Top Companies

Software as a Service (SaaS) – This is the most simplest looking cloud offering. Most of us have been using services without realising it to be a cloud platform. This model delivers software as a service over the Internet, eliminating the need to install and run the application on the customer’s own computers and simplifying maintenance and support. The best example could be Gmail, Office 365 or even Hotmail.

Top players of the market for SaaS are presented on Figure 10.7.

Key SaaS features:

1. SaaS vendors provide users with software and applications via a subscription model.
2. Users do not have to manage, install or upgrade software; SaaS providers manage this.
3. Data is secure in the cloud; equipment failure does not result in loss of data.
4. Use of resources can be scaled depending on service needs.
5. Applications are accessible from almost any internet-connected device, from virtually anywhere in the world.



SAAS

Legal	Social	Collaboration	CMS	ERP
DirectLaw Advologix Fios Certifi	Ning Zembly MS Yammer Jobvite	Box.net Dropbox Huddle Google	Crownpick Clickability SpringCM	SAP Plex Epicor Activant

Security	CRM	HR	Backup	Financial
Cisco McAfee Barracuda AppRiver Qualys Veracode	SalesForce NetSuite MS Dynamics Oracle On Demand Huddle	Taleo Workday SalesForce SAP SuccessFactors	EMC Mozy CA ArcServer Dell DataSafe Symantec Online backup	NetSuite QuickBooks FreshBooks SaaS

Figure 10.7 – Top SaaS Companies

10.2.2 Values and risks

Values and advantages are covered partly in previous parts, but it would be important to make a short summary, before describing the risks:

1. SaaS vendors provide users with software and applications via a subscription model.
2. Reduced setup costs can be considered as a major advantage for cloud computing, since the costs involved in setting up a data centre are not very high.
3. In addition to the IT industry, even small scale businesses can adopt this environment (model).
4. Considering cloud computing from the aspect of power management, it serves as a virtual server which is easier to implement in comparison to physical servers.
5. Hardware management failure can also be localized and rectified with relative ease.
6. Various data centres are spread throughout the country and thus it makes easy for the businesses to use preferred sites.
7. The assessment of data can be done any time and is highly beneficial for the IT industry in reducing workloads.
8. The cloud computing environments are easily scalable.
9. Backup recovery is very easy in Infrastructure as a Service (IaaS) Providers, hence there is efficient incident response whenever data needs to be recovered.

Having a clear list of advantages is one side, another is to understand the risks and disadvantages. In order to create awareness among the users of cloud computing regarding the serious threats and vulnerabilities involved in cloud computing environments, a study on various risks is imperative. The different risks are discussed below.

Security Risks

The state of preventing a system from vulnerable attacks is considered as the system’s security. Security risks involved with the governmental use of cloud computing have various risk factors. Seven important identity factors for risk in a cloud computing model are: Access, Availability, Network load, Integrity, Data Security, Data Location and Data Segregation

Access

The data in a private organization allows only the authenticated users to access the data. The access privilege must be provided only to the concerned customers and auditors in order to minimize such risks. When there is an access from an internal to external source, the possibility of risk is more in case of sensitive data. Segregation of the data is very important in cloud computing as the data is distributed over a network of physical devices. Data corruption arises if appropriate segregation is not maintained. Currently, there are no federal policies addressing how government information is accessed.

Availability

Availability plays a major role in cloud computing since the needs of the customers should be attended on time. A research from the University of California had tracked the availability and outages of four major cloud vendors. It was found that overload on the system caused programming errors resulting in system crashes and failures. Due to the lack of backup recovery Apple, MobileMe, Google Gmail, Citrix and Amazon s3 reported periods of unavailability ranging from 2 to 14hrs in a span of just 60 days. This resulted in a loss of confidence among the customers and the vendors. Natural disasters can also present significant risks. A lightening strike at one of Amazon.com's facilities caused the service to go offline for approximately 4 hours. This component of the cloud was difficult to replace immediately and resulted in delays.

Network Load

Cloud network load can also prove to be detrimental to performance of the cloud computing system. If the capacity of the cloud is greater than 80%, then the computers can become unresponsive due to high volumes .The computers and the servers crash due to high volume motion of data between the disk and the computer memory. The percentage of capacity threshold also poses a risk to the cloud users. When the threshold exceeds 80%, the vendors protect their services and pass the degradation on to customers. It has been indicated that in certain cases the outage of the system to the users are still not accessed. Flexibility and scalability should be considered pivotal when designing and implementing a cloud infrastructure. Money and time also plays an important role in the design of the infrastructure.

Customers will always have expectations on the durability and the efficiency of the system. Going forward the customers will also demand the need of interoperability, ability to switch providers and migration options Another risk factor of cloud computing is the implementation of the application programming interfaces (API).

Integrity

Data integrity affects the accuracy of information maintained in the system. In a cloud computing model data validity, quality and security affect's the system's operations and desired outcomes. The program efficiency and performance are addressed by the integrity. An apt example for this would be that of a mobile phone service provider who stored all the customer's data including messages, contact lists etc in a Microsoft subsidiary. The Provider lost the data and the cloud was unavailable. The customers had to wait until they got the necessary information from the cloud and the data was restored.

Data Security

Another key criterion in a cloud is the data security. Data has to be appropriately secured from the outside world. This is necessary to ensure that data is protected and is less prone to corruption. With cloud computing becoming an upcoming trend, a number of vulnerabilities could arise when the data is being indiscriminately shared among the varied systems in cloud computing. Trust is an important factor which is missing in the present models as the service providers use diversified mechanisms which do not have proper security measures. The following sub section describes the risks factors in cloud environments.

Data Location

Data Location is another aspect in cloud computing where service providers are not concentrated in a single location but are distributed throughout the globe. It creates unawareness among the customers about the exact location of the cloud. This could hinder investigations within the cloud and is difficult to access the activity of the cloud, where the data is not stored in a particular data centre but in a distributed format. The users may not be familiar with the underlying environments of the varied components in the cloud.

Data Segregation

Data Segregation is not easily facilitated in all cloud environments as all the data cannot be segregated according to the user needs. Some customers do not encrypt the data as there are chances for the encryption itself to destroy the data. In short, cloud computing is not an environment which works in a toolkit. The compromised servers are shut down whenever a data is needed to be recovered. The available data is not correctly sent to the customer at all times of need. When recovering the data there could be instances of replication of data in multiple sites. The restoration of data must be quick and complete to avoid further risks.

10.3 Services for performing computing in Android and iOS applications on the cloud

The supporting infrastructure stands behind each on-line mobile application which is available on App Store or Google Play. If it is needed, this infrastructure should provide data storage and synchronization, real-time communication, push notification services, web app service hosting, etc. During last years there were developed lots of services that have filled the gap between existing cloud capacities and mobile application needs. Those services can help to build the important infrastructure and focus on application value proposition mainly, without distracting on purchasing, building and managing own server systems. Here we present the brief analysis of the most popular services, which can supply in almost all issues of online based mobile applications – Firebase, Amazon, Microsoft Azure. The PubNub and Pusher, which have a strict focus on realtime communications and push-notifications are also described in this section.

Firebase [8]. Cloud storage for Firebase let developers upload and share content, store and analyse data in a Google Cloud Storage bucket – an exabyte scale object storage with high availability and global redundancy. This storage let developer securely upload files directly from mobile devices and web browsers and even handle spotty networks. The Firebase gives a set of tools that can be applied on three stages of app development – *building* (Cloud Firestore, Cloud Fucntions, Hosting, Realtime Database, ML Kit, Authentification, Cloud Storage), *improving* (Crashlytics, Test Lab, Performance Monitoring) and *growing* (Analytics, Firebase A/B Testing, Remove

Config, App Indexing, Predicition, Cloud Messaging, Dynamic Links, Invites).

The first *building* set provide developer with:

1. flexible, scalable NoSQL cloud database to store and synch data to client and server-side development;

2. tools for offline application support for both mobile and web, which help to build the responsive apps that work regardless to network latency or Internet connectivity;

3. tools for automatically running backend code in response to events triggered by Firebase features and HTTPS request;

4. fast and secure hosting for web applications, static and dynamic content and microservices;

5. mobile SDK that brings Google's machine learning experience;

6. backend services, SDK, UI libraries to authenticate users by passwords, phone numbers, federated identity providers like Google, Facebook, Twitter, etc.

The *improving* set gives following functionality:

1. tool for clear, actionable insight into app issues with crash reporting solution;

2. cloud-based app-testing infrastructure that help to test app across wide variety of devices and device configurations and get results including logs, videos and screenshots;

3. tool for getting insight into the apps performance characteristics, such as startup time, HTTP/S network requests and provide ability to set custom characteristics monitoring.

The final set – *growing* provide developers with:

1. free and unlimited Google Analytics solutions which help to understand user behavior by creating reports on up to 500 distinct events and custom audience actions as well;

2. integration with BigQuery, Firebase Crash Reporting, FCM, Firebase Remove Config, Google Tag Manager services;

3. tool for testing changes to app's UI and other features;

4. tool for targeting "predicted" user groups - running A/B tests on users who are predicted to perform a certain action;

5. service that let developer change the behavior and appearance of app without requiring users to download an app update;

6. service that links to app content on Google Search;

7. tool that applies machine learning to app analytics data and created dynamic user segments based on predicted behavior of users in an app;

8. cross-platform messaging tool that let developer reliably deliver messages to users an no cost;

9. solution for app referrals and sharing via email or SMS.

Amazon [9]. Amazon provides developers with APIs and services, namely Amazon Drive, Login with Amazon, Mobile Ads, which help to store, synch, share various documents, set user authentication system used by Amazon.com and implement mobile-optimized ads from Amazon and brand advertisers.

Amazon Drive service can be available only after receiving invitation. Its API and SDKs for both Android and iOS use Login with Amazon for authentication. Additionally, Amazon requires providing the developed app details, whether it need read and/or write access, type of content in the customers account (images, video, document, etc.).

Login with Amazon let developer protect the customer data using the Amazon.com authentication system, which is based on OAuth 2.0 and has been widely adopted across various sites. As customers prefer to log in with known credentials the ability to log in using Amazon.com system can make app more user-friendly. One of the biggest benefits of implementing this service is that with Amazon Pay customers get quick and convenient way to check out without leaving an app or site.

The Amazon Mobile Ad Network provide developer with solution to monetization of mobile apps and games across platforms. Using cross-platform solutions, highly relevant mobile optimized ads from Amazon this service also helps to measure the success rate with actionable reporting tool.

Microsoft Azure [10]. Azure provides mobile developers with mobile backend services, data storage in durable and scalable cloud services, machine learning and cognitive services, tools for apps building, testing and distributing, analytics, crash report and notification services, tools for developing location-aware IoT and mobility solutions.

Azure Functions gives a fully manages compute platform that meet the high reliability and security requirements. The code gets all needed compute resources that scale on demand. It is mainly concentrated on microservices-friendly approach, that can be realized in most wide applied programming languages.

With Functions developer get:

1. web application backend – online requests are selected from queue, processed and further stored in database;
2. mobile application backend – HTTP API calls a mobile app, which further is processed by a function and results are stored in database;
3. real-time file processing – activity records are securely uploaded as .pdf files for further decomposition, processing and storing in database for easy queries;
4. real-stream processing – the huge amount of various data from app or device is processed in near real-time and stored in database for use in analytics dashboard;
5. automation of scheduled tasks – customer database is analyzed for duplicate entries every 15 minutes, to evade multiple communications being sent out to same customers.

Azure Cosmos DB has an idea of global distribution and horizontal scale at its core. This database provides native support for NoSQL and OSS APIs including MongoDB, Cassandra, Gremlin and SQL. If offers multiple well-defined consistency models, guarantees single-digit-millisecond read and write latencies at the 99% and guarantees 99.999 high availability [11]. Azure Cosmos DB provides with infrastructure that can help to build globally distributed mission-critical applications, serverless applications, instantly scale to accommodate diverse IoT workloads, generate the real-time personalized recommendations for customers, support in-depth queries over rapidly changing inventory, scale database to accommodate bursts of traffic, etc.

With Azure Machine learning service developers can:

1. build, train and deploy personal machine learning models;
2. faster identify suitable algorithms and hyperparameters;

3. increase productivity with experiment tracking, model management and monitoring, integrated CI/CD and machine learning pipelines;
4. deploy models to the cloud just with few lines of code;
5. integrate services with any Python environment;
6. use open-source frameworks.

Using Cognitive services developer get access to:

1. image-processing algorithms;
2. converting of spoken audio into text;
3. voice for verification or add speaker recognition to an app;
4. mapping complex information to solve intelligent recommendations and semantic search tasks;
5. processing natural language with pre-built scripts.

PubNub [12]. PubNub is a global Data Stream Network and realtime network-as-a-service company which provide developers with infrastructure to build realtime cross-device and cross-platform chat apps; monitor, control and stream data between smart devices, sensor networks, hubs, ect.; broadcast push notifications and accurate alerts, when something happens in realworld.

ChatEngine is a PubNub framework for chat development and serverless deployment, including an SDK, plugins, UI components and messaging.

PubNub provides the infrastructure and APIs for communication in any size IoT deployment. This service gives tools to send and receive data between devices, monitor and track device status in realtime, route device data back to any existing system (AWS, IBM, Microsoft Power BI, etc.), use TLS and AES encryption, automatically execute code to trigger alerts or device actions.

Pusher [13]. Pusher Channels provides realtime communication between servers, iOS and Android apps and devices. It can be used for notifications, chat, gaming, wep-page updates, IoT and other systems which requires realtime communication.

Pusher Channels give developers libraries for: web browsers, iOS and Android apps, PHP frameworks, cloud functions, bash scripts and IoT devices. It uses WebSockets, HTTP and provides fallbacks for devices that don’t support WebSockets. As this service provides a wide range of information on application activity the developer can debug, analyze and record all needed data. Every connection, publication and subscription is available and can be accessed via Channels dashboard.

10.4 Work related analysis

The cloud computing [1] is a cutting edge technology that enabled a shift away from computing as a bundled hardware and software product and by sending, processing, storing and managing tons of data on distinct infrastructure the customers gain the technical benefits as well as economical.

For years the cloud infrastructures are supporting mobile development process, with all means for sending, sharing and synchronizing data, creating realtime communications, testing and deploying online services, apps analytics and crash reporting, push notifications and more [8 –13]. Recently cloud providers have started to support IoT developers as well with various supporting tools [8 – 13]. The merge of IoT and cloud computing became obviously will bring benefits for both sides, for example clouds will make a new jump in growing their efficiency and IoT gets affordable, reliable and scalable infrastructure [3, 4].

There can be defined three main cloud services – IaaS, PaaS, SaaS, that have been widely provided by such big companies as Amazon, Microsoft, Google, etc. The full list of companies is presented in section 10.2. The “IoT - Cloud” tandem gave birth to new types of services, such as “Sensor-as-a-Service”, “Sensing-and-Actuation-as-a-Service”, “Database-as-a-Service”, “Data-as-a-Service”, “Network-as-a-Service” and much more [3].

As the cloud companies have a great concentration on business customers, during last years various researches were provided to clearly define advantages that this type of users receives [2, 6, 7].

In 2018 IaaS became the fastest-growing segment of the market, which is forecast to grow 27.6 percent in 2019 and reach $39.5 billion. By 2022, Gartner expects that 90 percent of organizations purchasing public cloud IaaS, will start using both the IaaS and PaaS capabilities from their providers.

This section provides an overview on: basic cloud computing architecture; IaaS, PaaS and SaaS service models as well as new SeaaS, SAaaS, DaaS, DBaaS and NaaS; basic dynamic interactions in the cloud; analysis of cloud computing market growth; advantages and risks that business may face while using basic service cloud models; basic services that iOS and Android mobile developers can implement in their project to get cloud features in their applications.

The following MSc and PhD courses and programs of ALIOT project EU universities have been taken into account:

– Coimbra University, Portugal: MSc program in Electrical and Computer Engineering [16], Programmable Electronic Devices (PhD Course) [17];

– KTH University, Sweden: MSc programs in Systems, Control and Robotics [18], Hybrid and Embedded Control Systems [19];

– Newcastle University, United Kingdom: MSc program Embedded Systems and Internet of Things (ES-IoT) [20].

Besides, the following courses have been analyzed to develop lecture material for this module: IoT Sensors and Devices; IoT Networks and Protocols (Curtin University, Australia) [21].

Conclusions and questions

Study material presented in this section is covering the basic topics of cloud computing organization, how growing IoT sector depend on cloud and is depended by clouds as well, introduction into main cloud computing service models with their benefits and risks, overview of most widely applied cloud services for mobile applications development.

The following questions can help to strengthen and assimilate the educational material that is covered in this section:

1. Give a definition of private cloud type.

2. Describe basic features of public cloud type.

3. Provide several benefits of merging IoT and Cloud computing technologies.

4. Define the levels of cloud computing architecture.

5. What new XaaS were generated due to merge of IoT and Cloud Computing.

6. Describe the Database-as-a-Service model.

7. Give a description to Network-as-a-Service model.

8. What type of service provide Sensor-as-a-Service and Sensing-as-a-Service models?

9. Describe the main difference between synchronous and asynchronous communications in cloud.

10. Describe the mechanism of messaging queues.

11. How the FIFO n-n communication model works?

12. Describe the FIFO n-1 and FIFO 1-n communication models difference.

13. Describe the causally ordered communication model.

14. Describe the basic principle of “map-reduce” approach.

15. Which type of issues the “map-reduce” approach can solve?

16. What is the fastest growing segment of cloud computing in the market?

17. What are the main values of moving the business to cloud?

18. What are the main risks of moving the business to cloud?

19. Name five of the Cloud computing big market leaders.

20. Name several services that Firebase provide for mobile application building stage.

21. Which services can be used to set the crash reporting tool?

22. Which services can be applied to get the app analytics?

23. Which services can be applied to set the app monetization?

24. Which provides give tools to build the IoT-mobile interaction?

25. Which services can be applied to implement the realtime communication features in the application?

References

1. Mell P. M., Grace T. NIST Definition of Cloud Computing. Special Publication (NIST SP). – 2011. – p. 800-145

2. Sullivan D. The Definitive Guide to Cloud Computing. – Realtime Nexus Publishers. – 2009. – 219 p.

3. Botta A., Donato W. Integration of Cloud Computing and Internet of Things: a Survey. – Future Generation Computer Systems (Vol. 56), Elsevier. – 2016. – p. 684 – 700.

4. Belgarum M.R., Soomro S. Challenges: Bridge Between Cloud and IoT. – Proc. of IEEE International Conference on Engineering Technologies and Applied Science. – 2017. – p. 1- 5 .

5. Chevrou F., Hurault A. On the Diversity of Asynchronous Communication. – Springer, Formal Aspects of Computing (Vol. 28, Issue 5). – 2016. – p. 847 – 879.

6. Weinman J. Cloudonomics: The Business Value of Cloud Computing. – John Wiley & Sons. – 2012. – 412 p.

7. Weinman J. Cloud Pricing and Markets. – IEEE Cloud Computing (Vol. 2, Issue 1). – 2015. – p. 10 – 13.

8. Firebase . – [electronic source: https://firebase.google.com/]

9. Amazon Developer Services and Technologies . – [electronic source: https://developer.amazon.com/].

10. Microsoft Azure. – [electronic source: https://azure.microsoft. com/].

11. Azure Cosmos DB. – [electronic source: https://azure.microsoft.com/en-us/services/cosmos-db/].

12. PubNub. – [electronic source: https://www.pubnub.com/].

13. Pusher. – [electronic source: https://pusher.com/].

14. IoT Sensors and Devices. - [electronic source: https://www.edx.org/course/iot-sensors-and-devices].

15. IoT Networks and Protocols. [electronic source: https://www.edx.org/course/iot-networks-and-protocols-2].

16. The Master program in Electrical and Computer Engineering [https://www.uc.pt/en/fctuc/deec/courses/mieec]

17. Programmable Electronic Devices [https://www.uc.pt/fctuc/deec/PhD_courses/Programmable_Electronic_Devices]

18. Master's programme in Systems, Control and Robotics [https://www.kth.se/en/studies/master/systems-control-robotics]

19. Hybrid and Embedded Control Systems [https://www.kth.se/student/ kurser/kurs/EL2450?l=en]

20. Embedded Systems and Internet of Things (ES-IoT) MSc [https://www.ncl.ac.uk/postgraduate/courses/degrees/embedded-systems-internet-of-things-msc/#profile].

21. Mobile App Development with Swift.- [electronic source: https://www.edx.org/professional-certificate/curtinx-mobile-app-development-with-swift].

11. INTEGRATION OF BIG DATA

AND IOT TECHNOLOGIES

Dr. O.N. Odarushchenko, Dr. A.Y. Strjuk (KhAI)

Contents

Abbreviations

AMQP - Advanced Message Queuing Protocol
CEP - Complex Event Processing
CSV - Comma-Separated Values
HDFS - Hadoop File System
IIRA - Industrial Internet Reference Architecture
IoT – Internet of Things
JSON - JavaScript Object Notation
MSc – Master of Science
MQTT – Message Queue Telemetry Transport
PhD – Doctor of Philosophy
PHP - Hypertext Preprocessor
RDBMS - Relational Database Management Systems
SQL - Structured Query Language
XML - Extensible Markup Language

11.1. Foundations of Big Data Systems for IoT

This section describes the integration of two from the most discussed concepts in information technology today: Big Data and Internet of Things. As is this book is intended for MSc-, PhD-students and engineers who will be involved in design and development of such integrated projects, we will provide an engineering overview of the field of Big Data from an IoT perspective. This chapter covers the following topics:

- Big Data characteristics.
- Big Data platform stack and tools.
- The most commonly used architectures of Big Data systems.
- Requirements for IoT Big Data systems.

To start with, we will consider characteristics of the Big Data and try to highlight the most important from the IoT point of view ones.

11.1.1. Big Data characteristics

Big Data is an umbrella term applicable for any collection of data sets so large or complex that it becomes difficult to process them using traditional data management techniques such as, for example, the Relational Database Management Systems (RDMS) [1].

A huge number of IoT sensors create massive amounts of data that must be handled. From a data management standpoint, the biggest challenges are - how to evaluate massive amounts of data arriving from different sources in different forms, and how to perform such an evaluation in a timely manner [1, 4, 24]. The characteristics of Big Data are often associated with **several** Vs – in different sources a quantity of specified Vs could change from three [1, 4] to ten [5] and (:)) even up to forty-two [2]. Figure 11.1 shows the V8 set of Big Data characteristics:

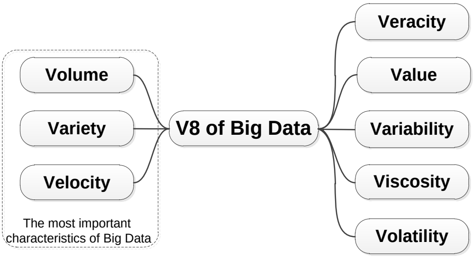


Figure 11.1 – Characteristics of Big Data

Volume: Volume refers to the scale of the data. For typical IoT applications volume could vary from gigabytes on the low bound to petabytes (10^{15}) or even exabytes (10^{18}) of data on the high bound.

Variety: Variety refers to the diversity of the data. Variety of data is a critical factor for database technology selection – selected data storage shall handle all of the anticipated data formats.

Velocity: Velocity refers to how quickly data is being generated, collected and analyzed. High quantity of sensors and other devices in IoT systems forces the necessity of real-time, predictive and prescriptive analytics.

Veracity: Variety refers to the accuracy, reliability, credibility and unambiguity of the data. The "dirty data" should not be accumulated in the system.

Value: Value refers to the how Big Data analyzing and collecting are valuable in accordance with business requirements of a stakeholder. This characteristic is important owing to a low value of the unprocessed sets of Big Data.

Variability: Variability refers to the variation in the rate of flow of the data. Usually, the velocity of the data is inconsistent and has periodic peaks and troughs.

Viscosity: Viscosity refers to the latency or duration of data transmission between the source and destination.

Volatility: Volatility refers to how long data are valid and should be stored.

Described characteristics define differences between Big Data and data found in traditional data management tools.

11.1.2. Big Data categories and types

Data science for the IoT segment deal with the following categories of data [1, 3]:

Structured. Structured data is used to refer to the data follows a predictable format, model or schema that define how the data is represented or organized. The most common example of structured data is data stored in table format such as Microsoft Excel or CSV files, where the meaning of each data item is defined. Structured data constitutes well with a traditional RDBMS. There is a lot of information that cannot be captured in a structured format.

Unstructured. Unstructured data has a high degree of randomness and variance. The most part (around 80%) of Big Data amount is unstructured data. Unstructured data is hard for processing and store with traditional programming means. Examples of unstructured data include video, audio, images, social media data, text and speech.

Semi-structured. Semi-structured data has some degree of variance and randomness in form but contains a certain schema and consistency. Documents in character format usually are considered as semi-structured. Examples of semi-structured data include JavaScript Object Notation (JSON) and Extensible Markup Language (XML), which are common data interchange formats.

Smart devices in IoT networks are able to generate data of all three categories.

By the content type, data in IoT networks could be classified as follows:

Natural language data. Natural language data is human-generated data, as a rule in unstructured form. The sources of natural language data include speech capture devices, different kinds of telephony and text-based communication services.

Machine-generated data. Machine-generated data is information that is automatically created by a machine, and that is not a result of a human choice. Machine-generated data is a major data resource in IoT. Examples of machine data are service and network logs, telemetry generated by sensors.

Graph-based data. Graph-based or network data is data that focuses on the relationship or adjacency of objects. Graph-based data is suitable to describe networks, such as social networks, information networks, biological networks and technological networks. Network data is represented as nodes connected via one or more types of relationship.

Geospatial data. Geospatial data represents information that has a geographic aspect. Records in this type of data have coordinates, an address, city, postal code or zip code, included with them. These data provide the link between place, time, and some descriptive information.

Audiovisual data. Audio, image and video are data types that the most challenging for data science as is complicated algorithms are necessary to process these data types. Another challenge of audiovisual data is the huge amount of data being produced and accumulated with real-time media.

Figure 11.2 shows interrelations between data with different content types and categories.

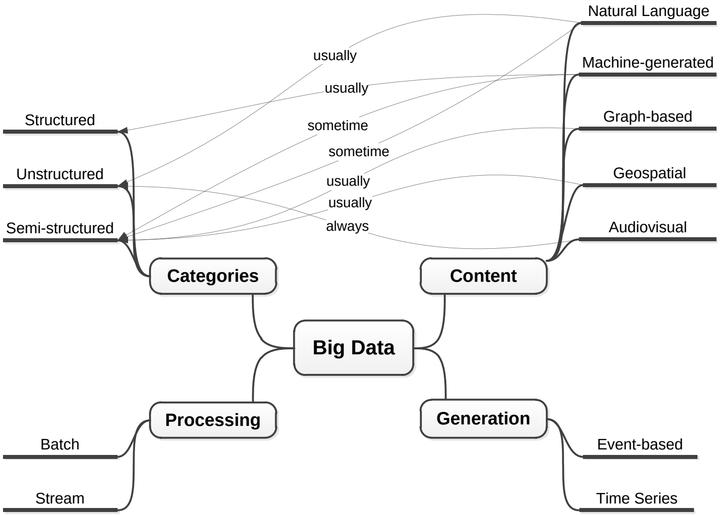


Figure 11.2 - Big Data Categories and Types

Data in IoT networks could be generated following the next two patterns:

Event-based data. Event-based data generated only when an IoT device detects a particular event. Event data contains three key information items, so-called behavior data: action; timestamp and state. The action is the event that's happening. The timestamp is the time when this event happened. The state describes all other information relevant to the threshold event.

Time series data. Time series data is a sequence of data items, typically representing measurements made over a periodic time interval.

By the analysis (processing) approach, data could be classified as follows [9]:

Batch data. Batch data is a data set that is fully available at the processing time. There are no strict requirements for the batch data analysis time, but in the case of batched Big Data, its amount may be too big to be loaded into the memory all at once.

Stream data. Stream data is a sequence of data elements ordered by time. The most important challenge of the stream data analysis is that it does not have access to all data and shall processes data as it arrives. Analysis of the stream data has to be provided in real-time or near real-time.

As a rule, IoT applications require both processing approaches to be implemented.

Big Data types define specialized techniques and tools which are used for data collection, data storage, data representation, data fusion, data processing and visualization.

11.1.3. Big Data Platform of IoT Services Platform

Figure 11.3 shows the three-tier Industrial Internet Reference Architecture (IIRA) that was published by the Industrial Internet Consortium. This reference architecture has three-tiers: Edge tier, Platform tier, and Enterprise tier [6, 7, 25].

The Platform Tier of IIRA receives, processes, and forwards data and control commands from the Edge tier to the Enterprise tier and vice versa. The Platform Tier is also responsible for data ingestion, data stores, and can store configuration and control data and provide non-domain-specific services such as data aggregation and analytics.

Although several reference models of IoT are endorsed by various organizations and regulatory bodies, all these models specify IoT from a layered perspective [4]. Every published IoT model includes a middleware layer (or a combination of layers) that has functionality similar to the Platform Tier functionality.

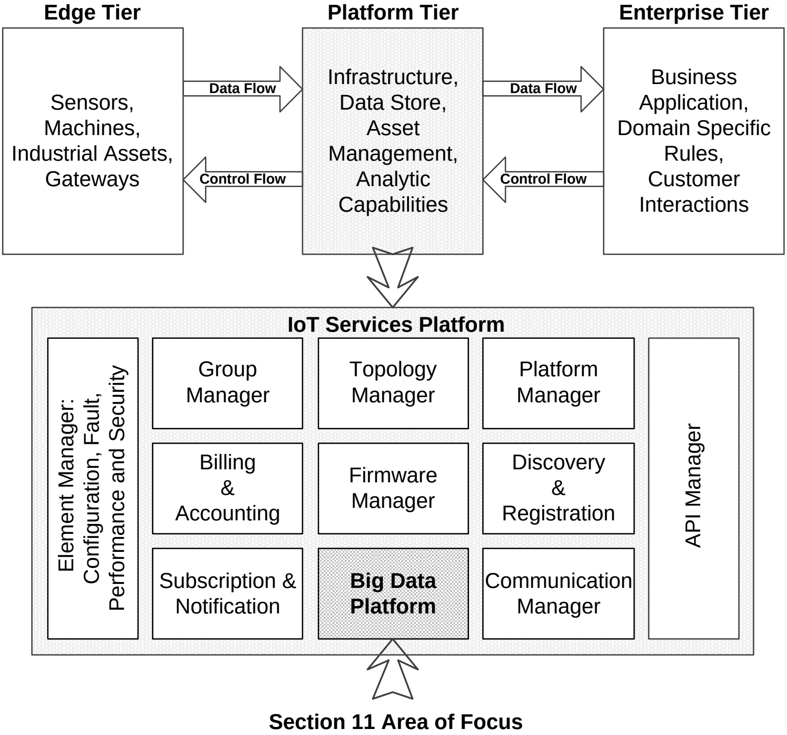


Figure 11.3 - Common IoT Services Platform Functions and Area of Focus for this Section

Core functions of the Platform Tier are implemented with IoT services platform. The purpose of the IoT services platform is to provide scalable vertical solutions for integration, connectivity, and data translation that are required by different technologies and applications.

The overall function areas of the IoT services platform are shown in Figure 11.3 [8]. A Big Data platform is an essential component of the IoT services platform. The Big Data platform shall provide means to efficiently store, retrieve, and process a massive amount of data. In the following chapter, we will consider the basic elements and tools of Big Data platforms.

11.2. Big Data platform stack and tools

A Big Data system has always been an important subsystem of an IoT system, as it has to provide a scalable, highly available, and fault-tolerant solution to deal with huge quantities of data in motion and in rest.

Currently, there exist a lot of software tools that provide the Big Data platform features. The diversity of these tools makes it difficult to select tools for implementation of a particular project of a Big Data system. Even more, some tools perform only a specific type of processing, while some others are able to perform a wide range of processing types. Nevertheless, based on similar goals and functionalities Big Data tools could be integrated into a layered Big Data platform stack (see Figure 11.4) which consists of the following layers [9, 10]:

Data storage: tools of this layer are used to store and retrieve Big Data. These tools include distributed file systems that maintain data on distributed servers and databases that support structured and unstructured data.

Data collection and distribution: this layer combines tools that are used to collect data arriving from different sources, perform data aggregation and integration, and distribute data among recipients within the platform.

Data processing: this layer contains tools for distributed processing of data. These tools are classified in accordance with their target application and input data type.

Presentation: tools of this layer perform fine-grain analysis of Big Data and provide the interface between a Big Data system and business applications of IoT system. Part II of this book thoroughly describes techniques and tools of this layer.

Control and service support: this layer combines tools that are used to manage the resources of a Big Data system, share them among the other tools within the system, provide security and testing of the system.

In the rest of this chapter, we will explore tools of each defined layers (except presentation layer) that answer three main questions: how to collect Big Data, how to store Big Data and how to process Big Data.

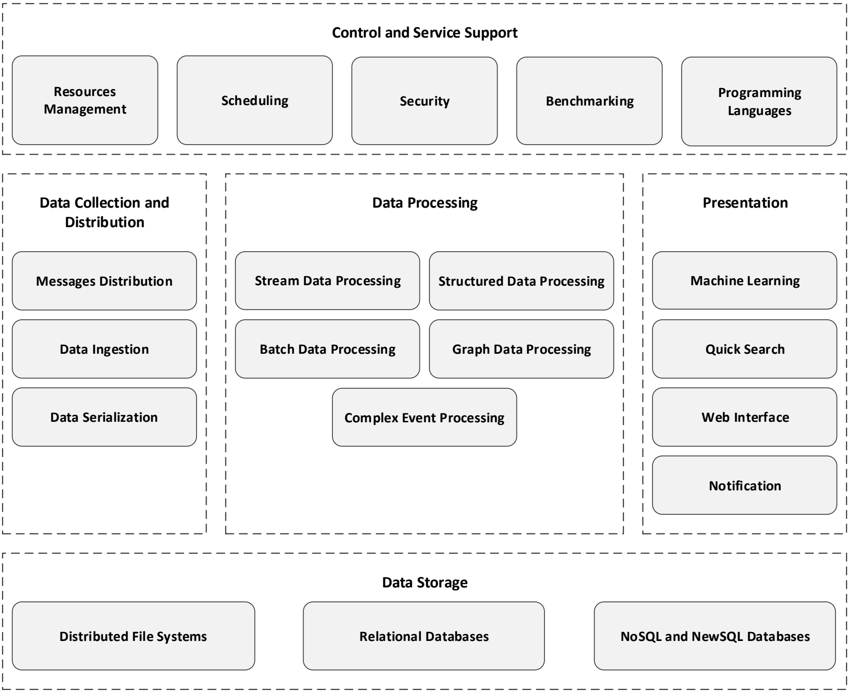


Figure 11.4 – Big Data Platform Stack

11.2.1 Data storage layer

Tools of the data storage layer could be categorized into three groups. Table 11.1 shows the ecosystem of this layer tools.

Distributed file systems. Distributed file systems make it possible to store, read and delete files on/from distributed disks, without involving users in details and complexity of these processes [9].

Distributed file systems have significant advantages that ground their wide usage in Big Data systems:

- They can store files larger than any one computer disk.
- They provide horizontal scalability – system resources (memory or storage capacity) could be increased by adding into a system a new server or servers.

Table 11.1 – Data Storage Ecosystem

Data Storage			
Distributed File Systems	HDFS: Hadoop File System		
	Red Hat GlusterFS		
	QFS: Quantcast File System		
	Ceph		
	…		
NoSQL and NewSQL Databases	**NoSQL**	**Document Stores**	MongoDB
			Couchbase
			Amazon DynamoDB
			…
		Key-value Stores	Redis
			Amazon DynamoDB
			Memcached
			…
		Column Stores	Cassandra
			HBase
			Azure Cosmos DB
			…
		Graph Stores	Neo4j
			Azure Cosmos DB

			Datastax Enterprise
			…
	NewSQL	**SQL on Hadoop**	Hive
			Impala
			Drill
			…
		Bayes DB	
		Sensei	
		Drizzle	
		…	
Relational Databases	Oracle		
	MySQL		
	Microsoft SQL Server		
	PostgreSQL		
	…		

At this moment the most-known and standart-de-facto distributed file system for Big Data projects is the Hadoop File System (HDFS). It is an open source implementation of the Google File System. Despite wide usage, Hadoop is not a replacement for database systems, as is a file system stores data as a set of bits, without knowing anything about content and structure of the data. Therefore, Hadoop have to be accompaned by database management system that provide access to the data in a way that allow using of the structure and content of the data.

NoSQL and NewSQL databases. To deal with the challenges of Big Data, a new group of databases has developed, these new databases have been grouped under the terms *NoSQL and NewSQL*. Initially, NoSQL means "No SQL support" since these databases did not support SQL, currently NoSQL is mostly interpreted as "Not only SQL" since some of these databases support a subset of SQL commands [4, 14].

NoSQL databases omit the constraints of the relational model, including consistency and schemas. The key benefits of NoSQL databases are the following:

- NoSQL provides schema-less data storage, it allows the storing of any types of data in multiple formats and in different schemas.

- NoSQL supports horizontal scalability.

- NoSQL could work in a clustered environment that is mainly built on commodity hardware.
- NoSQL spreads across multiple nodes with replication to multiple servers.

A lot of different types of NoSQL databases have developed, but they can be categorized into the following database types [11]:

Document stores **-** store every observation in a document, these databases generate a unique key for each row to associate it with semi-structured document data. This allows to use a more flexible data scheme.

Key-value stores **-** store data in the form of key-value pairs. The key is used as the unique ID to insert and retrieve data whereas the value contains the actual data. Stored data can be in any form, such as JSON, video, audio, log file, and so on.

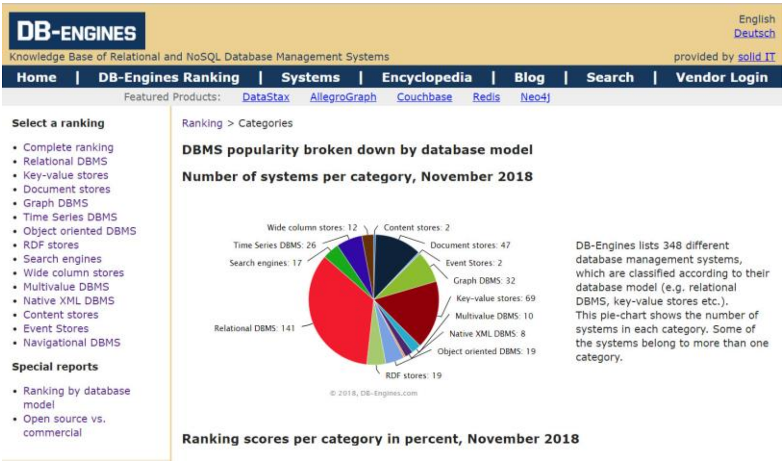
Column stores **-** data is stored in column form instead of row form, similar to relational databases. All cells of a column are stored together or contiguously on disk. Columnar storage is more efficient than row-oriented storage for analytic applications as is it allows algorithms to perform much faster queries.

Graph stores – store graph-based data, each record is interpreted as a node and all nodes are linked to form a relationship. It will help provide a more processable form of graph-based data.

New SQL **-** these databases retain key characteristics of the RDBMS (SQL interface and a relational data model) and scalability of NoSQL databases.

Relational databases (RDBMS) are still a good choice as a storage for structured data. In Big Data systems RDBMS could be used to store results of data analytics.

The DB-Engines [20] provides a monthly updated ranking of the most popular databases. Figure 11.5 shows that relational databases still dominate at the time this chapter was written, it also shows the relationship between different NoSQL and NewSQL models of storage.



DBMS popularity broken down by database model

Number of systems per category, November 2018

DB-Engines lists 348 different database management systems, which are classified according to their database model (e.g. relational DBMS, key-value stores etc.). This pie-chart shows the number of systems in each category. Some of the systems belong to more than one category.

Ranking scores per category in percent, November 2018

Figure 11.5 – Databases Ranked by DB Engines.com in November 2018

11.2.2 Data collection and distribution layer

Tools of this layer are used for collecting, aggregating, serialization and moving large amounts of data. Table 11.2 shows the ecosystem of this layer tools [12, 13, 14].

Table 11.2 – Data Collection and Distribution Ecosystem

Data Collection and Distribution		
Data Ingestion	Apache Flume	
	Apache Sqoop	
	…	
Messages Distribution	**Topic-based**	Apache Kafka
		Twitter Kestrel
		…
	Queue-based	Apache ActiveMQ
		Pivotal RabbitMQ
		…
Data Serialization	Apache Thrift	
	Apache Avro	
	Google Protocol Buffers	
	…	

Data ingestion tools - collect structured and unstructured data from multiple data sources and ingests it into a Big Data system. Apache Flume allows real-time ingestion of unstructured event-based streaming data in form of a data log. Apache Sqoop is used to import structured data from RDBMS which are outside of a Big Data system.

Messages distribution tool – provide connection between producers and consumers of data. Usage of a messaging system is a flexible and scalable solution if there is asymmetry either in the number of data producers and consumers or the velocity of produced and consumed data.

Topic-based messaging systems such as Kafka or Kestrel provide a higher-level programmability, high throughput, scalability, and durability. Kafka and Message Queue Telemetry Transport (MQTT) protocol are an efficient combination for end-to-end IoT integration from the edge to the data center [15].

Queue-based messaging systems such as RabbitMQ or ActiveMQ support Advanced Message Queuing Protocol (AMQP). These tools focused around consistency control and delivery guarantees but provide less throughput than the topic-based messaging systems [15].

Data serialization tools – provide a standard process for data processing tools to translate their built-in data structures into a standard format to store on disk or transfer across a network. Data can be serialized using many different formats, either human-readable (CSV, XML, JSON) or various binary formats. Human-readable formats could be recommended to use only for the simplest projects. Binary formats are generally preferred for storing or sending over a network large datasets owing to robustness, compact representation and architecture-independence of such a decision [13, 14].

Avro is a data serialization tool used for storage and sends data over a network. Avro provides a compact and fast binary format for data serialization. This format supports reach data structures and could be easily integrated with different programming languages.

Thrift and Protocol Buffers are data serialization tools that primarily provide data exchange over a network between applications written in different programming languages. Both tools support a variety of languages, including C++, C#, Java, JavaScript, Python, PHP, Ruby, Erlang, Perl, Haskell, Delphi, and other languages [13, 14].

11.2.3 Data processing layer

Data processing tools focus on the interpretation of structured and unstructured data in order to implement business requirements for Big Data systems. This requires an understanding of data and relationships between data items. Data analytics is performed with data processing tools usually focuses on past and present statistics of data. Tools of this layer mainly perform two types of analytics [14]:

- Descriptive analytics.
- Diagnostic analytics.

Simply stated, descriptive analytics answers question: What happened? Diagnostic analytics answers question: Why did it happen? Table 11.3 shows the ecosystem of this layer tools.

Batch data processing. Batch processing was one of the first use cases for Big Data tools such as Hadoop and MapReduce. During batch processing, data is collected for a period of time and processed in batches. The period of data collection could vary from hours to years, in accordance with business requirements. Therefore, batch processing systems have high latency. This latency could not be acceptable for a certain class of applications [13].

Table 11.3 – Data Processing Ecosystem

Data Processing	
Batch Data Processing	Hadoop MapReduce
	Apache Spark
	Google Cloud Dataflow
	…
Stream Data Processing	Spark Streaming
	Apache Storm
	Google Cloud Dataflow
	…
Structured Data Processing	Spark SQL

	Apache Hive
	Google BigQuery
	…
Graph Data Processing	Spark GraphX
	Apache Giraph
	Google Pregel
	…
Complex Event Processing	Apache Flink
	Oracle Event Processor
	TIBCO StreamBase
	…

MapReduce is a tool provided by Hadoop. While HDFS provides a distributed file system for storing Big Data sets, MapReduce provides a framework for batch data processing in parallel across a cluster of computers.

Spark is considered as the best replacement to MapReduce, as is it provides many advantages over MapReduce [13]. The most important advantages of Spark are the following [13]:

- Spark provides a simpler programming model and a more advanced API than that provided by MapReduce.
- Spark is significantly faster than MapReduce.
- Spark provides an integrated platform for different types of data processing, tools of this platform can be used for batch processing, stream processing, graph processing, interactive analysis and machine learning.

Cloud Dataflow is an example of proprietary data processing tools. Cloud Dataflow is a universal tool that is able to perform batch data processing as well as streaming data processing.

Stream Data Processing. Tools of this category perform processing of Big Data as it is collected in real-time or near real-time.

Spark Streaming is a Spark add-on, that runs on top of Spark core and extends Spark for stream data processing. Spark Streaming provides near real-time processing. Usage of Spark streaming within a system that requires both bath and streaming processing decreases the cost of development and maintenance.

Storm is an example of tools available for real-time stream processing. Comparably with Spark Streaming, Storm can provide lower latency and higher throughput.

Structured data processing. Tools of this category provide ability to analyze large amounts of structured data with SQL or SQL-like languages.

Spark SQL is a Spark library that runs on top of Spark, it provides a higher-level API for structured data processing. Spark SQL supports multiple query languages, including SQL and HiveQL [13].

Hive implements HiveQL that is a SQL-like language widely used as one of the interfaces for Hadoop MapReduce. BigQuery is a Google's product which supports SQL query to span distributed file systems and NoSQL databases [20].

Graph data Processing. Tools of this category focus on the processing of large-scale graph-based data.

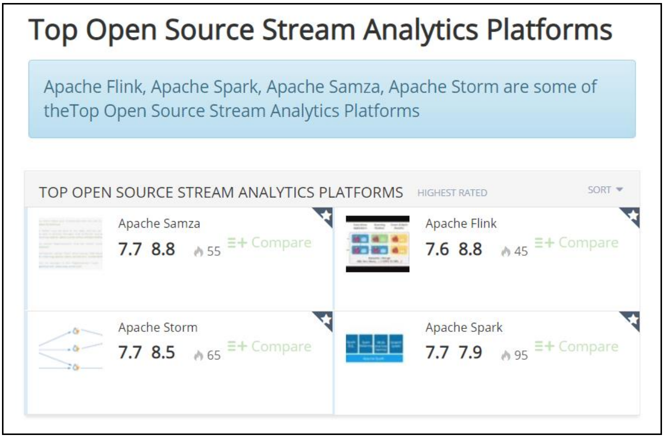
GraphX is a distributed graph analytics library that extends Spark for graph processing. GraphX allows performing iterative graph processing within a single Spark-based system [13, 14].

Giraph is a graph processing system designed to run over Hadoop using MapReduce [14,20]. Pregel is Google's scalable platform for processing large-scale graphs.

Complex Event Processing. Complex Event Processing (CEP) tools perform tracking and processing streams of incoming events in order to identify patterns and significant events and to derive timely actions and responses. CEP tools shall meet the following requirements: low latency; high throughput; ability to perform complex patterns analysis. Stream data processing frameworks can be used for CEP if they meet requirements for latency and throughput.

Flink is an open source platform which is widely used for CEP. There are also several proprietary platforms which are proposed as CEP capable.

The PAT RESEARCH [23] provides comprehensively wide analytics concerning Big Data software tools. Figure 11.6 shows the top of open sources stream analytics platforms rated by PAT RESEARCH.



Top Open Source Stream Analytics Platforms

Apache Flink, Apache Spark, Apache Samza, Apache Storm are some of theTop Open Source Stream Analytics Platforms

TOP OPEN SOURCE STREAM ANALYTICS PLATFORMS HIGHEST RATED

Apache Samza 7.7 8.8 55

Apache Flink 7.6 8.8 45

Apache Storm 7.7 8.5 65

Apache Spark 7.7 7.9 95

Figure 11.6 – Open Sources Stream Analytics Platforms Rated by PAT RESEARCH in November 2018

11.2.4 Control and service support layer

Control and service support layer combines tools related to the management of tools and services running on Big Data systems. Table 11.4 shows the ecosystem of this layer tools.

Resources management. Resources management tools manage computing resources (CPU, memory, storage, ports) of all nodes included into a cluster. The resources available on each cluster node are pooled together, then reasonably utilized and shared across multiple services [SDA, BDC]. Modern resources managers (Mesos, YARN, Ambari, ZooKeeper and other) are independent components that could be paired with most of the tools of distributed Big Data systems.

Scheduling. Scheduling tools automate the execution of repetitive tasks and jobs performed in a distributed data storage environment. Scheduling tools support services such as feed retention, replications across clusters, archival etc [9]. Execution of services could be triggered by time or data availability. Scheduling tools allow complex workflows to be constructed from lower level jobs.

Security. Security tools provide measures for Big Data privacy and security. The typical features are authentication and authorization of users and administrators, data encryption, and auditing of user and services activities related to data access [9]. Many Big Data tools include built-in basic security functions, dedicated tools such as Sentry, Ranger or Knox are able to provide comprehensive protection of Big Data systems.

Benchmarking. These tools provide benchmarking of Big Data clusters with standardized profiling suites. A profiling suite is a representative set of synthetic big data jobs. Job execution metrics are used to evaluate and optimize the Big Data cluster infrastructure and configuration [1].

Table 11.4 – Control and Service Support Ecosystem

Control and Service Support	
Resources management	Apache Mesos
	Hadoop YARN
	Apache Ambari
	Apache ZooKeeper
	…
Scheduling	Apache Oozie
	Apache Falcon
	Apache Falcon
	…
Security	Apace Ranger
	Apache Sentry
	Apache Knox
	…
Benchmarking	PUMA Benchmark
	Hadoop GridMix
	…
Programming Languages	Java
	Python
	R
	…

Programming languages. Big data tools are written in a number of programming languages, including Java, Python, C++, Scala, and others. For example, Hadoop itself and most Hadoop applications are written in Java, Spark is written in Scala. Nevertheless, most Big Data tools provide API at least for Java, Python, and R that are becoming the most popular languages for data processing [13].

11.3 Architectures of Big Data systems

During the last decade, dozens of Big Data tools and technologies have emerged (see Section 11.2); in order to implement a Big Data project, the hardest activity is selection the technologies to use in your project. To simplify the development of a consistent distributed architecture of the Big Data project several reference architectures have been introduced recently. Two of the most common data processing architectures that represent state-of-the-art real-time data processing are known as Lambda and Kappa. Let’s consider the Lambda and Kappa architectures in detail and highlight the pros and cons each of them.

11.3.1 Lambda architecture

Lambda architecture was proposed by Nathan Marz in 2012 [BDD]. To handle low-latency reads and updates in a scalable and reliable way, Lambda architecture unifies real-time and batch processing in a single framework. Lambda architecture, shown in Figure 11.7, consists of three layers: Batch layer, Streaming layer and Serving layer.

Batch layer stores all of the incoming data in its raw form and performs batch processing on the data. Batch processing occurs on some interval of time; the interval duration could vary from hours to days. The result of this processing is a batch view. MapReduce or Apache Spark are good examples of tools that can be used to implement this layer.

