Zbornik 19. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2016

Zvezek C

Proceedings of the 19th International Multiconference

INFORMATION SOCIETY - IS 2016

Volume C

Sodelovanje, programska oprema in storitve v informacijski družbi

Collaboration, Software and Services in Information Society

Uredil / Edited by
Marjan Heričko

http://is.ijs.si

10. oktober 2016 / 10 October 2016
Ljubljana, Slovenia

Zbornik 19. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2016

Zvezek C

Proceedings of the 19th International Multiconference

INFORMATION SOCIETY – IS 2016

Volume C

Sodelovanje, programska oprema in storitve v informacijski družbi
Collaboration, Software and Services in Information Society

Uredil / Edited by

Marjan Heričko

10. oktober 2016 / 10 October 2016
Ljubljana, Slovenia

Urednik:

Marjan Heričko
University of Maribor
Faculty of Electrical Engineering and Computer Science

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2016

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.77(082)(0.034.2)

MEDNARODNA multikonferenca Informacijska družba (19 ; 2016 ; Ljubljana)
Sodelovanje, programska oprema in storitve v informacijski družbi [Elektronski vir] : zbornik 19. mednarodne multikonference Informacijska družba - IS 2016, 10. oktober 2016, [Ljubljana, Slovenija] : zvezek C = Collaboration, software and services in information society : proceedings of the 19th International Multiconference Information Society - IS 2016, 10 October 2016, Ljubljana, Slovenia : volume C / uredil, edited by Marjan Heričko. - El. zbornik. - Ljubljana : Institut Jožef Stefan, 2016

ISBN 978-961-264-099-6 (pdf)
1. Gl. stv. nasl. 2. Vzp. stv. nasl. 3. Dodat. nasl. 4. Heričko, Marjan
287010304

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2016

Multikonferenca Informacijska družba (http://is.ijs.si) je z devetnajsto zaporedno prireditvijo osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Letošnja prireditev je ponovno na več lokacijah, osrednji dogodki pa so na Institutu »Jožef Stefan«.

Informacijska družba, znanje in umetna inteligenca so spet na razpotju tako same zase kot glede vpliva na človeški razvoj. Se bo eksponentna rast elektronike po Moorovem zakonu nadaljevala ali stagnirala? Bo umetna inteligenca nadaljevala svoj neverjetni razvoj in premagovala ljudi na čedalje več področjih in s tem omogočila razcvet civilizacije, ali pa bo eksponentna rast prebivalstva zlasti v Afriki povzročila zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so planetarni konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc. Predstavljenih bo okoli 200 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica, ki se ponaša z 39-letno tradicijo odlične znanstvene revije. Naslednje leto bo torej konferenca praznovala 20 let in revija 40 let, kar je za področje informacijske družbe častitljiv dosežek.

Multikonferenco Informacijska družba 2016 sestavljajo naslednje samostojne konference:

- 25-letnica prve internetne povezave v Sloveniji
- Slovenska konferenca o umetni inteligenci
- Kognitivna znanost
- Izkopavanje znanja in podatkovna skladišča
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Vzgoja in izobraževanje v informacijski družbi
- Delavnica »EM-zdravje«
- Delavnica »E-heritage«
- Tretja študentska računalniška konferenca
- Računalništvo in informatika: včeraj za jutri
- Interakcija človek-računalnik v informacijski družbi
- Uporabno teoretično računalništvo (MATCOS 2016).

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, SLAIS, DKZ in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in inštitucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V 2016 bomo četrtič podelili nagrado za življenjske dosežke v čast Donalda Michija in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel prof. dr. Tomaž Pisanski. Priznanje za dosežek leta bo pripadlo prof. dr. Blažu Zupanu. Že šestič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobilo ponovno padanje Slovenije na lestvicah informacijske družbe, jagodo pa informacijska podpora Pediatrične klinike. Čestitke nagrajencem!

Bojan Orel, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2016

In its 19th year, the Information Society Multiconference (http://is.ijs.si) remains one of the leading conferences in Central Europe devoted to information society, computer science and informatics. In 2016 it is organized at various locations, with the main events at the Jožef Stefan Institute.

The pace of progress of information society, knowledge and artificial intelligence is speeding up, but it seems we are again at a turning point. Will the progress of electronics continue according to the Moore's law or will it start stagnating? Will AI continue to outperform humans at more and more activities and in this way enable the predicted unseen human progress, or will the growth of human population in particular in Africa cause global decline? Both extremes seem more and more likely – fantastic human progress and planetary decline caused by humans destroying our environment and each other.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers at twelve conferences, round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal, which has 39 years of tradition of excellent research publication. Next year, the conference will celebrate 20 years and the journal 40 years – a remarkable achievement.

The Information Society 2016 Multiconference consists of the following conferences:

- 25th Anniversary of First Internet Connection in Slovenia
- Slovenian Conference on Artificial Intelligence
- Cognitive Science
- Data Mining and Data Warehouses
- Collaboration, Software and Services in Information Society
- Education in Information Society
- Workshop Electronic and Mobile Health
- Workshop »E-heritage«
- 3st Student Computer Science Research Conference
- Computer Science and Informatics: Yesterday for Tomorrow
- Human-Computer Interaction in Information Society
- Middle-European Conference on Applied Theoretical Computer Science (Matcos 2016)

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, SLAIS, DKZ and the second national engineering academy, the Slovenian Engineering Academy. In the name of the conference organizers we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the fourth year, the award for life-long outstanding contributions will be delivered in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to Prof. Tomaž Pisanski for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to Prof. Blaž Zupan. The information lemon goes to another fall in the Slovenian international ratings on information society, while the information strawberry is awarded for the information system at the Pediatric Clinic. Congratulations!

Bojan Orel, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Robert Blatnik
Aleš Tavčar
Blaž Mahnič
Jure Šorn
Mario Konecki

Programme Committee

Bojan Orel, chair
Nikolaj Zimic, co-chair
Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič Grega
Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah

KAZALO / TABLE OF CONTENTS

Zbornik 19. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2016

Zvezek C

Proceedings of the 19th International Multiconference

INFORMATION SOCIETY – IS 2016

Volume C

Sodelovanje, programska oprema in storitve v informacijski družbi
Collaboration, Software and Services in Information Society

Uredil / Edited by

Marjan Heričko

10. oktober 2016 / 10 October 2016
Ljubljana, Slovenia

PREDGOVOR

Konferenco “Sodelovanje, programska oprema in storitve v informacijski družbi” organiziramo v sklopu multikonference Informacijska družba že šestnajstič. Kot običajno, tudi letošnji prispevki naslavljajo aktualne teme in izzive, povezane z razvojem sodobnih programskih in informacijskih rešitev ter storitev.

Sprejem in uspešna uporaba na informacijskih tehnologijah temelječih storitev je v veliki meri odvisna od njihove kakovosti, kar vključuje tudi skrb za zaščito zasebnosti in zaupnosti osebnih podatkov, ki se uporabljajo pri za zagotavljanju uporabnikom prilagojenih storitev. Agilni pristopi in uporabniško naravnan razvoj dodatno prispevata k boljši uporabniški izkušnji. Prispevki, zbrani v tem zborniku, omogočajo vpogled v in rešitve za izzive na področjih kot so npr.:

- varovanje zasebnosti pri zunanjem izvajanju v informatiki;
- metode in tehnike anonimizacije geo-lokacijskih podatkov;
- hranjenje in obdelava podatkov na mobilnih napravah;
- kulturni, sociološki in formalin izzivi pri integraciji podatkovnih virov;
- analiza in napovedovanje ranljivosti v programski opremi;
- kriptografski algoritmi in računalništvo v oblaku;
- statična analiza kakovosti kode na osnovi konsistentne predstavitve programskih sistemov;
- izbira primernih agilnih pristopov in metod;
- nestrukturirano modeliranje primerov in notacija CMMN.

Upamo, da boste v zborniku prispevkov, ki povezujejo teoretična in praktična znanja, našli koristne informacije za svoje nadaljnje delo tako pri temeljnem kot aplikativnem raziskovanju.

FOREWORD

This year, the Conference "Collaboration, Software and Services in Information Society" is being organised for the sixteenth time as a part of the "Information Society" multi-conference. As in previous years, the papers from this year's proceedings address actual challenges and best practices related to the development of advanced software and information solutions.

The acceptance and success of advanced ICT-based services depends heavily on their quality, including their ability to protect privacy and confidentiality of personal data which are used to provide better services to end-users. User-centric and agile development approaches can also contribute significantly to improved user experience, whereas efficient quality assurance should not be limited to specific programming paradigms and platforms. Papers in these proceedings provide a better insight and/or propose solutions to challenges related to:

- Information privacy in IT/IS outsourcing;
- Methods and techniques for geolocation data anonymization;
- Data storage and processing on mobile devices;
- Cultural, social and legal issues in data integration;
- Software vulnerability prediction;
- Cryptography issues caused by cloud computing;
- Consistent representation of software systems to apply software quality static analysis;
- Selection of suitable agile method(s);
- Case management modelling and notation.

We hope that these proceedings will be beneficial for your reference and that the information in this volume will be useful for further advancements in both research and industry.

Prof. Dr. Marjan Heričko
CSS 2016 – Collaboration, Software and Services in Information Society Conference Chair

PROGRAMSKI ODBOR / PROGRAM COMITTEE

Dr. Marjan Heričko
University of Maribor, Faculty of Electrical Engineering and Computer Science

Dr. Ivan Rozman
University of Maribor, Faculty of Electrical Engineering and Computer Science

Dr. Lorna Uden
Staffordshire University, Faculty of Computing, Engineering and Technology

Dr. Gabriele Gianini
University of Milano, Faculty of Mathematical, Physical and Natural Sciences

Dr. Hannu Jaakkola
Tampere University of Technology Information Technology (Pori)

Dr. Mirjana Ivanović
University of Novi Sad, Faculty of Science, Department of Mathematics and Informatics

Dr. Zoltán Porkoláb
Eötvös Loránd University, Faculty of Informatics

Dr. Aleš Živkovič
Innopolis University, Faculty of Computer Science

Dr. Boštjan Šumak
University of Maribor, Faculty of Electrical Engineering and Computer Science