Streaming layer (also known as Speed layer or Real-time layer) analyzes data in real time. Streaming layer supports the serving layer to reduce the latency in responding the queries The result of this processing is reported as a real-time view. Stream processing tools like Apache Spark or Apache Storm are used at this layer.

Serving layer integrates results from the batch and streaming layers for relevant querying with low-latency. Low-latency NoSQL technologies such as HBase, Impala or Hive can be used to implement this layer.

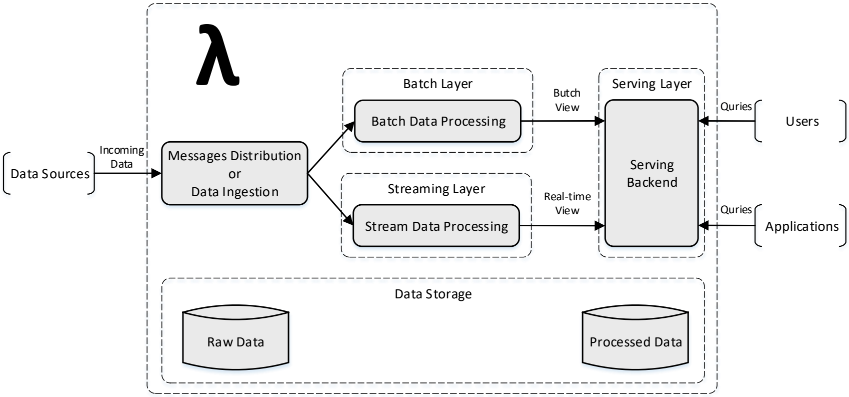


Figure 11.7 – Lambda Architecture

A widely known functional equation defines any query in Big Data system which is based on Lambda Architecture [19]:

query = function [(batch view), (real-time view)]

Lambda architecture can be deployed for those Big Data systems where [16]:

- Queries are required to be served on the ad-hoc basis using the immutable data storage.
- Near real-time responses are required and the system should be capable of handling various updates in the form of new data streams.
- None of the stored records shall be erased and it should allow the addition of updates and new data to the database.

The pros and cons of Lambda architecture are the following [16, 17]:

Pros:

- The batch layer of Lambda architecture manages historical data with the fault-tolerant distributed storage and ensures the low possibility of errors due to hardware failures and human mistakes.

- Lambda architecture provides a good balance of speed and reliability.

Cons:

- A data model of Lambda architecture is difficult to migrate or reorganize.
- Lambda is an architecture for asynchronous processing. Consequently, the computed results are not immediately consistent with the incoming data.
- The business logic is implemented twice in the streaming and batch layers. Therefore, the developers need to maintain code in two different frameworks and resulting operational complexity of systems implementing the Lambda architecture is high.

In summary, Lambda architecture achieves its goals but with the aid of high complexity and redundancy. Meanwhile, not all use cases require the combination of batch and streaming processing within the same system.

11.3.2 Kappa architecture

Kappa architecture was proposed by Jay Kreps in 2014 [18] as an alternative to Lambda architecture. The most important distinction of Kappa architecture is: All data flows through a single layer, using a stream data processing framework. For this architecture, incoming data is processed by a streaming layer and the results of processing are placed in the serving layer which is used to query the results. Figure 11.8 shows the basic diagram of Lambda architecture.

As a rule, Kappa architecture requires real-time message distribution, to capture and store events into a distributed and fault-tolerant unified log, Apache Kafka is commonly chosen for this role. The streaming layer can be implemented with real-time stream data processing tools such as Apache Storm or Apache Flink. NoSQL or NewSQL databases can be used at the serving layer.

A functional equation which defines any query in the Kappa architecture based Big Data system can be specified as follows [16]:

query = function (real-time view)

Kappa architecture can be deployed for those Big Data systems where [LVK]:

- Multiple events are logged into a unified system log.
- The order of the events and queries is not predetermined.
- The system shall be resilient and highly available.

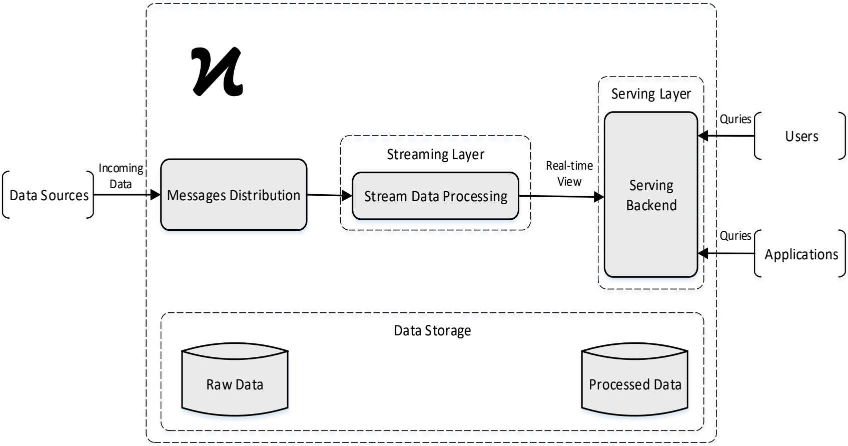


Figure 11.8 - Kappa Architecture

The pros and cons of Kappa architecture are the following [16, 17]:

Pros:

- Kappa architecture requires one code base, therefore development and maintenance are simplified
- The data model migration is simple, just a reprocessing of raw data is necessary.
- Kappa architecture can be used for horizontally scalable systems, just add an additional streaming layer if it is necessary.

Cons:

- The absence of the batch layer might result in errors during data processing or while updating the database that requires having an exception manager to reprocess the data or reconciliation.

- The architecture can only be applied for specific use cases and might not be able to manage intensive data processing.

Recently several new Big Data architectures have been introduced: Mu architecture, Zeta architecture, IOT architecture (iot-a) [26]. All these architectures aggregate the batch and the streaming processing; hence, could be considered as the adaptation of Lambda architecture for particular use cases and frameworks.

The choice between Lambda and Kappa architectures should be based on requirements for a developed Big Data system. Lambda architecture is more reliable as is it combines the benefits of batch and streaming processing to ensure fewer errors and is able to implement more sophisticated data processing. Kappa architecture provides the possibility to deploy a Big Data system which deals only with streaming events in a less expensive way.

11.4 Requirements for Big Data systems

The basis for successful Big Data project is clear and precise requirements for the developed system. In this section, we briefly describe a typical Requirements Engineering schema for Big Data applications (see Figure 11.9) and provide use cases that could help us understand the requirements for an IoT Big Data system.

11.4.1 Stakeholder Requirements for IoT Systems

Requirements for a Big Data system, that is integrated within an IoT system as a subsystem, should be derived from IoT system top-level requirements, so-called “stakeholder requirements”. Stakeholder is an individual or organization that has an interest in a system.

Business requirements specify the business objectives of a stakeholder and usually describe the following: business benefit or competitive threat, project cost, time to completion, project risks [7].

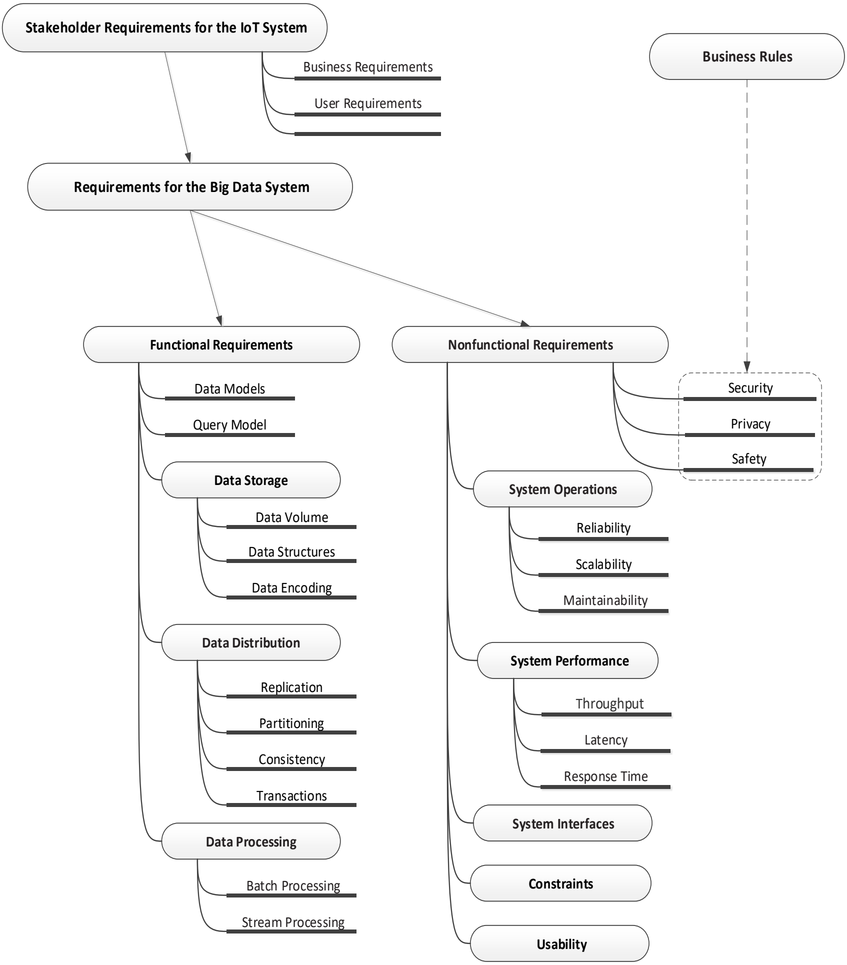


Figure 11.9 – Big Data System Requirements

User requirements define specific classes of system users, specify the generalized description of the expected system functionality and its inherent constraints from the user's point of view, and also include descriptions of the users desired product attributes or characteristics. User requirements become the most significant input for system requirements and guides the design, deployment, operations, and evolution of the system [7]. The common ways to represent user requirements are use cases. Typical IoT system use cases [IoT] help us understand the requirements on a build-in Big Data system.

Sensor network. The IoT system acts as a data gathering system for a set of sensors. The Big Data subsystem shall provide storage of collected data.

Alert system. The IoT system gathers and analyzes data from sensors. Alerts are generated when particular criteria are met. The Big Data subsystem shall support the descriptive and diagnostic analytics of data.

Analysis system. The IoT system gathers and analyzes data from sensors, but in this case, the analysis is predictive or prescriptive. The Big Data subsystem shall include components for predictive and prescriptive analytics using statistical techniques and machine learning algorithms.

Control system. The IoT system uses sensor data as the input for control algorithms that generate outputs for actuators. The Big Data subsystem shall implement complex event processing.

Table 11.5 (adapted from [15]) presents IoT use cases for different industries and provides the average values for such key characteristics as typical bandwidth, acceptable response time, and required analytics.

11.4.2 Functional requirements for Big Data systems

Functional requirements specify the functioning (behaviors) of the developed Big Data system under specific conditions. Functional requirements describe what the designer must implement to satisfy stakeholder requirements. During developing a Big Data system requirements specification, the following functional requirements should be considered as important to specify [20, 21, 22].

Data models requirements specify the data representation at each of the system layers. **Query model** requirements define the data querying features. **Data storage** combines requirements for the **data volume**, **structures**, and **encoding**. **Data distribution** – if distributed system is involved in storage and retrieval of data, specify requirements for **replication**, **partitioning**, and **consistency** of data, as well as **transaction processing**.

Table 11.5 – IoT use cases for different industries, adapted from [PLE]

Industry	Use Cases	Typical Bandwidth	Response Time	Analytics
Manufacturing	Operational technology. Brownfield. Asset tracking. Factory automation.	500 GB/day/factory part produced. 2 TB/minute mining operations.	Less than 1s	Recurrent neural nets. Bayesian networks/
Logistics and transport	Geolocation tracking. Asset tracking. Equipment sensing.	Vehicles: 4 TB/day/vehicle (50 sensors). Aircraft: 2.5 to 10 TB/day (6000 sensors). Assets tracking: 1 MB/day/beacon.	Stream: Less than 1s. Batch: Daily.	Rule engines.
Healthcare	Asset tracking. Patient tracking. Home health monitoring. Wireless health equipment.	1 MB/day/sensor.	Life critical: Less than 1s. Non-life critical: On each change.	Recurrent Neural Networks (RNN). Decision trees. Rules engines.
Agriculture	Livestock health and location tracking. Soil chemistry analysis.	512 KB/day/livestock head. 1000 to 10000 head of cattle per feedlot.	Stream: 1 s. Batch: 10 minutes.	Rules engines.

Energy	Smart meters. Remote energy monitoring (solar, natural gas, oil). Failure prediction.	100-200 GB/day/wind turbine. 1 to 2 TB/day/oil rig. 100 MB/day/smart meter.	Energy production: Less than 1s. Smart meters: 1 minute.	RNN. Bayesian networks. Rules engines.
Consumer	Real-time health logging. Presence detection. Lighting and heating. Security. Connected home.	Security camera: 500 GB/day/camera. Smart device: 1-1000 KB/day/sensor-device. Smart home: 100 MB/day/home.	Video: Less than 1s. Smart home: 1s.	Convolutional neural nets (image sensing). Rules engines.
Retail	Cold chain sensing. Point of sale (POS) machines. Security systems. Beaconing.	Security: 500 GB/day/camera. General: 1-1000 MB/day/device.	POS and credit transaction: 100ms. Beaconing: 1s.	Rules engines. Convolutional neural networks for security.
Smart City	Smart parking. Smart trash pickup. Environmental sensors.	Energy monitors: 2.5 GB/day/city (70K sensors). Parking spots: 300 MB/day (80,000 sensors). Waste monitors: 350 MB/day (200,000 sensors). Noise monitors: 650 MB/day (30,000 sensors).	Electric meters: 1 minute. Temperature: 15 minutes. Noise: 1 minute. Waste: 10 minutes. Parking spots: Every change.	Rules engine. Decision trees.

Data Processing defines requirements for implementation of the **batch** and(or) **stream** data processing [18, 19].

11.4.3 Nonfunctional requirements for Big Data systems

Nonfunctional requirements govern the interfaces, internal and external conditions or attributes of the system functioning. The most common nonfunctional requirements for Big Data systems are the following [20, 21, 22].

Requirements for **security, privacy,** and **safety** of the Big Data system, as a rule, are impacted by the "business rules" – government, industry or corporate regulations.

System operations requirements specify the **reliability**, **scalability**, and **maintainability** of the system.

Performance requirements define the **throughput**, **latency**, and **response time** of the system.

System interfaces determine the interaction of the system with external systems and interaction of the system's elements with each other. The following interface types should be specified for a Big Data system: user interface; hardware interfaces; software interfaces; communication interfaces.

Constraints limit the range of possibilities that are available to the system designer. Typical constraints prescribe to use proven frameworks or tools, existing design decisions or particular hardware.

Usability is the most important characteristic for such Big Data systems which include user-facing applications [21].

This section sets the foundation of the requirements engineering for Big Data systems. Due to the complexity and variety of the modern Big Data systems projects requirements on such a system could not be limited by the requirements which are presented in this section and shall be thorough specified for each particular Big Data system project.

11.5 Work related analysis

IoT is one of the fastest growing technologies. They are sources of a large amount of data that needs to be processed. Currently, there are a number of sources which are described separately: Big Data characteristics; Big Data categories and types; Big Data Platform etc. The development of new services requires an understanding of the relationship between Big Data characteristics, categories, types, platforms and tools. This section links these concepts. The characteristics of Big Data associated with several Vs – in different sources a quantity of specified Vs could change from three [1,4] to ten [5]. Updated analytic data are presented in [2], where a list of the 42 V's of Big Data and Data Science.

In data science and big data you will come across many different types of data, end each of them tends to require different tools and techniques. The main categories of data given in [1,3]. Three central questions concerning Big Data are how to classify Big Data, what are the best methods for managing Big Data, and how to accurately analyze Big Data. Although various methods exist to answer these questions, no single or globally accepted methodology is recognized to perform satisfactorily on all data and can be accepted since Big Data Analytics tools have to deal with the large variety and large scale of data sets. This question is analyzed in [6÷12]. After understanding how much and what kind of Big Data need to be processed, the developer proceeds to the choice of the IoT implementation model, platform. This question is deeply analyzed in [6÷11]. The sources [13,14,15] provide detailed information about the use of Apache Spark and Hadoop technologies. In order to implement a Big Data project, the hardest activity is selection the technologies to use in your project. To simplify the development of a consistent distributed architecture of the Big Data project several reference architectures have been introduced recently. Two of the most common data processing architectures that represent state-of-the-art real-time data processing are known as Lambda and Kappa [16,17,23]. The basis for successful Big Data project is clear and precise requirements for the developed system. This question is deeply analyzed in [18÷22].

The project partner universities and other research organizations are actively working on the subject of Big Data and Internet of Things and these studies were analyzed during the preparation of the lecture material. For example:

- University of Newcastle upon Tyne, UK “Bringing the ‘Internet of Things’ into everyday use”[24];
- Multidisciplinary Digital Publishing Intitute, Switzerland “Industrial IoT Monitoring: Technologies and Architecture Proposal” [25].

Conclusions and questions

This section has provided a brief introduction to integration of the Big Data and IoT technologies. The Big Data subsystem is an essential component for any IoT system owing to the volume and variety of data produced by numerous sources and necessity to process this data at near real-time to make decisions. We introduced the Big Data platform stack and identified the tool ecosystems of its layers. Based on this introduction we presented the most known and proven-in-use architectures of Big Data systems. We have also outlined the requirements for the Big Data systems integrated within the IoT system. It can be concluded that this section provides a sufficient basis for self-contained research and development of such a system.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. What are the V characteristics of Big Data? Briefly, define so much as you can.
2. Explain the difference between structured, unstructured, and semi-structured data.
3. This chapter categorized the Big Data Platform Stack into 5 layers. Name and define each of the 5 layers.
4. What are the key advantages provided by the distributed file system approach?
5. What are the four basic types of NoSQL databases?
6. Explain a possible role of the relational databases in Big Data applications.
7. What are the factors to implement the messages distribution?
8. How is stream processing different from batch processing?
9. How is complex event processing different from stream processing?
10. What are the specific architectural features of Lambda Architecture?
11. Explain the pros and cons of Lambda architecture.
12. What are the specific architectural features of Kappa architecture?
13. Explain the pros and cons of Kappa architecture.
14. This section described 4 typical use cases of IoT systems. Describe user requirements on a build-in Big Data system for each of the 4 use cases.
15. What are the key characteristics of the Big Data system performance?

References

1. Cielen, Davy, Meysman Arno D.B., Ali Mohamed Introducing Data Science. Big data, machine learning, and more, using Python tools, Manning Publications Co. Shelter Island, 2016, 322p., ISBN: 9781633430037.
2. Shafer T. The 42 V's of Big Data and Data Science.- [electronic source: https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html].-2018.

3. Mysore D., Shrikant K., Jain S., Big data architecture and patterns, Part 1: Introduction to big data classification and architecture. How to classify big data into categories. IBM developer Works, September 17, 2013.

4. Hanes, D., Salguero, G., Grossetete, P., Barton R., Hanry J IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, 543p.

5. Bhatt, C., Dey, N., Ashor, S., Amira Internet of Things and Big Data Technologies for Next Generation Healthcare, Studies in Big Data. Volume 23., Springer International Publishing AG 2017, 386p.

6. Gilchrist, A., Thailand, N., Industry 4.0: The Industrial Internet of Things, Banqken, Nonthaburl Thailand 2016, 259p.

7. Nath, S., Stackowiak, R., Romano, C., Architecting the Industrial Internet, The architect’s guide to designing Industrial Internet solutions, Packt Publishing, 445p.

8. Rayes, A.,Salam, S., Internet of Things—From Hype to Reality The Road to Digitization, Springer International Publishing AG 2017, 350p.

9. Taheri, J., Big Data and Software Defined Networks, Published by The Institution of Engineering and Technology, London, United Kingdom 2018, 504p.

10. Buyya, R., Dastjerdi, A.V., Internet of Things. Principles and Paradigms.- Morgan Kaufmann is an imprint of Elsevier, 2016 Elsevier Inc, 347p.

11.Harrison, G., Next Generation Databases. NoSQL, NewSQL, and Big Data, Apress Media 2015, 244p.

12. Somani Arun K., Deka Ganesh Chandra Big Data Analytics Tools and Technology for Effective Planning, CRC Press is an imprint of Taylor & Francis Group 2018, 414p.

13. Guller, M., Big Data Analytics with Spark. A Practitioner’s Guide to Using Spark for Large-Scale Data Processing, Machine Learning, and Graph Analytics, and High-Velocity Data Stream Processing, Apress Media 2015, 290p.

14. Ankam, V., Big Data Analytics, A handy reference guide for data analysts and data scientists to help to obtain value from big data analytics using Spark on Hadoop clusters, 2016 Packt Publishing, 325p.

15. Lea, P., Internet of Things for Architects, Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics,and security, 2018 Packt Publishing, 225p.

16. Samizadeh, I., A brief introduction to two data processing architectures — Lambda and Kappa for Big Data.- [electronic source:

https://towardsdatascience.com/a-brief-introduction-to-two-data-processing-architectures-lambda-and-kappa-for-big-data-4f35c28005bb].-2018.

17. Ounacer, S., TALHAOUI, M.A., Ardchir, S., Daif A., and Azouazi M. A New Architecture for Real Time Data Stream Processing, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 11, 2017.

18. Kreps, J., I love Logs. Event Data, Stream Processing, and Data Integration, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.,2015, 59p.

19. Marz, N., Warren, J., Big Data. Principles and best practices of scalable real-time data systems, 2015 by Manning Publications Co. , 330p.

20. Harrison, G., Next Generation Databases. NoSQL, NewSQL, and Big Data, Springer Science+Business Media New York , 2015, 244p.

21. Malaska, T., Seidman, J., Foundations for Architecting Data Solutions. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2018, 189p.

22. Kleppmann, M., Designing Data-Intensive Applications, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, 2016, 491p.

23. PAT RESEARCH B2B Review, Buying Guides & Best Practices. URL://www.predictiveanalyticstoday.com/.

24. James, P., Bringing the 'Internet of Things' into everyday use.
URL:https://www.ncl.ac.uk/press/articles/archive/2018/04/internetofthings (9 April 2018).

25. Raposo, D., Rodrigues, A., Sinche, S., Silva, S.J., Boavida, F., Industrial IoT Monitoring: Technologies and Architecture Proposal. URL:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6210632/ (21 October 2018).

26. Behera, R.K., Singh, K., Big Data Ecosystem-Review on Architectural Evolution. URL: https: // www.researchgate.net/ publication/323387483.

PART IV. IOT TECHNOLOGIES FOR CYBER PHYSICAL SYSTEMS

12. CPS AND IOT AS A BASIS INDUSTRY 4.0

Dr. H. I. Vorobets, Dr. O. I. Vorobets, Mr.S. V. E. Horditsa (ChNU)

Contents

Abbreviations

NTM – the newest technologies and means

IoS – Internet of Services

CPS – cyber-physical systems

SP – smart plants

IR – intellectual robots

CP – cybernetic part (cyber part)

ECM – embedded computer means

HPHSC – high-performance highly specialized calculator

CPO – cyber-physical objects

CC – cyber-components

ACS – automated control systems

ECS – embedded computer systems

NIST – National Institute of Standards and Technology

SNSS – Smart Networked Systems and Societies

M2M – machine-to- machine, inter-machine

ML – machine learning

DM – data mining

BPS – basic physical system

FPGA – field-programmable gate arrays

SRIS – self-reconfiguring intelligent computer system

HMA – hierarchical-modular approach

DCOS – designing complex objects and systems

GCP – GOOGLE Cloud Platform

AWS – Amazon Web Services

12.1 Basic principles for the organization and functioning of ecosystems of the Internet of things and cyber-physical systems

The world's leading scientists, economists, businessmen, managers accept that modern society comes into so-called post-industrial or informational stage of development, which has already been called the Fourth Industrial Revolution, or “Industry 4.0” [1,2]. The main feature of this process is the formation of self-regulated production systems based on the new concept of products added value creation using the newest technologies and means (NTM): Internet of things and services (IoT & IoS), cyber-physical systems (CPS), smart plants (SP), intellectual robots (IR), etc. [3–5]. The increasing of the systems functionality due to the widespread implementation of intelligent ("smart") algorithms is their peculiarity. It's been provided by the presence of a cybernetic part (CP, cyber part) as a compulsory component of production systems. Thus, the intellectualization of physical objects, industrial means, and technical and biotechnical systems foremost, which is achieved through the implementation of embedded computer means (ECM) and/or high-performance highly specialized calculators (HPHSC), leads to the formation of new technological ecosystems – IoT and CPS [6,7,19].

In the first approximation, the technology ecosystem is been understood as the synergy of applied problem-oriented software and the cyber-physical environment [6], which creates conditions for the self-organization and development of modern CPS due to the use of IoT technologies. In a broader sense, the prospects for the open dynamic ecosystems creation that combine heterogeneous ecosystems of humans, computing processes and smart things (Smart Things) are considered and interact with each other to achieve common business goals [7], including cloud computing technologies (Cloud computing). The term Elastic Systems is proposed for such ecosystems, and a new field of research, Elastic Computing, is predicted.

12.1.1 Evolution, Standards, Development Prospects for the IoT and CPS

For the first time, the use of the terms "Internet of Things" and "Cyber-Physical Systems" as the object of the subject area and the research direction is associated with the names of Kevin Ashton (1999) and Helen Gill (2006) respectively; however, currently there are no generally accepted terminology and standards. One of the latest works that can be considered the first attempt to develop standards guidance is the US National Institute of Standards and Technology (NIST) report on March 2019, edited by Christopher Greer [8]. In this paper, a comparative analysis of the essence and evolution of the definition, terminology and subject area of the CPS and IoT , which are currently used by various authors in open annual scientific publications, that increased from 32-35 in 2006 to 1,000-14,000 in 2017, is performed.

As the main criteria for the terminology development in both cases, the kits of deployed components and the functions performed by them are usually used. Note that although these research areas are based on entirely different concepts, in recent years a gradual assimilation of their definitions is observed and the creation of the hybrid model CPS/IoT is possible. In particular, the CPS arose on the basis of the development concept of improvement for the mechatronic [5,9], embedded [10] and distributed computing systems, the propagation of the cybernetic approach [11,12] for the functional improvement of physical objects and processes [13]. IoT technology was founded on the base of radio frequency identification techniques for the location and state of any physical objects or things, and was used to improve the logistics of their delivery [14–16]. Assimilation and a certain substitution of IoT concepts by the terms of CPS in the last decade took place, in our opinion, due to the rapid development of microelectronics, and primarily – minimizing mass and overall performance, reducing energy consumption and cost, improving the reliability and versatility of embedded computer systems [17,18]. The last one provided exactly the process of "intellectualizing" things by increasing the computing power of used ECM, and expanded the possibilities for connecting them to Internet communications using standard network protocols. Thus, more and more things acquire signs of "cyber-physical" and can support IoT technology.

However, while the process of convergence and hybridization of the CPS/IoT model is creating the preconditions for the development of new unified, hybrid discrete and continuous methods for the design, creation and operation of complex CPS and IoT devices [8], in our opinion, it can also have a negative impact on providing the reliability of such systems due to their excessive complexity and poorly protected access, especially for critical systems infrastructure. Obviously, the approach in the form of the problem decomposition into two components, where the CPS is the basis of intellectualized system for information processing and carrying out the problem-oriented functions, and the IoT acts as a set of existing and new network technologies and services to provide communication requirements for the information exchange between the objects of the CPS itself, and between the CPS and the environment or the person of user or operator, can be more effective during the analysis and synthesis of complex smart-systems. IoT technologies also have to perform the functions of maintaining the synthesis/configuration/ /reconfiguration of the distributed high-performance cyber-component of the branched CPS or synchronizing the interactions of a certain number of autonomous CPSs for their self-organization.

12.1.2 Conceptual diagrams of the Internet of Things and Cyber-Physical Systems

Let's analyze the currently known conceptual principles for implementing the architecture and functioning of the CPS and IoT. It is clear that the basic element to realize the functions, defined by the customer, must be things/objects, intellectualized to a certain rationally justified level; therefore in the pair of CPS and IoT the leading role is played by the CPS. Depending on performed applied tasks, CPS may also be autonomous, that is, they don't use or only partly/periodically use network communication tools as IoT technologies. Whereas systems based on IoT without using intelligent cyber-physical devices have no sense at all.

According to the definition of Edward A. Lee [13], who can also be considered one of the founders of the term and research on the CPS (his article "Cyber-Physical Systems – Are Computing Foundations Adequate?" was in program of NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, October 16-17, 2006 Austin), cyber-physical systems are understood as a combination of physical processes and calculations. However, physical processes reflect the dynamics of changes in states (cyber) of physical objects (CPO, which are the material formations of the surrounding world and the elements of CPS), and the environment where they exist. At the same time, computation is an information process for obtaining new knowledge from the observation and measurement results (quantitative data) of parameters of the objects status changes. In a more general sense, the term "calculation" may be interpreted as knowledge "awareness" about physical objects or the surrounding world, that is, the environment of physical objects existence. In general, the new knowledge is appointed to the system user – a human. However, if the system "object-process-calculator-object" is closed and functions autonomously, then we obtain a new quality of the material world – "smart" (intellectual) physical objects that "realize", "control" and "adjust" their existence in the surrounding environment.

The structural peculiarity of "smart" cyber-physics objects (CPO) is the presence of cyber-components (CC) in their composition. The hardware and software that provide a *controlled (conscious)* interaction of the physical component with the world based on mathematical information processing is understood as the cyber component. Due to the complexity of hardware implementation and, accordingly, to a set of implemented algorithms, CCs can be represented in different forms from simple digital or analog-digital automatons to microcontroller devices, or even complex computer systems. Accordingly, the functionality of the CPO will differ. The following questions arise: do all physical objects that contain a cybernetic component or even use it to improve the performance of their functions can be considered "smart" (intellectual); can a simple set of CPOs be a cyber-physical system?

The approach applied to the classification of intelligent devices and machines in [5] shows that the most simple mechanical machines that "make decisions" for the transition from one functionality to another, etc., due to changes in the state of the sensor element, which is sensitive to the parameters of the controllable environment or process, already have the elemental signs of "intelligence". The development of electronics and computer means allowed the implementation of complex algorithms for information processing in automated control systems (ACS) and led to the creation of mechatronic systems and intellectual robots [1-3], and miniaturization of the element base – to the development in the last decade of micro- and nanomechatronics [5]. The information amount that is "computed" in controllers and microcomputers of embedded computer systems (ECS), becomes comparable with the amount of data, processed by supercomputers 20-30 years ago. ECS are used not only in ACS by production systems and technological processes, but also to optimize the functioning of individual subsystems and modules of complex systems. More and more often, technology for artificial intelligence (artificial intelligence, AI), genetic algorithms (GA), fuzzy logic (FL) and so on are used in this case.

How do cyber-physical systems distinguish from mechatronic? According to [5], using artificial intelligence in robotic technical systems allows to recognized them as intelligent (or "smart") only if the used hardware and software solutions are aimed to obtain new information and knowledge, and they deal with the solution of functional tasks for the main purpose of the system – the implementation of the main target function. If the AI tools perform additional functions, then such a system is mechatronic, and it works by given algorithms in automated or automatic mode.

But in this case, the functional parameters of the CPO and CPS, describing their own essence and perfection, for example – energy efficiency, reliability, resistance to external influences, self-development ability, possibility of implementing their own adaptation to a changing environment and correction of the functions performed, etc., remain unnoticed.

The five-level 5C conceptual model of the CPS [20] (Table 12.1, Figure 12.1) solves the indicated contradiction to a certain extent. The abbreviation clearly describes the abstract functional essence of the 5C model: Connection –

Conversion – Cyber – Cognition – Configuration, and reveals both the stages of information processing and their principal content. In particular, the lowest level – connection – already involves some intellectualization of the used CPO in shape of "smart" sensor networks, implemented on the basis of intelligent sensors with the support of the "Plug & Play" function, that is, autoidentification and autoconfiguration of the system. At the second level – conversion – not only a simple data transformation into useful information is provided, but also the elements of the physical state self-analysis of components and the CPO system, and the degradation of their parameters are covered. However, the final decisions on the synthesis of new CPS configurations are based on the analysis of simulation results at the fourth level – cognition, that is, after decoding and visualizing the numerical calculations results and behavior modeling of the system at the third level – cybernetic. The fifth level – configuration – has to ensure the implementation of system "self-organization" function in accordance with the new synthesized model of its next state.

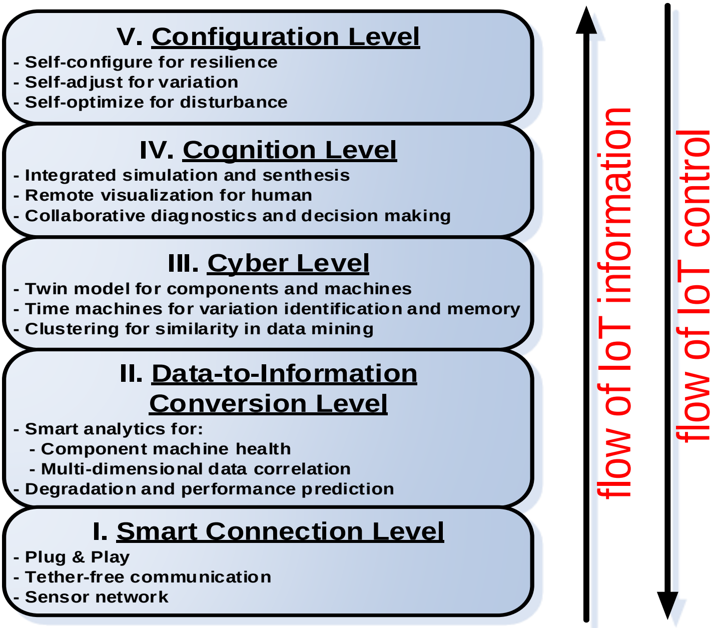

V. Configuration Level
- Self-configure for resilience
- Self-adjust for variation
- Self-optimize for disturbance
IV. Cognition Level
- Integrated simulation and senthesis
- Remote visualization for human
- Collaborative diagnostics and decision making
III. Cyber Level
- Twin model for components and machines
- Time machines for variation identification and memory
- Clustering for similarity in data mining
II. Data-to-Information Conversion Level
- Smart analytics for:
- Component machine health
- Multi-dimensional data correlation
- Degradation and performance prediction
I. Smart Connection Level
- Plug & Play
- Tether-free communication
- Sensor network
flow of IoT information
flow of IoT control

Fig. 12.1 – Five-level 5C conceptual model of CPS [20]

It is worth noting that in most cases, real CPSs are multifunctional, so they implement parallel processes of multithreading information processing from the lowest to the highest level of the 5C model. For example, in stand-alone cars the data is recorded simultaneously from several obstacle sensors and the received information is used at the same time for dynamic correction of at least two control functions – speed and direction of movement.

The basic conceptual IoT model (Fig. 12.2) provides the interaction of certain executive/information devices/objects/CPOs between each other or with the user via shared IoT Gateway [21].

Table 12.1 – Functional features of the 5C CFS model

Level	5C Abbre-viation	Functions	Attributes
V	Configu-ration	System self-configuring depending on the needs of the tasks performed to ensure its resilience	Self-configure for resilience
		Self-adaptation and system adjusting depending on the variation of input information and taken decisions	Self-adjust for variation
		System self-optimization in case of deviation from the given model of the trajectory or disturbance	Self-optimize for disturbance
IV	Cognition	Complex integrated simulation and synthesis of proposals for the needs of new system configurations	Integrated simulation and synthesis
		Remote (distant) access and visualization of information for the user	Remote visualization for human
		Collaborative complex system diagnostics and decision making about its functional	Collaborative diagnostics and decision making

III	Cyber	Input data processing for dual/twin component model and system	Twin model for components and machines
		Using Time Machine to identify variations and memory and synchronize data exchanges	Time machine for variation identification and memory
		System clusterization on the similarity principle in the intellectual analysis of data	Clustering for similarity in data mining
II	Conversion	Input data formatting and intelligent analysis for: Functional state determination of the system component; Correlation analysis of multi measurable/ dimensional data.	Smart analysis for: component machine health; multi-dimensional data correlation.
		Prediction of possible degradation and productivity of individual components and the system as a whole	Degradation and performance prediction
I	Connection	Auto identification, auto-connect, and autostart of ultimate smart devices and higher-hierarchy devices	Plug & Play
		Application of wireless communication technology and data transmission	Tether-free communication
		Information gathering concerning the environment conditions and the system itself using sensor networks	Sensor network

Most often servo drives take on the role of actuators, sensors and digital data converters of different physical nature signals, geo positioning devices – of information sources. Users (information consumers) can be both individuals and custom devices: computers, smartphones, personalized gadgets, etc. A mandatory requirement for all the elements of the IoT technology is the interfaces availability for commutation and data transmission through a communication gateway, based on network routers, switches, access points.

From the standpoint of the classical model, the process of information processing and decision-making in IOT technology is much similar to the stages described in the 5S CPS model. It also involves such processes as: 1) registering of measured signals from terminal devices; 2) formatting of the received data; 3) analysis of the results of mathematical information processing; 4) making a decision on the analysis results. However, system reconfiguration, according to the highest level of 5C IOT model, is not obligatory. Such a stage is possible, but depends on the technical solutions used in the system, and therefore IOT in the classical version can be considered a mechatronic system or ACS with the use of Internet communications, operating according to a given algorithm.

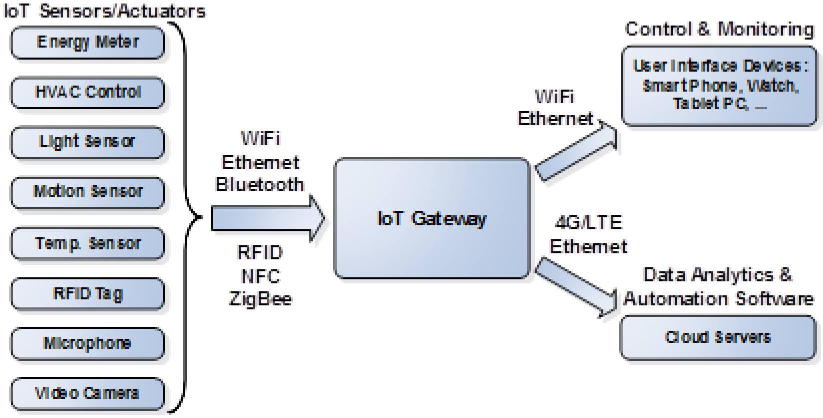


Fig. 12.2 – Conceptual IoT model [21]

If IoT end devices are implemented on the basis of CPO, then a hybrid CPS/IoT model can be proposed according to [8] (Fig. 12.3). Here, individual CPO or CPS interacts with each other either through a shared Gateway or through a local server. Similarly, IoT technology allows users to access CPS, or allows CPS to access cloud servers and services directly or indirectly through a local server.

Other approaches to the description of the CPS and IOT functional based on the convergence principle of systems [23] are also known, in which the fundamental difference between these technologies is not distinguished, and used terminology is associated only with certain scientific and industrial environments, countries or scientists. In particular, for example, the "Internet" environment is distinguished in the form of a virtual Cyber World that interacts with the Physical World, which includes IOT, and the CPS is considered as a system of reflections and interactions of the Physical and Cyber World components (Fig. 12.4.). However, in our opinion, such an approach does not correspond to reality. From the hardware-software, or even the model point of view, the "Internet" means are also material entities of the Physical World, which perform specific functions of information processing and communication between the objects. And the IOT model is also hard to imagine outside the cyberspace "Internet".

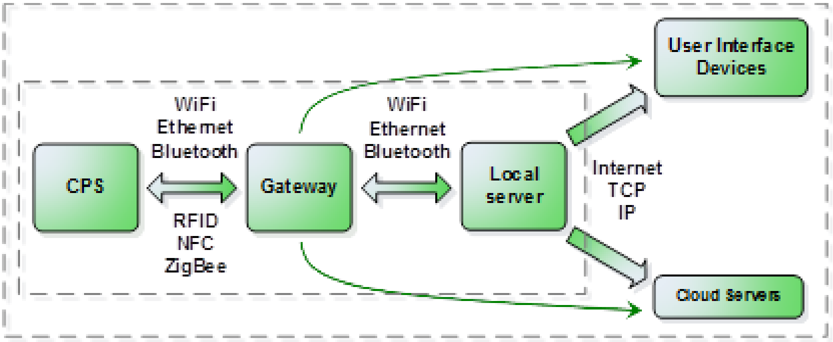


Fig. 12.3 – Conceptual hybrid CPS/IoT model

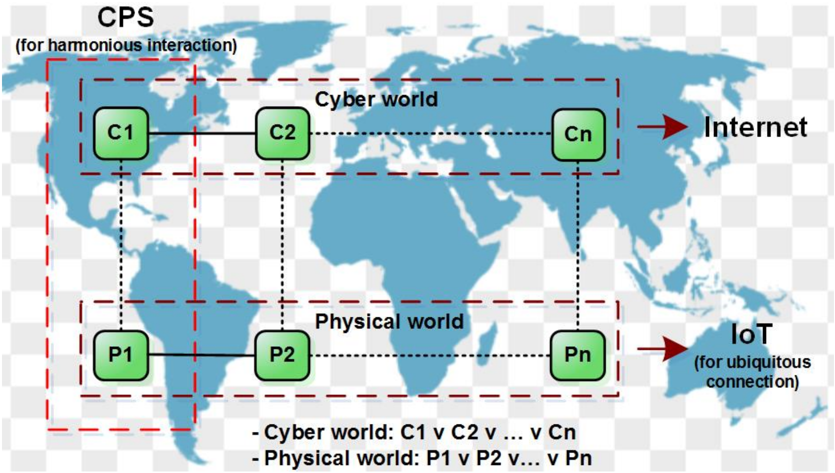

CPS
(for harmonious interaction)
Cyber world
C1
C2
Cn
Internet
Physical world
P1
P2
Pn
IoT
(for ubiquitous connection)
- Cyber world: C1 v C2 v ... v Cn
- Physical world: P1 v P2 v... v Pn

Fig. 12.4 – Conceptual CPS/IoT hybridization model [23]

While spreading the IoT model to the “Internet of Everything” (IoE) model [24], social networks and communications, IoT and CPS components, and various areas of human activities are included. Based on this approach, the concept of the "Smart World" and the network-centered ecosystem of "Smart Networked Systems and Societies" (SNSS) is formed (Fig. 12.5). Using network communications as the basis of SNSS provides in the future free direct communication not only for gadgets, but also for system components of different nature, including the person as a system object [25].

In [26], the basic concept of the CPS architecture is considered as a three-layer model: the lowest layer is level of information perception (the perceptual layer); the second one – the level of data transmission to the user (the data transmission layer); the highest third layer is the application layer for various industries: smart homes, cities, health care, industry, etc. Wherein, IoT technologies are considered to be the elements of the lowest level, providing the communication of terminal devices.

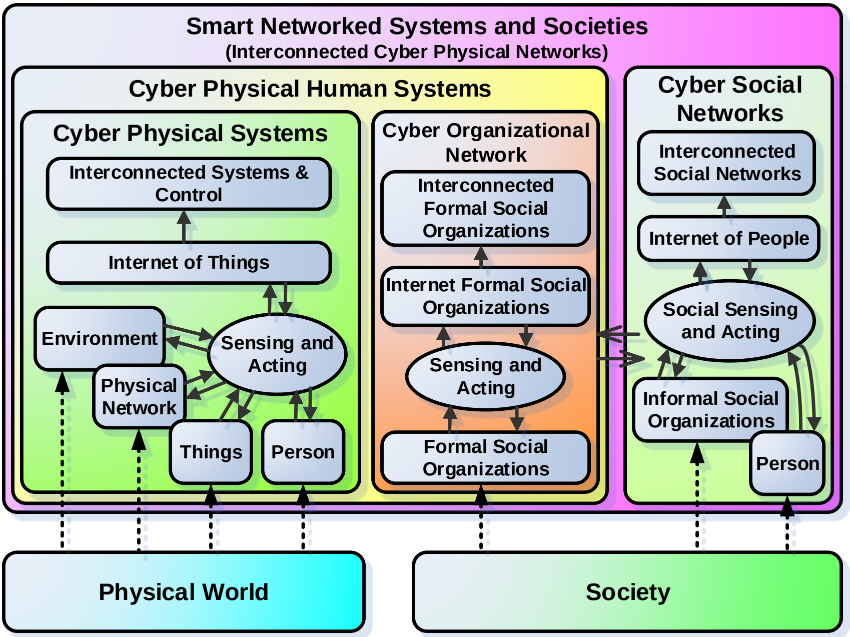

Smart Networked Systems and Societies
(Interconnected Cyber Physical Networks)
Cyber Physical Human Systems
Cyber Physical Systems
Interconnected Systems & Control
Internet of Things
Environment
Sensing and Acting
Physical Network
Things
Person
Cyber Organizational Network
Interconnected Formal Social Organizations
Internet Formal Social Organizations
Sensing and Acting
Formal Social Organizations
Cyber Social Networks
Interconnected Social Networks
Internet of People
Social Sensing and Acting
Informal Social Organizations
Person
Physical World
Society

Fig. 12.5 – Conceptual model of SNSS/IoE hybrid model [24]

In [27] a detailed analysis of the development stages of computer technology from the first computers, embedded computers and systems, personal computers, etc. to modern CPS and IoT technologies is given. Remarkably, IoT is considered there as a stage and a means where the incipience of so-called "resilient" or "elastic" computer systems begins. In a broader interpretation, resilience is understood as a "functional stability" of a system, i.e. the ability to "elastically" respond to external disturbances, while preserving the high ability of consistent execution of the maximum number of tasks provided for by the technical system order [28].

It's worth noting that the lack of accepted international standards for clear definitions (glossaries) and general ontology of the subject area, architectural decisions and design principles of CPS, IOT, IoE, SNSS and the technologies used by and for them often leads to a free and ambiguous interpretation of the basic principles and the substitution of concepts from one technology to another. Certainly, we don't mean synergy or convergence of related technologies, which contributes solving complex problems in the field of electronic engineering, automation and industrial computing.

The solution of this problem is facilitated by the work of specialists in many educational and in scientific institutions in the world. The leading role is played by scientists from NIST, NSF, The University of California in Berkeley [10, 39], Massachusetts Institute of Technology and other institutions in the United States, in particular C. Greer, M. Burns, D. Wollman and E. Griffor, E. Lee and S. Seshia, S. Sarma, D. Brock and K. Ashton, and others. At the University of Florida, for example, at the Department of Electrical and Computer Engineering, like in many other US educational institutions, the "EEL 6673 Cyber-Physical Systems Identification and IoT Applications" course was introduced [65].

In Europe, the leading schools of CPS and IOT are at the Newcastle University (NCU) [63] and at the Leeds Beckett University [64] in Great Britain, at the Royal Institute of Technology (KTH) in Stockholm, Sweden [34-38, 40, 61, 62], at the University of Coimbra in Portugal [19, 32, 60], at the University of Pisa, Italy [41]. The project "Cyber-Physical European Roadmap & Strategy, CyPhERS" [33] by the 7th EU Framework Program (FP7-ICT, ICT-2013.3.4) was coordinated by representatives of Germany (Bernhard Schätz from fortiss GmbH (Coordinator) and Thomas Runkler from Siemens AG (affiliate partner)), Sweden (Mart in Törngren from Kungliga Tekniska högskolan, KTH), France (Saddek Bensalem from the Université Joseph Fourier, Grenoble), Italy (Roberto Passerone of the Università degli Studi di Trento), Great Britain (John McDermid from The University of York).

Good studies of J. Wan and colleagues [29] on the development of CPS and IOT technologies from inter-machine (M2M) information exchange to modern smart technologies, P. Ray (Sikkim University, India) [30] on the peculiarities of architectural IOT decisions depending on the application field, R. Alur et al. [31] on the analysis of tasks related to the computer technology of IOT are useful for improving existing educational Master's courses and developing new programs for Doctors of Philosophy [34-38, 42].

In Ukraine, this problem both in terms of education and R&D is developing at the National Aerospace University "KhAI", ta Zaporizhzhya National Technical University, at Lviv Polytechnic National University, at Yuriy Fedkovych Chernivtsi National University and others, where relevant educational programs are provided and CPS and IOT conceptual issues are explored [59].

12.1.3 Motivation and examples of IoT and CPS for industry and the human applications

Primarily broad possibilities of using technical and hardware-software solutions for the needs of industry and the humanities are the motivation for modern research in the field of CPS and IOT. Due to the convergence of research results in microelectronics, computer engineering, cybernetics, network technologies, biotechnology, medicine, industry, it was possible already at the present stage to achieve a new qualitative obtaining level of collecting, processing and analyzing information for human needs in the form of intelligent data transmission systems, telemetry and telecontrol for household needs, electronic medicine, technology of new materials, ecology, social communication, etc., based on CPS, IoT, IoE, SNSS.

The conceptual map of the CPS proposed in [43, 44] reflects the analysis of the field of CPS research and applications of in three aspects (Fig. 12.6): 1) what is the essence of CPS? 2) the requirements for the reliable functioning of the CPS, and 3) the field of CPS application. A similar approach with the emphasis on the social dimension of the application of development results is considered to be a motivation for the IoT research [45].

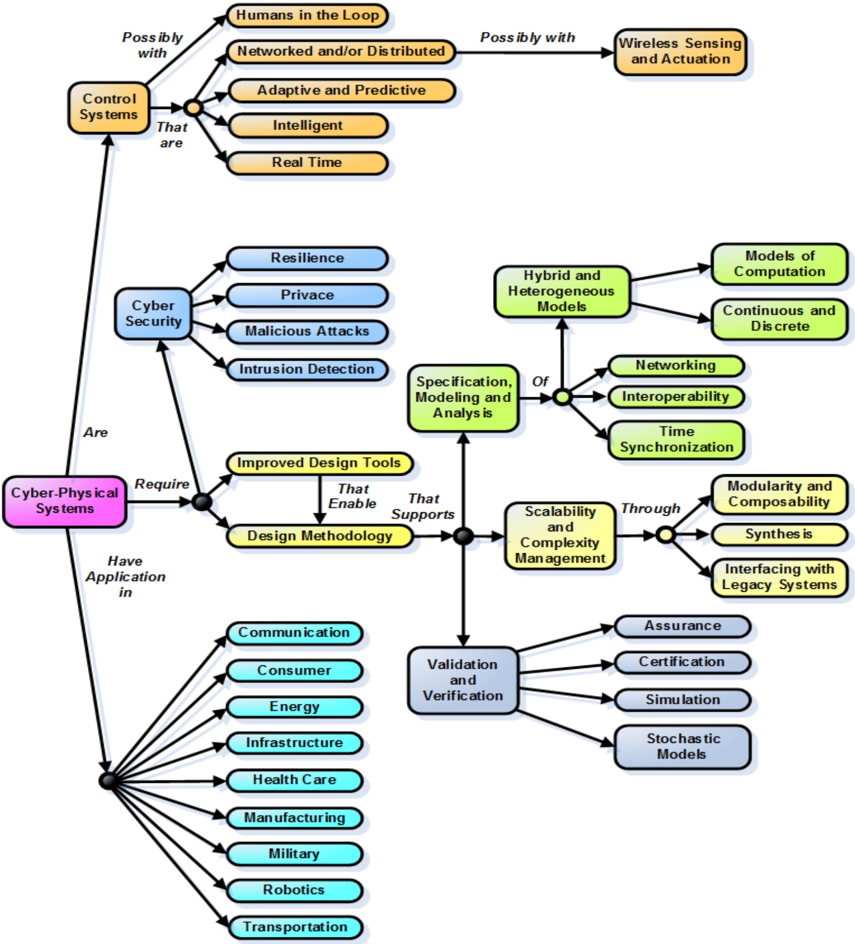


Fig. 12.6 – A generalized conceptual map [43] by aspects of the essence, needs and application areas of CPS

The essence of CPS [43] is the implementation of a control or monitoring network and/or distributed, adaptive, predictive, intelligent real time system. Human may be one of the links in the closed loop "receiving – processing – managing information – decision making" and act simultaneously as the beneficiary (participant) and the stakeholder (user) of the results, respectively, of the system implementation and operation.