Dr. Gregor Polančič
University of Maribor, Faculty of Electrical Engineering and Computer Science

Dr. Luka Pavlič
University of Maribor, Faculty of Electrical Engineering and Computer Science

Information Privacy and Information Technology Outsourcing

Domen Verber
Faculty of Electrical Engineering and
Computer Science,
University of Maribor
Maribor, Slovenia
+386 2 220 7434
domen.verber@um.si

ABSTRACT

In this paper we discuss the issue of information privacy in the regime of Information Technology Outsourcing (ITO). Nowadays, privacy in general, and information privacy in particular, is a very important and much debated issue. With ITO the companies contracted out IT infrastructure and/or IT related services, such as programing, to other companies. By doing this the responsibility for information privacy is shared by several parties. The owner of the data must protect the privacy of their customers even in the case of ITO. However, this may be in contradiction with the need for efficient utilization of data and may hinder the proper software development, testing and maintenance.

The paper contains a short introduction to information privacy and presents some real-world case studies related to this topic.

Categories and Subject Descriptors

K.4.1. [**Computers and Society**]: Public Policy Issues – *Privacy*
D.2.9. [**Software**]: Management - *Programming teams*

General Terms

Management, Performance, Security, Human Factors, Legal Aspects

Keywords

Information privacy, data protection, anonymization, information technology outsourcing.

1. INTRODUCTION

The rapid growth of the Internet, ever increasing number of mobile phones and smart devices, coupled with the new business practices, have raised far-reaching questions about the future of privacy. Computers and applications track us in almost everything we do. Data are collected when we click on some link in the web browser. With the support of our Loyalty Cards, the grocery store collects information about what we are buying. Data are stored about us in some medical records, financial records, school records, etc. In most cases, those data can be beneficial to us. Modern personal recommender systems can be very efficient and helpful, data records can speed-up the utilization of services, the doctors can devise better diagnostics, etc. However, some of those data are intimate to us and we do not want to reveal them to unauthorized companies and persons.

The paper discusses the issue of information privacy in the modern IT landscape [1]. Most companies today employ some sort of outsourcing (and offshoring) to reduce the costs. With the Information Technology Outsourcing, the companies are contracting out their IT infrastructure and/or IT related services, such as programing, to other companies. As a consequence, at least some of the data stored in the datacentres of the primary company, must be shared with the other IT providers. This raises the question of responsibilities for information protection and increases the complexity of assuring it significantly.

The term information privacy is strongly related to the Data Protection. The latter represents a much wider concept and is discussed only briefly.

In the first part of the paper some basic introduction to information privacy is given. In the second part some challenges are presented for assuring information privacy in the context of ITO, software development and maintenance. In the third part two case studies are demonstrated related to the practical implementation of information privacy.

2. INFORMATION PRIVACY

2.1 Introduction to information privacy

Information privacy (or data privacy) considers the relationship between the collection and dissemination of data. Information privacy is a part of a broader term, data protection, which is the process of safeguarding important information from corruption, unwanted exploitation and/or loss. In most cases, information privacy is related to personally identifiable information in combination with other attributes, such as financial records, medical history, religion and beliefs, shipping habits, web surfing behavior, etc. There are other sensitive data related with the business processes of some companies that must be protected also. Those are trade contracts, financial transactions and similar, made by other companies and persons.

Information privacy involves data storage and data processing technologies, and the public, legal and political issues surrounding them. Privacy concerns extend through the entire life-span of some information. It considers how such information is collected, stored, used and destroyed, either in digital or some other form.

Most countries have derived strict laws that protect personal privacy [3]. The EU Data Protection Regulation [4] promotes two main principles of data privacy: Privacy by design and privacy by default. Privacy by design means that each new service or product that makes use of personal data must take the protection of such data into consideration. IT developers must take privacy into account during the whole life cycle of the system or process development. Privacy by Default means that the strictest privacy settings apply automatically once a customer acquires a new product or service. No manual change to the privacy settings should be required on the part of the user. There is also a temporal element to this principle, as personal information must, by default, only be kept for the amount of time necessary to provide the product or service. Slovenia adopted most of the EU Regulations and has very progressive policies regarding information privacy.

2.2 Information privacy and IT solutions

The main challenge of data privacy is how to maximize the utilization of data while protecting personally identifiable information. For example, the end user wish to have access to the list of customers with their Personal Identification Numbers. This can be very convenient for unique identification of a person. However, the Personal Identification Numbers are considered as private information because they can reveal his or her birthdate and gender.

To maintain information privacy, first, we need to assure the data security. All potential measures to protect the privacy are useless if the data can be accessed by unauthorized parties. We need to consider all software, hardware and human resources to address this issue and implement the proper actions. The human resources are the most difficult to consider. A frustrated Data Administrator may expose the data to the public or even to some criminal group. We must also contemplate the employees and the end-users, who may, unintentionally or intentionally, expose the private information for no justifiable reason. It is the responsibility of the IT company to minimize such risks.

Nowadays, it is taken for granted that the data can be accessed everywhere: From the web, from mobile devices and remotely from personal computers. This presents an additional challenge. We must contemplate different scenarios to maintain information privacy and data security. The devices can be lost or stolen, the communication channels can be eavesdropped, a badly implemented web application can be hacked, etc. All sensitive data (not only the private ones) should be stored and transferred on the communication channels in encrypted form. By this, the data is protected if it is stolen or accessed unwarrantedly by the administrative personnel.

The laws and regulations related to information privacy and data protection are changing constantly. Therefore, the IT solution providers must reassess the compliance with information privacy and other security regulations continually. This may be difficult for applications that have already been in use for a long time and cannot be replaced or adapted easily.

2.3 Basic techniques for assuring the information privacy

The best approach to information privacy is to minimize the number of situations where privacy can be breached. To achieve this, we must remove all sensitive data from the basic parts of the user interface and show them only with the explicit request from the user. One example is the usage of the Personal Identification Number mentioned above. If possible, any personalized identification should be replaced by computer generated identification. With modern automatic identification techniques (e.g. RFID cards, bar codes, etc.), it is possible to implement seamless data identification and speed-up the processing of data without the need to expose some personal identification codes. However, if this is not possible, or if this can slow-down the business processes significantly, it is better to keep some of sensitive data available to the authorized end-user and make some obligatory contract with them to maintain the privacy.

A well-established practice in information privacy is Auditing. The software solution must keep track of who has accessed the sensitive data and when and whose data was seen. In some cases, the user must enter the reason why the data was accessed. Although this sounds reasonable, it is almost impossible to implement it entirely. For example, within the modern user interface, the end-user may see a list of several objects related with the sensitive data at the same time. However, some of the data can be seen only if the list is scrolled. It would be very cumbersome to implement the proper auditing in this case. The alternative is to show the data one by one, but this would diminish the user experience. Sensitive data can also be printed out or exported. In this case it is impossible to track the users with access. The end-user should be aware of Auditing. This would prevent any unnecessary and unwarranted access of the data. To enter a reason every time the data is accessed is not always practical and would slow-down the business process scientifically in most cases. Again, this can be avoided with proper authentication and authorization.

Data export and external access to the data with sensitive information should be made only with trusted parties and with clear and justifiable intention. We cannot track what happens to the data outside of our system. The printing of sensitive data should be forbidden or limited to obligatory documents. All such operations must be audited properly.

Both employees and the end-user should be educated about information privacy and the data security. Most confidentiality breaches are made unintentionally by the users who were not aware of the Regulations and the significance of information privacy.

3. INFORMATION PRIVACY WITHIN SOFTWARE DEVELOPMENT AND MAINTENANCE IN ITO

3.1 Information Technology Outsourcing

In general, outsourcing involves the contracting out of a business process and/or the assets to another party or company. One kind of business outsourcing is Information Technology Outsourcing (ITO), which is a company's outsourcing of its IT infrastructure and/or IT related services, such as programing, to other companies. With the expansion of Cloud Computing in recent years it has become more and more popular for the companies to transfer their IT infrastructure to the Cloud. This would reduce the costs of hardware and administrative personnel. Nowadays, the Cloud suppliers provide strong data protection. However, several controversial cases where other parties and even governments have access to the data has slowed down the migration. Most of the medium and large sized companies today still try to maintain their own datacentres.

The trend of ITO is also observed with the in-house software development. A lot of companies have reduced or even eliminated their IT departments and contracted them out to third parties.

There exist several models of ITO. In some cases, the company has no IT Department at all. In this case, the IT solution provider serves as the sole service provider. It maintains the software and the hardware, performs all the backups, trains the end-users, etc. In most cases, the companies maintain a small IT Department which is responsible for the smooth running of software and hardware in-house. The IT solution provider, in addition to maintaining the software, is responsible for all off-premise assets. Some companies have large IT Departments which maintain the hardware, and may run their own applications in parallel with several outsourced solution providers.

3.2 Data sharing and information privacy

Most business processes today rely on the acquisition and processing of some data. In the most common scenario, some sort of relational database is used. When the software development is outsourced, some of this data must be available to the IT solution providers. This may be in conflict with the requirements for the data protection and information privacy. Data protection can be sustained with well-established techniques (e.g. firewalls, VPN communication channels, etc.). This is in the main interest of both the owner of the data and the IT company. On the other hand, it is much more difficult to maintain the information privacy. For proper software development the IT company possesses the elevated priority level to access the data directly, although some security measures can be avoided easily with the customized version of the applications, etc. The owner of the data has no guarantee that the information would not be used improperly or even sold off.

Data anonymization and data obfuscation can be used to tackle this problem [5]. It is common practice that a copy of the database is used for the development, testing and training. In this database, all sensitive data can be replaced with some arbitrary content. For better data protection, the copy of the database and anonymization is performed by the owner; the outsourced solution providers have no direct access to the originals. However, there are some drawbacks to this scheme. Firstly, some faults in the application can be related to the content of the original data and may not be duplicated easily in the test environment. Secondly, in comparison, to copy the database as a whole, data anonymization can be a challenging and time consuming task, especially for the large data sets. The data size required for some tables can be duplicated if the data base system integrates some sort of change traces, and thirdly, it is almost impossible to assure complete anonymization. From the secondary attributes, and with some social engineering, it is possible to reconstruct the identity of a subject.