In order to improve the CPS development, it is necessary to improve used methods and design tools [50].. Wherein, typical design problems are solved at first – system specification creation, its modeling, analysis and synthesis of the prototype. However, already at this stage, direct application of network technologies is possible to accelerate the design development, for example, parallelization of modeling, as well as laying the basis for the IoT technologies usage in the functional of CPS design. In particular, when designing multifunctional systems, depending on their application area and solved problem-oriented tasks, specific requirements for its complexity, scalability of the system and its components, management features are laid in the CPS architecture.

The structural perfection and functional reliability of the designed CPS is determined on two levels: verification and validation qualities of hardware and software solutions, and using cyber security tools and technologies for data and information flows.

The application of CPS certainly is not limited with about 10 directions listed in [43], and it determines the attractiveness of developments and research in this area. However, the analysis of concrete application examples in any field leads to the question of the conceptual differentiation of CPO, CPS and ACS essences. Let's have a look at, for example, electronic systems for monitoring and the ignition angle setting in the car engines. Certainly, systems based on mechanical drives are simple automation systems. Systems of electronic angle control can be implemented: 1) with analog circuits, 2) using digital element base, 3) using microcontrollers, 4) as a board computer module of average complexity with monitoring of several auto-systems, 5) as a full-out on-board computer with remote control of auto-ignition and separate functions of auto-management – parking, for example, 6) as a separate system module for autonomous unmanned control of the car. The question eventuates – at what implementation stage the system can be considered a CPS, IoT or CPS/IoT?

If the term "cyber" (Greek – the art of management) is considered decisive, the use of simple mechanical or electronic servo moves the system to the level of CPS. Considering "calculation" to be the base term allows to speak about a car as a CPS from the third or fourth level. The third level can’t be completely determinative, since the microcontroller can be used in control systems for performing simple operations of counting the number of pulses and generating signals for switching electronic circuits, as well as for mathematical input data processing – the actual "calculations". The fourth level also can't be accepted without objection: in many CPS definitions, using remote access to data via Internet and/or Cloud resources is compulsory. However, the remote access on the fifth level can be implemented by standard TCP/IP network protocols, or by some specialized, for example, to increase the signal security in the system. At first glance decisions of the sixth level correspond to all requirements. But there are some task classes where CPS implementation doesn't require the using IoT technologies. For example, using unmanned vehicles that are CPS with powerful computing resources for the autonomous transportation of goods in warehouses doesn't require GPS navigation and traditional Internet. Mesh networks technology can be used in this case.

Likewise, it can be shown that similar uncertainties exist in other application areas for CPS and IoT technologies. For example, in e-Health, it may be the application of different levels for cardio-monitoring technical solutions. In robotics, it may be drones ranging from the simplest ones to the systems with a dynamically reconfigurable form during the flight [46], as well as other systems where artificial intelligence for machine learning (ML) and algorithms for obtaining new data and knowledge (data mining (DM) algorithms) can be applied [26]. In everyday life it may be different variants of air conditioning systems, climate control systems, lighting control systems, etc.

Thus, the problem of smart analysis and synthesis of CPS, IoT, IoE, SNSS and the following tasks are actual:

- systematization of existing technical solutions in the field of CPS using IoT technologies;
- classification and standardization of these solutions;
- systematic approach application to the design and technical analysis and synthesis of CPS using IoT technologies;
- creation of intellectual methods for:
- analysis of the class and volume of the solved tasks by the designed system;
- complexity level estimation of projected systems;
- definition of rational values of necessary hardware and software resources;
- selecting the necessary IoT services for the effective CPS functioning;
- adaptation and improvement of existing information models of mass service systems, Petri networks and others (AI, GA, FL, etc.) for developing new approaches and methods for CPS analysis and synthesis;
- analysis of a cyber-component as an element of the functional and design of the CPS, as well as a self-compliant CPS as a design object;
- creating a method and decision support system to solve the problem of rational decomposition of the CPS design task according to the levels and elements of the 5C model;
- solving optimization problems concerning the projected system functional.

Such a list is far from complete and depends on the specifically solved tasks of designing the CPS.

On the other hand, carried out condition analysis of the modern understanding of the essences of CPS, IoT, IoE, SNSS and the structure of this manual allows to overview these tasks more generalized. Due to the content of other sections, the systems examined there can be analyzed from the point of view "to be cyber-physics" and the essence and volume of used IoT technologies.

12.1.4 IoT services and technologies for CPS

One of the main problems of CPS is the necessity to protect not only user information, but also technological and service one, which is used to ensure the stable operation and reliability of the system [26]. Failure of certain management modules in the widespread used CPS can move them to elements of "critical infrastructure systems". The point is that both unmanned cars and household electrical and other CPS and IoT devices at controllability loss are "equally" really, not potentially dangerous to humans, like nuclear reactors or chemically hazardous production.

The threat of loss of controllability can be both as natural that is due to the aging and failure of individual elements of the system, so as artificial and acquired, resulting from, respectively, external interference or mistakes made during design. The complication of the used functional algorithms, program code, structural and circuit decisions leads to increasing of probability of system controllability loss. Using miniaturized ECS, intelligent sensors and IoT technologies allows to implement complex finding algorithms for the information loss in communication channels, the failure of individual system components, to identify third-party intervention attempts, or even to predict the functional state of the system during its life cycle.

In [27], the authors demonstrate some solutions and developments to ensure high information stability of the system (Fig. 12.7), to increase its reliability to external influences. The problems of adaptation, resilience and long-term reliability and security are discussed in the context of the IoT technologies usage in automotive CPS (auto CPS).

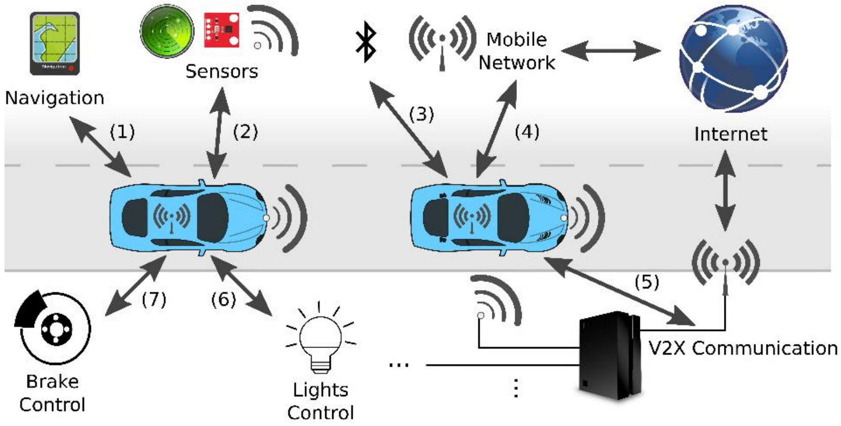


Fig. 12.7 – The information vulnerability model of the car CPS [27]: 1 – the absence of a GPS signal, 2 – noise, and sensor faults, 3 – authentication failure in the local network, 4 –- loss of mobile traffic packets, 5 – interference, denial of flooding service, 6 – wrong control because of radiation, replay attack, 7 – delayed access control message.

The analysis of possible errors, failures and information losses on the stages of obtaining, initial processing and transmitting information about the CPS status is used to create a cybernetic model of the system. Henceforward, model-oriented approach of decision-making and generation of control commands involves parallel processing of signals from real sensors and from CPS model (Fig. 12.8) [27]. Neural and Bayesian networks, fuzzy logic methods, rules for logical inferences of new data and knowledge and decision making are used to analyze the models.

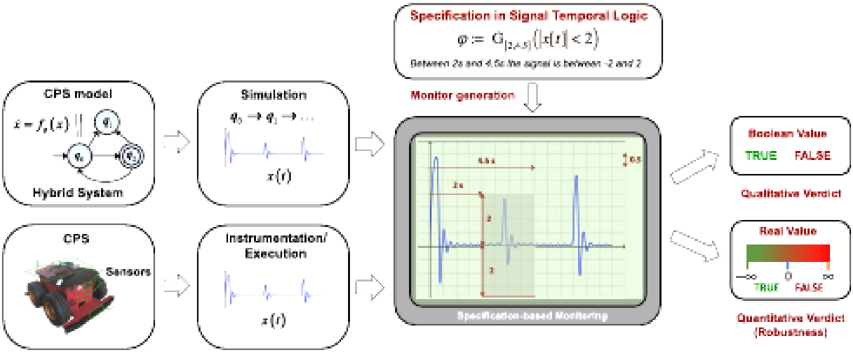

Specification in Signal Temporal Logic
$\varphi := G_{[2,4.5]}(|x[t]| < 2)$
Between 2s and 4.5s the signal is between -2 and 2
Monitor generation
CPS model
$\dot{x} = f_q(x)$
q_0
q_1
q_2
Hybrid System
Simulation
$q_0 \to q_1 \to \dots$
$x(t)$
CPS
Sensors
Instrumentation/ Execution
$x(t)$
4.5 s
2 s
2
2
0.5
Specification-based Monitoring
Boolean Value
TRUE
FALSE
Qualitative Verdict
Real Value
-∞
0
∞
TRUE
FALSE
Quantitative Verdict (Robustness)

Fig. 12.8 – Model of CPS/IoT functional simulation [27]

Dynamic analysis of the received signals in the time or frequency domain allows to verify system parameters in the form of Boolean functions or physical values of analog variables. In this case, IoT services are used to provide simulations of CPO models by cloud or fog technology. At the same time, the following problems remain relevant in the CPS / IoT design process: 1) verification and monitoring of the parameters and status of the IoT components themselves with minimal energy and hardware costs, 2) improving the IoT component resilience, 3) establishing the necessary requirements for CPS/IoT architectural solutions to provide the required reliability.

The first question is proposed to be solved in order to self-heal the system using a certain informative redundancy of data from sensors and a virtual model. To do this, a set of measures is proposed (Fig. 12.9): a) the development and expansion of an informative redundancy model, b) the introduction of self-adaptive fault detection, c) the application of fault diagnosis, d) restoration of the system’s functionality taking into account the information currently available.

Increasing the elasticity and controllability of the system can be achieved using machine learning (ML) methods to analyze the vulnerability of system protection and detect low power anomalies. The results of such training and detection should be taken into account in the architecture synthesis of the hybrid CPS/IoT system, which is characterized by the possibility of "self-treatment".

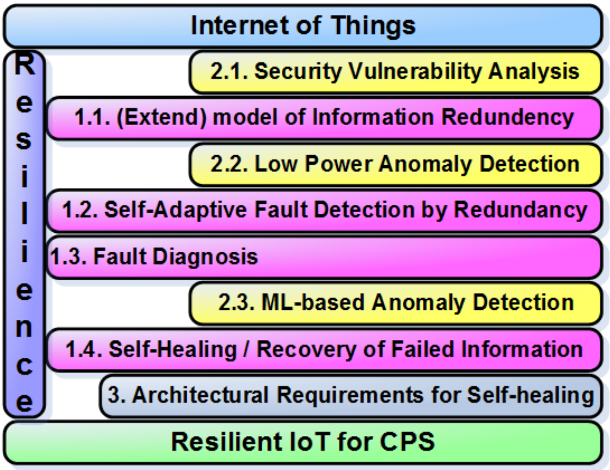

Internet of Things
Resilience
2.1. Security Vulnerability Analysis
1.1. (Extend) model of Information Redundency
2.2. Low Power Anomaly Detection
1.2. Self-Adaptive Fault Detection by Redundancy
1.3. Fault Diagnosis
2.3. ML-based Anomaly Detection
1.4. Self-Healing / Recovery of Failed Information
3. Architectural Requirements for Self-healing
Resilient IoT for CPS

Fig. 12.9 – Services. to increase the functional stability of the CPS/IoT system

12.2 System approach for the analysis and synthesis of IoT and CPS structures

The wide range of CPS and IoT applications from Smart Home projects, intelligent robotics and manufacturing, intelligent traffic monitoring systems in populated localities and aviation [43,46], and up to telemedicine and defense industry, imposes specific requirements for design, development and research of such systems and facilities. On the one hand, used and created CPS and IoT systems must be universal regardless of the application range of designed systems, and on the other hand, take into account their functional differences and individual characteristics.

The system analysis methodology meets these requirements. In dissimilarity to simple system tasks, the objective function in this case may be multi-parameter, which is due to the multifunctionality of most CPSs and their architectural complexity. Problem decomposition can be performed by: levels of the conceptual 5C model; architectural components – basic physical component, cyber component, IoT tools; objective function parametricity of a specific projected CPS. The objective function key goal is determined at each hierarchical level/sublayer of compositional structure. In classic case, these are minimization of resources, energy consumption and/or time parameters for solving typical problem-oriented tasks.

12.2.1 Setting problem-oriented tasks

The problematic CPS orientation is determined by the specifics of its application area and type of the tasks, the manipulation nature for data and physical objects, the information exchange with the environment. ECS can also be seen as a specific CPS that performs tasks to ensure the functionality of some its "technical shell" – the basic physical system (BPS), which is the "environment" for ECS. Industrial automated technological lines for automobile or semiconductor production, robotic mechatronic systems and complexes, medical or environmental monitoring information-measuring and control devices, etc. can play as BPS.

Problem-oriented tasks that arise in various fields, as a rule, relate to the tasks of a clearly defined class [44]: computing, control, data processing, etc. A limited number of algorithms is used to solve them; which also determines the features of the structural and functional solutions of ECM. However, for certain types of information-measuring, technological and other systems, the question arises of simultaneous solving the problems of modeling processes, collecting information, managing the peripheral devices, etc. Such problems take place, for example, in the functioning process of automated systems of small-scale semiconductor production, biochemical synthesis, monitoring and modeling of environmental processes in real time, etc. In particular, for the synthesis of semiconductor crystals, hetero-structures, Schottky-barrier diodes with specified characteristics, optical filters of the ultraviolet, visible and infrared radiation ranges, including for fiber-optic communication lines, there are possible tasks when it is important to provide both a dynamic correction of control signals based on the results of technological process parameters control and to simulate the dynamics of the process itself.

Similarly, there are diverse tasks that require computing resources to work with remote objects, it's provided by IoT technologies. In telecommunication systems and data transmission systems, the tasks of primary and noise-protective coding, compression of digital streams, cryptographic protection of information, etc. are solved.

The analysis results of the specifics of the problem-oriented task are decisive in the future for the architecture synthesis, the necessary computing resources assessment, the choice of IoT technologies for the designed CPS.

12.2.2 Definition and study of the target function of the CPS synthesis problem

The quality criterion for the synthesized CPS architecture or its components may be the target function (TF), which determines the relationship between the purpose and the optimization parameters of the proposed technical system (TS) solution. The phase state of the TS is called the hardware-software configuration k_i from the full set of possible CPS configurations $K=\{k_i\}$ designed to solve given types of problems – signal measuring, information processing , modeling objects and processes, control, etc., at certain intervals of the system life cycle. The complete set of phase states forms the phase space $\Phi=\{\varphi_i\}$ in which the CPS functions. The dimensionality of the phase space is determined by the number of types of the tasks performed in parallel, that is, it's proportional to the configuration dimension of the phase states. It can vary at certain points in time. It depends on the input values of the system information parameters and how the sets of $K=\{k_i\}$ are combined into the CPS executable functions $F=\{f_i\}$. The complete set of the executable functions $F=\{f_i\}$ corresponds to the set of executable programs algorithms $A=\{a_i\}$, and the appropriate hardware resources $AR=\{r_i\}$. Autonomous functioning of CPS (or a cyber-component, as a part of CPS) provides system self-organization in terms of dynamic self-configuration of its architecture in the phase space, depending on the change in values of input signals and system states parameters in real time. Various approaches are possible to describe the functioning of CPS, including classical approaches widely represented in literature, such as graph method, matrix methods, and others.

There are three main classical realization variants of the ECM TS structure according to the peculiarities of the implemented algorithms: 1) in the form of a linear system with sequential execution of problems *Z*, and, appropriately, consistent transitions between system states *S*; 2) as a parallel structure where certain problems *Z** of the set *Z* are realized simultaneously, which corresponds to the synchronization of certain system states; 3) as a combined structure with parallel-sequential execution of the set of task *Z*.

The problem of ECM TS synthesis depending on conditions of the implemented processes can be formulated as a system analysis task, determined by the objective minimizing function of the used hardware resources at satisfactory system performance, or the maximum speed function of the synthesized TS at a satisfactory value of the resources used.

In the first case, it is necessary to ensure the maximum combination of modules, and, accordingly, elements that perform certain procedures in the total space of states *S*. Then the synthesis problem is reduced to finding the optimal FPGA programmable environment by the value of the corresponding target function, taking into account the limitations of the minimum required number of FPGA elements to implement specific procedures of the executed algorithms.

The second case is more complicated, since optimization of the objective function requires taking into account both the dimensions of the executed algorithms with the implemented states and the synchronization of these algorithms.

12.2.3 Self-organization principles of the cyber physical systems

The basic principles of self-organization of cybernetic systems are laid down in the works of N. Wiener and V. Glushkov [11, 12]. One of the basic concepts in the systems self-organization is the question of adaptation as a process, or adaptability, as the functional ability of system adapting to the variability of conditions or external factors affecting its functioning. As a means of implementing the adaptability of the system to external disturbances, the structural-functional or structural reconfiguration of the system appears [48. Separately, we distinguish the architectural reconfiguring of the system, since for robotics this term defines the mechanical complexity of the system components and the interconnections between them, and for computer means it additionally characterizes the types and format of commands and data and the internal machine language for describing the interaction between individual modules.

In the general case, ***configurability*** means the ability of functional and structural adaptation of the system to the implement (execute) a certain set of elementary operations, actions, or more complex functions from a given kit. This ability is determined by the functional algorithms that are embedded in the system design process, and are implemented during its startup and debugging by appropriate specialists.

Reconfigurability is the ability for multiple variability and functional-structural adaptation of the system to a given set of functional algorithms, which is implemented at certain fixed moments during the system life cycle and carried out by the system operators.

Dynamically configurable/reconfigurable systems have the ability to adapt to real-time operating conditions, but the time moments at which their functional-structural configuration/reconfiguration occurs depend on predefined parameters and/or logical conditions of the implemented algorithms. What is important, the specified conditions and parameters are clearly defined at the programming stage of the executed algorithms. Such systems have an initial degree (or predispositions) to self-organization, which is also laid down at the stage of system design, and are actually analogous to means and devices for automatic regulation and control of processes and objects.

Self-configurability is the next (or second) level of system self-organization and is implemented as the ability of a functional-structural system self-adaptation to the problems being solved and functional algorithms without operator's influence, however, as in dynamic systems, it is limited by predefined values of the input parameters of the problem or certain logical conditions and their determined activating time.

Self-reconfigurability is the highest degree of self-organization of the system, which, after debugging, setting the initial parameters and conditions, and starting up, functions autonomously and is able to independently provide the implementation or even synthesize the necessary configuration to ensure the stable CPS functionality. Such systems can function in real time with insufficient certainty, or complete uncertainty or lack of predictability for behavior over time of input parameters, internal states and the effects of external factors on the system

Self-reconfiguring systems certainly contain additional hardware and software for in-depth system self-analysis (own states, input signals, critical and/or limit values of functional parameters, etc.). They have artificial intelligence to implement the self-learning functions, and can use, for example, fuzzy logic apparatus of decision making in critical conditions or assigned operating modes.

The questions of the self-reconfiguration of CPS hardware in the literature are considered as a way to implement the self-organization of the system by constructive restructuring, first of all, of its mechanical components. Full self-configuration as a process of self-organization of a system implies the presence in the system of technical capabilities for making constructive changes in the structure , in functions of individual modules or the whole system, and the correction of its own architecture. Such system capabilities, robust by controllability principles, are laid down during the CPS design process in the form of using special swivel joints, mechanical actuators, switching couplings, etc., depending on the purpose and functions of the system. The system for analyzing and controlling the mechanical reconfiguring of robot-like systems is implemented on the basis of the CPS cyber component exclusively at the level of ECM software.

Taking into account the differences in architecture, the self-reconfiguration of ECM as elements of the CPS structure is more complicated, since it is implemented by synthesis of functional special processors with architecture, adapted to the implemented algorithms, and requires circuit design correction of individual system modules [50].

12.2.4. 3S model of CPS

The functional organization of the CPS cyber-component is to provide the necessary level of ***self-analysis*** of the system as a whole and the ECM as a local CPS, its ***self-adaptation*** to the conditions of the external environment (in the broadest sense) and ***self-organization*** of the system as a multiprocessor object. For such a model regarding the cyber component, the term “3C-model” is proposed (Eng. – 3S-model: self-analysis, self-adaptation, self-organization). It is worth noting that at present, the process of synthesis of special processors requires the involvement of several software products that do not have a common interface and interact at the level of the operator-developer of the system. In this sense, full autonomy of the system in self-configuration mode is still impossible to ensure. An urgent question is the creation of software for the through-design of configuration files of special processors in one software product. Currently, the possibility of moving from an object-oriented approach of designing such systems to a so-called event-oriented approach is discussed in the literature. In our opinion, the transition to full functional-oriented design of ECM based on field-programmable gate arrays (FPGA) will become promising in the near future.

In the simplest case, to implement the functions of the 3S-model of the cyber component, the CPS should contain 3 parts: system control module (CM) with fixed organization; analytical module (AM) of control, analysis and decision making; module of reconfiguration environment (RM) with appropriate hardware and software. Depending on the problems or problem classes being solved, and with currently available hardware and software resources, the self-reconfiguration can be viewed in the following aspects of the architecture implementation.

1. A *co-processor* model for implementing a self-configuring environment. Its essence is that in software environments, depending on the problems or classes of problems to be solved, the configuration file of the "problem" coprocessor (Pcpr) is loaded. Performed functions or tasks are prepared as executable program files in a high-level language or in a specialized language for the selected class of tasks. Such files, according to the recommendations of the MA, are dynamically uploaded to the Pcpr for execution.

2. A model of complete *self-reconfiguration using a library of reconfiguration files*. In this case, the MA in dynamic mode decides to load one or more files of the hardware configuration into the reconfigured environment, depending on the input conditions and instantaneous states of the system, which take place in real time of the system functioning.

3. *Network distributed or server model of self-reconfiguration* provides remote preparation of configuration files for different types of real-time tasks. The analytical module in dynamic mode generates a request for the synthesis of a reconfiguration file or a set of such files necessary for the further system functioning when solving new problems or problems that arose at the previous stage. The generated request is transmitted by IoT means to a remote server or cloud in the form of a reconfiguration file synthesis task. Such a system, described by a complete *self-reconfiguration model based on the synthesis of new reconfiguration files implicating* IoT technologies, can be called a self-reconfiguring intelligent computer system (SRIS).

12.2.5 Examples of structural solutions

Developing unified CPS technical solutions is a difficult task because of the variety of their application areas. However, this task is greatly simplified by the hierarchical-modular approach (HMA) to designing complex objects and systems (DCOS). The essence of HMA is to decompose the task, and, therefore, the general algorithm of its solution and the structural decision of TS into separate segments with hierarchical subordination from the simplest to the most complicated. That's exactly the kind of ideology that is embedded in the discussed above 3–5 level CPS and IoT conceptual models.

For example, consider structural organization of the CPS cyber-component (CC) as a simplified model. The hierarchy proposed in [48]: Structure – Device – Module – Process – Function – Procedure/Action (S – D – M – P – F – A), allows to sufficiently detail an arbitrary algorithm for its hardware implementation. The basis of this hierarchy is the “Module” object as a functionally completed node to implement a certain simple process. The implemented “Process” is described by some logical/arithmetic “Function” consisting of a sequence of elementary “Procedures/Actions”. A certain set of modules implements a specific subroutine of a generalized functional algorithm and is considered to be a separate "Device" in the general "Structure" of the CP TS. Thus, module-oriented technology can be considered as an approach to the unification of structural (hardware) solutions, which is a reflection of the algorithm sets implemented by multitasking TS.

Unlike typical computing systems, embedded systems have the specialization (SP) and multifunctionality (MF) properties of solvable types and classes of tasks within CPS. MF covers many functions for a particular class of tasks or problematic orientation of the ECM or CPS as a whole. SP is considered as the hardware-software system adaptability to solve certain types of outlined problems. There is a one-to-one accordance between MF and SP for the respective problem or task classes. The multifunctionality of the embedded systems is a prerequisite for ensuring the performance of the CPS and permits the parallel implementation of most functions except the functions of general control and process synchronization.

Generalized structure of the CPS cyber-component (Fig. 12.10) usually contains: central processing unit (CPU); logical analysis module (LAM) of the input conditions $X(t)=\{x_i(t) \mid \mathrm{i}=\overline{1,I}\}$, and the outputs $Y(t)=\{y_m(t) \mid \mathrm{m}=\overline{1,M}\}$ received when executing control commands $U(t)=\{u_h(t) \mid \mathrm{i}=\overline{1,H}\}$; programming environment (FPGA) with programming/reconfiguration (PR) tools; built-in storage in the form of a specific library (LB); external/network access (I/O Ext) to the reconfiguration files of executable tasks $Z=\{z_j \mid \mathrm{j}=\overline{1,J}\}$.

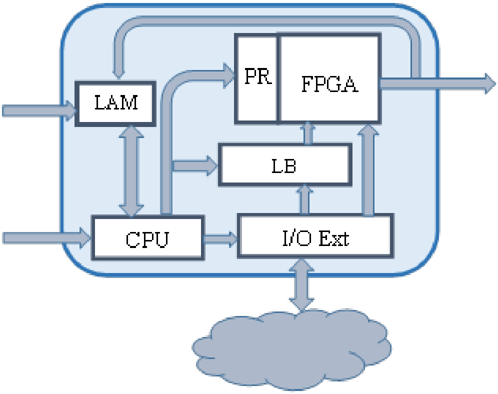


Fig. 12.10 – Generalized structure of the CPS cyber-component

For correct intellectual control of CPS processes, the set of its physical states must be reflected by the relevant states of CP in the multitask space $S(t) = \{s_k \mid k = \overline{1, q}\}$ (Fig. 12.11), which is determined by the parameter S = <LNZ> according to the dimension of the phase space of states.

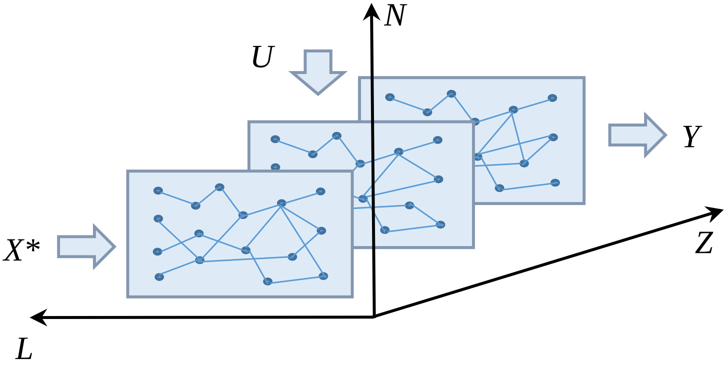


Fig. 12.11 – Model of displaying technical system state space by cyber component

In [49] V. Golembo and O. Bochkaryov propose a multi-contour model of CPS interaction, namely its cyber components, with physical space (PS) based on the HMA approach (Fig. 12.12). Each individual contour k_l corresponds to a certain level of cyberspace interpenetration into the physical world, or to a certain level of CPS adaptation to the change dynamics of PS properties. For each circuit, a generalized target CPS function $F_{CPS} \rightarrow \{F(k_1), F(k_2), \ldots, F(k_n)\}$ also undergoes a certain correction.

The conceptual model of contour interaction can be described as [49]:

$$M_K = \{(P \times K(p)), (U \times K(u)), K(c)\} \times \{R(g), R(q), R(e)\},$$

where $P=\{p\}$ is the set of states PS; $U=\{u\}$ is the set of users; $R\ (g)$ is the set of computing resources; $R(q)$ is the set of communication resources; $R(e)$ is the set of energy resources. The contours of interaction are described for general functional purpose: $K(p)$ – contours of CPS components interaction with PS; $K(u)$ – contours of CPS components interaction with users; $K(c)$ are contours of internal service interaction of CPS components.

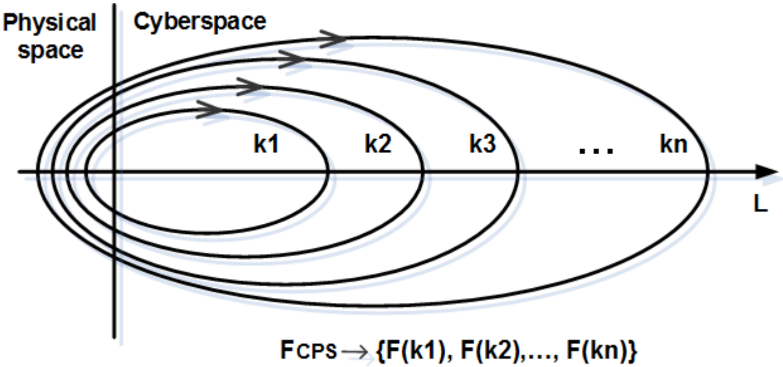


Fig. 12.12 – A multi-contour model of CPS interaction with PS

It is suggested to use the principle of multi-loop feedback to describe the processes of contour interaction. Solving the general problem of structural adaptation of the contours k implies the solution search for optimization problems of two types [49]: 1) the F_{CPS} maximization problem, which consists in finding the best "quantitative" relation between the obtained information about the system states and the information aimed to control the PS transition to required state; 2) the task of balancing (matching) the information content of physical processes models $\{M_i\}$ with the indicators of the information amount that "rotates" in these contours. This solves the problems of rational correlation for the involved hardware and software resources between the modules of parameter measurement and control. An estimate of the information amount, entering the components of the i-th contour, can be obtained on the basis of simplified form of Friis transmission equation [49]:

$$R'(k) = [f_1(R(k)) \times f_2(T(p))] \,/\, [f_3(L(k))]^2,$$

where *R(k)* is an indicator of the information amount, sent to the interaction contour k; *T(p)* is a characteristic time of PS, the information about is transmitted; *L(k)* is the scale of the contour k (for example, in the form of delay estimation of information transmission by the communication components of the contour); *f1, f2, f3* are scaling linear transformations. It reflects the fact that the amount of received information becomes smaller, that is, the information loses its relevance the longer it is been transmitted along the interaction contour.

A prospect for assessing the need to implement a particular hardware module or functionality is O. Kharkevich's indicator of the information expediency degree [49], which is defined as the change in the probability of reaching a goal by obtaining additional information.

Based on the model [49] and the estimates, V. Golembo and O. Bochkaryov propose a scenario of automated synthesis and configuration of CPS using unified structural modules. Such a scenario can be implemented as a software package using client-server and IoT technologies.

12.3 Data processing in the CPS

Information processing in CPS is implemented at all levels of conceptual models. Each level has its own peculiarities regarding the format and volume of the input/output data, how they are transformed and the hardware and software solutions used. In particular, various approaches and tools are used for primary processing of data from sensors, their translation between levels and storage. Control signal forming modules for analog and digital servo actuators also require different approaches. However, using a hierarchical-module approach to the CPS structural organization as a whole or their individual components, the S-D-M-P-F-A information processing model and the multi-contour model of CPS interaction with PS an unified approach can be offered to quantify the resources required for implementing the various functional modules of the embedded systems: logic blocks, memory elements, arithmetic devices, trigger keys, etc. For this purpose, the individual subtasks of designing functional modules are described in terms of the functional of their own phase state spaces.

12.3.1 Estimation of computing resources

For example, consider to evaluate the resources of a reconfigurable module implemented by means of programmable FPGA arrays. Let the model representation of the CPS ECM (Fig. 12.11) describes the task set of the executed TS in the form of separate planes *Z* of a three-dimensional space *S*.

The set of states for a separate task from the set *Z={z_j* | $\mathrm{j}=\overline{1,J}$ *}* is represented by nodes on the z_j planes, the transitions between them – by the corresponding edges.

The resulting graph, in the simplest case, reproduces the algorithm of the executable task z_j for some process, which is described by the corresponding set of functions $\mathrm{F(t)}=\left\{\mathrm{f_n}(t)\,|\,n=\overline{1,N}\right\}$. Each function uses standardized sets of procedures/actions/acts $A=\left\{a_l(t)\,|\,\ell=\overline{1,L}\right\}$, i.e. $\mathrm{f}_n(t)=g(A)$. If the elementary procedure is performed by a certain type of FPGA elements, then some process $p_q \in P$ implemented by some module *M*, where $P(t)=\left\{\mathrm{p}_q(t)\,|\,q=\overline{1,Q}\right\}$for a separate task requires such hardware resources:

$$R_{p_q}(t)=\sum_{n=1}^{N} f_n(t)=\sum_{n=1}^{N}\sum_{l=1}^{L} a_{nl}(t), \tag{12.1}$$

where $a_{nl}(t)$ are the coefficients of the matrix *P* with dimension *P=<LN>*, which take weight values according to the number of used FPGA elements types. In the case of the condition "one module *M* – one process *q* ", the structural complexity η of the synthesized module *M* is determined by the parameter R_{p_q} : $\eta(M(P_q))=R_{p_q}$ written in (12.1). When several processes are executed in parallel in one module, the amount should be written:

$$\eta(M(P))=\sum_{q=1}^{Q} P_q(t)\cdot \tag{12.2}$$

When problems of multitasking TS are solved by single-process modules, the FPGA reconfigurable environment must be able to display *Z* reconfiguration files, which requires the determination of common minimum required resources Φ for various functional types of TS algorithms:

1) for a TS operation sequential algorithm –

$$\Phi_{ser}=\min_{\delta}(\max_{1\le z\le Z}\eta(M(P)))=\min_{\delta}(\max_{1\le z\le Z}(\sum_{q=1}^{Q} P_q(t))), \tag{12.3}$$

that with a minimum reserve ratio $\delta_{ser}=R_0-R_{P_{\max}}$, where R_0 – the total resource of the selected FPGA environment, provides the configuration file download of the task $z_{j,}$, which requires the maximum amount of resources $R_{P_{\max}}$ in the reconfiguration matrix;

2) for a parallel algorithm of TS operation –

$$\Phi_{par} = \min_{\delta} \eta(M(P)) = \min_{\delta} \sum_{j=1}^{J} \sum_{q=1}^{Q} P_{jq}(t), \tag{12.4}$$

that with a minimum reserve ratio $\delta_{ser} = R_0 - R_{P_{\max}}$ provides simultaneous loading of a set *Z** from all the configuration files of tasks from the complete set of *Z*, which must be executed in parallel and require total resources $R_{P_{\max}}$, into the reconfigurable matrix;

3) for the combined algorithm $\Phi_{complex}$, described by an expression similar to (12.4), for each new step of structure reconfiguring with a new set *Z** and may take an intermediate value between Φ_{ser} and Φ_{par}, however, the reserve ratio $\delta_{complex} = R_0 - R_{P_{opt}}$ is determined for the structure optimally packed by the resources $R_{P_{opt}}$ used for the sets *Z**. The criterion for optimal packaging is minimizing the number of unused base elements of the selected FPGA. It should be borne in mind that for the second and the third algorithm type the additional resources may be used to synchronize parallel processes and to input/output of certain intermediate results of information processing.

For the second and the third algorithm type, the process matrix of resources used for TS with structural complexity $\eta(M(P))$ is transformed into a three-dimensional tensor with coefficients $a_{jnl}(t)$: $V = \left\{ a_{jnl}(t) \mid j = \overline{1,J};\, n = \overline{1,N};\, \ell = \overline{1,L} \right\}$. The number of resources for implementing the parallel and combined types of TS operation algorithm is described by the expression:

$$R_P = \sum_{j=1}^{J} R_{p_q}(t) = \sum_{j=1}^{J} \sum_{n=1}^{N} \sum_{l=1}^{L} a_{jnl}(t). \tag{12.5}$$

The limiting factors for deciding on the choice of FPGA type for synthesis of reconfigurable environment are the features and available resources of the element base, which is developed for a specific series of FPGAs and allows implementing elementary procedures/actions to perform algorithms of solved problems: the number of basic universal logical blocks

(LUT), switching and trigger elements (Flip Flops), buffer elements and multiplex fragments (Number of BUFGMUXs), input/output lines/buses (IOBus), and their configuration options for inter-element layout. The implementation features of inter-element/inter-module connections, as well as choosing the way to synchronize the signals from different modules can significantly affect the overall cost of resources for the project. The latest factor may significantly differ for different FPGA series.

Thus, the generalized task of system analysis for the synthesis of CPS multitasking TS is to determine the minimum required but sufficient resources of the FPGA to ensure the full system functionality, and is formulated as the problem of finding the minimum of the objective function $F^* = f(\delta)$ for the corresponding types of functional algorithms, which is limited by the basis $B = \{b_k\}$ in the *K*-dimensional space.

The discrete states of the *K*-dimensional space $B = \{b_k\}$ are points corresponding to the configuration of specific types of FPGAs. Therefore, the search algorithm for the formulated problem solutions in geometric interpretation is reduced to finding the points which are the closest to the convex polyhedron of the required resources Φ for different functional types of TS algorithms built in the same *K*-dimensional space.

The final decision to implement a complete CPC TS project depends on the related components, synthesized by the manufacturer in one shell with a programmable environment, including processor core (CPU), additional memory modules (RAM), and switching interfaces with peripheral devices, etc. The presence of such components in FPGAs, offered by such firms as Xilinx and Altera, simplifies TS synthesis and provides it with greater flexibility, including the ability to dynamically reconfigure the programmed environment.

12.3.2 Transmission, processing, display, storage of data

The quality of CPS technical solutions further determines the efficiency of information processing. The application of the CPS structural organization according to the S-D-M-P-F-A model and the CPS/PS multi-contour interaction (Fig. 12.12) for principles of multi-loop feedback allows to describe the recycling of information flows and IoT control flows in such systems according to the conceptual CPS diagram (Fig. 12.1). As a rule, classical data recycling is implemented in each CPS loop/information processing cycle: receiving data from sensors – transmitting data to a specific node – arithmetic/logical data processing – generating a control action – changing the object state – changing data from sensors.

In modern CPS, very large data arrays are generated, and their operative processing requires significant computing power. This problem is solved by complicating the structural solutions of the cyber component, in particular at the expense of arithmetic-logical devices and memory, or by temporarily-spatial separation of the stages of obtaining primary data and their mathematical processing. The latter is implemented through IoT technologies, including using access to cloud technologies for data storage and processing.

In autonomous systems with automatic control, the recycling of information flows is implemented in real time. Therefore, the structure complication of the cyber component and CPS as a whole is mainly applied, although remote data processing is possible depending on the requirements for the dynamics of the serviced physical processes.

In automated CPS systems, the presence of an operator/user is assumed in the “processing-control” cycle, therefore, cloud technologies are mainly used, as well as additional means of visualizing the results of data processing and analysis. Well-known companies provide for this wide range of capabilities in the form of basic software platforms: Microsoft Azure (Azure) [52], GOOGLE Cloud Platform (GCP), Amazon Web Services (AWS) [51]. Azure-based CPS/IoT structural solutions allow using specialized machine learning modules, working with databases (Fig. 12.13), processing large amounts of data from sensors or other devices using mobile Web technologies (Fig. 12.14).

The implementation of Amazon S3 and Azure for Windows & Linux and Java-oriented programming languages expands the opportunities of application development for CPS/IoT solutions [52].

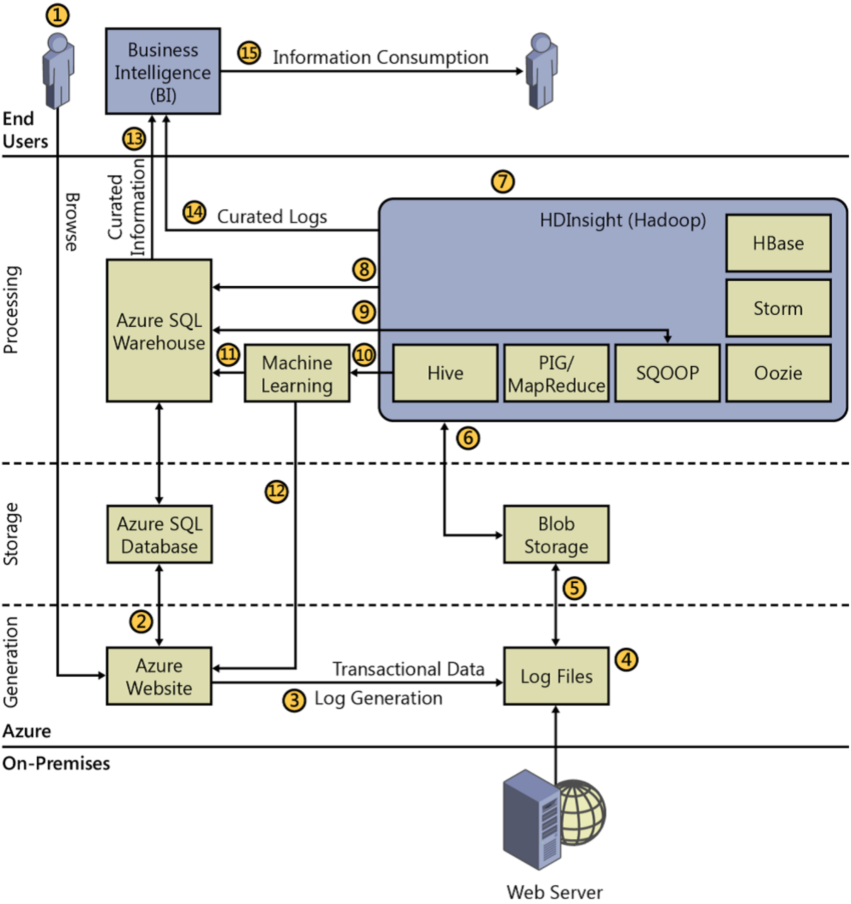


Fig. 12.13 – An example of a CPS cloud service implementation based on the Azure platform using ML technology [52]

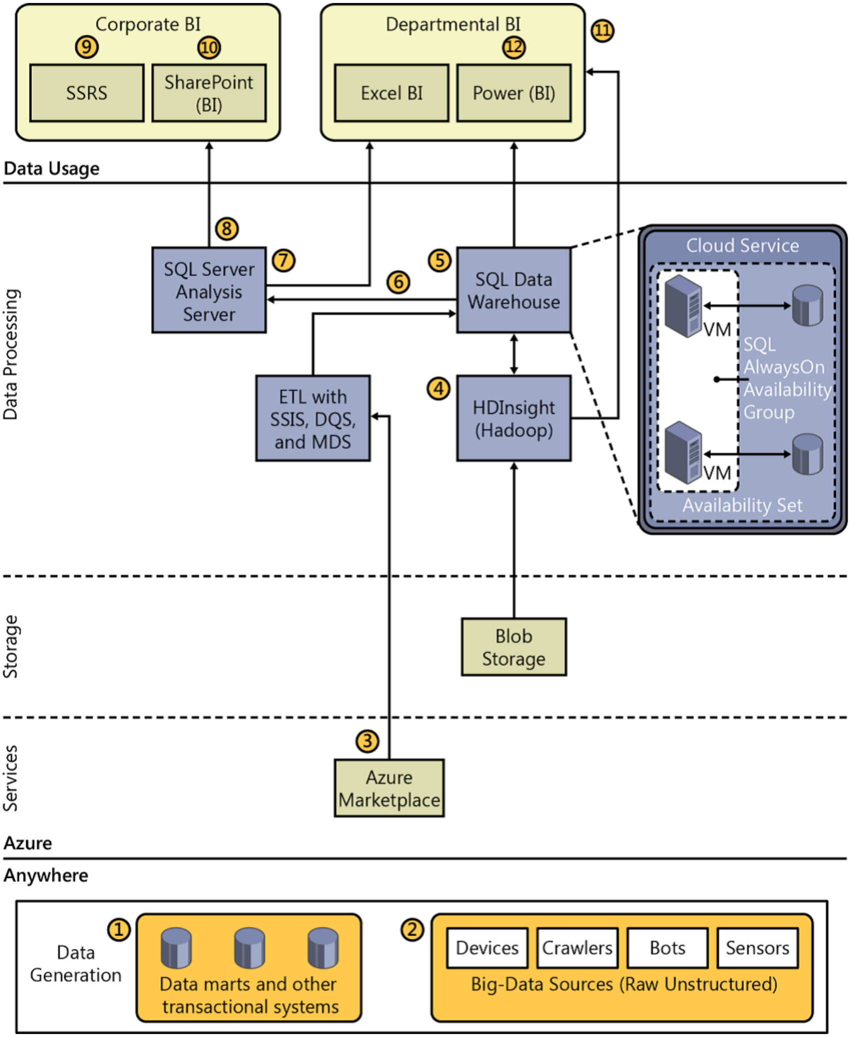


Fig. 12.14 – An example of a CPS cloud service implementation based on the Azure platform for working with BigData and mobile devices

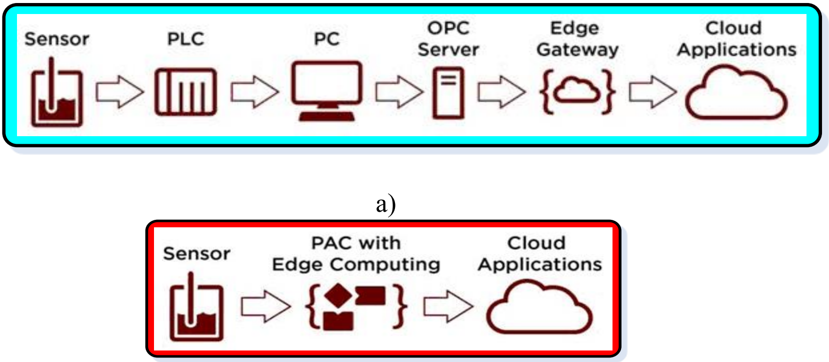
12.3.3 Parallel, cloud, fog, edge calculations and resources

The need to transmit large amounts of data between different levels of multi-level CPS models reduces the effective system performance. At the same time, there is a problem of inter-level coordination of the transmitted data formats and the corresponding protocols of intermodular information exchange in the S-D-M-P-F-A model. However, the unification of formats and protocols allows to abstract from the features of data conversion at each level or in a module and to standardize information exchange processes. This approach expands the physical limits of CPS/IoT devices “communication” and reduces the time and linear intervals of data transmission. It also allows to redistribute hardware and software resources, system computing load and to optimize functional CPS algorithms through the effective use of IoT technologies, in particular Edge Computing. Resource optimization is implemented by methods of parallel, cloud, fog and edge computing.

When parallel processing of information is capable or significant computing resources are needed, distributed GRID systems, computing clusters, or modern Cloud technologies [53] are often used. With smaller amounts of data or the ability to sequentially process them to accomplish similar tasks, specialized CPSs can use quite powerful resources on personal computers (PCs) or modern microcontrollers with Cortex-based ARM architecture. For example, CUDA software can increase PC processing power in the presence of NVidia GPU software reconfiguration. Similarly, the capabilities of embedded systems implemented on the basis of the Cortex kernel are expanded when using the variants of a 32-bit operating system in the created modules. The presence of built-in Ethernet interfaces and mobile OS allows to implement powerful computing complexes already on the basic Raspberry Pi platforms. [53] describes an approach to implementing CPS with the makings of artificial intelligence on Arduino XXX platforms with ARM processors.

Fog calculations [55] are essentially a kind of cloud technologies where sensor data goes through several transformation stages: from sensors and industrial programmable logic controllers to network switches on server gateways (Fig. 12.15, a). This reduces the computational load on the end devices, but each subsequent communicational link creates a potential risk of distortion or information loss.

Edge Computing [54, 55] technology is understood as the shift of computing from the Cloud communications layer to the limit of receiving data from sensors (Fig. 12.15, b). This is achieved by using smart sensors. They provide the initial data processing, convert the input data into a standard formatted binary code and make a decision on the data transmission to modules of the higher hierarchy and storing them in the cloud. Thus, it is possible to reconfigure the communication gateways in the hybrid CPS/IoT model (Fig. 12.3).



а)

б)

Fig. 12.15 – Classical (a) and optimized (b) due to IoT technology, Edge Computing model of data transmission in CPS [53]

Programmable environments (FPGAs) provide additional options for increasing the efficiency of embedded systems through software or hardware reconfiguration. In addition to the peripherals and interfaces developed in ARM architecture, FPGA has the ability of implementing storage nodes, arithmetic and logic units of varying complexity according to the requirements of calculation errors or data processing speed. At the same time, the problems of reconfigurable systems synthesis can be solved in the form of a complex approach – using hybrid structures such as ARM – FPGA, PC – ARM – FPGA, PC - FPGA and other systems with additional interface devices between separate functional units.

12.4 Mathematical and informational support of IoT and CPS technologies

The mathematical and informational support of CPS and IoT technologies is implemented at two levels – design and operation. These are mathematical models and software, which on the first level are intended for the analysis and synthesis of technical solutions, and on the second, respectively, to support the functionality of the synthesized structures offline or automated.

12.4.1 Stages and tasks of modeling of information processing

- The synthesis method and, accordingly, the design algorithm for the embedded reconfigurable computer means of CPS depends on their functional purpose and set of classes of problem-oriented tasks for the solution of which they are planned. However, one can describe the most generalized approach to the construction of such systems and the creation of hardware-software models of both the systems themselves and the process of their design:
- refinement of the set of problem classes $Z = \{z_1, z_2, \ldots , z_i, \ldots , z_n\}$ that the projected system is oriented to (research, technological, advisory, etc.) and physical objects described by real parameters and characteristics, which these tasks relate to ;
- specification of the problem orientation of certain types of tasks (measurement *M*, control *C*, data arrays processing *DP*, modeling *M*, correlation analysis *CA*, coding *CD*, data transmission *DT*, traffic compression *TC*, etc.) and marker identification of types, for example – $Pz_i=\{ M, C, DP, M, CA, CD, DT, TC \ldots\}$;
- selection of the set of characteristic parameters of physical objects or processes from the corresponding arrays of their descriptions, for example, for the type *M* (measurement) it can be temperature *T*, current *I*, voltage *U*, value of light flux *Φ*, or for other objects – data transmission speed *vT*, data volume *VD* , etc.: $M=\{[T, U, I, \Phi \ldots], [vT, VD \ldots]\}$;
- setting limit, allowable, control parameter values of an object or a process, calculation errors that are decisive in the selection of methods and the synthesis of information processing algorithms;
- selection of decision methods for each type and class of tasks, algorithmic optimization of the designed system according to the minimizing parameters of the methods due to the unification of the used algorithms;
- synthesis of the generalized system architecture at the level of structural decisions of information processing units, peripheral communication interfaces, data collecting and storage facilities, and displaying the results of their processing, as well as system programming;
- optimization of the structural and functional organization of the system according to the criterion of minimizing hardware costs by synchronization, timing and resource allocation of the various stages of data processing;
- solving the problem of system reconfiguring taking into account the results of its algorithmic and structural-functional optimization; substantiation of methods and means of hardware and software system reconfiguration;

- clarification of the characteristics of the functional units of the system and their circuit implementation using ARM processors, FPGA programmable environments and others.

The last stages of synthesis are traditionally implemented on the basis of previous system simulation with available software VHDL, Proteus, Altium Designer, mathematical Simulink packages, etc. Currently, powerful software tools have been created to automate the design process for reconfigurable systems, including by Xilinx, Intel (Altera), which are manufacturers of programmable structures. However, the question of creating industrial software samples for automated functionally oriented through-synthesis of systems, taking into account their problem orientation, as well as the synergy of CPS and IoT technologies, remains relevant. This refers to automatic design systems, from mathematical description of the model or even a set of input parameters to obtaining its program code, ready for program reconfiguration of the structure. Although the first steps in this direction have already been taken, for example, the use of the µC language for the direct programming of CPS and IoT base platform controllers, or their simulation in the Cisco environment, however, the process of synthesizing technical solutions requires user intervention.

12.4.2. Functional IoT and CPS algorithms (in terms of application)

According to the model of V. Golembo and A. Bochkarev [49] of multi-contour CPS interaction with the environment (Fig. 12.12), information processes occur simultaneously in two spaces: cybernetic and physical. For CPS, cyberspace or the cyber world, in the broad sense of the word, means not only computer (computing) means, but also a variety of technical and not only (for example, mathematical models) tools, that can be used to process information and "manage the objects of animate and inanimate nature" [11,12]. The physical space (environment) in terms of “information” and “control” primarily encompasses the physical processes of their interaction “in” and “between” the objects of the physical world. The functioning of objects, and accordingly the algorithms describing them, in cyber- and physical space have the same nature – physical, and are provided at the transformation level of physical signals of different nature. The connection between the cyber and the physical world is realized through sensors, attenuators and measuring transducers of various types of physical signals.

Different perceptions of the physical world as real, and cyberspace as virtual, by simple users and even specialists/CPS developers, in our opinion, are due to differences in the presentation of information - in analog form in the physical world, in discrete - in cyberspace. Although mathematical models in both cases are our, abstract or with a certain level of virtualization, ideas about real objects and processes.

Thus, it is advisable to consider generalized CPS and IoT functional models at the level of information data flows and control, as shown in Figures 12.11, 12.12. In such a model, the following streams come to the CPS input [49]: 1) the input data x_i from the set of all possible input data options $D_x(t)=\{x_i(t) \mid \mathrm{i}=\overline{1,I}\}$, and u_h commands from the set of commands $U_c=\{u_h(t) \mid \mathrm{i}=\overline{1,H}\}$; 2) the current state of physical processes s_p from the set of states $S(t)=\{s_p(t) \mid \mathrm{p}=\overline{1,P}\}$. At CPS output, we obtain the output y_m from the set of all possible outputs $D_y(t)=\{y_m(t) \mid \mathrm{m}=\overline{1,M}\}$, the guiding influences a_{j} from the set of guiding influences $A(t)=\{a_j(t) \mid \mathrm{j}=\overline{1,J}\}$.

As a result, the functional algorithm for the dynamic interaction of CPS with its environment can be represented as a cortege [49]:

$$<D_x, U_c, D_y, S, A, T_d, T_p>,$$

where $T_p:(D_x, U_c)\times S\rightarrow D_y$ i $T_d:(D_x, U_c)\times S\rightarrow A$ – operators of mapping input data, commands, and states of physical processes, respectively, into the output data D_y and into the guiding influences A. The type of operators (T_d, T_p) is determined by the structure and parameters of the CPS. The latter ones are determined by the search for solutions to the optimization of system analysis minimax problem for the CPS objective function $F^* = f(\delta)$ for the corresponding types of functional algorithms, which is limited by the basis $B=\{b_k\}$ of the K-dimensional configuration space of system solutions.

12.4.3. Mathematical models of CPS

Functional modeling aims to provide a synergy of interaction between elements of physical and cyber CPS space by synthesizing generalized functional algorithms. On their basis, structural solutions of the CPS and IoT models are synthesized. The creation of mathematical models requires further decomposition of the problems of analysis and synthesis of systems, and movement to the level of specific physical processes for which the designed system provides information and management support. Therefore, the description of mathematical models is relevant for concrete examples of CPS and IoT implementation. In the general case, it is advisable to analyze the system structuring and determine the classes of problems and models that may be required for mathematical modeling.

The structural organization of CPS can be analyzed according to the conceptual model of V. Golembo and O. Bochkarev [49] (Fig. 12.16). The proposed model details structural components at the levels of physical space, cyberspace and the interaction between them by means of the sensor system and the control system. In cyberspace, structural units of high-performance, including remote computing, and tools for local computing by embedded computer means are distinguished. IoT technologies enable communication of computing tools at the levels of local and advanced CS and PS interaction loops.

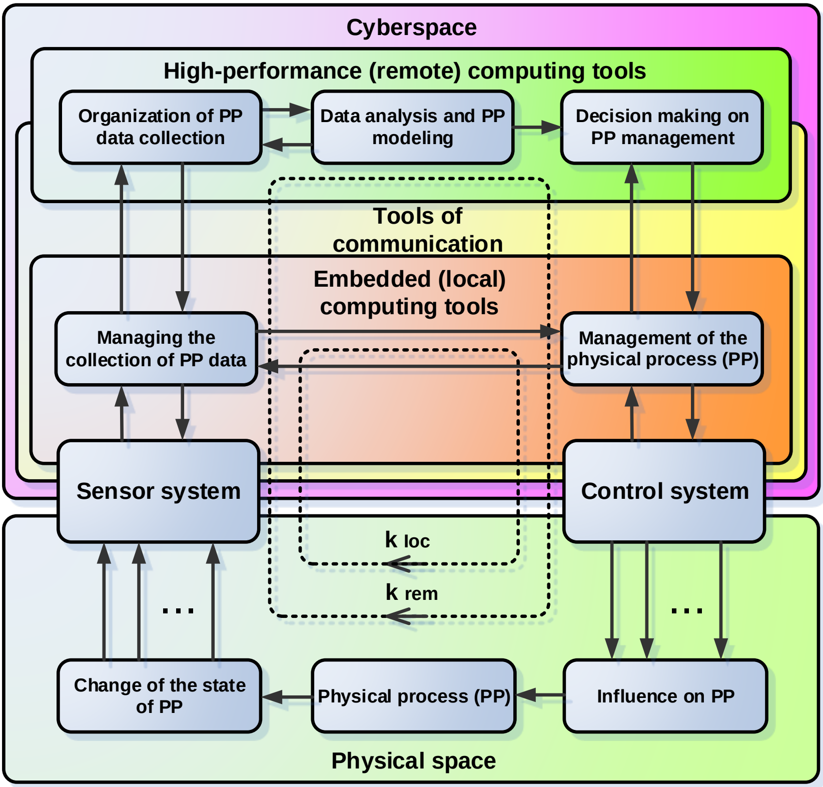

Cyberspace
High-performance (remote) computing tools
Organization of PP data collection
Data analysis and PP modeling
Decision making on PP management
Tools of communication
Embedded (local) computing tools
Managing the collection of PP data
Management of the physical process (PP)
Sensor system
Control system
k loc
k rem
...
...
Change of the state of PP
Physical process (PP)
Influence on PP
Physical space

Fig. 12.16 – Information model of multi-contour interaction of CPS with physical space

The given conceptual level of CPS structure detail is considered as basic in the process of detailing it. So, for example, the component "sensor system" contains measuring, computing and communication components from a set of basic components. The “physical process” (PP) component provides working with processes of various physical nature: thermal, diffusion, concentration control and management, measurement and correction of light fluxes and others. The same can be said about the module “data collection”, “management of the physical processes”, “decision making”, etc.

Along with the structural modules for processing and converting information between CPS components, the regularities of feedback links and cyclic multi-level hierarchical interaction are implemented. This requires additional mathematical apparatus to determine the interaction functions and the dynamics of transient processes in structural elements.

The generalized mathematical model of information processes at the basic level of the CPS structural organization (Fig. 12.16) can be described as a multi-parameter functional [49]:

$$M_F = \{[(S(p), K_e(S)),(A(p), K_e(A))], C, (Kh(S), K_h(M), K_h(A))\}\times P,$$

where $P=\{p\}$ – the set of physical processes which CPS interacts with; $S(p)$ – sensor system; $A(p)$ – executive system; K_e – embedded (local) computing tools, in particular $K_e(S)$ – PP data collection management tools, $K_e(A)$ – PP management tools; C – communication means; K_h – high-performance (remote) computing tools, in particular $K_h(S)$ – tools for organizing PP data collection, $K_h(M)$ – tools for data analysis and PP modeling, $K_h(A)$ – decision-making tools for PP management). Almost all components of the M_F model are also multi-parameter functionals, and their parameters describe the structural elements of the lower hierarchical level in the designed system. Also, these components can be grouped by the higher hierarchy level functionals. For example, two functional clusters of components can be distinguished from the above set [49]:

1) components that provide the functionality of the internal local CPS contour – $[(S(p), K_e(S)), (A(p), K_e(A))]$; their set may be sufficient for the implementation of simple autonomous CPS, which operate by coprocessor or library reconfguration models;

2) components of the external contour of interactions in CPS with the involvement of network access to high-performance computing tools – $(K_h(S), K_h(M), K_h(A))$.

Similarly, it is possible to group components by the levels of the 5C model, or by close functions performed, or by the similarity of the used mathematical models of the processes. For example, in the first cluster, subclusters of data collection $S(p), K_e(S))$, and control $(A(p), K_e(A))$ can be distinguished.

To move from the conceptual model of MF to a specific mathematical model, it is necessary to choose a calculation procedure and a basis of a calculus system for the corresponding processes (process calculus), describe the alphabet of events in the form of elementary operations, and establish an unambiguous correspondence between actions on CPS components and algebraic operators of calculus processes.

12.4.4 Information models of mass service systems (MSS) in CPS

Most of modern CPS and IoT technologies are designed to function in multitasking mode and search for solutions to multi-parameter problems. An important indicator of the technical perfection of such systems is the optimization of their hardware-structural complexity and time parameters of information processing. In computer systems, the improvement of these parameters is achieved by balancing the computational load between the system components due to software solutions, as well as using conveyor processing of informative data streams and control commands.

In CPS, a more generalized approach is possible in the form of mass service systems (MSS) information models that can be extended to both cyber resources and physical CPS components or their groups. The essence of MSS functioning lies in the organizational optimization of service requests from objects to minimize the number of requests in the queue, the time of service and the number of uncompleted requests. MSS application can be effective in CPS designed to solve the problems of logistics, maintenance and data collection from large arrays of sensors in technological complexes, environmental monitoring systems, medical devices, etc. Also interesting are application examples of MSS models and IoT technologies for synchronizing tasks of spatially distributed CPS, so-called multiagent systems. These are kits of the similar CPO that perform a common task, such as a set of quadcopter, humanoid robots, industrial robots, etc.

IoT tools and technologies can be used here for: 1) Ethernet/Internet data routing between cyber components using standard TCP/IP and UDP protocols; 2) implementation of wireless LANs IR, Bluetooth, WiFi, Zigbee or others; 3) development of our own non-standardized data exchange protocols on specialized radio channels.

Promising are the studies of the principle of functional CPS completeness, which determines its perfection, versatility and ability to solve various tasks on the functioning principles of MSS and IoE. According to the law of requisite variety, proposed by William Ashby [49] for cybernetic systems, CPS allows to maximize a person's ability to manage physical processes by combining almost all types of cybernetic devices in their entirety. The M_F model and the MSS theory can serve as a basis for the research and application of the law of necessary diversity in the context of CPS.

12.4.5 Models of Petri Networks for IoT and CPS technologies

The events of the real physical world in most cases are asynchronous. In multi-tasking systems, this can cause collisions when processing large arrays of input data. Therefore, an important question in the functional design of CPS and hybrid CPS/IoT structures is the correct synchronization of information and control flows at all levels of the conceptual system model. The formalism of extended Petri nets includes multi-marker markup and model time, allowing modeling dynamic behavior of CPS system objects [56, 57]. Such problem-oriented extended Petri nets are been developed and used in practice: E-, M-, comby-, FIFO-nets, and others.

In [58], the influence of temporal topology dynamics and nodes mobility on the quality of connections and data transfer in communications of IoT technologies is investigated. High-level Petri nets have been used for modeling and formal analysis of communication processes. The authors proposed three schemes for unloading mobile traffic based on the mobility analysis and schemes of temporary contacts of network nodes to predict future data transmission capabilities.

The methodology of Petri nets should also be used to model the functionality of the lower levels of the CPS/IoT conceptual model, for example, sensor networks.

12.5 Work related analysis

The material presented in this section concerns the authors’ vision of the current state and development of CPS and IoT technologies, as well as their synergy with IoE, SNSS and is presented based on the publication analysis of world-famous scientific schools and materials of leading universities of USA, Europe, Ukraine. The structure and educational material presentation is based on the course “Technologies of the Internet of Things and Cyber-Physical Systems”, which was lectured by the authors and discussed with students of the specialty “Computer Engineering” in 2017-2019 at Yuriy Fedkovych Chernivtsi National University.

The greatest influence on the vision formation of the problem “CPS and IoT as a basis Industry 4.0” as played by scientists' publications from NIST, NSF, The University of California in Berkeley [10, 39, 46] and other USA institutions, in particular C. Greer, M. Burns, D. Wollman and E. Griffor, E. Lee and S. Seshia, S. Sarma, D. Brock and K. Ashton, and others. The course “EEL 6673 Cyber-Physical Systems Identification and IoT Applications” [65], which is taught to students at the University of Florida at the Department of Electrical and Computer Engineering, was also analyzed.

Scientific publications and best practices on CPS and IoT of leading schools in Europe are analyzed, including:

- Newcastle University [63] and Leeds Beckett University [64] in the United Kingdom,
- the Royal University of Technology (KTH) in Stockholm, Sweden [34-38, 40, 61, 62],
- the University of Coimbra in Portugal [19, 32, 60],
- the University of Pisa, Italy [41],

as well as reports on the «Cyber-Physical European Roadmap & Strategy, CyPhERS» project [33] by the 7th EU Framework Program (FP7-ICT, ICT-2013.3.4) coordinated by representatives from Germany (Bernhard Schätz from fortiss GmbH (Coordinator) and Thomas Runkler from Siemens AG (affiliate partner)), Sweden (Mart in Törngren from Kungliga Tekniska högskolan, KTH), France (Saddek Bensalem from the Université Joseph Fourier, Grenoble), Italy (Roberto Passerone of the Università degli Studi di Trento), Great Britain (John McDermid from The University of York).

To improve the Masters and PhD training courses using thorough reviews of J. Wan and colleagues [29] on the development of CPS and IOT technologies from inter-machine (M2M) information exchange to modern smart technologies, P. Ray (Sikkim University, India) [30] on the peculiarities of architectural IOT decisions depending on the application field, R. Alur et al. [31] on the analysis of tasks related to the computer technology of IOT will be helpful.

In Ukraine, this problem both in terms of education and R&D is developing at the universities which participate in consortium of «*Internet of Things: Emerging Curriculum for Industry and Human Application*» (ERASMUS+ project “ALIOT” 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP) project [59], and also at the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, at National University “Lviv Polytechnic” and others, where relevant educational programs are provided and CPS and IOT conceptual issues are explored.

Conclusions and questions

The becoming of a new scientific or educational direction always faces the problem of defining its own object and subject of research, and differentiating it among the other already existing areas. Now this applies to CPS and IoT. The presented material attempts to find an answer to these questions. It is shown that the synergy processes of achievements in the field of CPS and IoT allow solving complex issues of modern industry and the humanitarian sphere. They are the basis for the development of IoE, SNSS, and self-organizing cybernetic systems. One of the approaches for system analysis and synthesis of modern CPS is proposed and the crucial role of IoT technologies in this process is justified. The models of CPS and IoT and the possibilities for their improvement using methods of system analysis, Petri nets, the ideology of mass service systems are reviewed and analyzed.

For a better understanding and assimilation of training material, the search for new solutions and challenges when developing new CPS and IoT solutions, a discussion and search for answers to the following questions are proposed:

1. What is the essence of terms CPS, IoT, Industry 4.0?
2. What do we mean by "CPS and IoT ecosystems"?
3. What became the basis for the emergence of CPS and IoT as new scientific areas? What is their similarity and difference?
4. Is there a difference between mechatronics and cyberphysics?
5. Analyze the structural levels of the 5C CPS model?
6. What does the CPS concept map describe?
7. What are the differences and similarities between CPS and IoT conceptual models?
8. Justify the place of CPS and IoT among the tasks and applications of IoE?
9. What are the advantages and disadvantages of the SNSS era?
10. Analyze the advantages and disadvantages of a hierarchically- modular approach to the synthesis and analysis of CPS and IoT?
11. What CPS problems can be solved with IoT technologies?
12. What the resilience is? How is it provided for CPS by IoT technologies?
13. What are the self-organization principles of cybernetic systems?
14. What is the 3S model and what role does reconfiguration of embedded computers play in it?
15. Analyze the functional varieties of reconfigurable systems?
16. Justify the stages and tasks of modeling CPS and IoT structural solutions?
17. Analyze the features of the S-D-M-P-F-A structural model of the CPS cyber component and its relation to the multi-loop model of CPS interaction with PS?
18. Justify the formulation of system analysis problem for the CPS and IoT synthesis?
19. Justify the possible application areas of the described method for analysis and evaluation of computational resources of the projected CPS?
20. What is the role of Microsoft Azure (Azure), GOOGLE Cloud Platform (GCP), Amazon Web Services (AWS) software platforms in CPS and IoT design?

21. What is the essence of principle of recirculating information and control streams in CPS? What models describe it?
22. Justify the need for parallel computing in CPS? What is the role of IoT?
23. Make a comparative analysis of fog and edge calculations in CPS? What is the role of IoT?
24. Analyze the advantages and disadvantages of CPS and IoT synergy? What is the essence of the CPS/IoT hybrid model?
25. Analyze a generalized CPS and IoT functional model at the level of data flows?
26. Justify the stages and tasks of information process modeling in CPS and IoT?
27. Describe and analyze the information model of multi-contour CPS interaction with physical space?
28. Justify the possibility of using MSS information models to describe the functionality of multiagent CPS?
29. Justify the possibility of using Petri nets to describe the functionality of sensor systems in CPS?
30. Analyze the possible impact of temporal topology dynamics and node mobility on the quality of connections and data communications in IoT communications?

References

1. K. Schwab, *The fourth industrial revolution*. Crown Publishing Group, Division of Random House Inc, 2017.

2. "Industrial Revolution 4.0. On the threshold of a new era" (Ukrainian), *Ua.korrespondent.net*, 2017. [Online]. Available: https://ua.korrespondent.net/business/web/3802445-promyslova-revoluitsiia-40-na-porozi-novoi-epokhy. [Accessed: 05- Jul- 2019].

3. S. Jeschke, "Everything 4.0? – Drivers and Challenges of Cyber Physical Systems", http://docplayer.net/7150740-Everything-4-0-drivers-and-challenges-of-cyber-physical-systems.html, 2013.

4. "The Fourth Industrial Revolution (Ukrainian)", *Uk.wikipedia.org*, 2019. [Online]. Available: https://uk.wikipedia.org/wiki/Четверта_промислова_революція. [Accessed: 05- Jul- 2019].

5. I. Kalyaev, V. Lohin, I. Makarov et al., Intelligent robots: a manual for universities / under the general ed. E.I. Yurevich /. M .: Engineering, 2007. (on Russion)

6. D. Robbins and M. Tanik, "Cyber-Physical Ecosystems: App-Centric Software Ecosystems in Cyber-Physical Environments.", in *Applied Cyber-*

Physical Systems, Springer Science+Business Media New York, 2014, pp. 141-147.

7. D. Moldovan, G. Copil and S. Dustdar, "Elastic systems: Towards cyber-physical ecosystems of people, processes, and things", *Computer Standards & Interfaces*, vol. 57, pp. 76-82, 2018. Available: https://www.infosys.tuwien.ac.at/Staff/sd/papers /Zeitschriftenartikel_2018_D_Moldovan_Elastic.pdf

8. C. Greer, M. Burns, D. Wollman and E. Griffor, *Cyber-Physical Systems and Internet of Things*. NIST Special Publication 1900-202, 2019, p. 61. . Available: https://nvlpubs.nist.gov/nistpubs/ SpecialPublications/NIST.SP.1900-202.pdf

9. V. Svyatnyi and D. Brovkina, "Modern tendencies in industrial complexes automation" (Ukrainian), *System research and information technology*, no. 1, pp. 32-39, 2016. Available: http://ela.kpi.ua/bitstream/123456789/15592/1/GM_Sviatny_Brovkina_N1_2016.pdf. [Accessed 5 July 2019].

10. E. Lee and S. Seshia, "Introduction to Embedded Systems - A Cyber-Physical Systems Approach", *Ptolemy.berkeley.edu*, 2019. [Online]. Available: https://ptolemy.berkeley.edu/books/leeseshia/releases/LeeSeshia_DigitalV1_08 .pdf. [Accessed: 05- Jul- 2019].

11. N. Wiener, *Cybernetics or control and communication in the animal and the machine*. Mansfield Centre, CT: Martino, 2013.

12. B. V. Glushkov, Introduction to Cybernetics.(Russian) Kiev: Publishing House of the Ukrainian SSR Academy of Sciences, 1964.

13. E. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?", *Ptolemy.eecs.berkeley.edu*, 2006. [Online]. Available: https://ptolemy.eecs.berkeley.edu/publications/papers/06/CPSPositionPaper/Lee_CPS_PositionPaper.pdf. [Accessed: 05- Jul- 2019].

14. K. Ashton, "That 'Internet of Things' Thing", *Rfidjournal.com*, 2009. [Online]. Available: http://www.rfidjournal.com/articles /view?4986. [Accessed: 05- Jul- 2019].

15. S. Sarma, D. Brock and K. Ashton, "white paper The Networked Physical World Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification", *Semanticscholar.org*, 2001. [Online]. Available: https://www.semanticscholar.org/paper/The-Networked-Physical-World-Proposals-for-the-Next-Sarma-Brock/88b4a255082d91b3c88261976c85a24f2f92c5c3. [Accessed: 05- Jul- 2019].

16. J. Esquer, F. González-Navarro, B. Ríos, L. Burtseva and M. Vargas, "Tracking the Evolution of the Internet of Things Concept Across Different

Application Domains",*www.semanticscholar.org*, 2019. [Online]. Available: https://pdfs.semanticscholar.org/6672/ 6950d756256ca479a068664ecb052b048991.pdf. [Accessed: 05- Jul- 2019].

17. Terasic.com.tw. (2019). *Terasic - All FPGA Main Boards - Arria 10 - Heterogeneous Extensible Robot Open Platform*. [online] Available at: https://www.terasic.com.tw/cgi-bin/page/archive.pl? Language=English& No =1173 [Accessed 5 Jul. 2019].

18. J. Morra, "Xilinx Adapts to an Adaptive Future of Computing", *Electronic Design*, 2019. [Online]. Available: https://www.electronicdesign.com/industrial-automation/xilinx-adapts-adaptive-future-computing. [Accessed: 05- Jul- 2019].

19. "A “made in Portugal” technology for IoT growth", *https://www.aicos.fraunhofer.pt*, 2016. [Online]. Available: https://www.aicos.fraunhofer.pt/en/news_and_events_aicos/news_archive/older_archive/a-_made-in-portugal-technology-for-iot-growth.html. [Accessed: 05- Jul- 2019].

20. B. Bagheri, S. Yang, H. Kao and J. Lee, "Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment", *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1622-1627, 2015. Available: 10.1016/j.ifacol.2015.06.318.

21. N. Suda, “Reconfigurable Architectures and Systems for IoT Applications”, Dissertation Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy. Arizona state university, 2016. – 83 p. – N. Suda, *Repository.asu.edu*, 2019. [Online]. Available: https://repository.asu.edu/attachments/ 164110/ content/Suda_asu_0010E_15651.pdf. [Accessed: 20- Jul- 2019].

22. "Opto22 - 2173 Your IoT Primer: Bridge the Gap between OT and IT", *Opto22.com*, 2019. [Online]. Available: https://www.opto22. com/support/resources-tools/documents/2173-your-iot-primer-bridge-the-gap-ot-and-it. [Accessed: 20- Jul- 2019].

23. G. Chen, “Internet of Things towards Ubiquitous and Mobile Computing”. *Microsoft Research Asia. Faculty Summit, 18-19 October, Shanghai, China. October 17*, 2010. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/ 2010/07 /Guihai-Chen_Oct19.pdf

24. R. D. Sriram, "Toward Internet of Everything: IoT, CPS, and SNSS", OntologPSMW. *Ontologforum.org*, 2019. [Online]. Available: http://ontologforum.org/index.php/ConferenceCall_2015_03_12. [Accessed: 20- Jul- 2019].

25. R. D. Sriram, "Toward Internet of Everything: Architectures, Standards, & Interoperability", 2017. [Online]. Available: https://www.brighttalk.com/webcast/15321/256305

26. Nam Yong Kim, Shailendra Rathore, Jung Hyun Ryu, Jin Ho Park and Jong Hyuk Park, "A Survey on Cyber Physical System Security for IoT: Issues, Challenges, Threats, Solutions”.", J,Inf Process Syst, vol. 14, no. 6, pp. 1361-1384, 2018. Available: 10.3745/JIPS.03.0105 [Accessed 3 August 2019].

27. D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique and E. Bartocci, "A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems", *IEEE Access*, vol. 7, pp. 13260-13283, 2019. Available: https://publik.tuwien.ac.at/files/publik_ 275267.pdf.

28. V. Harchenko, N. Zahorodna and R. Kozak, "Fundamentals of security and resilient computing", *Diit.edu.ua*, 2017. [Online]. Available: http://diit.edu.ua/sites/tempus/files/full/1%20(1).pdf. [Accessed: 05-Jul- 2019].

29. J. Wan, M. Chen, F. Xia, L. Di and K. Zhou, "From machine-to-machine communications towards cyber-physical systems", *Computer Science and Information Systems*, vol. 10, no. 3, pp. 1105-1128, 2013. Available: https://pdfs.semanticscholar.org/ 3902/a278567a7f29660e9a46ea4377c66d3414c0.pdf.

30. P. Ray, "A survey on Internet of Things architectures", *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291-319, 2018. Available: 10.1016/j.jksuci.2016.10.003.

31. R. Alur et al., "Systems Computing Challenges in the Internet of Things", *Cra.org*, 2019. [Online]. Available: https://cra.org/ccc /wp-content/uploads/sites/2/2015/09/IoTSystemsChallenges.pdf. [Accessed: 20-Jul- 2019].

32. "CPS Week 2018 | Cyber-Physical Systems Porto, April 10-13", *Cister.isep.ipp.pt*, 2018. [Online]. Available: http://cister. isep.ipp.pt/cpsweek2018/l.pdf. [Accessed: 05- Jul- 2019].

33. "CyPhERS. Cyber-Physical European Roadmap & Strategy. Research Agenda and Recommendations for Action", *Cyphers.eu*, 2013. [Online]. Available: http://cyphers.eu/sites/default/files/ d6.1+2-report.pdf. [Accessed: 05- Jul- 2019].

34. J. Deshmukh, "How can CPS education provide what the industry needs?", https://www.kth.se/polopoly_fs/1.518392.1550156534!/ CPS%20Ed%20Workshop_JyoDeshmukh.pdf.

35. M. Grimheden, "Mechatronics Education at KTH (and Embedded Systems)", https://www.kth.se/polopoly_fs/1.518408.1550157760! /CPSED2014_Berkeley_MartinGrimheden.pdf.

36. A. Pinto, "A short list of skills needed to build autonomous systems", https://www.kth.se/polopoly_fs/1.518405.1550155301! /CPSEd-2014-Public_Alessandro.pdf.

37. J. Jensen, "Industrial Needs of CPS Education", https://www.kth.se/polopoly_fs/1.518393.1550156534!/CPSED%202014%20Jeff%20C.%20Jensen%20_print.pdf.

38. M. Törngren and M. Grimheden, "Towards curriculum guidelines for Cyber-Physical Systems", https://www.kth.se/polopoly_fs/ 1.518411.1550157306!/CPSed_CPScurriculum_TorngrenGrimheden.pdf, 2014.

39. "Design of Robotics and Embedded systems, Analysis, and Modeling Seminar (DREAMS)", *Ptolemy.berkeley.edu*, 2018. [Online]. Available: https://ptolemy.berkeley.edu/projects/ embedded/seminar/#1d9c8a. [Accessed: 05- Jul- 2019].

40. M. Törngren, "Roadmapping efforts for research, education and innovation in Cyber -Physical Systems", https://ptolemy.berkeley.edu/projects/chess/pubs/1080/CPS_roadmaps_UCB_ MartinTorngren.pdf, 2014.

41. S. Torre, "Enabling Technologies for Industrial Internet of Things (ET-I2oT 2019)", *Unipi.it*, 2019. [Online]. Available: https://www.unipi.it/index.php/engineering/item/6869-summer-school-internet-of-things. [Accessed: 03- Aug- 2019].

42. J. Rajamäki, "Industry-university collaboration on IoT cyber security education: Academic course: "Resilience of Internet of Things and cyber-physical systems" - IEEE Conference Publication", *Ieeexplore.ieee.org*. [Online]. Available: https://ieeexplore.ieee.org/document/8363477/. [Accessed: 03- Aug- 2019].

43. "Cyber-Physical Systems - a Concept Map", *Cyberphysicalsystems.org*. [Online]. Available: http://cyberphysicalsystems.org/. [Accessed: 03- Aug- 2019].

44. "Chess - Center for Hybrid and Embedded Software Systems", *Chess.eecs.berkeley.edu*. [Online]. Available: https://chess.eecs.berkeley.edu/. [Accessed: 03- Aug- 2019].

45. G. Sowa and A. Marchlewska, "The Internet of Things: Technological and Social Aspects", *Journal of Applied Computer Science Methods*, vol. 8, no. 1, pp. 17-27, 2016. Available: 10.1515/jacsm-2016-0002 [Accessed 3 August 2019].

46. E. Ackerman, "Spring-Loaded Drone Collapses Mid-Flight to Zip Through Windows", *IEEE Spectrum: Technology, Engineering, and Science News*, 2019. [Online]. Available:

https://spectrum.ieee.org/automaton/robotics/drones/spring-loaded-drone-collapses-midflight-to-zip-through-windows. [Accessed: 03- Aug- 2019].

47. "Leading Cloud Market Platforms (Russian)", in *Information Technologies and Systems 2017 (ITS 2017)*, Minsk, 2017, pp. 8-9. Available: https://www.bsuir.by/m/12_100229_1_119488.pdf

48. H. Vorobets and V. Tarasenko, "Self-configuring computer tools (Ukrainian)", in *Cyberphysical Systems: Achievements and Challenges: Proceedings of the Second Science Seminar*, Lviv, 2016, pp. 114-120. Available: http://195.22.112.37/bitstream/ntb/39386/1/20-114-120.pdf

49. V. Golembo and O. Bochkaryov, "Approaches to Building Conceptual Models of Cyberphysical Systems (Ukrainian)", *Ukrainian Journal of Information Technology*, vol. 864, no. 1, pp. 168-178, 2017. Available: http://science.lpnu.ua/uk/scsit/vsi-vypusky/vypusk-864-nomer-1-2017/pidhody-do-pobudovy-konceptualnyh-modeley-kiberfizychnyh. [Accessed 3 August 2019].

50. V. Melnyk, I. Lopit and A. Keith, "Information exchange protocol for computer devices automatic creation in reconfigurable hardware platforms of the cyber-physical systems computing nodes (Ukrainian)", in *Cyberphysical Systems: Achievements and Challenges: Proceedings of the Second Science Seminar*, Lviv, 2016, pp. 17–22.

51. "Object Store Features – Amazon S3 (Russian)", *Amazon Web Services, Inc.*. [Online]. Available: https://aws.amazon.com/ru /s3/features/. [Accessed: 03- Aug- 2019].

52. "Web Apps Create and deploy mission-critical web applications that scale with your business | Microsoft Azure", *Azure.microsoft.com*. [Online]. Available: https://azure.microsoft.com/en-us/services/app-service/web/. [Accessed: 03- Aug- 2019].

53. B. Briggs and E. Kassner, *Cloud Architectural Blueprints*. Redmond, Washington: Microsoft Press, 2016. Available: http://www.interface.ru/iarticle/files/39116_50835073.pdf

54. "Edge computing primer: IoT intelligence starts at the edge", *Processonline.com.au*. [Online]. Available: https://www.processonline.com.au/content/industrial-networks-buses/article/edge-computing-primer-iot-intelligence-starts-at-the-edge-736246826. [Accessed: 03- Aug- 2019].

55. "Fog vs Edge Computing: What's the difference?", *Info.opto22.com*. [Online]. Available: http://info.opto22.com/fog-vs-edge-computing. [Accessed: 03- Aug- 2019].

56. E. Ciortea, "IoT analysis of manufacturing using Petri Nets", *IOP Conference Series: Materials Science and Engineering*, vol. 400, p. 042010,

2018. Available: 10.1088/1757-899x/400/4/042010. https://iopscience.iop.org/article/10.1088/1757-899X/400/4/ 042010/pdf

57. L. Chen, L. Shi and W. Tan, "Modeling and Performance Evaluation of Internet of Things based on Petri Nets and Behavior Expression", *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 18, pp. 3381-3385, 2012. Available: https://maxwellsci.com/print/rjaset/v4-3381-3385.pdf. [Accessed 3 August 2019].

58. A. Ghosh, O. Khalid, R. Rais, A. Rehman, S. Malik and I. Khan, "Data offloading in IoT environments: modeling, analysis, and verification", *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1-23, 2019. Available: https://jwcn-eurasipjournals.springeropen.com/track/pdf/10.1186/ s13638-019-1358-8.

59. "Internet of Things: Emerging Curriculum for Industry and Human Applications ALIOT – An official web-site of Erasmus+ ALIOT project 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP", *Aliot.eu.org*. [Online]. Available: http://aliot.eu.org/. [Accessed: 03- Aug- 2019].

60. Internet Of Things Course - Immersive Programme Master in City and Technology [https://apps.uc.pt/search?q=Internet+of+Things]

61. "Master's programme in Information and Network Engineering | KTH | Sweden", *KTH*. [Online]. Available: https://www.kth.se/en/studies/master/information-and-network-engineering/master-s-programme-in-information-and-network-engineering-1.673817. [Accessed: 03- Aug- 2019].

62. "Master's programme in Communication Systems | KTH | Sweden", *KTH*. [Online]. Available: https://www.kth.se/en/studies /master/communication-systems/description-1.25691. [Accessed: 03- Aug-2019].

63. "Embedded Systems and Internet of Things MSc - Postgraduate - Newcastle University", *Ncl.ac.uk*. [Online]. Available: https://www.ncl.ac.uk/postgraduate/courses/degrees/embedded-systems-internet-of-things-msc/#profile. [Accessed: 03- Aug- 2019].

64. "Education in green ICT and control of smart systems : A first hand experience from the International PERCCOM masters programme", in *12th IFAC Symposium on Advances in Control Education, ACE 2019*, Philadelphia, United States, 2019. Available: https://hal.archives-ouvertes.fr/hal-02176670/document

65. *Catalog.fiu.edu*, 2019. [Online]. Available: https://catalog.fiu.edu/2018 _2019/graduate/College_of_Engineering_and_Computing/Graduate_Electrical _and_Computer_Engineering.pdf. [Accessed: 03- Aug- 2019].

PART IV. IOT TECHNOLOGIES FOR CYBER PHYSICAL SYSTEMS

13. IOT TECHNOLOGY IN THE PROBLEMS OF SYNTHESIS AND ANALYSIS OF CPS

Dr. H. I. Vorobets, Dr. O. I. Vorobets, Mr.S. V. E. Horditsa (ChNU)

Contents

Abbreviations

SoC – Systems on a Chip

OS – Operating System

MMU – Memory Management Unit

MPU – Memory Protection Unit

PLC – programmable logic controllers

TLS – transport layer security

DSP – Digital Signal Processing

WCF – Framework Communication Foundation

NFC – Near-Field Communication

BAN – Body Area Network

PAN – Personal Area Network

LAN – Local Area Network

CAN – Campus / Corporate Area Network

MAN – Metropolitan Area Network

WAN – Wide Area Network

SN – Sensor Network

ICPV– Intellectual Converters of Physical Values

BM DDP – Basic Module for Digital Data Processing

PM –Peripheral Module

CS – Computer System

MCM – Mathematical Coprocessors Module

PFLM – Program File Libraries Module
VM RMM – Visualization Means and display the Results of Monitoring and Modelling

BF – Basic Format

AF – Advanced Format

SoC – System-on-a-Chip

AP SoC – All-Programmable System-on-Chip

BLE – Bluetooth Low Energy

RFID – Radio Frequency IDentification

DSDV – Destination Sequence Distance Vector

CSS – Complex Security System

OTA –Over The Air

WSN – Wireless Sensor Networks

DoS attack – Denial of Service attack

RTOS – Real-Time Operating Systems

POSIX – Portable Operating System Interface

CAD system – Computer-Aided Design System

13.1 Modern elemental and technological base for CPS and IoT

The modern development of CPS and IoT technologies is primarily due to the achievements of microelectronics in the element base development of computer technology. Increasing the bit capacity of processors, increasing their speed performance and miniaturizing sizes allows transferring high-performance computing technologies from supercomputers to microprocessors. The development of micro technologies and the creation of systems on a chip (SoC), programmable logic structures CPLD, FPGA, solid-state SSD memory modules, as well as an improved ARM architecture of microprocessors made it possible to implement high-grade high-performance computer systems in the form of micromodules for embedded systems.

The process of CPS designing and developing provides for substantiation of conceptual questions: development of models and practical implementation of the “physical” and “cybernetic” system parts, and solving the problem of their communication [2]. A wide range of microprocessors and programmable industrial controllers with extended instruction sets and built-in communication modules is now available for creating cyber components [3,4]. Ready-made micromodular solutions that are easily adapted to solve arbitrary information and signal processing tasks, as well as training materials for mastering the design technology of CPS and IoT [5,6] are offered.

13.1.1 Evolution of microcontroller facilities and systems. 32- and 64-bit ARM architecture

The greatest opportunities for implementing CPS projects at present are possessed by 32- and 64-bit microcontrollers with ARM architecture, which are manufactured in three versions for: A – high-performance applications using Linux or Windows; R – focused on real-time systems and use in network equipment and embedded control systems; M – for energy-efficient low-power microcontrollers, including IoT devices [5,7].

Versions up to 2010, the last of which is ARMv3, used 26-bit memory addressing, and starting with ARMv4 – 32-bit (full ARM) with additional processing commands for 16-bit half-words. In the ARMv4T modification, for the first time, and up to the seventh version, a set of Thumb commands with missing system commands was implemented. It does not allow creating on their basis OS or system-independent programs that work without OS, however, it significantly reduces the amount of memory used for programs.

The ARMv5 version and its modification ARMv5xM distinguishes from ARMv4 by an extended command system, and ARMv5T from ARMv4T – by improved ARM and Thumb code reconciliation. In the ARMv5TE modification additionally sets of DSP commands for digital signal processing are implemented. The ARMv5TEJ has Jazelle Java virtual machine technology.

ARMv6 architecture was an improved version of ARMv5TEJ with extended sets of ARM and Thumb commands. Its advanced version of ARMv6T2 received an expanded Thumb-2 set with the number of 32-bit commands as close as possible to the ARM set. In ARMv6K version, a set of commands to simplify the implementation of the OS kernel and system software received the extension, and in ARMv6Z and ARMv6KZ – for the implementation of virtual machines and highly reliable systems. However, ARMv6K/KZ versions do not support the Thumb-2 command system and are compatible with ARMv6T2 only at the level of the basic version of ARMv6.

Starting with the ARMv7 version, Cortex hardware cores adapted for mobile devices, in particular tablets and smartphones, have been implemented. The ARMv7-A modification with the Cortex-A core implements ARM, Thumb/Thumb-2, and ThumbEE command sets, and maintains software compatibility from the previous versions up to the latest one. The Cortex-A core has a built-in memory management unit (MMU), has possibilities to place several cores and a graphics processor (Mali) on one chip, and is designed for high-performance mobile devices. NVidia has created a family of Tegra processors with a common solution - ARM + GPU nVidia, similar to personal computers.

In ARMv7-R (Cortex-R) versions, ThumbEE support is optional, and a simplified Memory Protection Unit (MPU) is used instead of MMU. ARMv7-R is designed for real-time systems and high-performance industrial controllers of high reliability, which do not require virtual memory.

It is worth noting that the versions of ARMv6-M and ARMv7-M with the Cortex-MX core are not compatible at all at the system level with other architecture versions, since they are intended for low-power microcontrollers of small and medium performance. They only implement Thumb (ARMv6-M, Cortex-M0 and -M1 kernels) and Thumb-2 (ARMv7-M, Cortex-M) command systems, and the ARM sets are not supported. However, such controllers are much more productive than 8-bit and 16-bit ones, and can be successfully used at lower levels of CPS and IoT models, providing format compatibility and necessary data processing accuracy with higher levels.

Since 2012, the ARMv8 architecture of 64-bit processors for version A was launched into production, while for the R and M versions, 32-bit was left.

Note that the versions of processor cores in ARM may not coincide with the versions of architectures, in particular, ARM7TDMI and ARM920T cores implement ARMv4T version, ARM926EJ-S core – ARMv5TEJ. Jazelle technologies and the ThumbEE command system are already outdated and out of use.

ARM Cortex-M3/M4 processors are designed for high-speed, low-power systems. 32-bit STM32 microcontrollers, implemented on their basis, are widely used both for working with peripherals and for creating basic platforms in modules of the higher hierarchy. [5].

EMW3165-P (Cortex-M4 32-bit core), RTL8710AF (Cortex-M3 32-bit), ESP32 (Xtensa® DualCore 32-bit LX6), ESP8266EX (Xtensa® SingleCore 32-bit L106)) processors have good parameters for WiFi communications in CPS modules and IoT technologies for connecting to peripherals and the Google Cloud Platform [6].

13.1.2 Principles of synthesis of CPS based on industrial microprocessor modules

The quality and efficiency of CPS functioning depends on the following main aspects: 1) communication with the periphery; 2) end devices functionality; 3) reliability of data exchange with modules of the higher hierarchy. The need to transfer computing load from the cloud (Cloud) to remote devices (Edge) to reduce IoT traffic increases the hardware and software requirements for computing power and minimizing the power consumption of peripheral modules. For such tasks, NodeMCU V3, Yotster Alfa, WeMos ESP32 with WiFi data channel are offered.

For industrial solutions programmable logic controllers (PLC) by SIMENS (representative office in Khmelnitsky), and by OVEN (PLC xxx) (Kharkov) made in accordance with European standards are offered in Ukraine. Their programming is carried out in the CODESYS environment [4]. They are intended for local automation systems and “complete” scalable solutions; construction distributed control and dispatch systems using both wired and wireless technologies.

The Ukrainian representative office of the French company Legrand in Zaporizhzhia offers comprehensive IoT solutions for smart homes.

One of the leaders in the world in PLC development is Bedford Associates [8], which in 1973 developed the MODICON technology (MOdular DIgital CONtrol). With the development of technology, the Fieldbus network was implemented in PLC (IEC 61158), but the basic architecture of IEC 61131 considers each PLC in the network as logically independent one with its own individual configuration. For IoT needs, the PLC now has: 1) the ability to work with enhanced transport layer security (TLS), 2) sufficient memory capacity for processing Internet protocol (IP) stacks, 3) advanced control algorithms, for example with Kalman filtering 4) additing Digital Signal Processing (DSP) instructions is supported. Some devices add special network processing equipment that offloads packet processing tasks from the main 32-bit processor - for example, the Renesas RX600. The Infineon Technologies XMC400 Series MCUs come with built-in EtherCAT support for real-time distributed control algorithms.

For the Siemens PLCs S7-1200 (FW 4.2+) and S7-1500 (FW 1.8+), the PDSql library was created, which allows to work with Microsoft SQL Server 2005 or higher and to execute all the basic MySQL commands.

Using a PLC for working with peripherals in CPS greatly simplifies system designing, since the hardware implementation is reduced to the development of a switching circuit, and requires appropriate software for IoT technologies.

In [9], Peter Papcun et al. considered a generalized approach for describing the functionality of PLC-based CPS interaction with the cloud (Fig. 13.1) using the Microsoft Asure Cloud service. Gevechi et al. [10] proposed management as a service for the case of PLC-based industrial automation (Fig. 13.2).

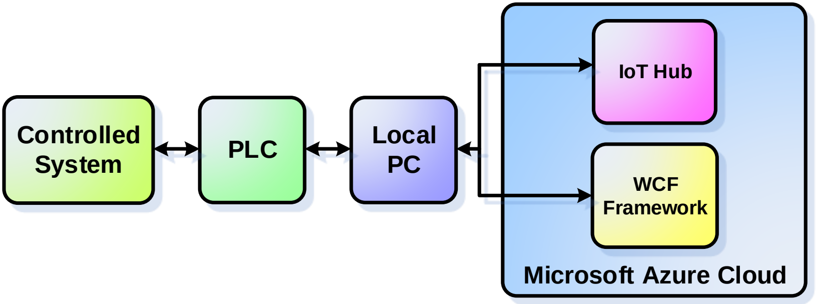


Fig. 13.1 – Proposed Architecture for Cloud Management in IoT [9]

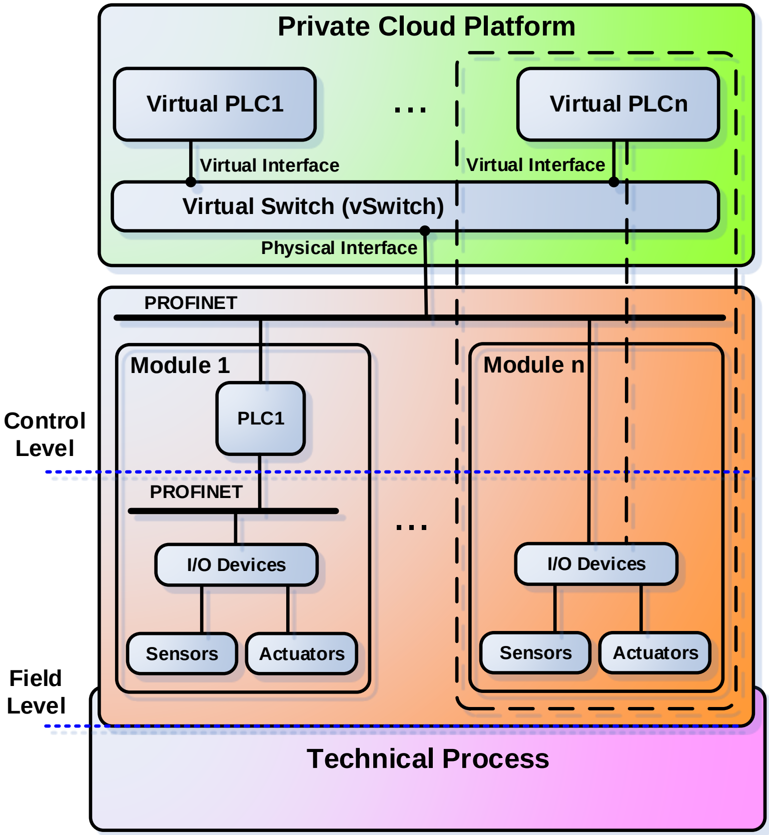


Fig. 13.2 – PLC-based cloud management model as a service for industrial automation [10]

To test the proposed ideas, real and virtual experiments were conducted. The real physical model controlled with the PID-controller law was located in the laboratory at the Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Computer Science, Technical University of Kosice. The model can be managed locally using the PLC controller, or from the cloud using the Microsoft Azure cloud services – IoT Hub and Framework Communication Foundation (WCF) according to the same PID-controller law. The performed experiments allowed exploring the differences in the temporal characteristics of PLC and cloud control depending on the transient characteristics of the controlled processes.

13.1.3 Principles of synthesis of CPS on the basis of programmable logic environments CPLD, FPGA

The transition to a hierarchical-modular structuring of CPS and IoT stimulated the development of designing in computer engineering in two ways: 1) the creation of unified modules and assemblies with a universal set of functions that can easily be adapted to solve many specialized applied tasks using only software; 2) the creation of universal prototypes and software for testing new algorithms and technical solutions, and testing of industrial designs. The hardware redundancy of universal computer systems based on CPLD, FPGA programmable logic structures [11-14], previously prompted the development of specialized design solutions, today has become competitive, as well as cost-effective compared to the cost of new industrial developments. This situation is due to the transition of crystalline production from micro to nanoscale. Along with this, the ideology of design work has shifted from circuitry to programming. Although CPS and IoT both require certain technical solutions, such solutions mainly come down to parametric reconciliation of technical parameters and dynamic characteristics of unified modules at different levels of CPS and IoT models.

One of the most famous manufacturers of such modules is Intel [11], which now owns stocks of the company, previously known as Altera. Based on the FPGAs series Cyclone® V, Intel®Cyclone® 10 Intel®Stratix® 10 and others, Intel offers multipurpose modules, for example – Intel®Stratix® 10 TX Signal Integrity Development Kit, etc. and the main educational materials for them [12] .