4. CASE STUDIES

In this section we represent real-world examples of IT solutions where information privacy plays a significant role. For more than 25 years we were employed as an external solution provider to small and medium sized companies in this part of Europe. In most case we were the sole solution provider for almost all software solutions and with full administrative access to the data. In this case we have also envisioned the role of information privacy and have implemented all the measures by ourselves. At first, these were “common-sense” rules. Later on, we tried to comply with all the suggestions and requirements of the personal data protection legislation of Slovenia and the EU. At first, this was also true for the two examples presented here. However, several years ago, our clients became more aware of the issues of information privacy and we upgraded our cooperation to a higher level.

4.1 Academic Information System

The Academic Information System is responsible for the smooth implementation of academic activity in the university. It allows the academic community, university staff and public to access a wide range of information. Among them it keeps all the records of the students and their marks. Some personal attributes and the marks are considered personal data and should be protected from unattained disclosure.

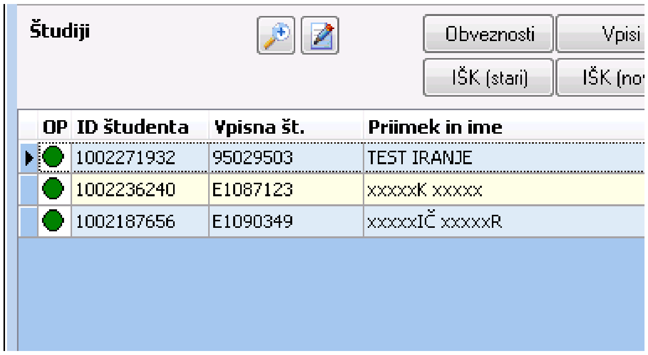
Each student has access to his or her own data. They may change some of the secondary personal attributes and some preferences. The teaching staff (subject leaders and their assistants) has direct access to the application forms of the exams for which they are responsible. For security reasons, their data are recorded separately and transferred into the student records at the end of the exam. Only the subject leader can do that

The university staff has full access and can modify all personal attributes for the students of a related Faculty and read-only access to some of the attributes of the other students.

At first, the unique personal number was one of the primary keys that identified a student as a person. It was (and still is) one of the attributes that is shown on the lists user interfaces. There was an idea to hide this attribute entirely; however, this would slow down some processes. Recently, the unique personal number of the students was replaced with a synthetic student identification number. The identification procedure was automated with the RFID cards

Once entered, the private attributes of the student can be changed only explicitly and the reason must be presented. All such changes are audited. In addition, all print-outs and exports where personal data of the students are presented are also recorded. The users have access to the audit trails. However, only the administrators can retrieve the details (e.g., to see which attributes were changed).

The development and testing of the applications is performed on a testing databased with some of the personal data obfuscated. This is demonstrated in Figure 1.



OP	ID študenta	Vpisna št.	Priimek in ime
●	1002271932	95029503	TEST IRANJE
●	1002236240	E1087123	xxxxxK xxxxx
●	1002187656	E1090349	xxxxxIČ xxxxxR

Figure 1. Obfuscated list of students in the test database.

The same database is used for the training and the testing performed by the end-users. The primary intention of obfuscation is to prevent unintentional exposure of personal data and not to isolate the IT provider form the client. If necessary, the tests can still be performed on the primary database.

4.2 Financial information subsystem of a bank

The financial information subsystem is used for tracking financial and other related transactions of a company. It provides bookkeeping, reposting, data analyze and other information of the financial records of a client of the company. A bank, as any other company, is also obliged to keep these records. Furthermore, there are some unique functionalities specific to the banks. Here, the private information is not some personal attributes of persons but the identities of those financial clients.

For obvious reasons, the banks have very strict data protection measures. As in the example above, all the development and testing is done on a separate database with some obfuscated attributes. Despite being a testing database, the outside access to it is protected heavily with time-changing encryption keys and firewalls. We are also isolated from the primary database and have direct access only in some exceptional circumstances. Because of the specifics in the database management system it is not very easy to prepare a copy. There a lot of database triggers that must be switched off during the copying of the data. Because all this is time consuming, the secondary database is updated only occasionally. The primary database also has direct access to some other information subsystems of the bank, which are not available on the test database. Instead of that, the client provided us with the working development environment inside the company that can be used on the production data if needed, under the supervision of their IT staff.

5. CONCLUSION

Privacy is a much debated issue today. IT solutions are obliged to maintain information privacy with the data they process. In the case of IT Outsourcing, the complexity to achieve this is much higher.

The requirements imposed by the information privacy sometimes contradicts the requirements for effective data processing and good user experience. While the strict information privacy is possible, in real-life scenarios it is sometimes better to make some compromises. Furthermore, the strict information privacy cannot be achieved only with the IT means. Probably, the biggest role for this lies in the persons that are involved with the solutions.

6. REFERENCES

[1] Solove D.J., Schwartz, P.M. 2011. *Privacy, Information, and Technology.* Aspen Publishers.

[2] Cullen, S., Lacity M., Willcocks, L.P. 2014. *Outsourcing: All You Need To Know*. White Plume Publishing.

[3] Solove, D.J. 2014. *Information Privacy Law*. Wolters Kluwer Law & Business.