Xilinx [13] also offers FPGA programmable environments in the form of functional modules for industrial tasks and university training courses. This includes, in particular, Xilinx Spartan-3 of various modifications, Xilinx Spartan-6 LX FPGA XC6SLX150-3FGG484I 10/100/1000 Gigabit Ethernet transceiver (PHY) 2 × 512 MB DDR3 SDRAM Clock frequency 125 MHz [14]. The list of possible uses of the latter one is impressive:

- cryptographic hardware module;
- digital signal processing;
- Built-in educational platform;
- integrated industrial platform;
- integrated design system;
- emulation platforms;
- FPGA graphics;
- image processing;
- IP (intellectual) cores;
- low energy consumption design;

- parallel processing;
- rapid prototyping;
- reconfigured calculations;
- System-on-a-Chip (SoC) development.

However, Zybo (Zynq ™ Board) modules [15] (Fig. 13.3) have even greater possibilities for implementing stand-alone solutions with flexible architecture. It's multifunctional, ready to use, with built-in software based on the Zilnx-7000 chip of the Z-7010 family. Z-7010 is based on the Xilinx All All-Programmable System-on-Chip (AP SoC) architecture, which closely integrates the dual-core ARM Cortex-A9 processor with FPGA programmable logic arrays on field-effect transistors.

Combined with the rich assortment of multimedia and communication peripherals available on Zybo, Zynq Z-7010, a very powerful functionally complete CPS with IoT technology support for applicated tasks and of own system reconfiguration modes can be implemented. Built-in storage devices, video and audio input/output, dual USB-slots, Ethernet and SD slot provide developers with the ability to implement complex projects with the need to connect additional equipment. Six Pmod connectors are available for smart CPS and IoT projects, which allows system scaling and integrating of other projects.

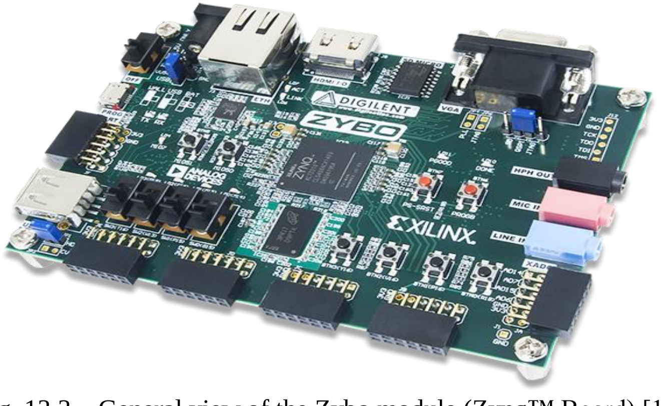


Fig. 13.3 – General view of the Zybo module (Zynq™ Board) [15]

For projects requiring specialized computing power for processing audio and video signals, Xilinx offers the Nexys Video Artix-7 FPGA module [16] (Fig. 13.4).

Nexys Video has several components, which makes it ideal for developing audio/video applications [16]. This is in particular:

- the most powerful chip in the Xilinx® Artix-7Artix®-7 family – the XC7A200T FPGA;
- Mini DisplayPort source, which provides a unidirectional, high-bandwidth board, low-latency audio/video channel.

The Nexys Video module is designed with a high level of integration with peripherals, which makes it suitable for educational, domestic and industrial applications.

Peripheral communication devices such as the on-board Ethernet, USB-UART, and high-speed USB allow Nexys Video to be integrated into large systems.

The technical solution proposed in Nexys Video, where the base board owns a rich set of service components – switches, buttons, LEDs and an OLED-display, FMC connector, four Pmod ports – allows using this module for prototyping, digital system testing, express analysis and verification of " fast "hardware-software solutions without additional involvement of extraneous equipment. This board also has good capabilities for scaling CPS and IoT projects.

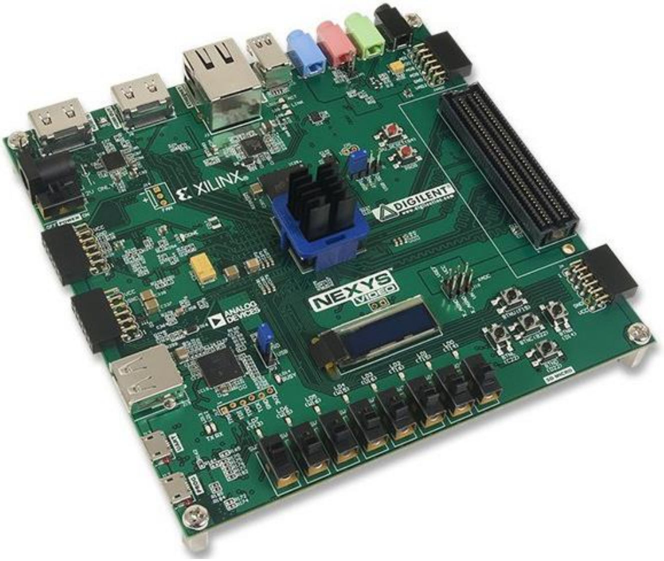


Fig. 13.4 – General view of the Nexys Video Artix-7 FPGA module [16]

In terms of analysis and synthesis of new CPS and IoT hardware solutions, the Nexys Video Artix-7 FPGA is compatible with both ISE® and Vivado® tools. It is supported in the free WebPACK™ versions of these tools so projects can be implemented at no extra cost. The free version of Vivado even includes the ability to create "soft" MicroBlaze™ processors [16].

The Nexys A7 module has a good set of hardware and software solutions for educational purposes (Fig. 13.5). It is a new version of the previously popular DDR Nexys 4 board, and uses the same Xilinx® Artix™ -7 (FPGA) programmable array of cells. The main purpose of the Nexys A7 is a ready-to-use platform for developing, simulating, testing circuitry solutions in a variety of application areas in a student audience. The Artix-7 FPGA architecture is optimized for high performance logic. Due to the high power of FPGA and a wide range of interfaces – USB, Ethernet and other ports, using Nexys A7 one can implement projects ranging from simple combinational circuits to powerful embedded processors. Additional peripherals, including an accelerometer, temperature sensor, digital microphone MEMs , speaker amplifiers and many I/O devices, allow the Nexys A7 to implement a wide range of laboratory work and CPS projects and IoT technologies without the need for other components.

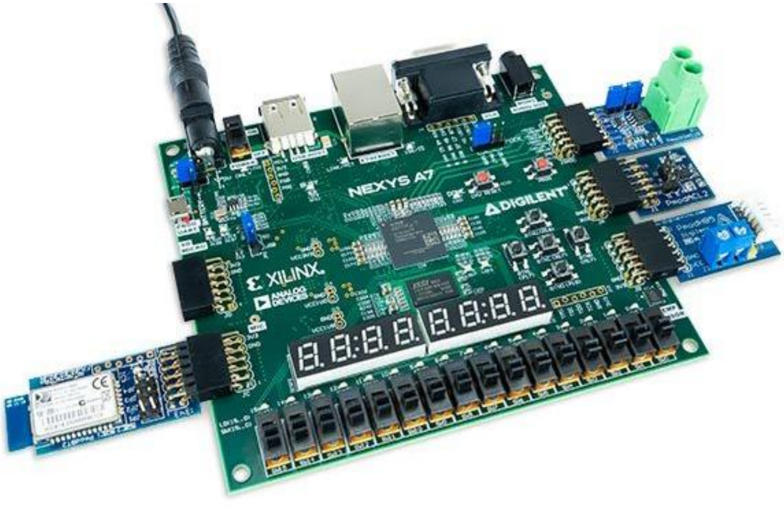


Fig. 13.5 – General view of the Nexys A7 module [17]

From a programming viewpoint, it is worth noting that Nexys A7-100T modification is compatible with Vivado® Xilinx design suite, as well as ISE® toolkit, including ChipScope ™ and EDK. Xilinx ISE support has been discontinued in favor of Vivado® Design Suite [17]. Nexys A7-50T modification is only compatible with Vivado® Design Suite.

In addition, Xilinx offers free WebPACK™ versions for users, so projects can be implemented at no extra cost. Nexys A7 is not supported by Digilent Adept.

13.2 Interfaces of open systems and network protocols IoT

The IoT information model is considered as three-level architecture: device (D) – gateway (G) – database/system (S) (Fig. 13.6). Information flows in such a model can be transmitted through four data transmission channels:

1. **Device-to-device (D2D)** – direct contact between two smart objects when they exchange information instantly without intermediaries. It is implemented in multi-agent systems, industrial works for coordination of complex joint actions.

2. **Device-to-Gateway (D2G)** – communications between intelligent sensors and gateway nodes that have more powerful computational parameters. Functions performed: data consolidation and relaying to the appropriate database for analysis. There are various IoT gateway protocols depending on the computing gateway capabilities, network power, reliability, data generation frequency and their quality.

3. **Gateway-to-data system (G2S)** – data transfer from the gateway to the appropriate data system. The choice of transmission protocol depends on the analysis of data traffic: collisions, congestion, security requirements, number of parallel connections.

4. **Between data systems (S2S)** – the transfer of information within data centers or clouds. The protocols for this type of connection should be simple to deploy and integrate with existing programs, have high availability, power and reliable disaster recovery.

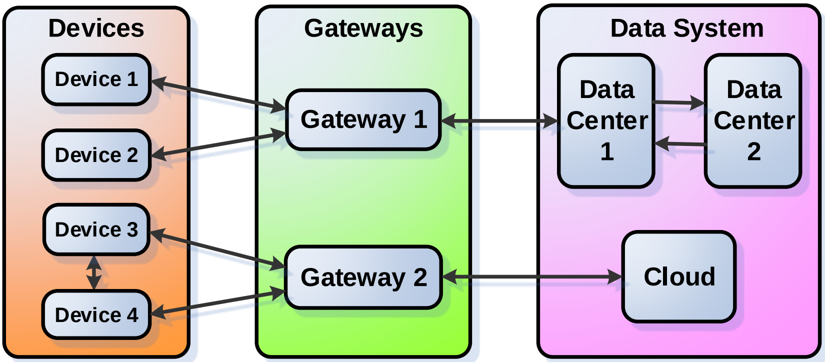


Fig. 13.6 – Three-layer D-G-S IoT architecture

Types of IoT networks are slightly different from the standard computer networks, and even more when considering the hybrid CPS/IoT model. They are divided into categories according to distance ranges (Fig. 13.7).

Nanonetwork is a set of small devices (no more than a few micrometers in size) that perform very simple tasks: sensing (probing), computing, storing, actuating. They are used in biometric, military and other nanotechnologies.

NFC (Near-Field Communication) – a low-speed network for connecting electronic devices at a distance within 4 cm of each other; application – contactless payment systems, identification documents and keys.

BAN (Body Area Network) – a network for connecting body-worn or near-body devices in various positions, embedded inside the body (implants).

PAN (Personal Area Network) – a network for connecting indoor(room) devices.

LAN (Local Area Network) – a network covering the area of one location / building.

CAN (Campus / Corporate Area Network) – a network that connects smaller local networks within a limited geographical area (enterprise, university).

MAN (Metropolitan Area Network) – A large network for a particular urban area operating on microwave transmission technology.

WAN (Wide Area Network) – a network that exists in large geographical areas and integrates various small networks, including LAN and MAN.

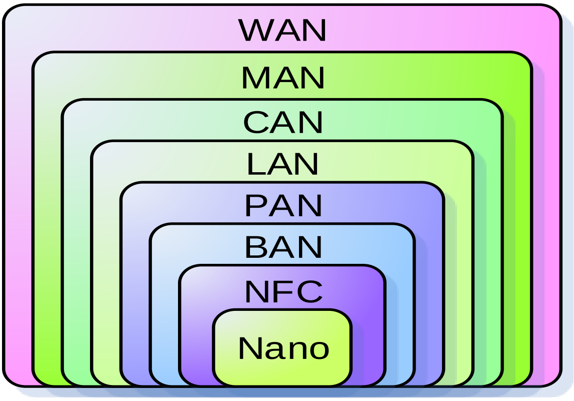


Fig. 13.7 – Types of IoT Networks

13.2.1. Sensor networks, nonstandard protocols of physical level in CPS

Sensor networks (SN) operate at the “smart connection” level of the 5C model and are designed to collect information about the state of the environment and the system itself. In addition, the “smart connection” level implements auto-identification, auto-connection and autorun features of smart devices and devices of the highest hierarchy. To ensure the full functionality of the sensor network, terminal devices are implemented in the form of intelligent sensors and intellectual converters of physical values (ICPV). ICPV itself consists of a sensor element that is sensitive to changes in the measured physical value, and a microcontroller circuit for processing the measured data. Depending on the type and functionality of the built-in controller, ICPV can perform analog-to-digital conversion of the measured signal, its formatting, storing or transmitting of the received digital data to the higher hierarchy module. The latter is implemented in most cases using wireless IoT technologies.

An example of CPS using a branched sensor network would be a monitoring system of ecological environmental parameters [11]. The basis of the system is the basic module for digital data processing (BM DDP) (Fig. 13.8) with functions of adaptive tuning according to the type of tasks being performed and, if necessary, for its own tuning at the software and hardware levels. It has an extensive system of interfaces and significant processing power.

The peripheral module (PM) covers arrays of heterogeneous ICPV. Its purpose is the primary processing of information signals, their formatting and marshaling the received data to the BM DDP for basic processing. Note that the separation of ICPV into separate structural units, and the separation of the functions of primary information processing and formatted data processing, unlike other similar technical solutions, allows unifying the presentation of information and standardize the requirements for CPS and SN basic units.

The modes of multifunctionality and accelerated computing can be provided due to the own power of the base module of the data center, or using software packages of a computer system (CS) or the cloud. For this purpose, the structural solution (Fig. 13.8) provides modules of mathematical coprocessors (MCM) and program file libraries (PFLM).

The interface buses for communication with ICPV and servers of the higher hierarchy level, as a rule, should be divided. This requirement is set out of reason that for certain types of ICPV non-standardized communication protocols, that take into account the functional features and the purpose of the “smart” sensors, may be used.

Thus, according to the generalized model of the system (Fig. 13.8), ICPV is entrusted with the tasks of identifying the types of measured signals, normalizing them and scaling the corresponding scale of the measuring converter. The functions of the BM DDP include the tasks of general data processing and measuring channels synchronizing. Processes of fast data processing are carried out using FPGA-based mathematical coprocessors, which, depending on the input information flows, are implemented in a programmable MCM environment. The set of performed tasks is limited by the library list of program files.

For the operative display of controlled data and the possibility of using the system in a mobile version or in production, it is advisable to equip it with visualization means and display the results of monitoring and modeling (VM RMM).

The measurement process can be activated in ICPV by the base program-loader, or initialized by a request from the BM DDP. In both cases, the control commands contain a complete description of the measurement process control procedures and instructions for the primary processing of information signals. The data format and the data exchange protocols for the ICPV – BM DDP channel are also selected.

Packet exchange mode allows both the accumulation of primary data in BM DDP memory, and its initial sorting and express analysis. The ICPV module provides only sorting of data from multi-parameter sensors and their short-term storage.

It should be noted that using ARM architecture processor cores of the same rank both the BM DDP and the ICPV significantly expands the possibilities for correcting the measurement control software and allows for the real time adjustment of the ICPV functioning modes by reloading the control program. This approach provides an extension of the system functionality, and increases its reliability and survivability.

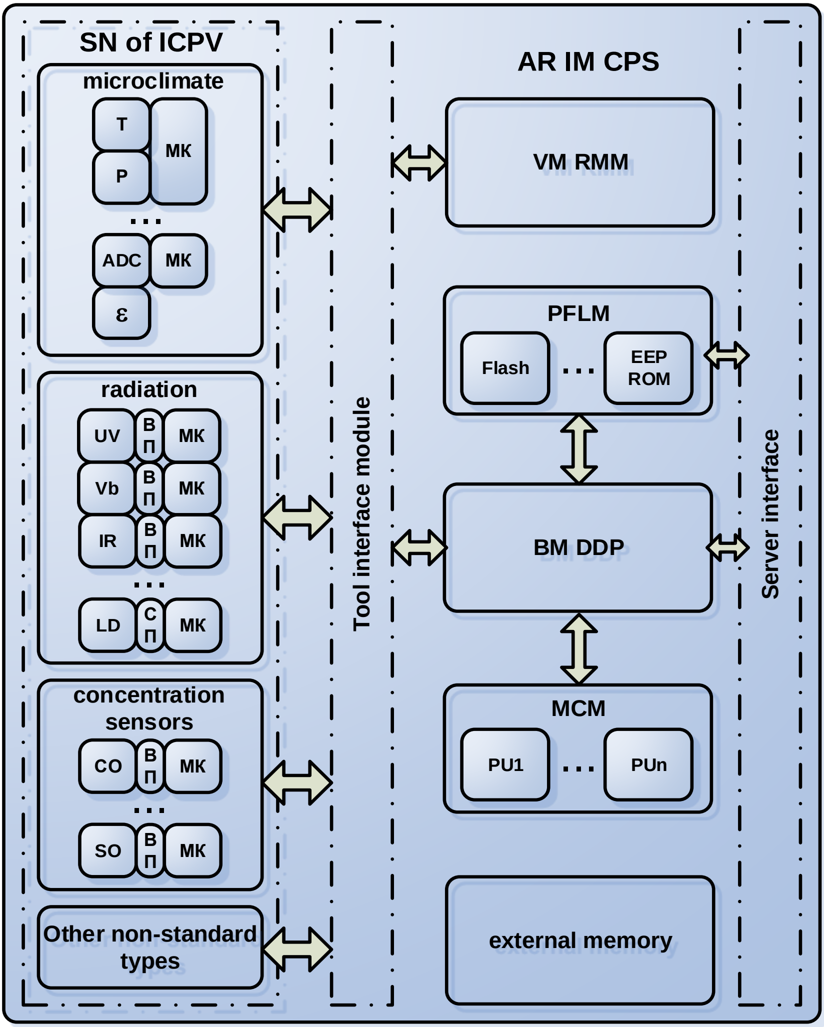


Fig. 13.8 – Generalized block diagram of adaptively adjusted information-measurement monitoring system of ecological environmental parameters

In addition, the using powerful ARM architecture core in ICPV allows one module to manage measurements of different classes' sensors – both analog and digital. It also enables the implementation of a peer-to-peer data exchange network with neighboring ICPVs. If necessary, it is possible to implement an algorithm for indirect control of remote SN access points through available points, if such a measuring network is implemented according to standard structural solutions.

Data formatting, and exchange protocols. Intelligent sensor for primary and secondary data processing, as well as the accumulation and transmission of data in a specific format, transmits data using a non-standard protocol. The forming packet protocol of 60 bytes size for data exchange with smart sensors can be represented in the basic (BF) or advanced (AF) format (Table 13.1).

Table 13.1 – Data transmitting formats between ICPV and BM DDP

	"10"	Command	SourceID	DestinationID	Number of Packet	Data	CRC	SME	"13"
BF	1	1	1	1	3	50	1	1	1
AF	1	1	4	4	3	112	1	1	1

Assigning code words in basic format is as follows:

- "10" – start-byte of the data packet;
- Command – command that is sent from/to the device (in this case it is a command of request-response type or marker transmission, etc.);
- SourceID – identifier of the device that initiates the message sending;
- DestinationID – identifier of the device to which the data is being sent;
- Number of Packet – packet number for reliable data synchronization;
- Data – array of data;
- CRC – checksum;
- SME – byte that indicates the beginning, middle and end of data transmission;
- "13" – end-byte of the packet transmission.

Application of ID code and ARM processors with advanced functionality allows implementing ICPV network self-identification technology according to Plug & Play principles, as well as monitoring SN activity using BM DDP. At the same time, the protocols for switching network elements and identifying access points do not provide for the use of the Host controller for network management, but implement "chain viewing" and active points identification and send configuration information to the BM DDP in real time upon its request.

13.2.2 Mesh networks, Zigbee protocols in CPS

The variety of implemented distributions in CPS at different levels requires the diversity of network technologies provided by IoT technology for CPS. Another problem is the rapid increase of communications number in the world, especially wireless, due to the introduction of IoT for CPS. According to forecasts [19], the number of "things" that will communicate with each other with people via radio communication will increase from 4,000,000,000 in 2020 to 125,000,000,000 in 2030. Broadband will reach more than 2.7 billion subscribers during this time. It is clear that in such a system, decentralized-type communications, characterized by high efficiency, fault resilience, adaptability, and self-organization ability, will have greater advantages. These requirements are met by mesh-networks organized by a cluster structure. The coverage area is theoretically not limited, and network nodes may act as a repeater (transport channel) or a subscriber access point. Reliability is ensured by automatic traffic redirection along a new route in case of any of the nodes failure, which guarantees not only the delivery of traffic to the recipient, but delivery in the minimum time. A special feature of mesh is the use of special protocols that allow each access point to create tables of network subscribers with monitoring the transport channel status and supporting dynamic routing along the optimal route. High scalability is implemented automatically by integrating new access points when they are turned on.

The IEEE 802.15.4 International Standard is the basis for protocols like ZigBee, WiFi, Bluetoot, 6LoWPAN. ZigBee is a communications technology for implementing low-speed, low-power wireless private networks with low power consumption and cost and high bandwidth. It is used in the information transmission both in everyday life and in industrial control devices and automation equipment, CPS, IoT. ZigBee specification is provided by ZigBee Alliance. A ZigBee network can contain a huge number of nodes (up to 65000) and connect them as a single management network in any industry [20].

It is known that mesh functionality was first laid down in the Bluetooth 5.0 standard as a platform for deploying scalable smart systems.

It can be stated that ZigBee is one of the most common technologies for building wireless networks of Internet of Things (IoT) (an open ZigBee standard). A ZigBee network with a mesh topology (mesh-network) has its own IEEE 802.15.4/Zigbee communication protocol stack, which does not support the Internet Protocol (IP). A computing network of objects based on the ZigBee stack, for interacting with external devices located on an IP network, is connected to the Internet through a dedicated IP gateway - Gateway ZigBee. The new ZigBee IPv6 standard has now been created. Networks based on the new Zigbee IPv6 standard can be connected to an IP network through a router rather than a dedicated gateway. The Gateway ZigBee repackages data from one format to another and provides interconnectivity between networks based on heterogeneous MQTT/ZigBee technologies – HTTP/TCP/IP.

13.2.3. IR, Bluetooth, RFID for local data transmit in CPS

Radiation and radio frequency data transmission technologies are similar in nature, but differ in implementation technology and application possibilities.

Devices based on infrared (IR) electromagnetic radiation are mainly implemented in the near (0.74-2.5 μm) short-wave area and are used in household complexes in open communication channels, or fiber-optic transmission lines. The disadvantage of open channels is a significant "exposure" of the information flow by the natural background. Special modulation schemes are used to avoid this.

In CPS/IoT systems, IR devices can be effectively used at the stage of obtaining and converting primary information as sensors. In particular, in systems for accurate measurement of the chemical composition of materials or gases, spectrometric studies, in data transmission systems over short distances. Although open data transmission channels based on laser emitters also were implemented.

The development of Bluetooth technology was began in 1989 by Ericsson. Later, a consortium was created that included many IT giants: Intel, IBM, Nokia, Toshiba and others. As a result of the merger, the Bluetooth Special Interest Group (SIG), a non-profit organization, was formed, which in 1998 released the very first specification of the standard, so far it is IEEE 802.15.1. [20].

Bluetooth technology has progressed significantly and has been expanded to provide not only traditional short-range audio transmission, but also as mesh connections for CPS, IoT and M2M-communications. The Bluetooth low-power standard has been updated to provide energy-efficient implementation of ZigBee and Bluetooth applications [20].

With the development of Bluetooth 4.0 version, the technology includes several protocols: standard Bluetooth, high-speed based on Wi-Fi, and the most promising - Bluetooth low energy (Bluetooth Low Energy, BLE).

Low-power technology is the most suitable and promising for CPS. Bluetooth LE consumes 10-20 times less energy and is quite capable for transmitting data 50 or more times faster at a distance of more than 100 meters than classic Bluetooth solutions. BLE has high security, reliability, low connection delay when connecting and low power consumption. An important feature of this standard is the adaptability of frequency tuning, that is, error correction occurs during signal transmission, BLE quickly changes its operating frequency, choosing the most optimal for eliminating obstacles, overflow problems and to reduce interference.

The Bluetooth 5.0 specification was created orienting on the Internet of Things. This finally showed that the standard seeks to “capture” the device market. Compared to the previous “integer” version 4.0, the data transfer speed was increased almost to the speeds of HSPA and LTE of earlier versions, while the power consumption remained at the same level. An important indicator for building the Internet of things is energy efficiency. In mobile devices, Bluetooth can be used to control various systems: lighting, home automation, security system and many others. An interesting option is applicating system of two technologies: Bluetooth NFC, where one of the technologies provides longer range, and the second serves to establish a fast and secure connection between two devices due to the shorter range (for example, wireless headphones with NFC-chip).

RFID (Radio Frequency IDentification) is a method of automatic radio frequency identification of objects, which is carried out through radio signals, or recorded data, called transponders or RFID tags. The system consists of a reader (interrogator) and a transponder (RFID-label or RFID-tag). According to the reading range, the RFID systems can be divided into short-range identification groups (reading is carried out at a distance of up to 20 cm), medium range (from 20 cm to 5 m) and long-range (from 5 m to 300 m). It is used as information sensors in CPS/IoT trade, logistics, etc.

13.2.4. Network protocols and computer network programming for CPS

As noted in [20], the main features of a mesh network both wired and wireless, are flexibility and the self-organization ability. To build routes with several switches and to determine the network topology, a mesh network requires a routing protocol. Such a protocol can be obtained from routing protocols for special networks. They are divided into proactive and reactive protocols. In proactive protocols, each node is responsible for accounting for individual or multiple routing tables. Such tables determine the topology of the entire network, or a specific fragment of it. Such routing protocols are also called "tables managed." The routing information in them is constantly updated for each node. Proactive routing protocols are used for networks with a small number of nodes. The most famous example of such a proactive routing protocol is presented by the DSDV (Destination Sequence Distance Vector) protocol [21].

In reactive routing protocol only routes to specific requests are fixed. Thus, they are characterized by reduced overhead. Their functional algorithm contains three main phases [21]:

1) opening the route necessary to search for a possible version of the existing route to an unknown destination;
2) route maintenance, which is used to detect disconnects and search for alternative routes;
3) gradual search using the step-by-step oncoming to limit the number of links passed when routing search is enabled.

In the absence of node activation, passive nodes do not generate any control or routing of information traffic. In these types of networks, nodes that do not actively participate in communication flows, moreover, the route is maintained equally as long as required for the initial source node. Among all available reactive routing protocols, the Dynamic Source Routing Protocol (DSR) [22] is well-known.

The AODV vector routing protocol [23] is designed for special networks. It supports unicast, multicast and broadcast communications. Installation of the route-vector by the sender node S0 is performed in a dialog mode "request-response" with the environment, and, when the destination node D0 is found, it is prescribed in the routing table of the sender. The general searching algorithm for AODV protocol is as follows:

1. The sender S_0 generates and transmits the REQuest Route (RREQ) packet to the nearest D_x node. The main fields in RREQ indicated by the sender S_0 are the recipient's code D_0 and the state n_j of the counter of the transition to the environment D_0 (D_j | j = 1 ÷ N).
2. If node D_x is an addressee of D_0, then it gives the REPly (RREP) acknowledgment to sender, if D_x is not an addressee of D_0, it transmits RREQ to the next node, increasing the counter value by one.
3. The S_0 host-sender builds a routing table based on the RREP responses and uses received information for subsequent transmissions.

The disadvantage of the protocol is the need to complete the current process of detecting the route before starting communication with D_0 node. One of the key features of AODV protocol is the destination serial number, which is also used for non-cycle routes. This number is generated by each node to maintain updated routing table entries. If there are two different routes from source S_0 to destination D_0, then S_0 selects the route with the highest number of destinations.

AODV protocol is used to check the integrity of the route on mobile LANs. If specific node is absent in the network for a certain period of time, he is sent a greeting-request. Lack of response to such requests after a specified number of hits is perceived as a loss of a particular RERR link and the routing table is reformatted. However, with a postback of REPly responses, the record may be restored if the sender needs it. Thanks to these mechanisms, the AODV protocol can quickly adapt to dynamic communication conditions, and it can be used with low power CPS / IoT devices and limited memory resources.

13.2.5. Methods of information protection in IoT technology for CPS

Information security issues in IoT and CPS technologies can be considered in the following aspects: communication security, device protection, device control and control of network interaction. Using this approach, a powerful and easy to develop complex security system (CSS) can be created. To protect the communication channel, technologies of encryption, authentication and reliability of service requests from remote elements, key management to verify the authenticity of the data and the reliability of the channels for their receipt are used. Protection of IoT and CPS devices is provided at the stage of transfer to the customer by embedding remote access tools for testing them – the so-called "over the air" (OTA) capability.

Threat objects in IoT are RFID, NFC, and WSNs (Wireless Sensor Networks) technologies. Desynchronization, information leaks, and repetition of attacks are dangerous for an RFID-system. Desynchronization attacks allow tracking tags, determining their location, blocking the data transfer from the tag to the reader. When using NFC-technology, the denial of service attacks (DoS attacks) or listening, replacing or inserting incorrect data are possible, and it requires secure communication channels. WSNs are vulnerable to attacks at the protocol stack level. This is realized by powerful extraneous radio frequency interference, or by spoofing, partial or complete replacement of traffic.

Botnets are dangerous for IoT devices. The Hide`N Seek botnet (HNS, January 2018) attacking IoT devices compromised 24,000 devices in a year. HNS is built on the basis of a modified version of Mirai and uses a decentralized peer-to-peer architecture and its own mechanism for P2P communications. Bots are able to execute commands to extract data, execute code, and interfere with the operation of devices. HNS is distributed through a combination of dictionary brute-force attacks and encoded list of credentials, finding devices with open Telnet ports in the network. The DoubleDoor botnet uses a combined firewall bypass (Juniper Netscreen) with bugs exploitation directly in the target devices.

Effective protection against interference to the IoT program code and substitution of sensor readings, anti-fraud, identification management, transaction management, element state verification of various systems, ensuring data integrity is the use of blockchain technology in CPS/IoT.

Dudykevych V. B. et al. [24] propose a paradigm and concepts for building multi-level CSS for CPS, which is oriented to the development of the conceptual foundations of secure interaction of levels and components in the space *“confidentiality – integrity – authenticity”* in accordance with the creation stages and functional implementation of CPS in subject areas (Fig. 13.9), which will allow for the secure exchange, processing, storage of measurement and service information.

The management system of CPS complex security is based on the *"plan-execute-check-act"* model and the *"object-threat-protection"* concept.

The CSS CPS creation methodology covers:

- systematic approach – principles of hierarchy, structuring, integrity, which provide the basis for CSS creation for CPS in the segment of optimal combination of normative-methodical, organizational, information, technical (hardware) and software providing at the stages of life cycle of automated systems;
- synergistic approach – the property of emergence, which manifests one of the integrity facets of information security in the CFS and assumes the presence of properties that are inherent in the complex security system of the CFS, but not inherent in its individual elements – complex security systems: cyberspace, communication environment, physical space.

Based on the analysis and classification of threats, it is proposed to develop their models, and models of countermeasure and security ensuring of CPS and the communication environment, as a basic element of IoT. The criteria for information security in CPS is the level of confidentiality ensuring for data and the results of their processing, the integrity of databases and knowledge developed as observations results , functional dependability of technical means, etc.

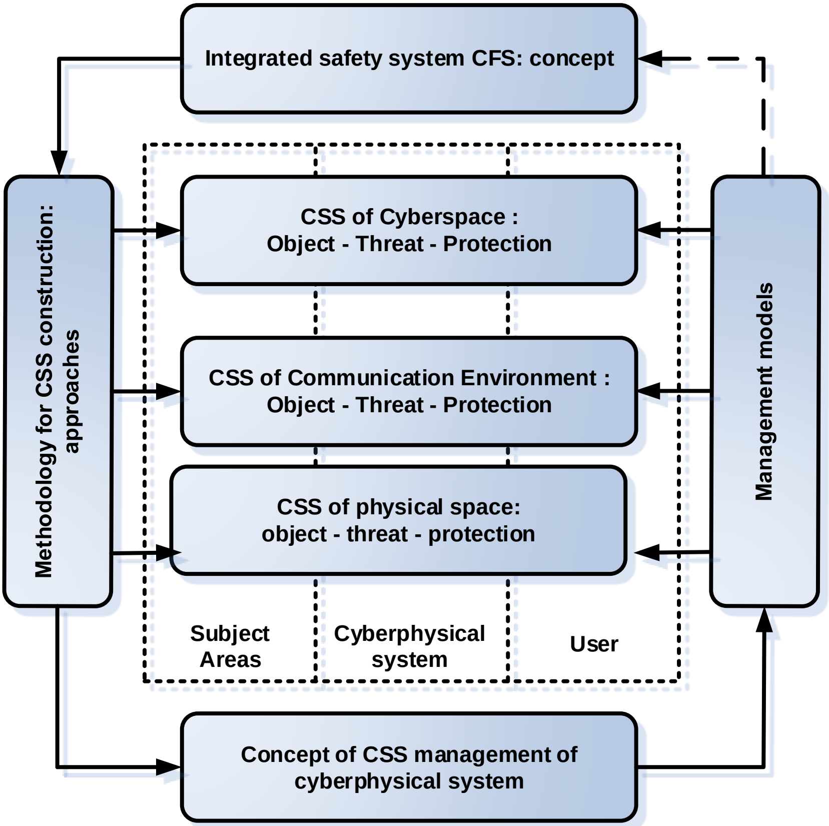

Integrated safety system CFS: concept
Methodology for CSS construction: approaches
CSS of Cyberspace : Object - Threat - Protection
CSS of Communication Environment : Object - Threat - Protection
CSS of physical space: object - threat - protection
Management models
Subject Areas
Cyberphysical system
User
Concept of CSS management of cyberphysical system

Fig. 13.9 – The paradigm structure of building a multi-level CSS for cyber-physical systems

Complex security system for the physical space is developed on the basis of the concept *“object - threat - protection”* in accordance with the segments [24]: sensors embedded in heterogeneous physical objects; special sensors built into the devices (electronic and aerospace reconnaissance, remote monitoring of the planet’s ecosystem parameters, monitoring emergency situations, detecting moving and stationary objects in military situations, searching for objects); MEMS devices (a wide range of applications, in particular in security systems) – requirements for the parameters of sensors in order to ensure the accuracy of the selection, recording and transmission of information in the cyber space, the processing of measurement data by systems and the transfer of information to the management of intellectual objects in physical space.

The proposed paradigm and concept of building CSS for CPS can be the basis for developing a system of multilevel protection of information and CPS and IoT technical objects.

To create a complex security system for the communication environment, it is proposed to segment data transmission technologies from the lowest to the highest level of the CPS conceptual model - ZigBee, Wi-Fi, Bluetooth, WiMAX, LTE, cloud computing, and developing measures for detecting and preventing extraneous interference to ensure data integrity, functional stability of the system.

13.3 Specialized software packages for simulation and synthesis of IoT and CFS

CPS and IoT software can be used at different stages of the project lifecycle. However, the success of the project is already determined at the stage of analysis and synthesis of technical solutions. In the first part of this stage, it is necessary to carry out modeling and research of the proposed technical solution. On the second - its schematic and design development, and on the third - system programming. Let's look at some features of the related software.

13.3.1 Ptolemy II

The Ptolemy II project [25, 26] is intended for modelling, simulation, design and research of complex embedded real-time systems. The basic principle of the project is to use well-defined computational models that control the interaction between components. The main problem being solved is the use of heterogeneous hybrid computing models. The software is implemented by Java in the form of a fully functional software system called Ptolemy II. Work is being conducted at the Center for Industrial Cyber-Physical Systems (iCyPhy) at the Department of Electrical Engineering and Computer Science, University of California, Berkeley, under the supervision of Professor Edward Lee.

The project website [25] provides more complete information on developments, software and training materials [26].

13.3.2 RTOS

Real-time operating systems (RTOS) [27] are designed to develop embedded computer software which is the foundation of CPS and IoT. Such systems are focused on minimizing the used hardware resources, and, accordingly, the programming code size. It allows to implement functionally complete projects based on modern microcontrollers and programmable logic structures. The software market now offers dozens of such OSs, in particular: Windows IoT (formerly Windows Embedded); Linux-based (Brillo, Huawei LiteOS, OpenWRT, Ostro Linux, Raspbian Linux); open OS Apache, Mynewt, FreeRTOS, TinyOS and others. Some operating systems are focused on mobile platforms, or are developed for specialized tasks, for example, NuttX is designed for unmanned aerial vehicles and drones operated with the APM/ArduPilot and PX4 UAV platforms. Let's look at some of them that can be used in CPS and IoT projects.

RIOT OS for embedded platforms is energy efficient and compatible with many wireless protocols. It is undemanding to resources and can run on MCU modules consuming 1.5 Kb of RAM and 5 Kb of flash memory. It differs in multithreading, dynamic memory allocation system, multi-platform, partial compatibility with POSIX (portable operating system interface) and C++ language support, although such set of options is more typical for Linux-systems than for RTOS.

A special feature of the RIOT OS architecture: an error in one of the modules does not “destroy” the entire system, increasing its reliability. The OS is licensed under the LGPL, and can be used in many applications. Applications written for Linux or OS X can be moved to devices with RIOT OS.

The main advantages of TinyOS are low power consumption, and simple but well-developed component architecture. Its specificity lies in the provision of mechanisms for the parallel execution of tasks in extremely limited resources. The OS is written in the C dialect – nesC – and is distributed under the terms of the BSD license.

TinyOS fits for experiments with low-power wireless networks and mesh networks. It is not intended to run resource-demanding software; it works with Cortex-M cores, and is tested on radio chips.

Zephyr OS is developed by the Linux Foundation and runs on x86, ARM, ARC architecture. It's designed for testing in a QEMU-based emulator, and comes under the Apache 2.0 license. Project participants were Intel, NXP Semiconductors/Freescale, Synopsys and UbiquiOS. Zephyr is based on River Rocket OS and Viper systems. The last one is a cut-down version of VxWorks. The core is designed to consume minimal resources (from 8 to 512 KB of RAM depending on the layout), which allows to use it in portable systems, from built-in sensors and chips for clothes to “smart” wireless gateways for IoT devices.

Minoca is one of the youngest RTOS. Its code is open under the GPLv3 license. The compilation is prepared for x86, ARMv6, and ARMv7 architectures, including generated boot images for Raspberry Pi 2, Raspberry Pi, BeagleBone Black, Asus C201, PandaBoard and Galileo boards, as well as QEMU-based emulator. The system is modular – the core subsystems are separated from each other. Device drivers are not attached to the core and are issued as universal executable files that are independent of the core's version.

13.3.3 Features of FPGA programming by Altera

Altera, which has been a division of Intel since 2016, offers Altera Quartus II programming environment for project developers using FPGA and CPLD chipsets. This CAD allows designing circuitry and logic of embedded modules in AHDL, VHDL, Verilog and other programming languages. The Altera Quartus II programming environment allows simulating projects, programming microcircuits, and much more. The latest enhanced version of Quartus II is called Altera Quartus Prime. There are paid subscriptions and free, but quite functional are Quartus II Web Edition, or the latest Quartus Prime Lite versions of the CAD. However, even free versions are often enough for professional work.

There is no doubt, there're some differences between Quartus II and Quartus Prime. First of all, it should be noted that for different FPGA series you may need different CAD systems. Detailed information and software versions can be obtained from the manufacturer [28].

13.3.4 Features of FPGA Programming by Xilinx

Xilinx [29] offers to technical solutions developers of embedded systems, CPS and IoT the software package ISE® design Suite, which supports Xilinx FPGAs of different families. This package works on Windows XP/7/Server and Linux, as well as for Spartan-6 device on Windows 10.

For new designs starting with Virtex®-7, Kintex®-7, Artix®-7 and Zynq®-7000, Xilinx recommends Vivado® Design Suite.

ISE Design Suite: Embedded Edition includes Xilinx Platform Studio (XPS), Software Development Kit (SDK), large plug-and-IP repository, including MicroBlaze™ Soft Processor and peripherals, and complete RTL-to-bitstream design process. Embedded Edition provides basic tools, technologies and a convenient design process for optimal results. The toolkit contains smart clock generators for dynamic power reduction, group design technology for multi-site development teams, saving the project for repeatability, synchronization, and partial reconfiguration for greater system flexibility, size, power optimization and cost savings.

The system version of ISE Design Suite: System Edition is based on Embedded Edition while adding System Generator for DSP™ to design high-performance DSP systems using Xilinx software, providing system modeling and automatic code generation with Simulink® and MATLAB® (The MathWorks, Inc.).

The ISE WebPACK release provides a complete design process and instant access to ISE features and capabilities at no cost. It also has several applications, more information can be found on the link [29] on the ISE WebPACK Design Software page.

13.3.5 Means for the synthesis and analysis of analog and digital circuits Altium Designer

Altium Designer is a comprehensive computer-aided design (CAD) system for electronic equipment developed by the Australian Altium company [30,31]. It allows implementing projects of electronic means at the level of the circuit or program code with subsequent information transfer to the designer of FPGA or printed circuit board. The feature of the program is the design structure and end-to-end integrity of the development at various design levels. In other words, changes in development at the board level can be instantly transferred to the FPGA or circuit level and vice versa. Also it's worth to note the integration of ECAD and MCAD systems as the priority direction of the developers of this program. The development of a printed circuit board is possible in three-dimensional form with two-way transfer of information to mechanical CAD systems (Solid Works, Pro/ENGINEER, NX, etc.)

This package consists of two products based on a single integrated DXP platform, the ability to work with one or another of them depends on the type of license purchased:

- Altium Designer Custom Board Front-End Design – FPGA design, circuit design and simulation.
- Altium Designer Custom Board Implementation – PCB and FPGA design.

The Altium Designer software package includes all the necessary tools for developing, editing and debugging projects based on integrated circuits and FPGA. The circuit editor allows entering hierarchical and multi-channel circuits of any complexity, as well as conduct mixed digital-analog modeling. The program libraries contain more than 90,000 ready-made components, many of which have footprint models, SPICE and IBIS models, as well as three-dimensional models. Any of the above models can be created using the program’s internal tools.

The Altium Designer PCB Editor encompasses powerful tools for interactive component placement and multi-layered wire tracing. Tracing tools take into account all the requirements of modern technologies, for example, the features of tracing differential pairs and high-frequency sections of boards. The program includes an automatic Situs tracer. To improve 3D modelling functions, the C3D graphics core was licensed in 2017.

Work on all parts of the project is carried out in a single shell Design Explorer. Changes made at any stage of development are automatically transferred to all related stages of the project. Altium Designer has extensive import and export capabilities of exterior designing systems and supports almost all standard source file formats (Gerber, ODB++, DXF, etc.). All developments in the form of circuits, boards and libraries developed in the latest versions of P-CAD are fully supported.

13.4 Work related analysis

The challenges of analysis, synthesis and synergy of CPS and IoT take an important place in the training process for computer engineers, developers of automation systems and information smart-systems. For Ukrainian universities, where the new educational programs on the technologies of the Internet of things and cyber-physical systems are being introduced, the practical experience and methodological development of training courses of the partner-universities from the European Union on the project “Internet of Things: Emerging Curriculum for Industry and Human Application” project (ERASMUS project "ALIOT" 573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP) are interesting and useful. The choice of material and its structuring in this section was influenced by seminars held in 2017-2018, and teaching materials from universities in Coimbra (Portugal) [32, 35], KTH in Stockholm (Sweden) [33] and in New Castle (Great Britain) [34].

Conclusions and questions

The problem of CPS analysis and synthesis and the place of IoT technologies in this process are considered in terms of three aspects: analysis of the modern element base for the development of cyber components, providing a reliable interface between system components at all levels of the conceptual model, and the capabilities of CPS modeling and synthesis software. The material is summarized as an overview of the general approach to this problem. You can study the material more thoroughly and prepare for the workshop using the cited literature.

For a better understanding and assimilation of the learning material, searching for new solutions and challenges when developing new CPS and IoT solutions, discuss and find answers to the following questions:

1. What are the features of the micromodular approach to CPS design?

2. What controller types with ARM architecture have Thumb, Thumb-2, Thumb/Thumb-2, and ThumbEE command sets?

3. Why it's said that ARMv6-M and ARMv7-M controllers have no true ARM architecture?

4. From which ARM architecture version do processors have 32-bit addressing? For which CPS/IoT solutions can they be used?

5. From which ARM architecture version do processors have 64-bit addressing? For which CPS/IoT solutions can they be used?

6. What processors are available in CPS and IoT modules for WiFi communication? Describe their characteristics.

7. Justify the advantages and disadvantages of PLCs. What solutions are they used for?

8. Describe the architecture for cloud management technology in IoT. Justify its advantages and disadvantages.

9. Describe the architecture of cloud technology as a management service in IoT. Justify its advantages and disadvantages.

10. Why has the hardware redundancy of universal computer systems based on CPLD and FPGA programmable logic structures become competitive in the development market? Give examples of quantitative estimates.

11. Analyze a list of Xilinx FPGA possible uses. Which ones are the most suitable for CPS/IoT projects?

12. Describe the architecture and parameters of the Zybo (Zynq™ Board) modules. What tasks are they most suitable for?

13. Describe the architecture and parameters of the Nexys Video Artix-7 FPGA modules. What tasks are they most suitable for?

14. Describe the architecture and parameters of the Nexys A7. What tasks are they most suitable for?

15. Describe the advantages and disadvantages of three-layer D–G–S IoT architecture. What are its disadvantages in terms of CPS?

16. Describe the features of IoT networks compared to regular computer networks.

17. Describe, what types of sensor networks can be implemented in environmental monitoring systems?

18. Describe the CPS and IoT synergy by the example of the environmental monitoring system functioning.

19. Justify, in which cases it's advisable to use unstandardized data transmission protocols? Give an example of this solution.

20. Justify the features of use and characteristics of Mesh-networks and Zigbee protocols in CPS.

21. Justify the usage features and characteristics of IR and Bluetooth for information transmission in CPS.

22. Justify the features of use and characteristics of RFID for information transmission in CPS.

23. Describe the features of the AODV vector routing protocol. What are its advantages over DSDV?

24. Describe the paradigm structure of the multilevel CSS construction for cyber-physical systems.

25. Justify the advantages and disadvantages of the paradigm structure of multilevel CSF construction for cyber-physical systems?

26. Prepare a brief overview of the functionality and principles of using Ptolemy II software.

27. Prepare a brief overview of the functionality and principles of using the Altera Quartus II programming environment.

28. Prepare a brief overview of the functionality and principles of using Xilinx ISE WebPACK software.

29. Prepare a brief overview of the functionality and principles of using Altium Designer software.

30. Justify the advantages and disadvantages of applying 2-3 known to you RTOS for CPS/IoT projects.

References

1. S. Lin, "FPGAs, SoCs, Microcontrollers – A Quick Rundown of IoT Devices", *Bitcoin Insider*, 2019. [Online]. Available: https://www.bitcoininsider.org/article/53125/fpgas-socs-microcontrollers-quick-rundown-iot-devices.

2. I. Horváth and B. Gerritsen, "Cyber-Physical Systems: Concepts, Technologies and Implementation Principles", 2019. [Online]. Available: https://www.academia.edu/14501665/Cyber-Physical_Systems_Concepts_technologies_and_implementation_principles.

3. "FasTrak SoftWorks, Inc.: PLC WorkShop Suite for Siemens 505", *Fast-soft.com*, 2019. [Online]. Available: http://www.fast-soft.com/page.php?20.

4. "Programmable Logic Controllers", *Owen.ua*, 2019. [Online]. Available: https://owen.ua/ru/programmiruemye-logicheskie-kontrollery?gclid=Cj0KCQjwhJrqBRDZARIsALhp1WSlFvQMudQ7RK161lrLhqG09fiSgzUGMN8KrvENYYRPnqaIO79mVVsaAiZ4EALw_wcB.

5. "M2R", *M2rtechnomations.com*, 2019. [Online]. Available: https://m2rtechnomations.com/course-2.html. [Accessed: 04- Aug- 2019].

6. "IoT - Using Cloud IoT Core to connect a microcontroller (ESP32) to the Google Cloud Platform", *Nilhcem.com*, 2019. [Online]. Available: http://nilhcem.com/iot/cloud-iot-core-with-the-esp32-and-arduino.

7. "ARM architecture", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/ARM_architecture.

8. [6]"Reinventing the PLC for Industry 4.0", *Rs-online.com*, 2019. [Online]. Available: https://www.rs-online.com/designspark/reinventing-the-plc-for-industry-40-1.

9. P. Papcun, E. Kajáti, C. Liu, R. Zhong and I. Zolotová, "Cloud-Based Control of Industrial Cyber-Physical Systems", 2019. [Online]. Available: https://www.researchgate.net/publication/332606551 _Cloud-based_Control_of_Industrial_Cyber-Physical_Systems..

10. Givehchi O., Imtiaz J., Trsek H., Jasperneite J., “Control-as-a-service from the cloud: A case study for using virtualized PLCs”, In: Workshop on Factory Communication CIE48 Proceedings, 2-5 December 2018, The University of Auckland [PaperNr]-14 Systems (WFCS 2014), Toulouse, France, p. 1 - 4, 2014, DOI: 10.1109/WFCS.2014. 6837587

11. "Intel® SoC FPGAs Programmable Devices", *Intel*, 2019. [Online]. Available: https://www.intel.com/content/www/us/en/products/ programmable/soc.html.

12. "Intel® Quartus® Prime Standard Edition Handbook Volume 3", *People.ece.cornell.edu*, 2019. [Online]. Available: https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/Power_Estimation/qts-qps-5v3.pdf.

13. "FPGA", *Xilinx.com*, 2019. [Online]. Available: https://www.xilinx.com/ support/documentation-navigation/silicon-devices/fpga.html.

14. "FPGA", *Xilinx.com*, 2019. [Online]. Available: https://www.xilinx.com/ support/documentation-navigation/silicon-devices/fpga.html.

15. "Zybo Zynq-7000 ARM/FPGA SoC Trainer Board (RETIRED)", *Digilent*, 2019. [Online]. Available: https://store. digilentinc.com/zybo-zynq-7000-arm-fpga-soc-trainer-board/.

16. "Nexys Video Artix-7 FPGA: Trainer Board for Multimedia Applications", *Digilent*, 2019. [Online]. Available: https://store. digilentinc.com/nexys-video-artix-7-fpga-trainer-board-for-multimedia-applications/.

17. "Nexys A7: FPGA Trainer Board Recommended for ECE Curriculum", *Digilent*, 2019. [Online]. Available:

https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/.

18. H. Vorobets, R. Hurzhui and M. Kuz, "The computerized system with the reconfigurable architecture for monitoring environmental parameters", *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 674, p. 55, 2015. Available: 10.15587/1729-4061.2015.40899.

19. A. Cilfone, L. Davoli, L. Belli and G. Ferrari, "Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies", *Mdpi.com*, 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/4/99/pdf. [Accessed: 06- Aug- 2019].

20. A. Ahmadi, M. Moradi, C. Cherif, V. Cheutet and Y. Ouzrout, "Wireless Connectivity of CPS for Smart Manufacturing: A Survey", 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01952355/document

21. C. Fang, X. Liu, P. M. Pardalos, and J. Pei, "Optimization for a three-stage production system in the Internet of Things: procurement, production and product recovery, and acquisition," Int. J. Adv. Manuf. Technol., vol. 83, no. 5–8, pp. 689–710, 2016.

22. G. Schuh, C. Reuter, and A. Hauptvogel, "Increasing collaboration productivity for sustainable production systems," Procedia CIRP, vol. 29, pp. 191–196, 2015.

23. J. Bao, Q. Wang, X. Zhen, J. Zhang, and X. Ji, "A humanmachine interaction approach of block erection schedule with three-dimensional spatial constraints," Concurr. Eng. Res. Appl., vol. 24, no. 4, pp. 359–368, 2016.

24. V. Dudykevych, V. Maksymovych and G. Mykytyn, "The Paradigm and Concept of Construction of a Multiple Complex System of Security of Cyberphysical Systems", *Ena.lp.edu.ua*, 2019. [Online]. Available: http://ena.lp.edu.ua/bitstream/ntb/31196/1/02-3-7%20%281%29.pdf.

25. "Ptolemy Project Home Page", *Ptolemy.berkeley.edu*, 2019. [Online]. Available: https://ptolemy.berkeley.edu/.

26. Edward A. Lee, System Design, Modeling, and Simulation using Ptolemy II [Online]. Available: https://ptolemy.berkeley.edu/books/Systems/chapters/IGettingStarting.pdf

27. "Why RTOS and What is RTOS?", *FreeRTOS*, 2019. [Online]. Available: https://www.freertos.org/about-RTOS.html.

28. "Altera Quartus II | IT Department", *Information-technology.web.cern.ch*, 2019. [Online]. Available: http://information-technology.web.cern.ch/ services/software/quartus-ii. "Documentation: Configuration Devices", *Intel.com*, 2019. [Online]. Available: https://www.intel.com/ content/www/us/en/programmable/ support/literature/lit-config.html.

29. "ISE Design Suite", *Xilinx.com*, 2019. [Online]. Available: https://www.xilinx.com/products/design-tools/ise-design-suite.html.

30. "Altium Designer 19 - Best PCB Design Software for Engineers", *Computer Aided PCB Design Software*, 2019. [Online]. Available: https://www.altium.com/altium-designer/.

31. "Altium Designer Documentation | Altium Designer 19.0 User Guide | Documentation", *Altium.com*, 2019. [Online]. Available: https://www.altium.com/documentation/ru/19.0/display/ADES/Altium+Designer+Documentation.

32. Internet Of Things Course - Immersive Programme Master in City and Technology [https://apps.uc.pt/search?q=Internet+of+Things]

33. MSc Programme in Communication Systems. [https://www.kth.se /en/studies/ master/communication-systems/ description - 1.25691]

34. MSc Programmes to Embedded Systems and Internet of Things [https://www.ncl.ac.uk/postgraduate/courses /degrees/embedded-systems-internet-of-things-msc/relateddegrees.html]

35. "IPN - Instituto Pedro Nunes", *Ipn.pt*, 2019. [Online]. Available: https://www.ipn.pt/.

14. POWER-OVER-ETHERNET BASED TRANSDUCER NETWORKS FOR CYBER PHYSICAL SYSTEMS

I. M. Lobachev (ONPU), Dr, Assoc. Prof. M.V. Lobachev (ONPU),
DrS, Prof. V.S. Kharchenko (KhAI)

Content

Abbreviations

HDF5	Hierarchical Data Format - version 5
HVAC	Heating, Ventilation and Air Conditioning system
IoT	Internet of Things
JSON	JavaScript Object Notation
MCB	Microcontroller Board
PoE	Power over Ethernet
SDN	Software Defined Networking
SSH	Secure Shell Script
TCL	Tool Command Language
UPOE	Universal Power Over Ethernet
CAFFE	Convolution Architecture for Feature Extraction

14.1 Internet of Things and scalability of cyber physical systems

Sensors have become indispensable elements in today’s society, being utilized in nearly every electronic device that is being used. They are obvious part of control systems. Location, design and functionality of sensors are very important points for cyber physical systems. Due to this situation there has been a great advancement in the field, and today there is an innumerable variety of sensors for every application imaginable. As technology moves forward, so do the trends and user demands.

Among these trends there is the branch of IoT-based smart buildings as well as the relatively new concept of “mega buildings” [1] that we are seeing more and more of, along with their need to gather data on the state of the building and its affairs. In the age of rapid technological advancement, many telecommunication applications are being integrated into our lives, including smart phones and Internet of Things (IoT). Smart buildings (and houses) use these technologies to reduce energy consumption and increase safety as well as add supporting features, where applicable. The need for these buildings is growing as urbanization continues and resources dwindle.

Another trend is “ecological buildings” or tall wooden buildings which have been an object of interest particularly in areas with a high supply of lumber. Furthermore as noted in a report by the United Nations, by 2050 over 65% of the world’s population will live in urban areas according to their projections [2]. This will cause the size and number of megacities to expand drastically in the near future. Complex communication networks, controls and other services will allow to build smart cities that are able to manage and improve the public’s quality of life. The cities of today will continue to evolve into smart cities, expressing a need for smart and green-oriented buildings, and infrastructure. The necessity of smart buildings and, eventually, cities, has stimulated the growth of sensor networks for these purposes. This in turn has required a system with a data-oriented approach, along with a sufficient degree of autonomy, as well as adaptability to different scenarios.

With the currently present approaches, such large scale systems risk being infrastructure dependent, and potentially very costly both in installation and maintenance. It is a challenge for complex cyber physical systems which must be able to evolvable and adapt based on the changes in requirements as well as environmental parameters. By devising an architecture, that is able to modularly integrate elements of technology available today, a system that maximizes the strength of each individual component can be constructed.

This approach allows to increase the degree of scalability and deployability, as well as minimize redundant closed-loop data transmission to improve resource efficiency by means of sharing the load via a distributed computing approach as well as by minimizing the need to have custom connections and interface.

14.2 Conception and advantages of Power over Ethernet

14.2.1 Conception of PoE method. State of the art

A network architecture concept that uses Power over Ethernet (PoE) as a method for transferring data and power over a single medium, conjoined with the principals of neural networks as well as decentralized and remote computing for data processing. The main objectives of this section are to describe a flexible and scalable architecture concept of a system that would be able to unify the benefits provided by IoT, along with power and installation efficiency of PoE as well as to investigate the feasibility of integration of neural network approaches and relevant hardware.

The concept used a Cisco Catalyst 4507R+E switch and utilized cloud and on-board computing to provide an easily scalable and adaptable architecture that can be modularly integrated into existing solutions. The prototype set-up was tested on RaspberryPi microcontroller boards as sensor hubs, and used DigitalOcean as the cloud computing service of choice. The Movidius Neural Compute chip was used to deploy the neural networks for the relevant data processing and deep learning hosts. The server in this implementation acts as user interface, front end, and as the console unit back end. The proposed architecture shows great promise on the feasibility of creating modular systems that unite the concepts of IoT, PoE and Neural Network approaches.

PoE has demonstrated the possibility to reduce installation costs for certain types of applications, such as LAN, IP and VoIP [3]. Furthermore, the necessary infrastructure for a PoE-enabled system is already in place for any building that has internet access thorough Ethernet ports, as the same Ethernet connections can be used for both power and data transmission. The system architecture needs to be flexible enough to adapt to the vast variety of sensor types and configurations, in order to allow various peripherals to be connected or disconnected to an existing, operating network. One of objectives includes investigating the applicability and usability of some off-the-shelf smart sensors, applicable to structural monitoring applications. In addition, the research aims to base the new concept on the lessons learned from previous iterations, in order to improve aspects such as scalability, adjustability, and ease of deployment.

Once the prototype of the architecture concept has been developed, it was tested to verify the feasibility of use of such a system, and its possible deployment schemes. In particular the testing attempted to answer whether the new design concept is able to adhere to the paradigm of high degrees of scalability and adjustability. The prototype system passed the testing stage, the details of which can be found in subsection 14.5.

As a result the developed concept can be used to later create a commercial system, a hybrid sensor network (as enabled by the designed system architecture), combining the abilities and properties of both wireless sensor networks as well as hard-wired sensor networks and their respective sensor modules.

This area has been previously investigated from a number of different directions, primarily from the point of view of the type of network in questions, for example wired vs. wireless. Furthermore a separate criterion is the deployment scale, a single smart room or house versus an entire network, for example smart energy grid.

A good example of a smart sensor that has been previously developed is the Imote2 smart sensor platform developed by Jennifer Rice et.al. [4]. Their work focused on a wireless sensor network setup, differentiated from other solutions by employing a decentralized computing scheme to reduce the amount of data communication within the network; as a result, the architecture has reduced the amount of power consumed by the system through limiting the power usage of individual elements.

However, while the system does offer benefits over traditional wireless setups (where the design would attempt to mimic a wired configuration using wireless transmission), it still has its flaws. While the group managed to reduce the overall power consumption, the main problem becomes the maintenance of the energy sources, which are in these cases batteries, regardless of whether they are primary or secondary cells.

A number of other recently developed solutions target more specific applications, such as wireless sensor networks for home lighting systems [5], industrial wireless sensor network data transmission scheme in [6] or the case study on Greenorbs in [7]. Which looked into the issues of scalability and system dependence on individual nodes, an issue addressed in this work. Yet another solution for a semi-wireless approach is for structural health monitoring, has been implemented by Xu et. al. with their system named Wisden [8].

The system was then later improved and tested again by Chintalapudi et. al. who published their results in [9]. However while the system uses wireless sensors, which send the data to a local hub, or node as titled by the authors, it is then connected to a local PC via serial port, a hard-wired connection. The computer would then handle the bulk of the data processing and can be connected as a network to other computers employing a similar set-up.

This design shows a number of drawbacks, such as the need for the wiring, which defeats the purpose of using a wireless set up as well as the delay due to the serial connection employed when compared to faster protocols such as TCP/IP which can be employed in the architecture presented in this work.

A number of attempts to optimize the wireless configuration using neural networks has also been approached by Kulkarni et. al. with their approach to utilize neural networks for a different clustering algorithm in [10] however, this approach would present its own challenges when attempting to increase the deployment scale.

14.2.2 Advantages of the PoE method

One major disadvantage of wireless sensor networks is that in big concrete buildings, the signal faces a lot of interference, from both other electronic equipment as well as due to signal bouncing off of the walls, ceilings and floors of the rooms. These effects were also demonstrated by Sato et. al. in [11] as well as by Wallace et. al. in [9]where measurements were taken to gauge the behavior of wireless systems in such environments.

This increases the difficulty of having a high concentration of sensors in a single given area. Furthermore, current systems such as Wisden, while wireless in sensor-to-gateway transmission, still require a wired connection for gateway-to-computer as well as separate wiring for power. By utilizing PoE. and cloud computing as well as taking advantage of the computational power available on modern microcontrollers such as Raspberry Pi, BeagleBone or the vast selection of TI MCUs, it is possible to create an easy to set up, efficient system, with minimal wiring and local processing of time-sensitive data for prompt response to events.

Lastly the use of Ethernet based data transfer protocols directly, for the sake of the prototype UDP was employed, the transfer rates are more optimal than a serial connection such as RS 232 for example [12]. The introduction of neural networks in turn, when properly configured, can help offload the data processing from the main processor, and methods similar to [13], as discussed by Rao et.al. can be applied for the collected non-linear sensor data.

14.3 System Power of Ethernet based architecture

14.3.1 General architecture view

The concept required the architecture to take the matters of saleability, configurability, deployability and power efficacy into account. These were achieved by employing a hierarchical structure to ease expansion and saleability. While modular sensor hubs and PoE were used for deployability and configurability aspects, with the addition of a software control layer. Further power efficiency was added by implementing varying sensor hub classes, with the corresponding varying operation schedules, and semi-distributed data pro-cessing with a heavy emphasis on cloud computing. The general overview of the architecture can be seen in Fig. 14.1. The inter-system communication and linkage, in addition to providing the possibility of hierarchical and remote control, allows for system to be optimized with regards to power consumption and processing cycles, by only powering on the necessary hubs on a per-case basis, the role of a hub is to act as a local sensor module interface, with some additional configuration and local signal processing tasks.

The use of PoE allowed for both the data, power and control commands to be sent via a single cable, which is already present in almost any building that has an internet connection infrastructure.

The setup of the prototype used to test the validity and feasibility of the concept proposed in this work has used a Cisco Catalyst 4507R+E switch [14] to handle the routing and supply the power to the sensor hub network [15].

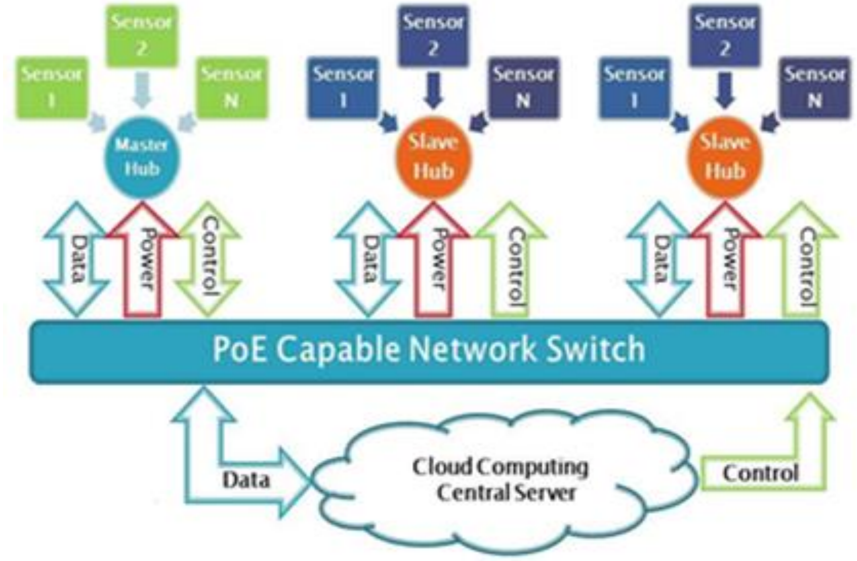


Fig. 14.1 - General architecture view

The Catalyst switches of these series are capable of delivering up to 60W per port using Universal Power Over Ethernet (UPOE), which essentially doubles the per-port power output specified by the PoE standard [16]. In this setup a hub is a Microcontroller Board (MCB) capable of receiving PoE, either by means of a shield or an integrated adapter, and is capable of housing the Linux kernel.

While the current prototype used an Ubuntu distribution, a lighter operating system, such as Puppy Linux or ArchLinux, can be used on MCBs with less resources, as well as scenarios where the size of the hub has higher priority over the amount of local processing that is necessary.

A good example of such a scenario would be the slave class hubs, which will be discussed later in this chapter. Due to their task being non-resource-demanding, the proposed scheme of optimization can be used to reduce costs. The booting sequence of the operating system on the MCBs used in the experiment was modified to reduce booting time, by preventing programs and services unnecessary for the functionality of the system rom loading.

A high level overview diagram of the system concept can be found in the Fig. 14.2 below. The figure outlines the overall set-up that was assembled, while later a lower level description will be provided to accompany the discussed details.

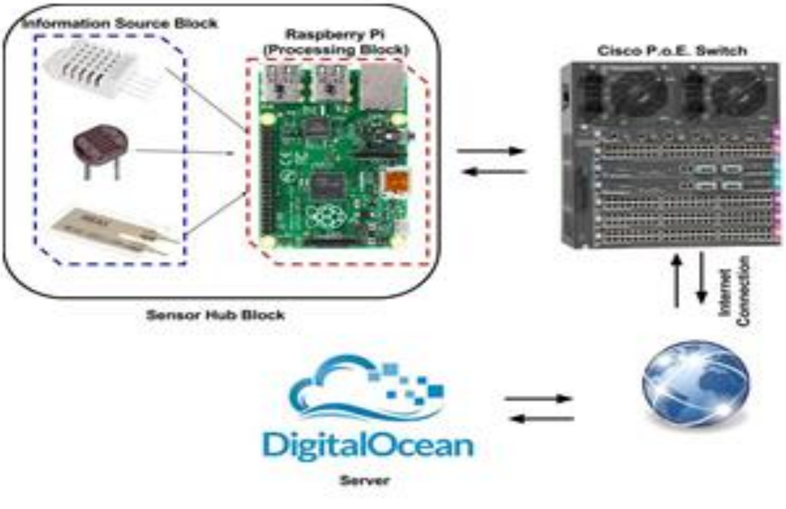


Fig. 14.2 - A high level diagram of the implemented system set-up

14.3.2 System requirements

There is a short list of requirements for the components that can be used to set up the system. In particular the Microcontroller Board (MCB) should have:

1). **Sufficient processing capabilities** - The board, which will serve as a sensor hub module needs to be able to run either Linux, or some operation system that is capable of connecting to the internet and executing python scripts (in the current iteration). It also needs to able to process and package the data into HDF5 format for sending.

2). **An interface to connect the sensors** - The board needs the capability to connect the various types of sensors that are necessary for a given task. The connection and corresponding peripherals can be either wired or wireless by nature, depending on the boards capabilities and other requirements and restraints of a given application.

3). **Sufficient storage** - This aspect needs to be considered for MCBs that will operate in semi-autonomous and autonomous modes as described in subsection 14.3.4. The storage requirement in this case will be dictated by the amount of information to be processed and stored, which in turn is determined by the application. In our set-up the dedicated storage for the data was 4Gb.

4). **PoE capable** - last major requirement is the ability to operate with PoE. this can be accomplished either by a built in circuit that the board has, or by a means of a shield, such as the Pi POE Switch hat for Raspberry pi, or a Cisco Catalyst Power Splitter for many of the other boards.

14.3.3 Sensor hub classes

There are two general classes of sensor hubs in this system set-up. A master class sensor hub and a slave class sensor hub. The naming convention was used due to the behavior of the system as described below.

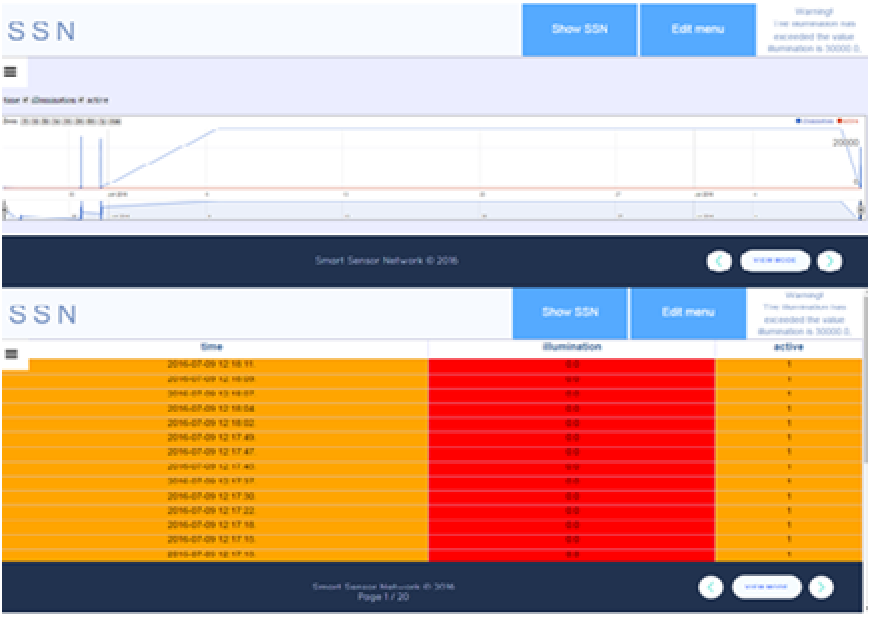
1. **Master** - is a sensor hub class which can operate in either fully autonomous or semi-autonomous mode of operation. In order to assign this class to a hub it needs sufficient processing power, the amount is dictated by the exact application and performance constraints. While in possession of this class the hub may issue commands to the slave class hubs assigned to it, such as to gather more data, and is also capable of bi-directional communication with the server.

2. **Slave** - is a sensor hub class which is capable only of one mode of operation. However, while it is unable to perform complex or resource demanding tasks locally, it provides the benefit of a lighter build. This in turn means that the hub may have much smaller physical dimensions and may be

more economical, which provides a more financially efficient solution on larger scales.

14.3.4 Configuration and operation modes of the system

As seen in Fig. 14.3 the edit menu provides the user with the ability to configure all of the sensor hub modules via the web interface, remotely.


SSN
Show SSN
Edit menu
Warning!
The illumination has exceeded the value illumination is 30000 0.
20000
Smart Sensor Network © 2016
SSN
Show SSN
Edit menu
Warning!
The illumination has exceeded the value illumination is 30000 0.
Smart Sensor Network © 2016
Page 1 / 20

time	illumination	active
2016-07-09 12:18:11	0.0	1
[illegible]	0.0	1
[illegible]	0.0	1
2016-07-09 12:18:04	0.0	1
2016-07-09 12:18:02	0.0	1
2016-07-09 12:17:49	0.0	1
2016-07-09 12:17:47	0.0	1
[illegible]	0.0	1
[illegible]	0.0	1
2016-07-09 12:17:30	0.0	1
2016-07-09 12:17:22	0.0	1
2016-07-09 12:17:18	0.0	1
2016-07-09 12:17:15	0.0	1
[illegible]	0.0	1

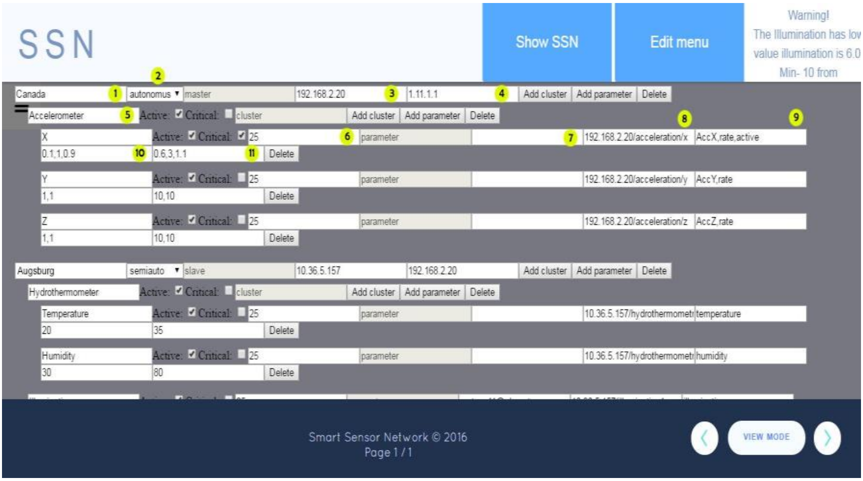
Fig. 14.3 - A screenshot of the varying views and representation of numerical data collected

The changes made there will be carried over into the master configuration file, which is stored on the server. The configuration file uses JavaScript Object Notation, JSON, format to store all the settings and relay them to the respective modules.

Once any changes have been made the server will issue an update notification to the MCBs that have been affected, once the MCBs receive the notification and have prepared to receive information the new configuration file for them will be created, sent and applied, upon which the changes will have taken effect.

The individual configuration file is created by taking a relevant snippet of the JSON master list file that contains the information of all the MCBs, including the ones that were affected. That snippet is then used to create the file that will be sent. If desired, all of the settings can be altered by directly accessing the back end as well, assuming sufficient access privileges are present.

While a MCB acts a sensor hub, it can house a number of logical clusters of sensors, for example a sensor that provides multiple types of data, such as an accelerometer or a humidity and temperature sensor such as DHT22 or even a visual data stream. The hierarchy of the system takes into account what is considered a “parameter”, as is seen in Fig. 14.4.



Warning!
The Illumination has lov
value illumination is 6.0
Min- 10 from

Canada | autonomus ▾ | master | 192.168.2.20 | 1.11.1.1
Accelerometer | Active: ☑ Critical: □ | cluster
X | Active: ☑ Critical: ☑ | 25 | parameter | 192.168.2.20/acceleration/x | AccX,rate,active
0.1,1,0.9 | 0.6,3,1.1
Y | Active: ☑ Critical: □ | 25 | parameter | 192.168.2.20/acceleration/y | AccY,rate
1,1 | 10,10
Z | Active: ☑ Critical: □ | 25 | parameter | 192.168.2.20/acceleration/z | AccZ,rate
1,1 | 10,10
Augsburg | semiauto ▾ | slave | 10.36.5.157 | 192.168.2.20
Hydrothermometer | Active: ☑ Critical: □ | cluster
Temperature | Active: ☑ Critical: □ | 25 | parameter | 10.36.5.157/hydrothermometr | temperature
20 | 35
Humidity | Active: ☑ Critical: □ | 25 | parameter | 10.36.5.157/hydrothermometr | humidity
30 | 80

Smart Sensor Network © 2016
Page 1 / 1

Fig. 14.4 - A screenshot of the web view of the user interface and configuration window

The numbers highlighted in the Fig. 14.4 are used to reference the fields they are located at, where: 1 is the name of the Master or Slave module, 2 is the mode of operation, 3 is the IP address of the module, 4 is the IP address of the corresponding switch or master module, 5 is the name of the cluster, 6 is the pin or port that the cluster is connected to on the board module, 7 is the e-mail address of the person to contact in case of anomalies, 8 is the path within the HDF5 file, 9 are the headings in the HDF5 file that will be monitored, 10 are the minimum thresholds for the specified parameters, 11 are the maximum thresholds for the specified parameter.

The MCB, which, as long as it meets the requirements outlined in Section 0, is non-discriminant of the type and model used, communicates directly with the server. The server is located on a cloud computing service host, in the case of this implementation DigitalOcean was used, however depending on the application the server can be both outsourced to a third party or hosted locally depending on the available resources and performance needs. In addition to the previously implemented functionality of a master/slave status of the sensor hubs, which focused more on power consumption control, the system now also has three modes of operation, designed to allow it to adapt to any scenario. These modes are:

1. **Autonomous** (Fig. 14.5) – The sensor hub functions individually, conducting all of data processing locally, and performing actuation and/or executing scenario scripts as is dictated by its local code. The hub will periodically send its stored information to the server.

When operating in this mode the hub will also periodically probe the server to determine whether it's operational and whether internet is present. If either one of them does not respond the error will be logged and brought to the attention of the person in charge once the connection is restored.

2. **Semi-autonomous** (Fig. 14.6) – the most common mode of operation, typically used by the master class sensor hubs.

When operating in this mode the hub will only process time-critical information locally, and act based on the results. Data that is categorized as non-time-critical will simply be periodically forwarded to the server by the hub.

3. **Follower** – The most basic mode in which the hub will simply execute the commands it was given, while forwarding all of the collected data to the server for processing.

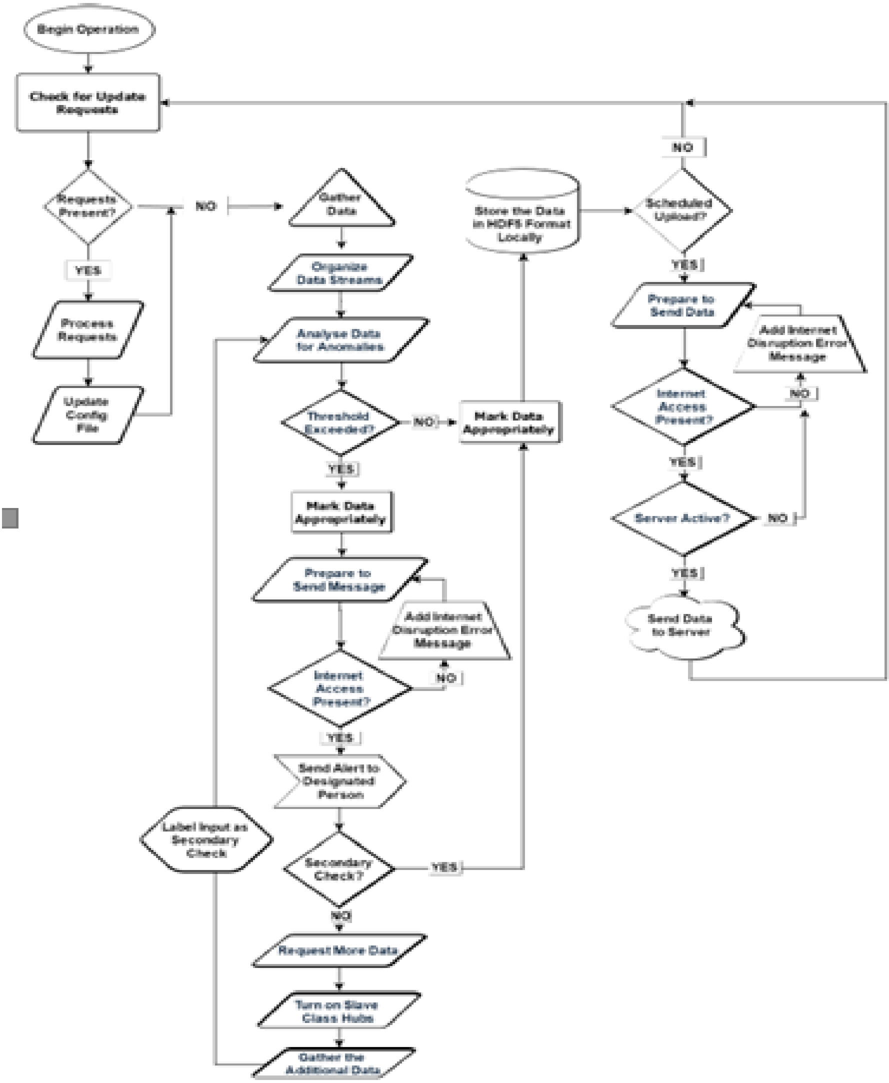


Fig. 14.5 - A diagram displaying the high level algorithm of autonomous modes of operation

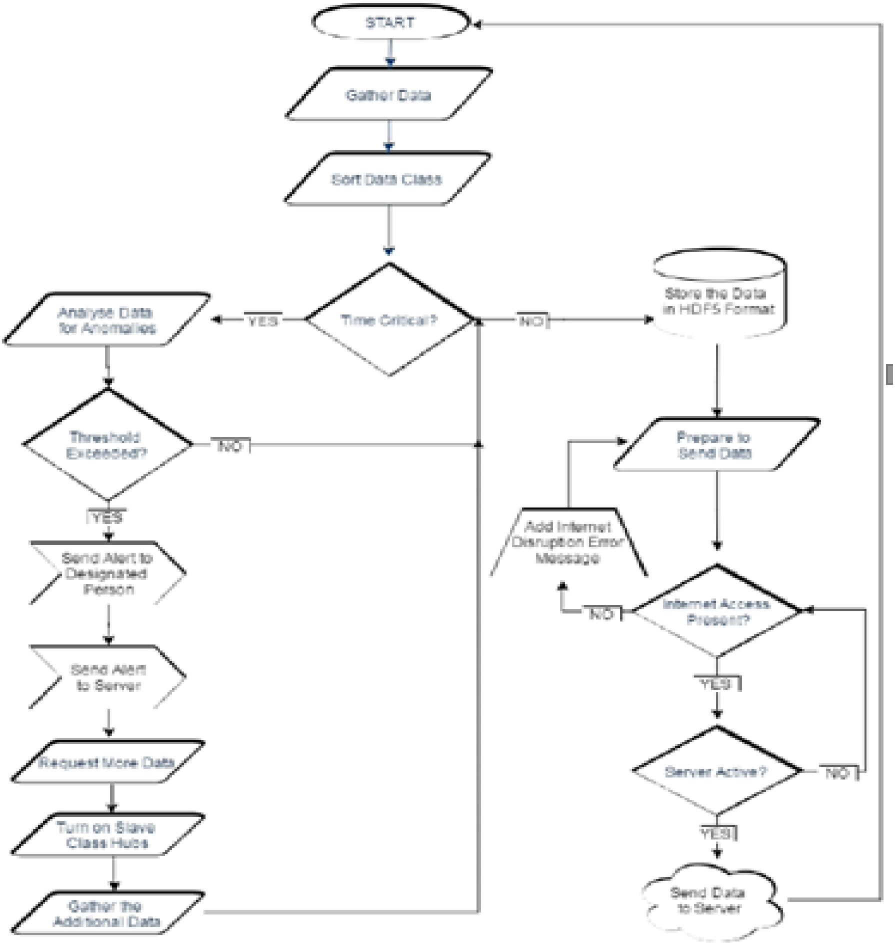


Fig. 14.6 - A diagram displaying the high level algorithm of semi-autonomous modes of operation

14.3.5 Parameters, organization and data processing

Each parameter has maximum and minimum threshold values, if the current value of the reading exceeds either of the defined boundaries an alarm is triggered. The numerical data is stored and can be inspected via two modes, as seen in Fig. 14.5, Fig. 14.6. The analysis algorithm marks the concerning values on the web page for manual inspection if needed and sends a notification to the corresponding e-mail, in addition the error is kept in a log of alarms which is present on the main header tab, as can be seen in Fig. 14.3, Fig. 14.4.

The alarm will also trigger an additional data request. This request will wake up all of the slave hubs that are under the master hub which recorded the anomaly. Once awake the slave hubs will collect an additional set of data to either confirm or deny the presence of the anomaly that was picked up from the readings provided by the master class hub.

By unifying the main control under one server, and allowing it to communicate with any of the switches or hubs from a single location, the scalability becomes a trivial matter. To facilitate direct access, the switch must first be configured to allow Secure Shell (SSH) [14,15]. Once that is done the switch can be accessed remotely by any machine that has access to the internet and the terminal, through the server virtual machine.

To further automate the process, a macro can be saved for each switch if necessary by writing a simple script which will pull the necessary credentials from a configuration file containing the information on the switches, hubs and sensors [17,18]. Due to the need for security, as the switches may potentially have a lot of control within a building, measures need to be taken to secure the information.

However due to the focus of this work being the improvement of scalability, adaptability and responsiveness this aspect will be further discussed in Section 14.5 as future work. One of the primary uses for automatic SSH scripts is to issue commands to the switch remotely and autonomously, such as in the case where additional slave hubs need to be activated. For this purpose Tool Command Language, TCL, is used on the switch side. It is invoked by the server once an SSH connection is established and can be configured to perform a variety of tasks in addition to controlling the power supply.

A CRON job is used to perform a series of regular, periodic checks on the non-time-critical data, the server also constantly listens to any alerts from the hubs. The script can be activated in two cases, either an alert about an anomaly is received directly from the master hub, which processed the critical data locally, or derived from processing the information on the server. Each hub, if requesting additional data, will transmit its ID along with the request, which will be used to look up the required credentials.

Once found, a script will be run to issue the command to the corresponding switch and slave hubs. When the data is processed on the server the IP address of the hub that detected the anomaly will be used to find the required credentials.

14.3.6 Data processing and presentation

As mentioned previously, data is collected and stored in HDF5 format. The use of the format allows to assign meta data to the gathered measurements, which simplifies the process of parsing the data for further processing, as well as cross platform import, in cases where different parties want to use different software to analyse the data locally.

In addition once the data has been arrange correctly, the new transition times of file transfer become shorter, as the files are a lot more compact and light weight. In order to achieve this, the raw data is first gathered from the individual sensors. A preassigned name is used to label the source of information, the name is obtained from the JSON configuration file. In addition the type of data that is being gathered, as well as its safety constraints are also used as the meta data for the measurement.

Once formatted correctly, the newly arranged data is stored in its own sub directory of the local database, as seen later (Fig.14.7) along with the other measurements. On the server side the data is the extracted, parsed into arrays and displayed to the user.

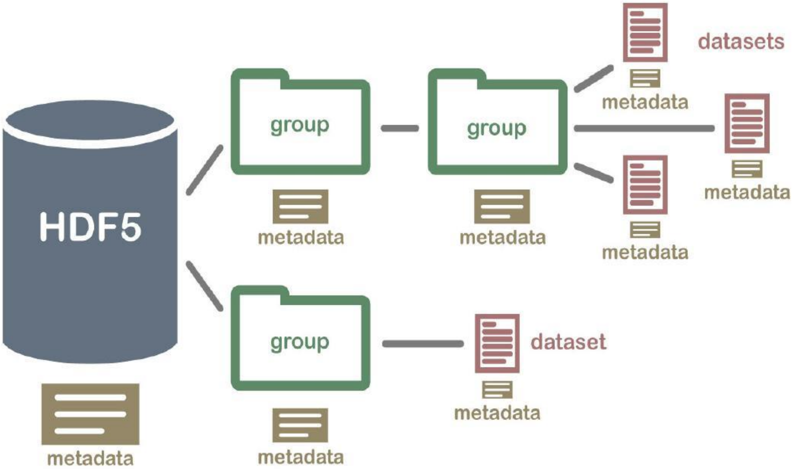


Fig. 14.7 - A diagram displaying the structure of the HDF5 data format [19]

14.3.7 The systems hierarchy

Hierarchy is a very important aspect in a project that emphasizes scalability such as the one discussed in this work. As such a few approaches are used to introduce the concept of hierarchy into the system. The database format that is used to send, store and process the data is Hierarchical Data Format - version 5

(HDF5), which is great for fast transfer of data due to its ability to keep the file sizes low and well organized. The latter point also makes it great for processing, as accessing any piece of information from the database or parsing it becomes simple when using HDF5.

Hierarchy is also employed in other aspects of the design, as can be seen in Fig. 14.8, the hierarchy determines the amount of control a unit has. For example, a server has full authority, while a single salve hub has none. This "chain of command" makes propagation of commands rather simple and clean.

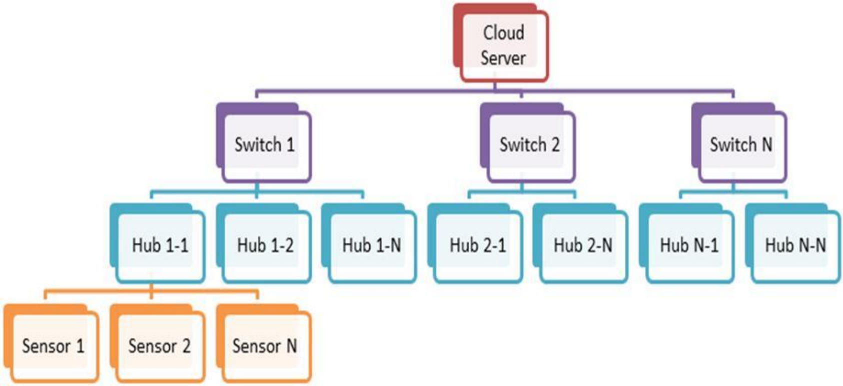


Fig.14.8 - A high-level diagram outlining the hierarchy of the system

14.4 Incorporation of neural networks

To demonstrate the modularity of the system, and to further enhance its applicability and local data processing performance the possibility of introducing the concept of neural networks was investigated. The Movidius Neural Compute Chip by Intel [20] was used as the mobile host for the neural network due to its low power consumption, USB interface and mobility. As the main goal is to have a mobile, distributed ad-hoc network, the hubs need to be modular, and consume the least amount of power possible. That was the reason for choosing a low power processor MCB, however typical hosts for neural networks in addition to having rather strict hardware and interface protocol requirements, also have high levels of power consumption, such as the NVDIA GPUs cards which can consume in the 100s of watts per unit.

Furthermore the available interface methods would allow at most one unit per hub, with an elaborate adaptation hardware, which does not fit the needs of the system. Instead the Movidius NCS consumes approximately 1W of power, and we are able to deploy multiple units per hub, which in turns allows us to have a flexible system where multiple neural networks, for different data types, can be deployed nearly simultaneously. Furthermore, by being able to offload the actual processing of the data to the neural network hosts, the processors primary jobs becomes managing and parsing the data flows, as well as interaction with the adjacent hubs in the network.

The on-going development of this aspect has so far allowed to local data processing such as image processing and pattern recognition which can be employed for resource optimization within a building, in the case of green building applications. While the efficiency of employment of neural networks and deep learning are still being investigated, currently the benefits of sharing the processing load with the neural chip when processing numerical or graphical data has already been seen, due to the repetitive nature of the received data.

For the setup the binary result data employed the 14x40x2 architecture (14 inputs, 40 hidden neurons and 2 outputs), where the learning process is carried out using the back propagation algorithm with the employment of the ReLU function within the hidden neurons. In order to facilitate non-linearity output layer, contrary to the hidden layer, uses Sigmoid, which allows to obtain the probability of class assignment. The training of the neural network used the Adam Optimization method, as it showed the beset results based on trial runs. In addition the binary signal analysis used the mean squared error approach for the weighting function, as seen below:

```
# Define loss and optimizer
cost = tf.losses.mean_squared_error(y, pred)
optimizer =
tf.train.AdamOptimizer(learning_rate).minimize(cost)
```

The two currently investigated frameworks that have been applied are Tensorflow and CAFFE, where in the former the DNNClassifier was employed. The CAFFE framework is being tailored more towards processing of the graphical input data, since the framework also includes a number of established algorithms which can be used as basis, such as AlexNet for example.

The diversified use model is supported by the mechanical properties of Movidius sticks, which can have more than one unit attached to a single hub, while having different trained matrix employed.

14.5 Testing of the network

For the network testing, the experimental setup that can be seen in Fig. 14.9 was used. The units of the Y-axis in Fig. 14.10 are in milliseconds, and display the latency statistics as will be discussed below. A number of components in this system were tested. The connection speeds between the machine and the server.

Due to the fact that the main limitations of the connection speed are the Internet Service Provider imposed limitations and the number of hops the transmission has to go through, which usually correlates with the geographic location of the communicating parties. The average time for the European setup was 42ms while the average Canadian time was 190ms.

Due to the server being located in Germany these results are to be expected. For more optimal times either a service that is geographically close or personal servers can be used if necessary.

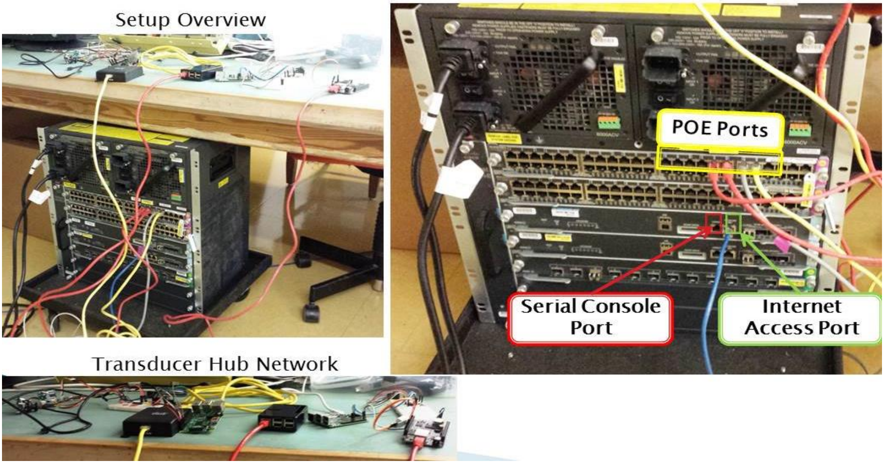
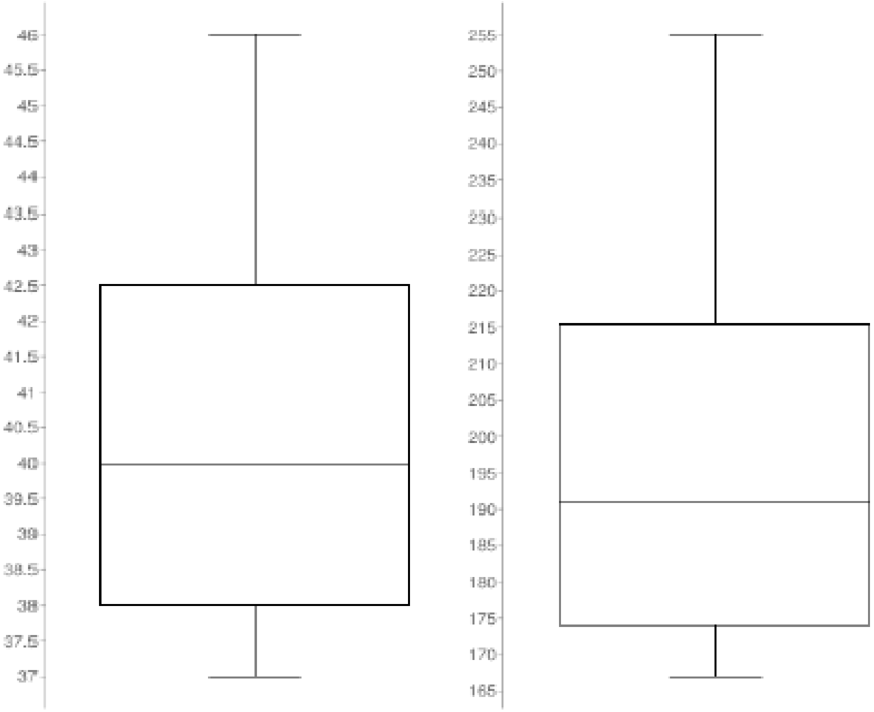


Fig. 14.9 - Experimental Setup for Local Testing of the Prototype System



(a) Germany

(b) Canada

(b) Canada

The system was also tested for scalability by connecting two setups from two continents. The first setup was done in Vancouver, Canada while the second one was done in via access in Germany. Each system

Fig. 14.10 - Latency test results for the two cities where the tests were conducted

consisted of at least o

ne master and at least one slave module. Each of the modules had a number of sensors connected to them. While Some of the sensors varied, an illumination sensor, a vibration sensor, temperature and humidity sensor and power sensor were used in different configurations on all of the MCBs.

After setup, to test operational stability and performance a latency test was performed. The test was carried out over the course of 15 days, at approximately that same time according to local time, the peak demand period of afternoon was used as studies have shown that around 14:00 is when usage is the heaviest according to [21].

During the measurement three samples were taken and recorded. The resulting latency statistics can be found in Figure 14.9 where sub-figure (a) displays the latency measurement results obtained in Germany and sub-figure (b) the ones obtained in Canada. Over the course of testing there has been minimal package loss, even on long transitions in case of Vancouver measurement (11%). During the testing there were no errors detected on the server side with neither the data nor the request handling. The system had no issues handling the requests from both setups and operating with both sets of MCBs, even if their requests arrived nearly simultaneously.

As a test of adaptability, a different microcontroller board was used BeagleBone. After a simple configuration of the output pins, no other changes were necessary. The sending, receiving and processing schemes as well as the control modes all performed as expected without any errors, and did not require any excessive remodeling or re-coding.

A full scale test of the system is planned to be conducted as part of the Green Campus initiative at UBC, Canada. There the system is planned to be used to monitor the energy consumption and generation around a section of the new building that has been built,as well as its structural integrity of one of the sections and the corresponding interior climate.

For this purpose vibration sensors, LDT0-028K was chosen as the sensor of choice as it has shown to perform well for the purposes of earthquake detection and building vibration [22]as well as parasitic power generation as discussed by P. Glynne-Jones in their work outline in [23].

Large area of effect strain sensors, such as a long-gage fiber optic sensor, which have been shown to perform well as civil structure monitoring device as explained in their overview paper by H.N. Li et. al.[24] as well as in the paper by S. Li et. al. who discussed the feasibility of a distributed system of such sensors in [25]. As well as temperature and humidity sensors and power measuring sensors will be used. The alpha prototype has tested the correctness of operation of the vibration sensor, temperature and humidity sensor and the power measurement sensor by assembling a circuit for each and calibrating their outputs. Once the obtained readings matched the expected values, which were obtained by performing a control measurement by industry manufactured devices, the operation level was deemed feasible.

14.6 General integration flow

In order to integrate this architecture into a specific implementation the following steps have to be taken.

Access what initial level of depth and consequently hierarchy is needed. This is in part dictated by the scale of initial deployment, and the number of nodes present.

Then it is necessary to identify what types of data will be processed and select the appropriate sensors. During this step it is also necessary to either compile a list of recently trained neural networks, or train them based on the necessary data sample.

The last step in the planning stage is establishing what kind of network architecture will be most beneficial for the specific application. For example a local server with a closed sandbox network versus a cloud-based architecture that will use an internet connection to communicate. Once these two steps are complete the hardware deployment can begin.

After that it is necessary to establish the communication protocol and the back end, along with the deployment of control algorithms relevant to the hierarchy depth selected earlier.

The testing would entail ensuring a smooth data flow, checking even response times, as well as node-to-node communication. Failure simulation should also be conducted to ensure the system can continue functioning even in cases of spot-based or even cluster-based failures.

14.7 Work related analysis

Detailed analysis of publications related to application and development PoE based systems has been done in subsection 14.2.1. Additionally it’s need to underline that a lot of EU universities including ALIOT project partners and universities of USA conduct research and implement education MSc and PhD programs in the Internet of Things, PoE networking in context of cyber physical systems for different domains. Development of the described methods and solutions are based on analysis of these programs and providing some of the educational topics and research directions.

In particular, the following courses and programs have been considered:

- Coimbra University, Portugal: IoT course for MSc [26]. The courses represents a new stage in the digital evolution and focuses on the Internet of Things for smart transport and cities, and the development of tools to transform city infrastructure;

- KTH University, Sweden: three MSc programs including:

a) IoT related topics in Information and Network Engineering [27],

b) Communication Systems [28],

c) Embedded Systems [29];

- Newcastle University, United Kingdom: MSc Programme on Embedded Systems and Internet of Things (ES-IoT) MSc [30].

Conclusion and questions

As a result, a system that matches the set goals has been developed. The system is capable of operating by utilizing Power over Ethernet, which drastically simplifies installation procedures as well as reduces power consumption costs. The removal of power bank configuration, used by similar alternatives, allows for much longer maintenance periods of 1-2 years on average for electrical inspections compared to approximately 6 months to replace power sources in aforementioned solutions.

The dependability of the verdicts issued by the hubs is reinforced by validating any spotted anomaly either via the subordinate slave hubs, which will collect the same data and a statistical result can be obtained to avoid false positives, or via other master hubs which can perform the same action in their area of effect. Also as discussed previously the system can perform "self-checks" for connectivity to both the higher members of the hierarchy as well as its subordinates.

This allows it to spot any units or modules that may have gone "offline" and therefore have them serviced in a much more timely manner rather than by a more manual inspection method. The introduction of neural network modules allows to perform a lot more of the data processing locally, which will greatly benefit any data flows that are time critical, for example the ones that would issue an evacuation warning or signal.

This in turns allows to reduce the amount of required bandwidth, and as a result may also reduce the dependence on an external internet connection which may be an issue in certain geographical locations. However, the general inter-communication algorithm presented in this work allows to integrate a number of application-specific failsafe algorithms to improve the reliability of the system's operation in a given environment.