[4] EU data protection regulation. 2016. http://www.eudataprotectionregulation.com/. Visited on: 2016/09/18.

[5] Ferrer, J,D., Sanchez D., Comas, J.S. 2016. *Database Anonymization: Privacy Models, Data Utility, and Microaggregation-based Inter-model Connections*. Morgan & Claypool Publishers.

A survey on geolocation data anonymization

Matija Heričko
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
matija.hericko@student.um.si

Balaji Palanisamy
University of Pittsburgh,
School of Information Sciences,
Pittsburgh, USA
bpalan@pitt.edu

Tatjana Welzer
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
tatjana.welzer@um.si

Marko Hölbl
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
marko.holbl@um.si

Prashant Krishnamurthy
University of Pittsburgh,
School of Information Sciences,
Pittsburgh, USA
prashant@sis.pitt.edu

Vladimir I. Zadorozhny
University of Pittsburgh, School of Information Sciences, Pittsburgh, USA
vladimir@sis.pitt.edu

ABSTRACT

The advancements in positioning technologies and mobile devices have made it possible for location-based services to become very popular, since they provide contextualized information for users depending on their position. Despite the big numbers of users that use these services, many are wary of their risks and have concerns about their privacy. Data anonymization plays an important part in location-based services. Since the services do not have strict regulations, it is up to the data anonymization methods and techniques to protect the users' privacy. In this paper, we present a survey of data anonymization in the context of geolocation and location-based services. We provide an overview of recent work in the research field, summarise the methods, architectures and configurations used in the research and provide some open problems, challenges and direction for further research.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; D.2.8 [Software Engineering]: Metrics complexity measures, performance measures

General Terms

Theory

Keywords

data anonymization, data generalization, location-based services, geolocation data

1. INTRODUCTION

The Internet of Things (IoT) paradigm, where everything and everyone is connected, enables us to witness significant advances in wireless network communication and positioning technologies, such as Wi-Fi, NFC, RFID, 3G/4G network, Bluetooth, etc. Additionally, the new paradigm facilitates devices that support network communication and geo-positioning [1].

These advancements, together with the growth of the network infrastructure, provide an excellent platform for applications which make use of the devices' geolocating ability. We are, therefore, witnessing an increase in location-based services, which use the geo-spatial location information to deliver on-line location enhanced information [2].

Location-based services require users to submit their geolocation along with their query, so that the service can contextualize the response based on the users' location. Examples of some frequently used location-based services are navigation, point of interest application ("where is the nearest ATM?"), traffic alerts, weather information, location-based games, etc. [3, 4].

However, the convenience of these services is accompanied by some security concerns, because of the sensitive nature of the users' location information. If the user wishes to use location-based services he/she must send his/her location along with his/her request (or query) to an untrusted third party server, his/her privacy can be intruded easily. If the server has malicious intent it can easily use the location information for its own malicious actions, or the data can be forwarded or sold to some other third party. Users should be aware of the risks that accompany location-based services and should take steps to protect their privacy [1 ,5, 6].

In this paper we conduct a survey of data anonymization, which is one of the ways to protect user privacy. We survey the field of data anonymization in the context of geolocation data. More specifically, we look into geolocation data that are used by location-based services. With this review of the data anonymization we wish to determine which different methods are available to achieve the required data anonymity level. We also review briefly the different metrics for measuring the achieved level of anonymity and examine the environment in which it is used.

The rest of the paper is structured as follows: In Section 2 we overview and discuss related work, in Section 3 we present current work, as well as methods of achieving user privacy, in Section 4 we discuss some open issues and future research directions of data anonymization in location-based services, and in Section 5 we conclude the paper.

2. RELATED WORK

Data anonymization plays a big role in preserving the privacy of users and is, therefore, often an important security requirement in many different technological areas. Due to the importance of data anonymization, many researchers tackle the problem in different application areas. The difference between areas is the techniques used to achieve data anonymity and the environment in which it is used [7, 8]. In [7] Parmar and Jinwala surveyed the area of wireless sensor networks and they observed the approaches to data aggregation. The objectives of data aggregation in wireless sensor networks are end-to-end privacy preservation and aggregation at intermediate nodes. The technique most used in wireless sensor networks is privacy homomorphism and its variants, which assures privacy and helps with data aggregation, but affects integrity and freshness negatively. They have concluded that data aggregation could possibly be used in cloud computing and that there is need for more protocols that provide integrity and freshness.

Dhand and Tyagi in [9] further reviewed the data techniques to achieve data aggregation in wireless sensor networks. They identified several cluster-based approaches which minimise communication requirements and, at the same time, maximise network lifetime. They have divided the protocols into homogenous and heterogeneous and each of those groups further into single-hop protocols and multi-hop protocols. The authors have concluded that data aggregation extends the network resources, since it lowers the data that needs to be transmitted.

Data anonymization is also an important topic in the field of big data. A survey on big data privacy was done by Vennila and Priyadashini in [8] where big data sets are sent to a cloud. They have observed that traditional privacy models and data anonymization approaches are not applicable to big data sets.

In [10] the authors surveyed the field of location-based wireless services and they classified services based on various attributes. They analysed the usage trends of services, technologies used by the services, protocols and standards used and architecture. They have mapped the requirements with the technical aspects with the purpose of increasing the awareness.