Furthermore by placing the central client of the system in the cloud, an online processing service, a high degree of scalability has been achieved, as well as a high degree of controllability, where the control addressee can be targeted from a single hub to a whole branch enveloping a building or a floor. This in turn is a very important aspect for the IoT field.

Furthermore the data transfer speeds as well as throughput have been increased by integrating an Ethernet based transfer protocol, which by nature provides higher bandwidth and can be driven at higher data rates when compared to the earlier models utilizing RS-232. Nevertheless, nowadays many other models also utilize TCP/IP based communication from their systems, where the distinguishing factor becomes the organization and positioning of the nodes. This matter feeds back into and is resolved by the hierarchical structure discussed in this work.

While the system could be further improved by increasing the security of the transmitted data, as well reducing the size of the MCBs used. It can nevertheless benefit smart buildings, large structures that wish to reduce the amount of power consumed by the Heating Ventilation and Air Conditioning, HVAC, systems, as well as buildings that apply new building technology and materials and need to analyze their performance.

As can be expected the system functions well with the big data approach and has been tested to be operational in a tans-continental set up, where the gathered data was accessible from any machine that had access to the internet, and the processing and transfer of the data was completed in a timely and efficient manner, due to high processing capabilities of cloud computing machines and small size of the files that used HDF5 database format.

The system has improved its aspects of usability, adaptability and scalability, however for commercial use there are still a number of improvements that could be made that would benefit the performance.

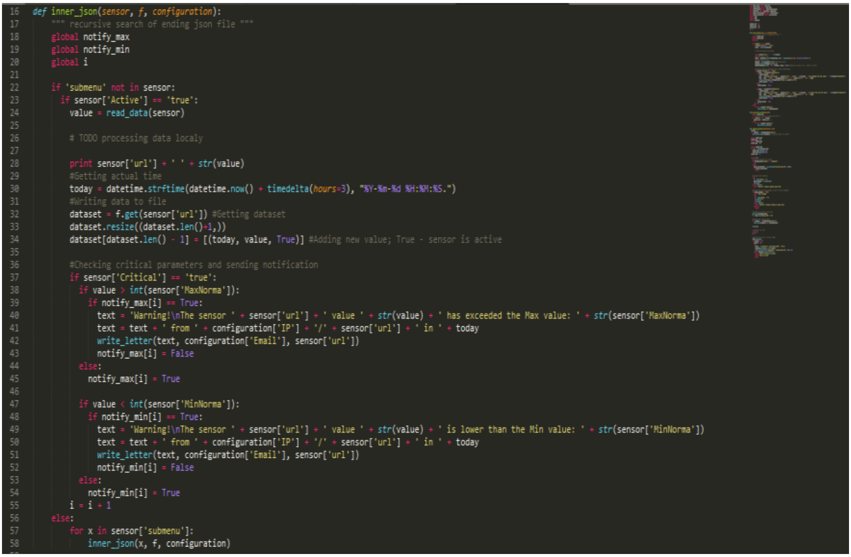
The major aspect to be considered is the enhancement of security and, in case of in-house servers, more elaborate data routing and handling. While a certain level of security it is obtained by securing the server the system can benefit from employing Software Defined Networking, SDN, to improve internal routing, which would in turn improve and optimize data handling as well as packet control to ensure no tampering or unauthorized access occurs. In their work Kapil B. [31] outline how the employment of SDNs can benefit a network, and quantify the performance benefits of such a transition from the traditional hardware defined networks.

Another improvement that can be made to the system is further segmentation and automation of access to different levels of the system. This includes adding the capability to select an "area of effect" which can be defined based on the application, but in case of buildings it would define what portion of the building is affected by a certain action. These functions, as well as more diverse data processing and automation can be further expanded by adding more layers to the learning algorithms and developing the classes of the classes and matrixes of the neural networks.

In order to better understand and assimilate the educational material that is presented in this section, we invite you to answer the following questions.

1. Describe what neural network models can be used to process sequential numeric data.
2. Describe what neural network models can be used to process graphical information.
3. What protocols can be used to transfer data over an intra-net?

4. How can further optimization be implemented leveraging the resource efficiency of HDF5 and Neural Netowrks?
5. What other areas could benefit from integration of such architecture?
6. Why is having a scalable system important for fast-growing cities?
7. What protocol optimizations can be implemented to further increase the throughput of the data?
8. What other approaches for Neural Network distribution can be employed to increase resource efficiency?
9. What kind of safeguards can be put in place to counteract malicious data input into the training sequences of the Neural Networks during their fine-tuning process? (EX: placing specially designed graphics to render image recognition networks useless)
10. What benefits would a localized system have over a centralized one in this type of architecture?
11. Please comment the following example code snippet (Fig.14.11) from a semi-autonomous algorithm flow described in subsection 14.3.4 (Fig.14.6)



```
def inner_json(sensor, f, configuration):
    """ recursive search of ending json file """
    global notify_max
    global notify_min
    global i

    if 'submenu' not in sensor:
      if sensor['Active'] == 'true':
        value = read_data(sensor)

        # TODO processing data localy

        print sensor['url'] + ' ' + str(value)
        #Getting actual time
        today = datetime.strftime(datetime.now() + timedelta(hours=3), "%Y-%m-%d %H:%M:%S.")
        #Writing data to file
        dataset = f.get(sensor['url']) #Getting dataset
        dataset.resize((dataset.len()+1,))
        dataset[dataset.len() - 1] = [(today, value, True)] #Adding new value; True - sensor is active

        #Checking critical parameters and sending notification
        if sensor['Critical'] == 'true':
          if value > int(sensor['MaxNorma']):
            if notify_max[i] == True:
              text = 'Warning!\nThe sensor ' + sensor['url'] + ' value ' + str(value) + ' has exceeded the Max value: ' + str(sensor['MaxNorma'])
              text = text + ' from ' + configuration['IP'] + '/' + sensor['url'] + ' in ' + today
              write_letter(text, configuration['Email'], sensor['url'])
              notify_max[i] = False
          else:
            notify_max[i] = True

          if value < int(sensor['MinNorma']):
            if notify_min[i] == True:
              text = 'Warning!\nThe sensor ' + sensor['url'] + ' value ' + str(value) + ' is lower than the Min value: ' + str(sensor['MinNorma'])
              text = text + ' from ' + configuration['IP'] + '/' + sensor['url'] + ' in ' + today
              write_letter(text, configuration['Email'], sensor['url'])
              notify_min[i] = False
          else:
            notify_min[i] = True
        i = i + 1
    else:
        for x in sensor['submenu']:
            inner_json(x, f, configuration)
```

Fig.14.11 Example code snippet from a semi-autonomous algorithm flow

References

[1] R. Goldstein and D. Neuman, “Mega-buildings: Benefits and opportunities of renewal and reused the essential role of existing buildings in a carbon neutral world,” in Proceedings of the American Institute of Architects National Convention and Design Exposition held in Miami, Florida, USA, 10-12 June, 2010.

[2] United Nations, Department of Economic and Social Affairs, Population Division, “World urbanization prospects: The 2014 revision, highlights (st/esa/ser.a/352),” 2014. [Online]. Available: https://¬esa.un.org/-unpd/¬wup/¬Publications/¬Files/¬WUP2014-Highlights.pdf

[3] COMMSCOPE INC, Laying the groundwork for a new level of Power over Ethernet, 2015. [Online]. Available: http://¬www.commscope.com/¬Docs/¬POE_Groundwork_WP-107291.pdf

[4] J. A. Rice, K. Mechitov, S.-H. Sim, T. Nagayama, S. Jang, R. Kim, B. F. Spencer Jr, G. Agha, and Y. Fujino, “Flexible smart sensor framework for autonomous structural health monitoring,” Smart structures and Systems, vol. 6, no. 5-6, pp. 423–438, 2010.

[5] M. Magno, T. Polonelli, L. Benini, and E. Popovici, “A low cost, highly scalable wireless sensor network solution to achieve smart led light control for green buildings,” IEEE Sensors Journal, vol. 15, no. 5, pp. 2963–2973, 2015.

[6] O. Kreibich, J. Neuzil, and R. Smid, “Quality-based multiple-sensor fusion in an industrial wireless sensor network for mcm,” IEEE Transactions on Industrial Electronics, vol. 61, no. 9, pp. 4903–4911, 2014.

[7] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, “Does wireless sensor network scale? a measurement study on greenorbs,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 10, pp. 1983–1993, 2013.

[8] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, “A wireless sensor network for structural monitoring,” in Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 2004, pp. 13–24.

[9] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, “Monitoring civil structures with a wireless sensor network,” Internet Computing, IEEE, vol. 10, no. 2, pp. 26–34, 2006.

[10] U. M. Kulkarni, D. V. Kulkarni, and H. H. Kenchannavar, “Neural network based energy conservation for wireless sensor network,” in 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon). IEEE, aug 2017.

[11] H. Sato, H. Domae, M. Takahashi, and M. Abe, “Reflection and transmission control of electromagnetic wave for concrete walls,” Electronics and Communications in Japan (Part II: Electronics), vol. 83, no. 11, pp. 12–21, 2000. [Online]. Available: http://¬dx.doi.org/¬10.1002/¬1520-6432(200011)83:11<12::AID-ECJB2>3.0.CO;2-E

[12] W. Townsend, “The barretthand grasper-programmably flexible part handling and assembly,” Industrial Robot: an international journal, vol. 27, no. 3, pp. 181–188, 2000.

[13] H.-X. Rao, Y. Xu, B. Zhang, and D. Yao, “Robust estimator design for periodic neural networks with polytopic uncertain weight matrices and randomly occurred sensor nonlinearities,” IET Control Theory & Applications, vol. 12, no. 9, pp. 1299–1305, jun 2018.

[14] D. Barnes and B. Sakandar, Cisco LAN switching fundamentals. Cisco Press, 2004.

[15] Cisco Catalyst 4500E Supervisor Engine 8-E Configuration Guide (Wireless), Cisco IOS XE Release 3.7E, 2nd ed. Cisco INC, 2014. [Online]. Available: http://¬www.cisco.com/c/en/us/td/docs/switches/lan/-catalyst4500/XE3-7-0E/wireless/configuration-guide/¬b_37e_4500sup8e_cg.html

[16] Cisco Systems INC, Cisco Catalyst UPOE Power Splitter, 2015. [Online]. Available: https://-developer.cisco.com/fileMedia/download/99c67d92-8089-44b9-980a-9bc304abf739

[17] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, SSH, the secure shell. O’Reilly, 2005.

[18] 2016. [Online]. Available: http://www.openssh.com/manual.html

[19] F. G. Osorio, M. Xinran, Y. Liu, P. Lusina, and E. Cretu, “Sensor network using power-over-ethernet,” in Computing and Communication (IEMCON), 2015 International Conference and Workshop on. IEEE, 2015, pp. 1–7.

[20] Intel Corporation, Movidius Neural Compute Stick User Guide, 2017. [Online]. Available: https://-movidius.github.io/¬ncsdk/¬

[21] N. Brownlee and K. C. Claffy, “Understanding internet traffic streams: dragonflies and tortoises,” IEEE Communications magazine, vol. 40, no. 10, pp. 110–117, 2002.

[22] F. R. d. C. Alves, “Low-cost vibration sensors: tendencies and applications in condition monitoring of machines and structures,” Ph.D. dissertation, INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA, 2015.

[23] P. Glynne-Jones, M. Tudor, S. Beeby, and N. White, “An electromagnetic, vibration-powered generator for intelligent sensor systems,” Sensors and Actuators A: Physical, vol. 110, no. 1â€“3, pp. 344 – 349, 2004, selected Papers from Eurosensors {XVI} Prague, Czech Republic. [Online]. Available: http://-www.sciencedirect.com/¬science/¬article/¬pii/¬S0924424703005995

[24] H.-N. Li, D.-S. Li, and G.-B. Song, “Recent applications of fiber optic sensors to health monitoring in civil engineering,” Engineering structures, vol. 26, no. 11, pp. 1647–1657, 2004.

[25] S. Li and Z. Wu, “Development of distributed long-gage fiber optic sensing system for structural health monitoring,” Structural Health Monitoring, vol. 6, no. 2, pp. 133–143, 2007.

[26] Internet Of Things Course - Immersive Programme Master in City and Technology [https://apps.uc.pt/search?q=Internet+of+Things]

[27] Master's programme in Information and Network Engineering [https://www.kth.se/en/studies/master/information-and-network-engineering/master-s-programme-in-information-and-network-engineering-1.673817]

[28] Master's programme in Communication Systems [https://www.kth.se /en/studies/ master/communication- systems/ description -1.25691]

[29] Master's programme in Embedded Systems [https://www.kth.se /en/ studies/master/embedded-systems /description-1.70455/

[30] Related Programmes to Embedded Systems and Internet of Things (ES-IoT) MSc [https://www.ncl.ac.uk/postgraduate/courses /degrees/embedded-systems-internet-of-things-msc/relateddegrees.html]

[31] K. Bakshi, “Considerations for software defined networking (sdn): approaches and use cases,” in Aerospace Conference, 2013 IEEE, 2013, P. 1–9.

Part IV. IOT TECHNOLOGIES FOR CYBER PHYSICAL SYSTEMS

15 MODEL-BASED SYSTEMS ENGINEERING FOR THE CYBER-PHYSICAL SYSTEMS

Dr. R. K. Kudermetov (ZNTU)

Contents

Abbreviations

CPS – Cyber-Physical System

CSL – Collaborative Sensing Language

DSL – Domain Specific Language

GQAM – Generic Quantitative Analysis Modeling

GRM – Generic Resource Modeling

MARTE – Modeling and Analysis of Real-Time and Embedded Systems

MBD – Model-Based Design

MBSE – Model-Based Systems Engineering

NFP – Non-Functional Properties

OMG – Object Management Group

PAM – Performance Analysis Modeling

RTE – Real-Time Embedded Systems

SAM – Schedulability Analysis Modeling

SE – Systems Engineering

SysML – Systems Modeling Language

UML – Unified Modeling Language

VSL – Value Specification Language

15.1 Modeling methodologies for CPS

You can make a huge list of domains in which it is possible and even necessary to use CPS, let’s name just a few of them – any kind of industry, medicine, scientific research; all areas of the Internet of Things – smart cities, smart homes, smart transportation, smart energy; military applications and space systems; many intersections of named domains. The variety of domains makes it necessary to take into account various technologies, features, standards and traditions when designing control and automation systems, and, consequently, this diversity affects the nature of CPS [1]. Therefore, there is a need to introduce into traditional design methods in separate domains design the methods and technologies from other domains and to further adopt and implement Systems Engineering (SE) approaches such as system analysis and verification at all stages of the life cycle of the designed systems, allowing to designing successful systems regardless of domain areas.

There are many definitions of the term cyber-physical systems. Since this section will address the development aspect of CPS, we will adhere the concept in which the authors of [2] nest the following sense: "A cyber-physical system (CPS) is an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. As an intellectual challenge, CPS is about the intersection, not the union, of the physical and the cyber".

In this section the methods of the one of the modern approach to SE – the Model-Based Systems Engineering (MBSE) and its applying for CPS are studied.

A large number of CPS applications require design technologies that can cover various industrial areas, such as the automotive industry, industrial control, medicine, mobile communications, etc. Each domain has its own view on the basic technical and physical details. Taking into account the various tasks inherent in the development of CPS, it is obvious that CPS needs to improve methods and specifications of interdisciplinary modelling that can support static analysis, validation, modeling, performance analysis and implementation technologies.

15.1.1 Rationale MBSE approaches for analysis, specification, design and verification CPS

Model-based design (MBD) was identified as a powerful design technology for CPS [3]. The models are the basis of the MBD process. The specifications of the system and its basic components are defined as models capable of reflecting the evolution of the system. These models can be used for early project analysis; assist in the separation of problems, traceability, tracking, impact analysis, formal verification, modeling and synthesis. Using models, one can identify structural defects earlier instead of the prototyping phase with a much higher cost. In addition, automated or semi-automated processes can also help synthesize model implementations, such as automatic code generation and software synthesis on heterogeneous platforms [4]. However, the internal heterogeneity and complexity of CPS does not make it possible to adequately take into account all the problems associated with CPS using a single modeling language or modeling tool. MBD is part of an even more fundamental methodology for creating complex systems – Model-Based System Engineering.

15.1.2 An overview the general-purpose modeling languages and its benefits for CPS

The design of cyber-physical systems consisting of software as well as digital and analogue hardware – is still a great challenge that is caused by the increasing complexity and the multidisciplinary requirements which are typical for mixed-signal applications. One issue is to connect different application domains of the system in a whole design process. To cope with this, many different design languages have been developed [5].

In [6], the authors introduce an abstraction language in the form of a domain specific language (DSL) to implement controllers at the mission level, called the Collaborative Sensing Language (CSL).

In our opinion, at present, the UML [7] remains the still general-purpose way of modeling CPS, because it is the most universal, accepted in wide circles of developers and scientists, has a large number of automated tools for its use and is in continuous development. One of the fundamental properties of the UML language is the mechanism of its expansion with the help of stereotypes. Using this mechanism, many profiles were created and standardized, covering almost all domains where it is necessary to design complex systems, for example, SysML™ [8], MARTE [9], UML profile for System on a Chip (SoCP) [10, 11], UML Profile for Schedulability, Performance, & Time (SPTP™) [12], UML Profile for Telecommunication Services (TelcoML™) [13], UML profile for Service Oriented Architecture Model Language (SoaML®) [14], UML Testing Profile 2 (UTP2) [15], SysML Extension for Physical Interaction and Signal Flow Simulation (SysPhS) [16], Robotic Interaction Service (RoIS™) [17], etc.

Domain-specific profiles, such as the UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) extends the UML with generic features required for real-time and embedded systems. However, MARTE should be expanded to cover the modeling of the complete CPS and, in particular, the physical models that are usually left outside the scope of classical digital (cyber) models. SysML is another extension dedicated to systems engineering and has been successfully used to cope with physical models [18].

15.1.3 Technology platforms for CPS modeling

Today, most engineering tools suites are vertically integrated, but have limited support for integration across disciplinary boundaries. The issue is how to create the foundations and technologies for semantically precise model and tool integration that enable reuse of existing tools in the domain-specific design flows [19]. In [19] authors proposed the integration technologies developed for CPS design automation tool chains. These integration technologies based on three integration platforms: the Model Integration Platform, Tool Integration Platform and Execution Integration Platform. These platforms include domain agnostic methods and tools for co-modeling CPS artifacts and engineering processes that can be instantiated in domain-specific contexts. The Model Integration Platform and Tool Integration Platform are implemented into experimental design automation tool suite, OpenMETA, for complex CPS in the vehicle domain.

In recent years, several projects have been carried out aimed at integrating model based methodologies and tools to overcome the problem of multidisciplinarity inherent in CPS [1]. In CHESS project [21] the profile CHESSML has been developed. CHESSML is defined as a collection-extension of a subset of the standard OMG languages (UML, MARTE, SysML), the purpose of which is the possibility of defining platform independent models (PIM), platform specific models (PSM) and analysis models according to the CHESS methodology.

In the CONTREX project [22], a UML/MARTE methodology for distributed, mixed-critical embedded systems has been proposed. This modeling effort is focused on extending the standards to integrate aspects related to distributed networks and mixed- criticality systems, which are not fully addressed in the standards. While CPSs modeling is not undertaken in the project, the outputs of this project can serve as a foundation for new research activities related to CPSs oriented Model-Based methodologies in the near future.

The CONCERTO project [23] proposes a methodology for enabling correct-by-construct component assembly for multicore systems. The automatic generation of virtual prototypes has been made possible, along with introduction of support for separation of concerns using a meta-model based approach. New run-time monitoring mechanisms have been developed to analysis extrafunctional properties such as energy consumption. Finally, the project enables iterative development by enabling back propagation from platform-specific to platform-independent models.

The COMPASS project provides tools and techniques to support a model-based approach to developing Systems of CPSs, also called Systems of Systems (SoSs) by introducing the COMPASS Modeling Language (CML) [24]. They extend SysML by the addition of formal CML notations. COMPASS augments CPSs modeling by means of additional tools and techniques to enable informal SoS development to be undertaken under the guidance of CML analysis techniques, some of which can be presented at the SysML level.

The DESTECS project [25] proposes a methodology for defining co-models allowing discrete event (DE) and continuous time (CT) models to be co-simulated. The DE and CT models are linked through a common interface specification that identifies shared (monitored/controlled) variables, design parameters and events. While the project supports co-simulation, verification is not supported.

The INTO-CPS project [1,23,24] tool chain will provide powerful analysis techniques for CPSs, including generation and static checking of Function-Mockup Interface FMI interfaces; model checking; Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) simulation, supported by automatic code generation.

Many UML modeling tools have plugins for using SysML and MARTE, for example Modelio, Papyrus, MagicDraw, Cameo Systems Modeler, etc.

15.2 MARTE profile of UML foundations

The use of UML profiles and stereotypes is a consequence of the fact that standard UML elements, and any other formalized languages, are not always enough to describe all possible use cases. In this regard, some extensions of languages are used and conventions for their use. The UML profiles SysML and MARTE considered in this section are extensions of the UML language. So, the SysML is the extension of UML for systems modeling, and MARTE is an extension of UML for modeling embedded real-time systems.

15.2.1 An introduction to UML profiles

In this subsection, we will look at the general mechanisms for extending the UML language and how this is implemented in the UML profile MARTE. The extension mechanism of the UML language is the method of making changes to it. This mechanism was provided by the language developers, thus ensuring the extension of the language is a property of the UML language itself. This fundamental property of the language allows the language to be expanded with new elements so that the syntax and semantics of the newly created elements do not contradict the UML language. Most computer modeling tools (for example, Magic Draw, Modelio, Papyrus) in the UML language contain tackles to support the extension, and they handle user-entered elements as well as basic UML elements.

There are three mechanisms for extending the UML language:

– a tagged value is a tag value pair that can be used to add properties to model elements in UML. Tagged values are shown in the form {tag = value} where tag is the tag name and value is a literal value;

– a constraint is a packageable element which represents some condition, restriction or assertion related to some element (that owns the constraint) or several elements. Constraint is usually specified by a Boolean expression which must evaluate to a true or false. Constraint must be satisfied (i.e. evaluated to true) by a correct design of the system;

– a stereotype is a profile class which defines how an existing metaclass may be extended as part of a profile. It enables the use of a platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass. Because stereotype is a class, it may have properties. Properties of a stereotype are referred to as tag definitions. When a stereotype is applied to a model element, the values of the properties are referred to as tagged values.

The process of stereotyping is shown in Fig. 1. Here the stereotype «Microprocessor» is defined, which has the tags {type}, {name}, etc., and the constraint {name must be unique}. In this model, two specific classes MCU (Microcontroller Unit) and DSP (Digital Signal Processor) are stereotyped by the stereotype «Microprocessor». When creating instances of these classes mcu and dsp, you must observe the constraint {name must be unique}, and tags get specific values, for example, {name="TMS570LS3137"}, {name = "TMS320VC549"}.

If the modelling is carried out in a specific domain in which some objects are very often in focus, it is expedient for them to create stereotypes that include characteristic properties that can be expressed as suites of tags and constraints. For example, if we develop an application in which a particular classes, say Interface, Service and Semaphore, often used, then to modelling this application we can introduce stereotypes «Interface», «Service» and «Semaphore» (Fig. 15.2).

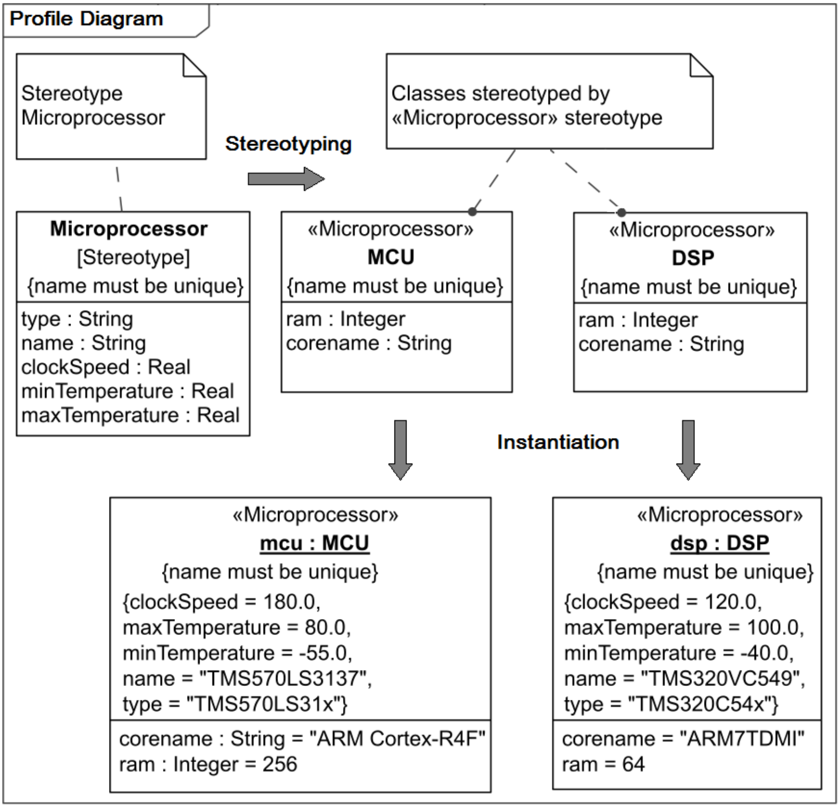


Fig. 15.1 – Example of stereotyping process

Several stereotypes can be grouped into a special package called a profile. A profile is a UML model with a set of ready-made stereotypes, tagged values, constraints, and base classes. Applying a profile to an application is similar to the normal import of a standard package. One or several profiles can be applied to a package containing classes or stereotypes created on the base of metamodel that is extended by a profile. Applying a profile means that it is allowed, but not necessarily, to apply stereotypes defined as part of a profile. It is possible to apply multiple profiles to a package as long as they do not have conflicting constraints. In diagrams, the use of a profile is indicated a dashed arrow than open arrowhead pointing away from the package of the application towards the profile. This arrow is labelled with the keyword «apply». Once again we recall, since profiles are based on standard UML, they do not define a new language and can be supported by standard UML tools. An example of the use of the profile is shown in Fig. 15.3. The previously created Sample profile (Fig.15.2) is used in the package Application. There is a relation «apply» between the profile package and the application package. Classes «ExampleInterface», «ExampleService» and «ExampleSemaphore» are created using the Sample profile.

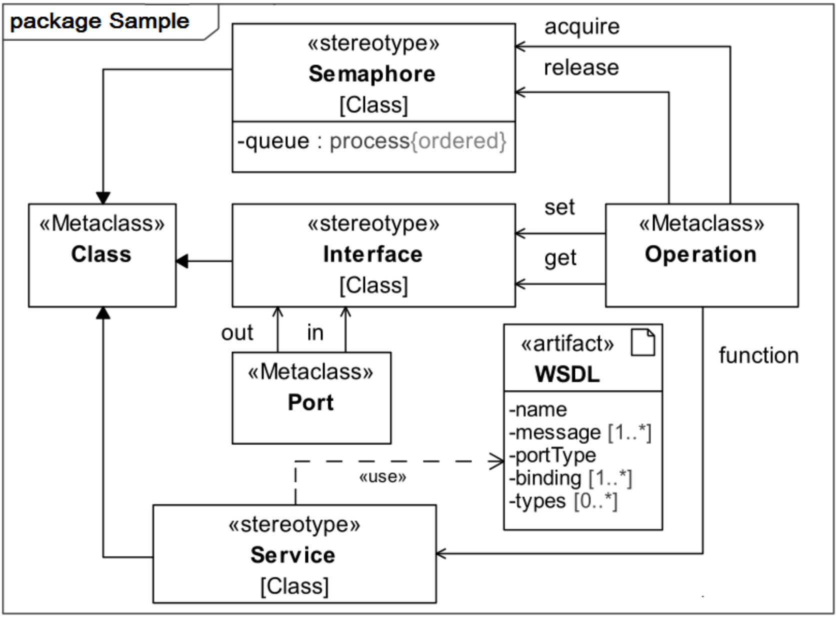


Fig. 15.2 – An example of the introduction of new stereotypes

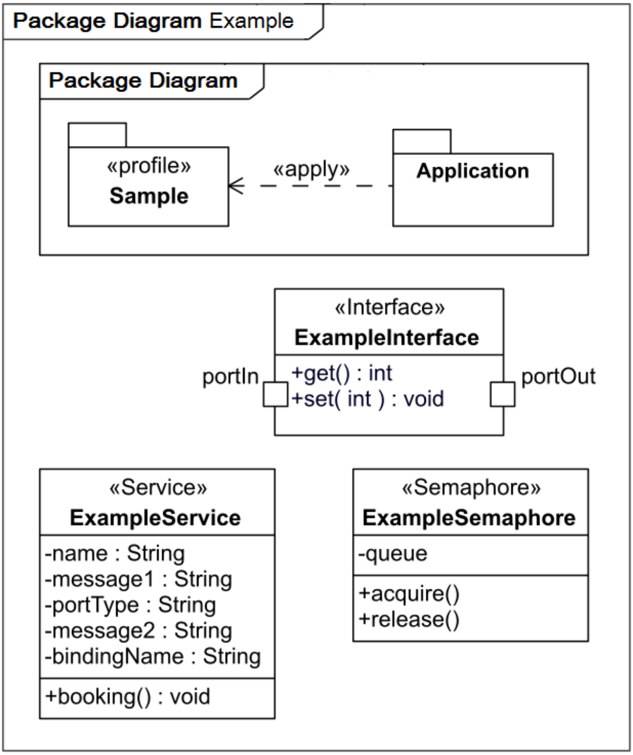


Fig. 15.3 – An example of using a profile with again introduced stereotypes

MARTE is a UML profile for creating models of real-time embedded systems (RTE). It consists of a set of domain extensions that correspond to the general concepts of the UML language and provides the developer with first-class language constructs for RTE modeling. Many of these extensions relate to non-functional aspects (quantitative and qualitative) of RTE applications. Structurally, the MARTE is a hierarchy of profiles and sub-profiles and consists of four main parts (Fig. 15.4).

The top package is basic and consists of four basic UML profiles:

– the non-functional properties (NFP) profile provides the ability to declare and apply constructions that describe the non-functional properties of model elements. It is complemented by the textual Value Specification Language

(VSL), which allows to specify algebraic expressions. Using NFP it is possible to declare non-functional properties as UML data types, and using VSL it is possible to specify values for these types;

– the Time profile allows to determine the time and its presentation in models. The Time profile supports chronometric, logical and synchronous time modeling;

– Generic Resource Modeling (GRM) profile allows modeling of system infrastructure resources, representing a set of resources and computer platforms that implement elements of RTE systems;

– the allocation modeling (Alloc) profile is designed to simulate the distribution of system functions between entities that implement these functions. This can be a distribution in space and in time, as well as taking into account the current non-functional characteristics.

Based on the basic concepts of the top-level profiles package two categories of profiles "MARTE design model" and "MARTE analysis model" were built. The first is designed to support model-based system design, and the second one is for model-based analysis, such as verification, validation, and optimization.

15.2.2 Specifying non-functional properties

MARTE has formal semantics, which allows non-functional requirements to be fixed in more formal ways and with the necessary details to perform system modelling and analysis at early stages of development.

For modeling of the system’s non-functional properties the MARTE profile has following features:

1. A standard library of basic physical data types that represent physical quantities, such as length, mass, duration, energy, etc.;

2. The ability to specify specific literal values for such physical data types (for example, "105 us");

3. The ability to extend the basic types of libraries in accordance with the domain under consideration. An example of the defining new types HighPower and Rate is shown in Fig. 15.5.

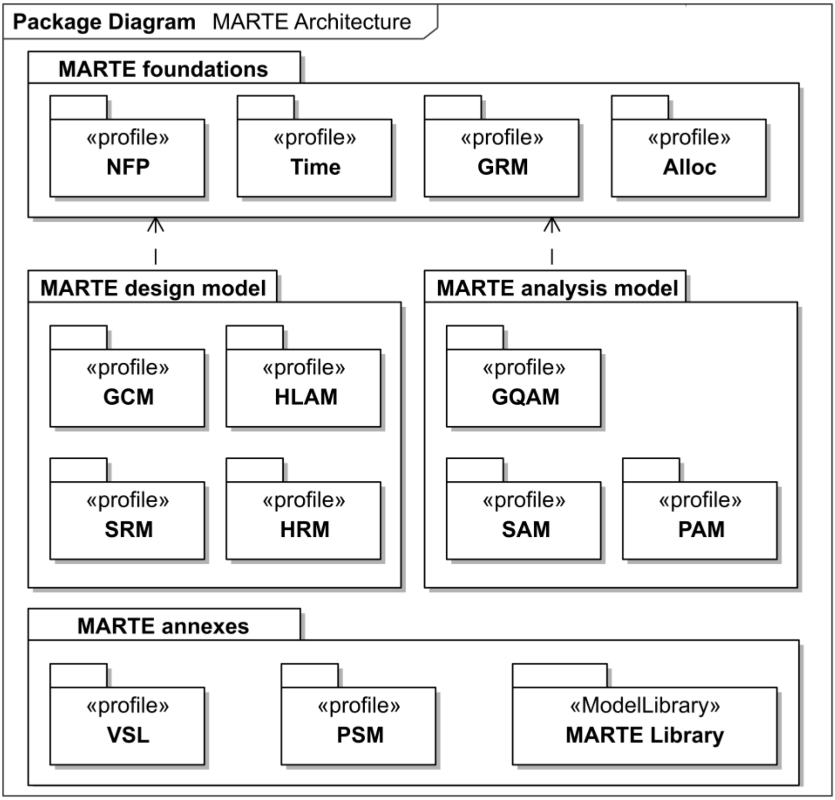


Fig. 15.4 – MARTE's architecture description

15.2.3 Modeling time and resources

Two main peculiar properties distinguish RTE software from other kinds of software:

1) the need for *timely* response to environmental events;

2) the physical characteristics of *resources* (processor performance, memory size, throughput, reliability, etc.) can determine the operation capability of the RTE.

These two concepts link cyberspace (virtual entities) and the physical space in which cyber-physical systems operate. The MARTE profile focuses on the time and resource concepts and is, therefore, important for model-based systems engineering of RTE and CPS.

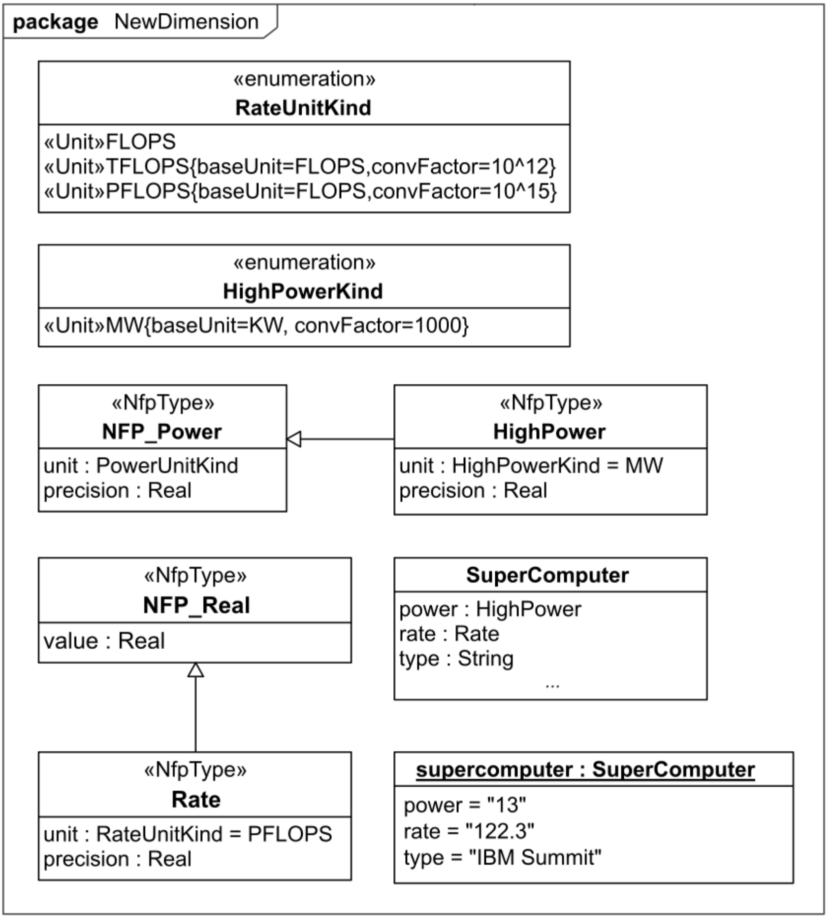


Fig. 15.5 – An example of the defining the new types

Time models. The time model in UML is too simple. The disadvantages of this model include the fact that it is assumed that there is a single source of time, which is used by all simulated distributed sub-systems and subjects; the ways of fixing the time of event occurrences and the process durations are not defined; the accuracy, resolution and some other characteristics are not defined. These shortcomings make it impossible to adequately model RTE end CPS.

MARTE provides two approaches to using the concept of time in models:

– an *explicit clock reference* approach, in which time information is presented relative to an explicit clock whose characteristics are precisely defined;

– the *implicit clock reference* approach, in which the reference to a centralized ideal clock is assumed.

The first approach is necessary for modeling systems that use such quantitative characteristics of time as a unit of time, resolution or offset relative to some basic moment, etc. In such systems, it is necessary to associate the time of the event occurrences and behaviour with an explicit reference to the clock. The second one is used when there is no need for explicit binding to exact clock or such binding is too expensive. In such systems, it is sufficient to assume that for the entire system there is a central clock, and all subsystems refer to this clock. The choice of approach depends on the preferences of the modeler and his understanding of the consequences of the choice and the system being modeled. For example, for the first approach, stereotypes can be used:

– «TimedInstantObservation», which allows to specify explicitly the reference time, for example, when an event occurred or will occur;

– «TimedDurationObservation», which allows to model the duration of the process between two event occurrences;

– «TimedProcessing» to modeling the computational processes that require to specify the moment of completion of the process;

– «TimedEvent» to modeling events that should occur at certain points in time, for example, periodic clock beats.

In the second approach, the «TimeObservation» stereotype, which is the UML 2 stereotype, should be used. In addition, MARTE has a special stereotype «ClockResource» for simple clock modeling.

In MARTE, time is represented as an ordered sequence of time instants. This sequence can be discrete or continuous (sometimes called the dense model of time). To support the various time models used in software development and computer science, MARTE distinguishes and provides three different alternative time conceptualizations:

– the *logical* model of time, which is not metrical and ordering events based on their causal relationships. This time model is very useful for qualitative analysis, for example, determining possible deadlocks in the system;

– the *synchronous* model of time is another non-metric representation of time and it is assumed that the system outputs in response to inputs are computed at certain time intervals which are tied to the beats of a reference clock;

– the *physical* model of time, which is considered as a monotone progression of time instants, i.e. similar to the time we are dealing with.

The key abstraction of MARTE is the concept of a "timed element", which is presented as a model element, inextricably associated with a reference clock and expressed through the stereotype «TimedElement». This stereotype is the parent for stereotypes «TimedInstantObservation», «TimedDurationObservation», «TimedPrcessing», «TimedEvent» and many others.

There are two interrelated stereotypes in MARTE «ClockType» and «Clock» to model the clock. The stereotype «ClockType» has a number of attributes that can be used to determine the characteristics of an instance of a clock, such as nature (continuous or discrete), unit (seconds, milliseconds, etc.), resolAttr (sec, ms, us) – the resolution of the clock, etc. Using the stereotype «Clock», which can be called an instance of the stereotype «ClockType», specific instances of the clock are tagged. Figure 15.5 shows an example of the use of the «ClockType» and «Clock» stereotypes.

The MARTE library has a useful element (class) IdealClock, which has the stereotype of «ClockType» and represents a clock with continuous time and no jitter. This utility element is used to modeling cases where the specific characteristics of clock are not required, but it is assumed this clock with perfect accuracy.

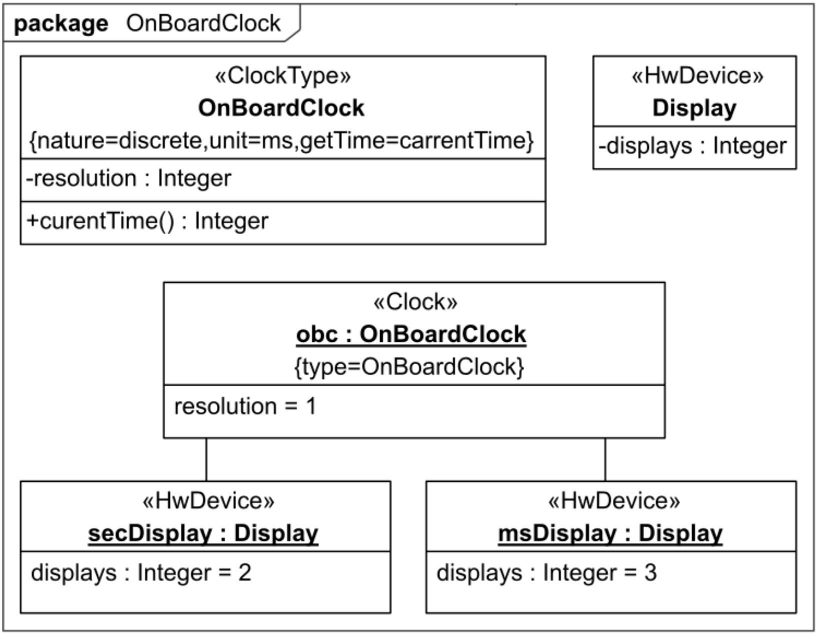


Fig. 15.5 – An example of the use of the «ClockType» and «Clock» stereotypes

Figure 15.6 shows an example of the use of the «TimeProcessing» stereotype, which is used to determine activity with a maximum duration of 150 milliseconds, and this duration is obtained as a result of measurements.

Resource Modeling. To represent the resource, MARTE uses a client-server pattern that models the relationship between the application and the underlying platform. In some cases, this may include three parties: a customer who needs resources; a resource broker who is responsible for managing and allocating resources and a server that actually provides resources.

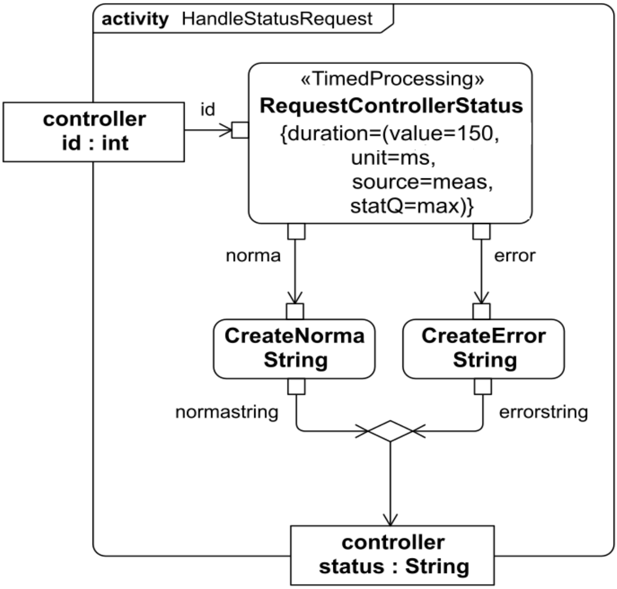


Fig. 15.6 – An example of the use of the «TimeProcessing» stereotype

To model resources, MARTE uses the following basic concepts:

– *resource* is an abstract concept that defines an element with *some kind of physical basis* that participates in the implementation of a software application. It can be a real physical device, such as a controller, processor, sensor, actuator, or network. But it can also be a *logical device*, which is an abstract representation of some physical resource, such as an operating system flow or message pack. Note that even logical devices have a physical basis: the operating system flow is a kind of "virtual" processor, which is realized by the actual physical processor, and the message packet is a specific memory block implemented in application on top of some type of physical storage;

– *resource services* are treated as resource providers accessed through service interfaces;

– *resource broker* may act as an intermediary between a client of resources and resources. Its role is to ensure compliance with the policy of using a certain type of resources. Thus, instead of directly assigning the resource, the client may need to ask the broker for the resource;

– *resource usage*. This concept reflects the characteristics of a particular resource use by some customer. These might include a list of resources used, the corresponding amounts, and some data related to a particular service, such as execution time or energy consumed.

The generalized resource concept is represented in MARTE by the stereotype «Resource», which can be used to tag model elements. MARTE also provides a set of specialized stereotypes based on the nature and purpose of the resource:

– «ProcessingResource» for processing resources, such as computers;

– «StorageResource» for different kinds of memories;

– «CommunicationMedia» and «CommunicationEndPoint» for communication facilities such as networks, busses;

– «ConcurrencyResource» for elements such as operating system processes and threads;

– «MutualExclusionResource» for devices that are used to synchronize the execution of concurrency resources;

– «DeviceResource» for various specialized devices such as sensors and actuators;

– «TimingResource» for clocks and timers.

Most of these stereotypes are additionally refined by more specialized stereotypes. The concept of a resource can be applied both to a classifier (class, interface), and to an object.

The general stereotype «Resource» has three attributes that are inherited by all its improved stereotypes:

– resMult, which indicates how many instances of the resource represents by the stereotyped model element. It can be used to represent collections of resources of the same type;

– isActive, which is Boolean attribute. Its true value means that the resource can be active independently of other objects of the model;

– isProtected is a logical attribute that indicates that access to an object is somehow protected by, for example, a resource broker.

Resource services are presented through the general «GrService» stereotype, which is detailed by two useful stereotypes:

– «Acquire» to indicate that a resource is being acquired;

– «Release» to indicate that the resource is being released.

Resource usage can be defined using the «ResourceUsage» stereotype.

Figure 15.7 shows an example of how requirements for application components are captured using resource stereotypes. For example, instances of c1 and c2 of the Controller class, which are stereotyped by the «HwComputingResource» stereotype, should have a performance of 5 mips and have 100 KB of RAM. Similarly requirements are defined for computer and data link environment.

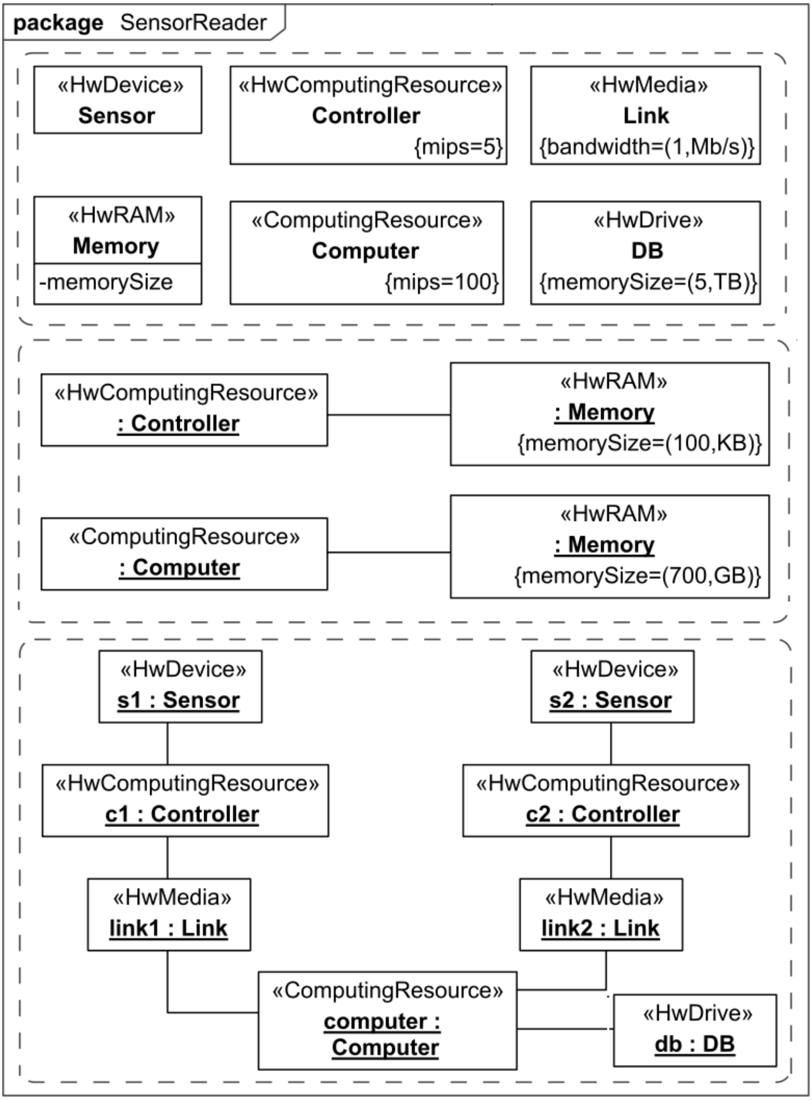


Fig. 15.7 – Example showing modeling an acceptable platform for an application

15.3 Modeling CPS with SysML and MARTE

15.3.1 The SysML profile

To support MBSE, the Object Management Group (OMG) consortium has developed and proposed Systems Modeling Language (OMG SysML ™) [8, 25]. It is a visual design language based on the widely accepted UML, which was originally used mainly for the development of complex software systems. SysML uses and extends many functions of UML 2, so that it can be used to develop all kinds technical systems. In particular, diagrams of requirements, blocks, parameters were introduced into SysML, activity diagrams were modified, properties of standard ports were expanded (Fig. 15.8).

The SysML allows representing system requirements as integral parts of a model. Requirements do not contain attributes and operations; they cannot have relation of generalization and association. However, requirements may have a sub-requirements. Special relations such as DeriveRqt, Refine and Trace allow building hierarchies of requirements and dependencies between requirements. Representing a requirement as an element of a model allows to associate it with a component of the system by the satisfy relation, i.e. explicitly indicate which component of the system implements this requirement.

The main structural element of SysML is a «block» with which you can represent any component of the system – functional, physical or virtual objects. Internal Block Diagram allows the developer to clarify the structural aspect of the model.

Parametric diagrams are used to represent constraints that can be expressed using mathematical formulas and logical relations. Using the concept of an allocation, a developer can link and match various modeling methods. Allocation is most often used to map system functions to structural elements of a modeling system.

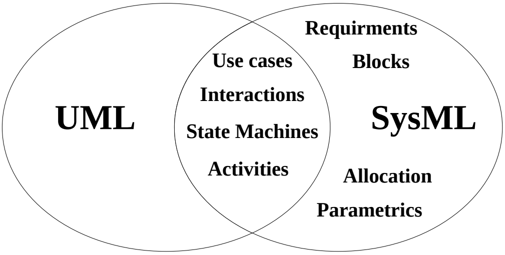


Fig. 15.8 – Comparison of SysML with UML

15.3.2 Methods of combining SysML and MARTE for modeling CPS

A CPS system is a heterogeneous technical system consisting of many heterogeneous physical components controlled by software. The main characteristic of the engineering of this system is that the system should be developed as a whole. This means that even after the decomposition of the system into subsystems, it is necessary to constantly evaluate and correct the relations between the parts and the whole. This is primarily due to the growing functional complexity of CPS, which are more functionally integrated than conventional technical systems.