3. DATA ANONYMIZATION METHODS AND TECHNIQUES

Data anonymization in location-based services is used to protect user privacy. User location information is anonymized in such a way that a service cannot infer the users' identity, interests, or any other specific information, but rather the data is so generalised that it can be used to describe a multitude of different users. On the other hand, data should still be specific enough to allow the user to enjoy the benefits and convenience of services contextualized to his/her location [1, 6].

With the wide spread of location-based services that are used daily by many users, data privacy became a big concern as services come with many hidden risks and threats to user privacy. Threats to privacy arise from a multitude of actions, such as the collection of personal information, unauthorised use of personal information, improper access to personal information, bad storage of personal information and other actions similar to or derived from these actions [11, 12].

Privacy is the users' right to have control over how information about him/her is collected, maintained, used, disclosed or shared [13], and we can classify location privacy into microscopic and macroscopic levels [14], where microscopic presents a single user query and macroscopic presents a whole journey with multiple queries. [15] also divides the macroscopic level further into journey-level and long term location privacy. Techniques to achieve location privacy can be divided into three major groups. These are anonymity-based schemes, obfuscation-based and false location or dummy generation-based [15, 16].

The difference between the schemes is that the anonymity-based and obfuscation-based can only provide location privacy for microscopic levels, and the dummy generation-based can provide for both the microscopic and the macroscopic levels. We can also classify the techniques into two categories based on the involved actors [15]. These categories are anonymization server-based (or centralised) schemes and mobile device-based (or decentralised) schemes [1]. As the names imply, the server-based schemes use a trusted server for the anonymization of the data, while the mobile device-based schemes do not use a server to achieve anonymization, but rely on the sharing of information between users [1].

Centralised schemes make use of the trusted anonymizing server to anonymize the query of the user. The server first removes sensitive information about the user (such as the name, age, etc.) and then it anonymizes location information by either cloaking, using dummy locations or confusing the path. The biggest disadvantage of a centralised scheme is that the data is gathered in a single location and, if the server is compromised, all the data that it holds is compromised as well [15, 17, 18, 19, 20]. Decentralised schemes do not use a trusted server to anonymize the queries, instead they use other methods. The most prevalent method uses peer-to-peer communication. In this method the users' device searches for neighbouring devices and uses their location to anonymize its own location information in the query. The biggest disadvantage of these schemes is that a user has to rely on neighbouring devices and, if there are not enough devices nearby, the location information cannot be anonymized. Another drawback is that the computational overhead may be too much for some smaller devices [1, 6].

Anonymity-based techniques try to preserve users' privacy by making his/her query anonymous with the use of different methods. The most popular and important method of these techniques is cloaking. Cloaking is divided further into spatial and spatio-temporal cloaking. Both of these methods make use of a metric called k-anonymity which was first proposed by Sweeney [21]. K-anonymity means that a user cannot be distinguished from (k-1) other users whose data is also in the same data set. Two other metrics that have gained traction recently are entropy-based metric and l-diversity metric. The basis for the entropy based metric is information theory, where the entropy is a measure of uncertainty or unpredictability. This means that for the entropy-based metric measures with what level of certainty can we define the real location among a group of locations? The L-diversity metric is based on a graph theory and it examines the l-neighbourhood graphs and the connections between neighbours to try and determine the user [22].

Anonymity-based techniques are the most researched area, and there are many different variants. Some researchers focus on cloaking of mobile users where the issues are the continuous queries of users and their movements [4, 5, 23, 24, 25, 26, 27, 28, 29]. Others research centralised schemes with a focus on microscopic or snapshot queries, where every query stands alone [15, 17, 18, 19, 20, 30, 31]. Less researched are some combinations, such as the hybrid approach to cloaking, which uses both a centralised and decentralised scheme [6], or a scheme that uses middleware [32] to provide privacy preservation [33, 34].

Obfuscation-based techniques try to prevent services from identifying the user, whether by adding some noise to his location information or by shifting the original location. The idea behind location obfuscation is that the real location is transformed into another space in which their spatial relationships are maintained to answer the location based queries. Obfuscation is not as frequent as the other two techniques, perhaps because it is similar to dummy generation. Maybe these two methods will be known under one name in the future. Nevertheless, it is a very active research field [35, 36, 37, 38, 39].

And, while obfuscation based techniques do their best to conceal the users' real location information, false location-based techniques do not try to conceal the location information, but rather they hide the users' location information in plain sight.

False location-based techniques protect user privacy either by reporting false locations to the location based-services, or by generating some dummy locations which are added to the real location and packaged into a query, so that the service does not know which location is the real one. In these false location-based methods there is a choice of using a random [40] or a carefully planned generator [41, 42] where the generator uses some other principles and techniques to generate the dummy data, such as soft computing techniques in [15].

So far there has not been a universal method developed, that would provide the desired privacy protection across all the different architectures and configurations. Each of the methods and techniques discussed in this paper has its strengths and weaknesses, and we must observe these qualities when deciding on which method to use. Another factor that we must pay attention to is also the rules and regulations of the country in which the location-based service provider is located, as that may also play a big role in protecting the privacy of the user.