Real-time embedded software should support communication between programs in subsystems and systems of a higher hierarchical level or external systems. This means that in a model-oriented design, the relations between the CPS subsystems models can be in refinement relation of or in peer-to-peer relation. In the first case, SysML can be used to cover high-level system architecture, and as a combination of UML and MARTE can be used to more clearly represent the parts implemented by software and supporting computer equipment. In the case of a peer-to-peer relationship between models SysML and MARTE are used at the same level of abstraction, either as separate but related models, or in the same model while simultaneously using these profiles. SysML is used to specify general system requirements, as well as to represent mechanical and other physical components that interact with software. The combination of UML-MARTE is used to model software components and information related to its allocation, resources and platforms.

There are three ways to use SysML and MARTE together [26]:

– *Disjoint models*,

– *Partitioned model*, and

–*Overlapping models*.

In the "*Disjoint models*" approach each profile is applied to different model, but the models belong to the same system. For example, models can represent different levels of abstraction. In this case, the top levels of abstraction are modeled using SysML, and the lower levels, especially if they are software modules, are modeled using UML and/or the MARTE profile (Fig. 15.9). Relationships between relevant models at different levels of abstraction may be tagged via external traceability links. For example, element Control in the system-level model is decomposed into a software component MotionControl and a hardware component Controller in a detailed level. If the models are at the same level of abstraction, but they model different parts of the system, as shown in Fig. 15.9. The link between them can also be represented by some external tracking mechanism. For example, component

TrafficControl may be a software driver for the physical part represented by Trafic Camera. Also, these elements can be linked through a common abstraction top level element.

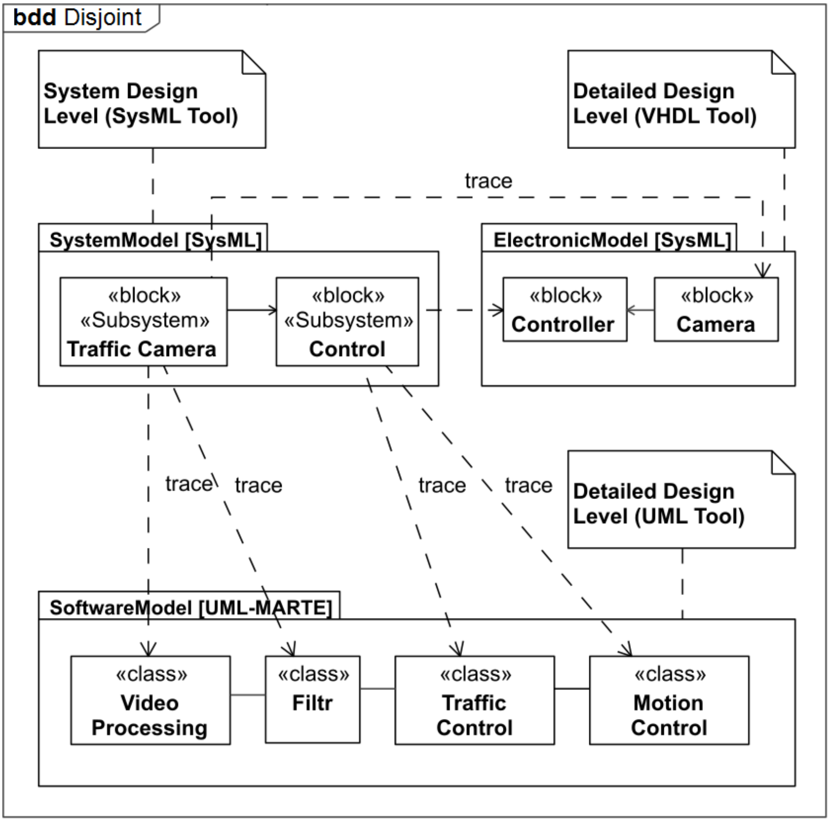


Fig. 15.9 – Joint use of SysML and UML-MARTE for models at different levels of abstraction

In the "*Partitioned model*" approach, both profiles are present in the same model, but each of them is applied to different parts of the model, so there are no overlapping elements. This approach is possible due to the fact that UML allows you to use multiple profiles in one model. In this case, it is possible to link the related parts of models with standard UML relationships, such as association, dependency, generalization, or implementation, or SysML profile relationships, such as allocation, track, etc.

The "*Overlapping Models*" approach involves concurrent and overlapping use of SysML and MARTE on the same model. Overlapping means that some model elements may be tagged with both SysML and MARTE stereotypes (Fig. 15.10).

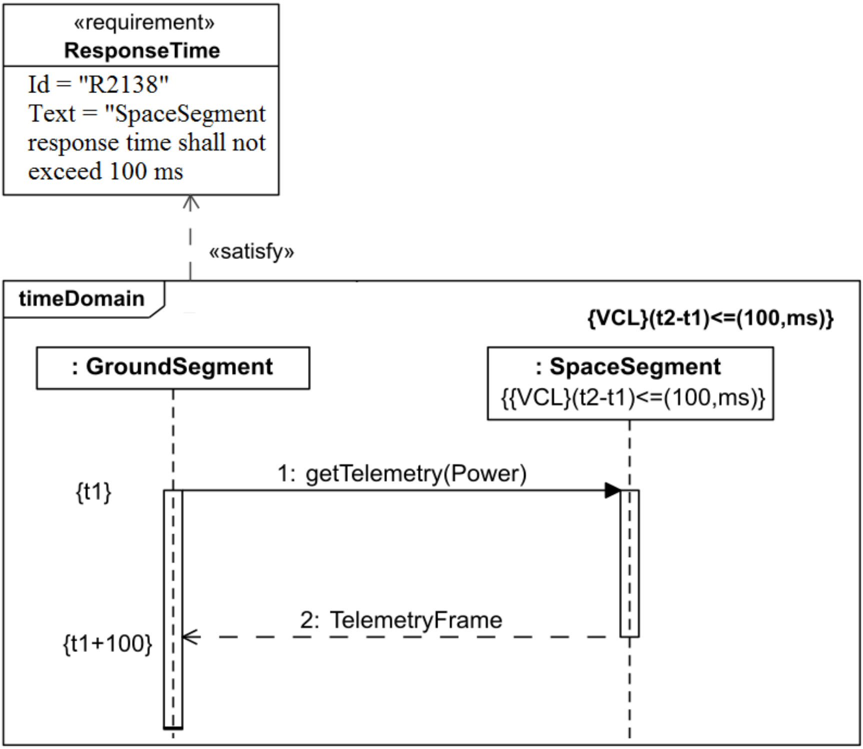


Fig. 15.10 – Example of combined use of SysML and MARTE to define requirements

15.4 Basics of model-based analysis of CPS

One of the goals of model-based design is to understand and predict the performance and key characteristics of the system being developed. For example, will the performance of microcontroller be sufficient for real time orientation determination of the robot manipulator? Will there be enough memory to store contextual information? Will network throughput ensure timely receipt of geodata from cloud servers? Accurate answers to these questions will reduce the strategic risks of the project. These problems are especially difficult when developing software for CPS, since single errors among thousands or millions of the program code lines, such as an uninitialized pointer or uncontrolled array boundaries, can drive on the collapse of the entire system.

Designers and analysts can use different languages and different models to represent the same system. Therefore, it is very important to ensure the mutual consistency of these models. Since the design model is the original source for all analysis models, the problem is that the analysis models are an accurate representation of the design model.

The most reliable way to ensure the consistency of these two types of models is to formally derive analysis models from a design model. This requires formal transformations from UML to different analysis languages. MARTE can help avoid these problems by providing a set of analysis-specific annotation subprofiles, which can be used to attach the necessary information related to a specific viewpoint directly to the design model [26].

A significant part of MARTE’s sub-profiles are those that provide model-based analysis. Central to this is a generic framework, called the Generic Quantitative Analysis Modeling (GQAM).

The system analysis model of the MARTE profile is based on a "demand-supply" pattern. The pattern consists of two main elements:

– the supply is represented by the under analysis, which consist of the application and its supporting platform, i.e. a set of hardware or software resources that are ultimately associated with physical devices (memory, processors, communication devices);

– the demand is represented by the workload imposed on the system by the environment in which it operates, i.e. these are use cases. Use cases are essentially functional requirements for the system, which in turn are related to quality of service requirements, such as maximal acceptable response time, reliability, safety, cost, or energy consumption.

Responsible design technical decision is not made the first time around, it is necessary to analyze multiple project alternatives, compare the results of their analysis and make a choice, and this is an iterative process. MARTE provides its own approach to the model analysis process. It consists of an extensible generalized model annotation structure (GQAM) and two specialized analysis subprofiles: the Schedulability Analysis Modeling (SAM) and Performance Analysis Modeling (PAM).

The analysis process (Fig. 15.11) starts with a design model, which is then annotated using the appropriate MARTE stereotypes. The annotated model is transformed into appropriate analysis-specific model. Further, this model is analyzed. The analysis results are evaluated and, if it is necessary, the cycle can be repeated using other configurations of the annotation values until a satisfactory combination is uncovered.

The central concept of the GQAM is the analysis context. This concept represents a situation that needs to be analyzed and serves as the starting point for analysis (Fig.15.12). A typical example of an analysis context is a sequence diagram describing a scenario or set of scenarios whose timing or performance characteristics need to be analyzed. An example of analysis context is shown in Fig. 15.13. Note, to completely specify an analysis context, at least one platform needs to be defined.

In GQAM, the analysis context is represented via the generic «GaAnalysisContext» stereotype. It has following attributes:

– workload – this specifies the workload imposed on the system (i.e., the "demand");

– platform – this points to a resource that represents the platform on which the real-time application executes;

– contextParams – this specifies an optional set of parameters to be used in analysis (see below);

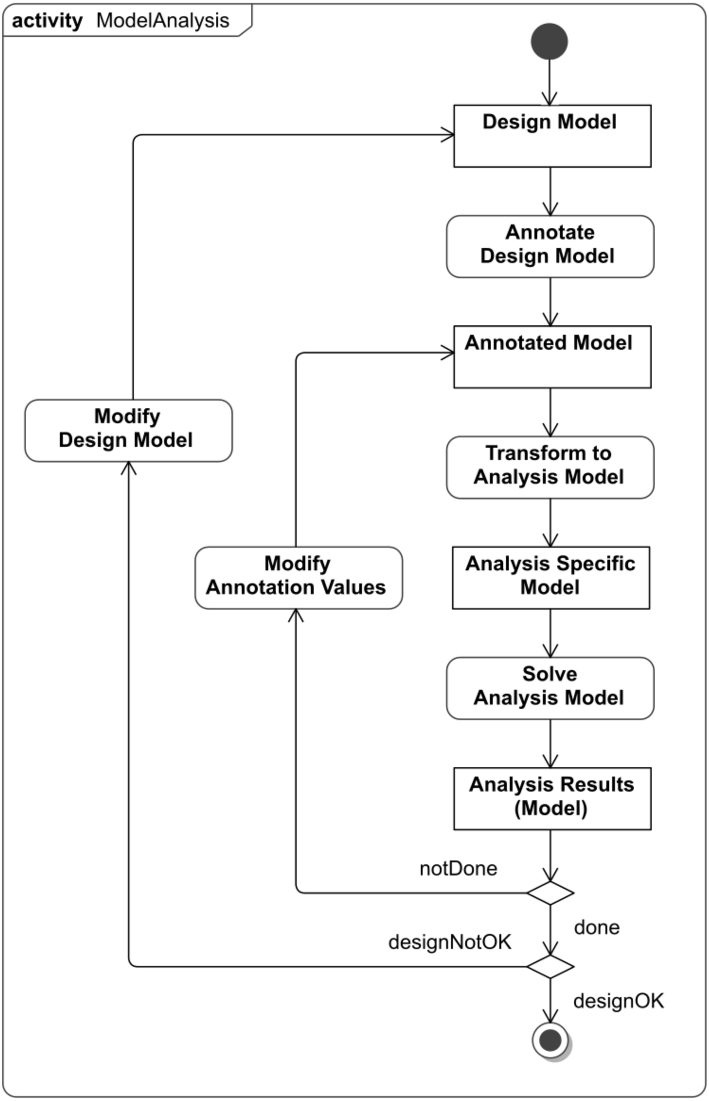


Fig. 15.11 – The general model analysis process based on MARTE.

– mode – this points to a state in a UML state machine that represents a mode of operation of a system in which the analysis context applies (e.g., a "Running" state of an engine); this allows modeling of systems whose behavior can vary with the operating mode.

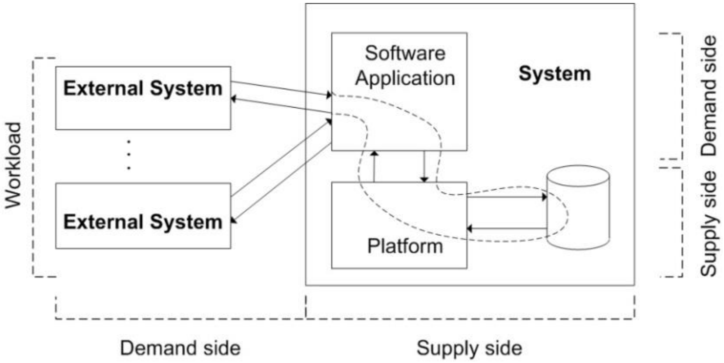


Fig. 15.12 – Conceptual view of an analysis context

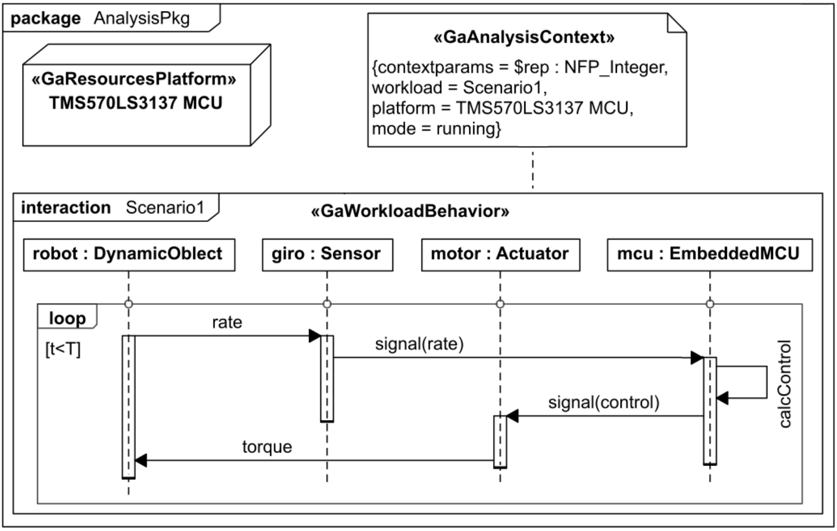


Fig. 15.13 – An example of analysis context

The GQAM is a subprofile of MARTE that realizes a generic conceptual framework for implementing analyses based on the demand-supply pattern. GQAM provides a number of stereotypes that are intended to be specialized to produce analysis-specific variants. These specific stereotypes are packaged in separate subprofiles. These specific stereotypes are packaged in two separate sub-profiles:

– SAM is a profile for analyzing the schedulability characteristics of certain categories of mostly cyclical real-time systems, using established methods such as rate-monotonic analysis. For the schedulability analysis the context of the system is stereotyped by the «SaAnalysisContext» stereotype, which is a specialization of the «GaAnalysisContext» stereotype (and inherits all its attributes), and adds the attributes isSched and optCriterion. The isSched attribute is used to store the result of the analysis. If its value is *true*, then specified configuration of schedulable resources is called schedulable, *false* otherwise it is not. The optCriterion attribute is typed by the OptimalityCriterionKind enumeration and is used to specify the criterion to be used for the analysis (meetHardDeadlines, minimizeMissedDeadlines, minimize-Meant-ardiness, undef, other).

– PAM is a profile for analyzing the performance characteristics of systems based on queuing theory.

The demand-side is represent by workload description, which can be modelled using «GaWorkloadEvent» stereotype. It has the following attributes:

– pattern is used to specify events whose occurrence is characterized by the MARTE library type ArrivalPattern;

– generator is used when an event is specified in the form of UML behavior (finite state machine, action). This attribute has the stereotype «GaWorkloadGenerator»;

– trace is used when the event occurrences described by a UML behavior and stereotyped as «GaEventTrace»;

– timedEvent is used when the event occurs at a specified time.

The supply side of the system in GQAM is represented by some scenario in which the system uses resources to perform its functions. A scenario is triggered by an external event and consists of an ordered set of steps, for each of which the system can use one or several platform resources. The scenario is stereotyped by the «GaScenario» stereotype, which has a number of attributes that specify the initiating events (cause), scenario steps (root), the demands to the host resources (hostDemand), and the occupancy of the resource hosting the scenario (utilization).

15.5 Work related analysis

In recent years, the growth of connected CPSs and IoT devices has increased tremendously due to the availability of high-capacity networks (3G and 4G/LTE networks), advanced sensors (e.g. RFID, NFC, etc.), protocols (e.g. IPv6, MQTT, etc.), mobile Internet and wearable devices [1]. Model-based design is a powerful design technique for CPS which emphasizes mathematical modeling to design, analyze, verify, and validate dynamic systems. A complete model of the CPS represents the coupling of its environment, physical processes, and embedded computations. Modeled systems may be tested and simulated offline, enabling developers to verify the logic of their application, assumptions about its environment, and end-to-end (i.e. closed-loop) behaviour [3]. Therefore, researching and teaching model methodologies for CPS, including related issues, is very important. To address the problems associated with the design and modeling of CPS, many standards (e.g. SysML [8], MARTE [9], SoCP [10], SPTP [12], etc.), methodologies (in projects CHESS[21], CONTREX [22], CONCERTO [23], etc.), tools (in projects COMPASS [24], DESTECS [25], INTO-CPS [1,23,24], etc.) and domain-specific languages (SysML[8], TelcoML[13], SoaML [14], UTP2 [15], etc.) have been developed to cover the design challenges of specific design domains.

These standards, methodologies, tools and languages are the result of the collective work of numerous experts in the domain of embedded real-time systems and cover various domains of the industry. However, there is a certain drawback of practical and training examples using SysML and MARTE described in publications, and the official documentations are structured as reference manuals and have a large amount (for example, the current version of the standard MARTE extends to almost 800 pages).

Conclusion and questions

This section outlines one of the approaches to support the processes of the MBSE in the design of CPS. This approach is based on modeling using standard UML extensions, namely the SysML and MARTE profiles. What the UML profile is, how to expand it and use it for specific tasks, in particular, for embedded real-time systems are described. The examples of using these profiles for CPS modeling in their development are considered. The common foundations of the MARTE framework (GQAM) for the analysis of systems are considered. They are based on the demand-supply pattern and two profiles SAM and PAM for analyzing the real-time embedded systems schedulability and performance, respectively. Acquiring skills of modeling using SysML and

MARTE standards will help create CPS and other embedded real-time systems, which is the main goal of the MBSE.

1. What is the purpose of systems engineering?
2. What is the difference between a good model and a good design?
3. What is the difference between modeling and simulation?
4. What is the difference between model-based and document-based design?
5. What is the purpose of applying an MBSE method?
6. What two types of profiles can be created in UML?
7. Which model elements can a profile contain?
8. What are three mechanisms for extending the UML language?
9. What are the stereotypes in UML used for?
10. What are the tags in UML used for?
11. What is the difference between slots and tags?
12. What are the constraints in UML used for?
13. What must modelers do before they can apply stereotypes to elements in their models?
14. On a diagram, how can a modeler tell that a stereotype has been applied to a model element?
15. What does MARTE add to UML?
16. What are the main subprofiles the profile MARTE consists of?
17. What pattern MARTE uses for modeling the relationship between an application and the underlying platform?
18. What does non-functional quality of the system mean and which ones do you know?
19. How to define new physical data types in MARTE?
20. What is the difference between a time model in UML and a time model in MARTE?
21. What three alternative models of time does MARTE offer?
22. What does the term "resource" mean in MARTE?
23. What pattern does the MARTE profile use to represent the resource?
24. are the three roles can be in the MARTE resource model?
25. What stereotypes does MARTE provide for modeling such resources as processors, memory, threads, mutual exclusions resources, specialized devices, clocks and timers?
26. What are five aspects of a system that SysML can represent?
27. What is the difference between «block» in SysML and «class» in UML?
28. What tools do you know for modeling with UML, SysML and MARTE?

29. What are the three ways to apply SysML and MARTE profiles together?

30. What pattern is used in MARTE for analysis?

31. What two subprofiles are used in MARTE for analysis?

References

1. I. Quadri, E. Brosse, A. Sadovykh and A. Bagnato, "Modeling Methodologies for Cyber-Physical Systems: Research Field Study on Inherent and Future Challenges", *Ada User Journal*, vol. 34, no. 4, pp. 246-253, 2015.

2. E. Lee and S. Seshia, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, 2nd ed. MIT Press, 2017.

3. J. Jensen, D. Chang and E. Lee, "A Model-Based Design Methodology for Cyber-Physical Systems", in *International Wireless Communications and Mobile Computing Conference - IWCMC 2011*, Istanbul, 2011, pp. 1666-1671.

4. H. Posadas, P. Peñil, A. Nicolás and E. Villa, "System synthesis from UML/MARTE models: The PHARAON approach", in *2013 Electronic System Level Synthesis Conference (ESLsyn)*, Austin, TX, 2013, pp. 1-8.

5. F. Slomka, S. Kollmann, S. Moser and K. Kempf, "A multidisciplinary design methodology for cyber-physical systems", in *4th International Workshop on Model Based Architecting and Construction of Embedded Systems, Wellington (New-Zealand),* 2011, pp. 23-37.

6. E. Pereira, K. Hedrick and R. Sengupta, "The C3UV testbed for collaborative control and information acquisition using UAVs", in *American Control Conference*, Washington, DC, 2013, pp. 1466-1471.

7. "Unified Modeling Language", *Object Management Group (OMG)*, 2017. [Online]. Available: https://www.omg.org/spec/UML/. [Accessed: 10-Jul- 2018].

8. "SysML Specifications - Current Version: OMG SysML 1.5", *SysML.org*, 2017. [Online]. Available: https://sysml.org/sysml-specifications/. [Accessed: 18- Jul- 2018].

9. "About the UML Profile for MARTE Specification Version 1.0", *omg.org,* 2009. [Online]. Available: https://www.omg.org/spec/ MARTE/1.0. [Accessed: 26- Jul- 2018].

10. "About the UML Profile for System on a Chip Specification Version 1.0.1", o*mg.org*, 2006. [Online]. Available: https://www.omg. org/spec/SoCP/About-SoCP/. [Accessed: 26- Jul- 2018].

11. "About the UML Profile for System on a Chip Specification Version 1.0.1", *omg.org*, 2006. [Online]. Available: https://www.omg. org/spec/SoCP. [Accessed: 10- Sep- 2018].

12. F. Boutekkouk, M. Benmohammed, S. Bilavarn and M. Auguin, "UML2.0 Profiles for Embedded Systems and Systems On a Chip (SOCs).", *The Journal of Object Technology*, vol. 8, no. 1, pp. 135-157, 2009.

13. "About the UML Profile for Schedulability, Performance, & Time Specification Version 1.1", *omg.org*, 2005. [Online]. Available: https://www.omg.org/spec/SPTP/. [Accessed: 18- Feb- 2019].

14. "About the UML Profile for Telecommunication Services Specification Version 1.0", *omg.org*, 2013. [Online]. Available: https://www.omg.org/spec/TelcoML/. [Accessed: 18- Feb- 2019].

15. "About the Service Oriented Architecture Modeling Language Specification Version 1.0.1", *omg.org*, 2012. [Online]. Available: https://www.omg.org/spec/SoaML/. [Accessed: 18- Feb- 2019].

16. "About the UML Testing Profile 2 Specification Version 2.0", *omg.org*, 2018. [Online]. Available: https://www.omg.org/ spec/UTP2/. [Accessed: 20- Jun- 2019].

17. "About the SysML Extension for Physical Interaction and Signal Flow Simulation Specification Version 1.0", *omg.org*, 2018. [Online]. Available: https://www.omg.org/spec/SysPhS/. [Accessed: 12- Apr- 2019].

18. "About the Robotic Interaction Service Specification Version 1.2", *omg.org*, 2018. [Online]. Available: https://www.omg.org/spec /RoIS/. [Accessed: 12- Apr- 2019].

19. F. Mallet, E. Villar and F. Herrera, "MARTE for CPS and CPSoS", in S. Nakajima, J.P. Talpin, M. Toyoshima and H. Yu (eds.), *Cyber-Physical System Design from an Architecture Analysis Viewpoint: Communications of NII Shonan Meeting*, Springer (Singapore), 2017, pp. 81-108.

20. J. Sztipanovits, T. Bapty, X. Koutsoukos, Z. Lattmann, S. Neema, and E. Jackson, "Model and Tool Integration Platforms for Cyber–Physical System Design", in *Proceedings of the IEEE*, vol. 106, no. 9, Sep. 2018, pp. 1501-1526.

21. Larsen, P.G., Fitzgerald, J., Woodcock, J., Gamble, C., Payne, R., Pierce, K. "Features of integrated model-based co-modelling and co-simulation technology", in *Software Engineering and Formal Methods - SEFM 2017 Collocated Workshops: DataMod, FAACS, MSE, CoSim-CPS, and FOCLASA, Revised Selected Papers.* vol. 10729, Springer, Lecture Notes in Computer Science, 2017, pp. 377-390.

22. "Composition with Guarantees for High-integrity Embedded Software Components Assembly (CHESS)", 2012. [Online]. www.chess-project.org [Accessed: 11- Feb- 2017].

23. "Design of embedded mixed-criticality CONTRol systems under consideration of EXtra-functional properties (CONTREX)", 2017. [Online]. https://contrex.offis.de/ [Accessed: 11- Feb- 2019].

24. J. Fitzgerald, C. Gamble, R. Payne, P. G. Larsen, S. Basagiannis and A.E.D. Mady, "Collaborative model-based systems engineering for cyber-physical systems, with a building automation case study", in *INCOSE International Symposium*, vol. 26, Sep. 2016, pp. 817-832.

25. P. G. Larsen, P.G., J. Fitzgerald, J. Woodcock, P. Fritzson, et al., "Integrated tool chain for model-based design of Cyber-Physical Systems: the INTO-CPS project", in *2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*, 2016, pp. 1–6.

26. S. Friedenthal, A. Moore, and S. Rick, *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann Publishers, Inc.: San Francisco, CA, 2012.

27. B. Selic, S. Gérard, S. *Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE: Developing Cyber-Physical Systems.* Morgan Kaufmann, Burlington, 2013.

Анотація

УДК 62:004=111

І73

Рецензенты: Dr. Mario Fusani, ISTI-CNR, Піза, Італія
Dr. Olga Kordas, KTH University, Стокгольм, Швеція
Viktor Kordas, KTH University, Стокгольм, Швеція

І73 Інтернет речей для індустріальних і гуманітарних застосунків. У трьох томах. Том 1. Основи і технології / За ред. В. С. Харченка. - Міністерство освіти і науки України, Національний аерокосмічний університет ХАІ, 2019. - 605 с.

ISBN 978-617-7361-81-6

Книга, що складається з трьох томів, містить теоретичні матеріали для лекцій та тренінгів, розроблених в рамках проекту Internet of Things: Emerging Curriculum for Industry and Human Applications / ALIOT, 573818-EPP-1-2016-1-UK-EPPKA2- CBHE-JP, 2016-2019, що фінансується програмою ЄС ERASMUS +. Том 1 описує проблеми, принципи і технології Інтернету речей (IoT) і Інтернету всього (IoE). Книга складається з 4 частин для відповідних магістерських курсів: основи IoT, IoE і Веб речей (розділи 1-4), наука про дані для IoT (розділи 5-8), мобільні і гібридні IoT обчислення (розділи 9-11), IoT технології для кіберфізіческіх систем (розділи 12-15).

Книга підготовлена українськими університетськими командами за підтримки колег з академічних закладів країн ЄС, що входять до консорціуму проекту ALIOT.

Книга призначена для магістрантів і аспірантів, які вивчають технології IoT, програмну і комп'ютерну інженерію, комп'ютерні науки. Може бути корисною для викладачів університетів і навчальних центрів, дослідників і розробників систем IoT.

Рис.: 172. Посилань: 606. Таблиць: 30.

Анотації розділів 1-15

У вступному розділі проаналізовано динаміку публікацій з Інтернету речей (IoT), існуючі навчальні програми з IoT для магістрів, докторів наук та пісді-університетської інженерної освіти. Представлено спільний проект з розробки навчальних програм ALIOT, що фінансується в рамках програми Erasmus+. Проект забезпечує адаптацію академічних програм в Україні та інших країнах до потреб європейського ринку праці. ALIOT охоплює різні сфери застосувань IoT, такі як системи охорони здоров'я, системи інтелектуального транспорту, системи екології та промисловості 4.0, розумні будівлі та місто.

У розділі 1 аналізується концепція IoT, завдання, методологічні проблеми та рішення в області застосування IoT, пропонується розширена концепція IoT. У ньому розглядаються методологічні та практичні питання впровадження систем і інструментів BDA і IoT в контексті кібербезпеки і оцінки та забезпечення безпеки. Аналізуються концепції безпеки та кібербезпеки з урахуванням атрибутів безпеки і кібербезпеки для критично важливих додатків. Обговорюються переваги і обмеження застосування технологій, заснованих на BDA і IoT, в критично важливих для безпеки системах, включаючи можливості їх використання для запобігання, моніторингу та мінімізації наслідків важких аварій. Описано промислові приклади, такі як система моніторингу аварій АЕС, яка базується на Інтернеті дронів (IoD), система медичного контролю на базі IoT, система прогнозування надійності програмного забезпечення. Сформульовано рекомендації та обмеження застосування BDA і IoT в критичних для безпеки системах.

Розділ 2 присвячений огляду технологій IoT/WoT, навчальній програмі для майбутніх фахівців з розробки технологій IoT/WoT і досвіду викладання дисципліни: «Технології та засоби розробки додатків WoT» в різних варіаціях. Огляд існуючих технологій WoT/IoT показує, що серверною мовою програмування інтерфейсу Web Thing API, побудованого з урахуванням архітектури REST, є JS в середовищі Node.JS. З урахуванням обраної мови програмування Web Thing API аналізуються технології для розробки WoT додатків з використанням JavaScript/Node.JS. Надано огляд курсів «Технології та засоби для розробки веб-додатків», доповнена розділами «Хмарні обчислення», «IoT/WoT» які викладаються в НТУ «ХПІ» на кафедрі системи інформації і в НАУ «ХАІ» на кафедрі комп'ютерних систем, мереж і кібербезпеки по курсах «Мережа речей» і «Промисловий Інтернет речей».

Інтернет речей активно охоплює різні сфери людської діяльності. У розділі 3 було проведено аналіз наявних рішень та ефективності щодо систем IoT. Проаналізовано стандарти та рекомендації у галузі архітектури, безпеки, технологій IoT-систем. Розглянуто особливості представлення моделей систем Інтернету речей, вимоги до їх організації. Проводиться аналіз та описуються основні показники для оцінювання систем Інтернет речей. Розглянуто особливості доменів Інтернету речей.

В розділі 4 розглядаються методи комунікації в області Інтернету речей, при цьому особлива увага приділяється протоколам прикладного рівня, що використовуються. Досліджуються такі питання, як архітектура мереж, затримки в бездротових сенсорних мережах, можливості стандарту Bluetooth 5.0 та хмарна архітектура для інтернету речей. Проведено порівняльний аналіз протоколів HTTP/HTTPS, MQTT, MQTT-SN, AMQP, XMPP, DDS і CoAP. Оцінені їх переваги та недоліки. Розглянуто методи визначення ефективної швидкості завадостійких кодів і порівняння їх швидкості, а також комплексний показник енергоефективності завадостійких кодів для пристроїв IoT.

Розділ 5 надає стислий огляд наукових методів аналітики даних для IoT. Наведено огляд характеристик даних, відповідних підходів та методів та алгоритмів наукових даних, що застосовуються до даних IoT. Описана екосистема IoT та IoE. Обговорюються моделі наукової аналітики, використовувані у вертикалях IoT, а також синтез даних та обробка даних з пристроїв IoT. Мета цього розділу - зрозуміти, як дані IoT від сенсорів до кінцевих пристроїв можна отримувати та аналізувати для виявлення інформації.

У розділі 6 розглянуто принципи і технології інтелектуального аналізу і обробки даних для IoT систем. Обговорюються особливості використання інтелектуального аналізу даних для IoT даних, моделі і методи інтелектуального аналізу даних для IoT, інтелектуальний аналізу потоків даних і масивних наборів даних.

У розділі 7 представлено результати розвитку та практичні досягнення на перетині технологій Інтернету речей та штучного інтелекту. Детально описано принципи функціонування глибоких нейронних мереж, наводяться приклади їх роботи, аналізуються основні аспекти проектування, використання та впровадження для систем Інтернету речей.

У 8 розділі розглядаються питання зберігання великих даних в контексті IoT-систем, а також досліджується взаємоз’вязок між узгодженістю даних і часовими затримками в розподілених сховищах даних NoSQL. Основна увага в роботі приділяється вивченню залежності продуктивністю (часом відгуку і пропускною спроможністю) нереляційних бази даних Cassandra і настройками узгоджености. У розділі представлені результати аналізу продуктивності (швидкості виконання операцій читання і запису) кластеру Cassandra, розгорнутого в хмарному середовищі Amazon EC2. Представлені кількісні результати, які дозволяють оцінити вплив різних параметрів узгодженості даних на продуктивність Cassandra при різних робочих навантаженнях. Наприкінці розділу запропоновано методику підвищення продуктивності Cassandra на основі оптимізації параметрів узгодженості з урахуванням співвідношення операцій читання і запису та їхньої інтенсивності.

Розділ 9 містить базову інформацію щодо розробки мобільних додатків для операційних систем Android та iOS, а також для сумісних з ними переносних пристроїв. Оскільки книга призначена для магістрів, аспірантів та інженерів із інформаційних технологій, у розділі описано різноманітні питання, починаючи з розгляду сучасних стандартів, що використовуються у мобільній та IoT індустріях, основ проектування інтерфейсів користувача, аналізу рекомендованих Google та Apple архітектур мобільних додатків та закінчуючи детальною інструкцією з завантаження додатку на два основних мобільних маркета – App Store та Google Play.

Десятий розділ містить інформацію про поєднання двох модерних технологій – Cloud комп’ютингу та Інтернету речей, а також про те, як його результати можуть бути використані під час мобільної розробки. Оскільки навчальний матеріал розрахований як на магістрів, аспірантів так і на досвідчених фахівців з області IT, у даному розділі проаналізовано сучасний економічний стан Cloud сервісів, розглянуто базові архітектури та інфраструктури, що можуть бути використані під час розробки мобільних додатків.

В розділі 11 описується інтеграція двох найбільш обговорюваних сьогодні концепцій інформаційних технологій: великих даних та Інтернет речей. Оскільки ця книга призначена для магістрантів, аспірантів та інженерів, які будуть брати участь в проектуванні і розробці таких інтегрованих проектів, надано технічний огляд області великих даних з точки зору IoT.

В розділі 12 приведено результати аналізу сучасного стану розвитку кіберфізичних систем та роль технологій Інтернету речей у їх становленні. Показано, що процеси синергії досягнень в галузі CPS та IoT дозволяють вирішувати комплексні питання сучасної промисловості і гуманітарної сфери. Вони є основою розвитку технологій IoE, SNSS, та самоорганізованих кібернетичних систем. Запропоновано один з підходів для системного аналізу і синтезу сучасних CPS та обґрунтовано визначальну роль у цьому процесі технологій IoT. Розглянуто і проаналізовано моделі CPS та IoT та можливості їх удосконалення використовуючи методи системного аналізу, мереж Петрі, ідеологію систем масового обслуговування.

Проблема аналізу та синтезу CPS і місце IoT технологій у цьому процесі розглядається у 13 розділі з точки зору трьох аспектів: аналізу сучасної елементної бази для розробки кіберкомпоненти, забезпечення надійного інтерфейсу між компонентами системи на всіх рівнях концептуальної моделі, та можливостей програмного забезпечення для моделювання і синтезу CPS. Матеріал викладено у вигляді короткого огляду для розуміння загального підходу до даної проблеми.

Розділ 14 аналізує концепцію Power over Ethernet (PoE) та методи передавання даних та енергії через спільне середовище, а також використання нейромереж для децентралізованої та віддаленої обробки даних. Для цього використовується свіч Cisco Catalyst 4507R+E, засоби хмарного та бортового комп’ютингу, щоб реалізувати легко масштабовну та адаптовну архітектуру з можливістю модульної інтеграції з існуючими рішеннями. Прототип був протестований на мікроконтролері RaspberryPi у якості сенсорного хаба і DigitalOcean як хмарного комп’ютингового сервісу. Чип Movidius Neural Compute використаний для розгортання нейромережі і релевантної обробки даних та глибокого навчання (Deep Learning). Запропонована архітектура показала переваги за рахунок гнучкості створення модульних систем і можливості поєднання концепцій IoT, PoE та нейромереж. Ці принципи можуть бути використано для розроблення кіберфізичних систем для різних індустріальних і гуманітарних застосунків.

Розділ 15 присвячений моделе-орієнтованому проектуванню кіберфізичних систем з застосуванням в рамках цього підходу двох методологій. Перша методологія базується на профілі MARTE мови UML, який призначений для моделювання вбудованих систем реального часу. Друга - використовує мову SysML, яка дозволяє моделювати фізичні та обчислювальні частини кіберфізичних систем. Поєднання цих методологій надає можливість всебічно моделювати і аналізувати функціональні і нефункціональні властивості кіберфізичних систем.

Аннотация

УДК 62:004=111

I73

Рецензенты: Dr. Mario Fusani, ISTI-CNR, Пиза, Италия
Dr. Olga Kordas, KTH University, Стокгольм, Швеция
Viktor Kordas, KTH University, Стокгольм, Швеция

I73 Интернет вещей для индустриальных и гуманитарных приложений. В трех томах. Том 1. Основы и технологии /
Под ред. В. С. Харченко. – Министерство образования и науки Украины, Национальный аэрокосмический университет ХАИ, 2019. – 605 с.

ISBN 978-617-7361-81-6

Книга, состоящая из трех томов, содержит теоретические материалы для лекций и тренингов, разработанных в рамках проекта Internet of Things: Emerging Curriculum for Industry and Human Applications /ALIOT, 573818-EPP-1-2016-1-UK-EPPKA2- CBHE-JP, 2016-2019, финансируемого программой ЕС ERASMUS +. Том 1 описывает проблемы, принципы и технологии Интернета вещей (IoT) и Интернета всего (IoE). Книга состоит из 4 частей для соответствующих магистрских курсов: основы IoT, IoE и Web of Things (разделы 1-4), наука о данных для IoT (разделы 5-8), мобильные и гибридные IoT вычисления (разделы 9-11), IoT технологии для киберфизических систем (разделы 12-15).

Книга подготовлена украинскими университетскими командами при поддержке коллег из академических организаций стран ЕС, входящих в консорциум ALIOT.

Книга предназначена для магистрантов и аспирантов, изучающих технологии IoT, программную и компьютерную инженерию, компьютерные науки. Может быть полезна для преподавателей университетов и учебных центров, исследователей и разработчиков систем IoT.

Рис.: 172. Ссылок: 606. Таблиц: 30.

Аннотации разделов 1-15

Во вводном разделе проанализирована динамика публикаций и описана необходимость создания новых учебных программ по Интернету вещей (IoT) для получения степени магистра, доктора наук и после-университетского инженерного образования. Представлен совместный проект по разработке учебных программ ALIOT, финансируемый в рамках программы Erasmus+. Проект обеспечивает адаптацию академических программ в Украине и других странах к потребностям европейского рынка труда. ALIOT охватывает актуальные области приложений IoT, такие как системы здравоохранения, интеллектуальные транспортные системы, системы экологии и промышленности 4.0, интеллектуальные здания и города. Представлено описание магистерских и аспирантских программ и тренинг-курсов.

В разделе 1 анализируется концепция IoT, задачи, методологические проблемы и решения в области применения IoT, предлагается расширенная концепция IoT. В нем рассматриваются методологические и практические вопросы внедрения систем и инструментов BDA и IoT в контексте кибербезопасности и оценки и обеспечения безопасности. Анализируются концепции безопасности и кибербезопасности с учетом атрибутов безопасности и кибербезопасности для критически важных приложений. Обсуждаются преимущества и ограничения применения технологий, основанных на BDA и IoT, в критически важных для безопасности системах, включая возможности их использования для предотвращения, мониторинга и минимизации последствий тяжелых аварий. Описаны промышленные примеры, такие как система мониторинга аварий на АЭС на базе IoT (Интернет дронов), система прогнозирования надежности программного обеспечения. Сформулированы рекомендации и ограничения применения BDA и IoT в критических для безопасности системах.

Раздел 2 посвящен обзору технологий IoT/WoT, учебной программе для будущих специалистов по разработке технологий

IoT/WoT и опыту преподавания дисциплины: «Технологии и средства разработки приложений WoT» в разных вариациях. Обзор существующих технологий WoT/IoT показывает, что серверным языком программирования интерфейса Web Thing API, построенного с учетом архитектуры REST, является JS в среде Node.JS. С учетом выбранного языка программирования Web Thing API, в учебном плане предлагаем набор технологий для разработки WoT приложений с использованием JavaScript/Node.JS. Дисциплина «Технологии и средства для разработки веб-приложений», дополненная разделами «Облачные вычисления», «IoT/WoT» и соответствующей учебной программой, преподается в НТУ «ХПИ» на кафедре «системы информации» и в НАУ «ХАИ» на кафедре компьютерных систем, сетей и кибербезопасности по курсам «Основы Интернета вещей» и «Промышленный Интернет вещей».

Интернет вещей активно охватывает различные сферы человеческой деятельности. В разделе 3 был проведен анализ доступных решений и производительности в отношении систем IoT. Были проанализированы стандарты и рекомендации в области архитектуры, безопасности, технологий IoT-систем. Рассмотрены особенности презентации моделей систем Интернет вещей, требования к их организации. Проведен анализ и описаны основные метрики, применимые для оценки критериев систем Интернета вещей. Рассмотрены особенности доменов Интернета вещей.

В разделе 4 рассматриваются методы коммуникации в области интернета вещей, при этом особое внимание уделяется используемым протоколам прикладного уровня. Исследуются такие вопросы, как архитектура сетей, задержки в беспроводных сенсорных сетях, возможности стандарта Bluetooth 5.0 и облачная архитектура для интернета вещей. Проведен сравнительный анализ протоколов HTTP/HTTPS, MQTT, MQTT-SN, AMQP, XMPP, DDS и CoAP. Оценены их преимущества и недостатки. Рассмотрены методы определения эффективной скорости помехоустойчивых кодов и сравнения их скорости, а также комплексный показатель энергоэффективности помехоустойчивых кодов для устройств IoT.

Раздел 5 представляет краткое введение в науку о данных для IoT. Предоставляется обзор характеристик данных, соответствующих подходов и методов и алгоритмов для обработки данных, применимых к данным IoT. IoT и IoE экосистемы. Обсуждаются модели научной аналитики, используемые в вертикалях IoT, а также слияние и обработка данных с устройств IoT. Одна из основных целей этой главы - дать представление о том, как данные IoT с устройств - от датчиков до конечных устройств можно извлекать и анализировать для раскрытия информации.

В разделе 6 рассмотрены принципы и технологии интеллектуального анализа и обработки данных для IoT систем. Обсуждаются особенности использования интеллектуального анализа данных для IoT данных, модели и методы интеллектуального анализа данных для IoT, интеллектуальный анализу потоков данных и массивных наборов данных.

В разделе 7 представлены обзор развития и практические достижения на пересечении технологий Интернета вещей и искусственного интеллекта. Подробно описаны принципы функционирования глубоких нейронных сетей, приводятся примеры их работы, основные аспекты проектирования, использования и внедрения для Интернета вещей.

В разделе 8 рассматриваются вопросы хранения больших данных в контексте IoT-систем, а также исследуется взаимосвязь между согласованностью данных и временными задержками в распределенных базах данных NoSQL. Основное внимание в работе уделяется исследованию зависимости производительностью (временем отклика и пропускной способностью) нереляционной базы данных Cassandra и настройками согласованности. В разделе представлены результаты анализа производительности (скорости выполнения операций чтения и записи) реплицированного кластера Cassandra, развернутого в облачной среде Amazon EC2. Представлены количественные результаты, которые позволяют оценить влияние различные настройки согласованности данных на производительность Cassandra при разных рабочих нагрузках. В заключении предложена методика повышения производительности Cassandra на основе оптимизации параметров согласованности с учетом процентного соотношения операций чтения и записи и их интенсивности.

Раздел 9 представляет базовую информацию по разработке приложений для мобильных операционных систем Android и iOS, а также совместимых с ними приложений для носимых устройств. Так как книга предназначения для магистров, аспирантов, а также инженеров из области информационных технологий, текущий раздел покрывает множество тем начиная с рассмотрения современных стандартов применимых в мобильной и IoT индустриях, основ проектирования пользовательских интерфейсов, анализа рекомендуемых Google и Apple архитектур для мобильных приложений и заканчивая подробной инструкцией о загрузке приложения на два основных мобильных маркета – App Store и Google Play.

В разделе 10 представлена информация о слиянии двух передовых технологий – Cloud компьютинга и Интернета вещей, и как его результаты могут быть применены в области мобильной разработки. Так как учебный материал рассчитан на магистров, аспирантов, а также опытных инженеров из ИТ отрасли, в данном разделе приведен анализ современной экономики Cloud сервисов, рассмотрены основные архитектуры и инфраструктуры которые могут быть использованы в процессе разработки мобильных приложений.

В 11 разделе описывается интеграция двух наиболее обсуждаемых сегодня концепций информационных технологий: больших данных и Интернет вещей. Поскольку эта книга предназначена для магистрантов, аспирантов и инженеров, которые будут участвовать в проектировании и разработке таких интегрированных проектов, представлен технический обзор области больших данных с точки зрения IoT.

В разделе 12 приведены результаты анализа современного состояния развития киберфизических систем (КФС) и роль технологий Интернета вещей в их становлении. Показано, что процессы синергии достижений в области КФС и IoT позволяют решать комплексные вопросы современной промышленности и гуманитарной сферы. Они являются основой развития технологий IoE, SNSS, и самоорганизующихся кибернетических систем. Предложен один из подходов для системного анализа и синтеза современных КФС и обоснована определяющая роль в этом процессе технологий IoT. Рассмотрены и проанализированы модели КФС и IoT и возможности их совершенствования с использованием методов системного анализа, сетей Петри, идеологии систем массового обслуживания.

Проблемы анализа и разработки КФС и принципы их решения описываются в 13 разделе. Место IoT технологий в этом процессе рассматривается с точки зрения: анализа современной элементной базы для разработки киберкомпонент, обеспечения надежного интерфейса между компонентами системы на всех уровнях концептуальной модели, создания и использования программного обеспечения для моделирования и синтеза КФС.

Раздел 14 обсуждает концепцию Power over Ethernet (PoE) и методы совместной передачи данных и энергии через общую среду, а также использования нейросетей для децентрализованной и удаленной обработки данных. Для этого используются свич Cisco Catalyst 4507R+E, средства облачного и бортового компьютинга, чтобы реализовать легко масштабируемую и адаптрируемую архитектуру с возможностью модульной интеграции в существующие решения. Прототип был протестирован на микроконтроллере RaspberryPi в качестве сенсорного хаба и DigitalOcean как облачного компьютингового сервиса. Чип Movidius Neural Compute был использован, чтобы развернуть нейросеть для релевантной обработки данных и глубокого обучения (Deep Learning). Предложенная архитектура показала преимущества в гибкости создания модульных систем и возможности объединения концепций IoT, PoE и нейросетей. Эти принципы могут быть использованы для разработки киберфизических систем для различных индустриальных и гуманитарных приложений.

Раздел 15 посвящен моделе-ориентированному проектированию киберфизических систем с применением двух методологий. Первая методология основана на профиле MARTE языка UML, который предназначен для моделирования встроенных систем реального времени. Вторая использует язык SysML, который позволяет моделировать физические и вычислительные части киберфизических систем. Совместное использование этих методологий дает возможность всесторонне моделировать и анализировать функциональные и нефункциональные свойства КФС.

Тетяна Олександрівна Білобородова, Артем Володимирович Боярчук,
Євгеній Віталійович Брежнєв, Дмитро Анатолійович Бутенко,
Валентина Олегівна Бутенко, Олександр Анатолійович Чеміріс,
Валентина Еммануїлівна Гордиця, Сергій Яковлевич Гільгурт,
Анатолій Вікторович Горбенко, Олег Олександрович Ілляшенко,
Вячеслав Сергійович Харченко, Марина Олександрівна Колісник,
Мирослав Петрович Комар, А-Ліан Кор, Василь Сергійович Коваль,
Равіль Камілович Кудерметов, Іван Михайлович Лобачев,
Михайло Вікторович Лобачев, Дмитро Андрійович Маєвський,
Олена Борисівна Одарущенко, Олег Миколайович Одарущенко,
Володимир Яковлевич Пєвнєв, Кріс Філліпс, Анатолій Павлович Плахтеєв,
Анджей Русинські, Інна Сергіївна Скарга-Бандурова,
Олексій Юрійович Стрюк, Ольга Михайлівна Тарасюк,
Володимир Антонович Ткаченко, Михайло Віталійович Цуранов,
Георгій Іванович Воробець, Олександр Іванович Воробець,
Лілія В'ячеславівна Висторобська

Інтернет речей для індустріальних і гуманітарних застосунків.
Том 1. Основи і технології

(англійською мовою)

Редактор *Харченко В.С.*
Комп'ютерна верстка *Ілляшенко О.О.*

Зв. план, 2019
Підписаний до друку 22.08.2019
Формат 60x84 1/16. Папір офс. No2. Офс. друк.
Умов. друк. арк. 35,17. Обл.-вид. л. 37,80. Наклад 150 прим.
Замовлення 220819_1

Національний аерокосмічний університет ім. М. Є. Жуковського
"Харківський авіаційний інститут"
61070, Харків-70, вул. Чкалова, 17
http://www.khai.edu

Випускаючий редактор: ФОП Голембовська О.О.
03049, Київ, Повітрофлотський пр-кт, б. 3, к. 32.
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців,
виготовлювачів і розповсюджувачів видавничої продукції
серія ДК No 5120 від 08.06.2016 р.

Видавець: ТОВ «Видавництво «Юстон»
01034, м. Київ, вул.. О. Гончара, 36-а, тел.: +38 044 360 22 66
www.yuston.com.ua
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців,
виготовлювачів і розповсюджувачів видавничої продукції
серія ДК No 497 від 09.09.2015 р.