4. OPEN PROBLEMS AND FURTHER DIRECTION OF RESEARCH

The field of data anonymization is fairly popular and well-known and there are many researches being carried out. Despite that, there are still many challenges and open problems that await future research.

One such open problem is the balance between the location-based services' precision and the user privacy. This is a particularly interesting topic, because of the delicate balance between the two opposing interests. Users wish for the location-based services to provide information with pin-point accuracy and, at the same time, not expose their location in such detail. And therein lies the dilemma because, if we want the service to provide precise information, we must provide it with the most detailed location information that we can, and in order to secure the location privacy of the user, we have to broaden the location to a range of multiple users, which diminishes the precision of the service. So this problem will continue to be a priority for researchers, as they will try to find the best balance between the services' precision and users' location privacy.

Another open problem that is interesting, but has not seen much research, is the problem of setting the desired level of privacy protection dynamically. This problem is interesting because it would give the users the power to choose what level of security they want for themselves. So far, there has been little done in the way of allowing users to choose their desired level of privacy. Often, users have to accept the implicit demands or terms of use of location-based services. On the other hand, if users take advantage of one of the methods discussed in this paper, they also simply have to accept the level of privacy protection that method is designed with, which leaves users with two absolute options, either 'full' exposure or 'full' protection. So we expect some research to go in that direction in the future.

Along with these two bigger problems, we also believe that the area of measuring the methods and examining their real-life application adequacy will also grow.

5. CONCLUSION

In this article we surveyed the area of data anonymization in the context of geolocation. Specifically, we have investigated mechanisms of protecting user privacy, classified them according to the architecture and techniques used, discussed some of these techniques, and reflected on some of the open research questions and problems. We have observed that the most popular method for protecting users' privacy is cloaking, which widens the location area of the user to a bigger field that encompasses multiple users and, while this is a good method for protecting the user's privacy, it also lowers the location-based service' precision, which is not ideal for the user experience. The research field of data anonymization will, therefore, continue to see many problems tackled and researches published.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency under the grant no. BI-US/15-16-067 and by the Research Programme P2-0057.

7. REFERENCES

[1] B. Niu, X. Zhu, Q. Li, J. Chen, and H. Li, "A novel attack to spatial cloaking schemes in location-based services," *Future Gener. Comput. Syst.*, vol. 49, pp. 125–132, Aug. 2015.

[2] R. Abbas, K. Michael, M. Michael, and R. Nicholls, "Key government agency perspectives on location based services regulation," *Comput. Law Secur. Rev.*, vol. 31, no. 6, pp. 736–748, Dec. 2015.

[3] W. Zhang, X. Cui, D. Li, D. Yuan, and M. Wang, "The location privacy protection research in location-based service," 2010, pp. 1–4.

[4] C.-Y. Chow, M. F. Mokbel, and X. Liu, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," *GeoInformatica*, vol. 15, no. 2, pp. 351–380, Apr. 2011.

[5] I. Memon, "Authentication User's Privacy: An Integrating Location Privacy Protection Algorithm for Secure Moving Objects in Location Based Services," *Wirel. Pers. Commun.*, vol. 82, no. 3, pp. 1585–1600, Jun. 2015.

[6] C. Zhang and Y. Huang, "Cloaking locations for anonymous location based services: a hybrid approach," *GeoInformatica*, vol. 13, no. 2, pp. 159–182, Jun. 2009.

[7] K. Parmar and D. C. Jinwala, "Concealed data aggregation in wireless sensor networks: A comprehensive survey," *Comput. Netw.*, vol. 103, pp. 207–227, Jul. 2016.

[8] S. Vennila and J. Priyadarshini, "Scalable Privacy Preservation in Big Data a Survey," *Procedia Comput. Sci.*, vol. 50, pp. 369–373, 2015.

[9] G. Dhand and S. S. Tyagi, "Data Aggregation Techniques in WSN: Survey," *Procedia Comput. Sci.*, vol. 92, pp. 378–384, 2016.

[10] D. Mohapatra and S. S.B, "Survey of location based wireless services," 2005, pp. 358–362.

[11] R. P. Minch, "Privacy issues in location-aware mobile devices," 2004, p. 10 pp.

[12] J. V. Chen, W. Ross, and S. F. Huang, “Privacy, trust, and justice considerations for location-based mobile telecommunication services,” *info*, vol. 10, no. 4, pp. 30–45, Jun. 2008.

[13] S. Saravanan and B. Sadhu Ramakrishnan, “Preserving privacy in the context of location based services through location hider in mobile-tourism,” *Inf. Technol. Tour.*, vol. 16, no. 2, pp. 229–248, Jun. 2016.

[14] R. Shokri, J. Freudiger, and J. Hubaux, “A Unified Framework for Location Privacy,” 2010.

[15] F. Tang, J. Li, I. You, and M. Guo, “Long-term location privacy protection for location-based services in mobile cloud computing,” *Soft Comput.*, vol. 20, no. 5, pp. 1735–1747, May 2016.

[16] H. Lu, C. S. Jensen, and M. L. Yiu, “PAD: privacy-area aware, dummy-based location privacy in mobile services,” 2008, p. 16.

[17] Baik Hoh and M. Gruteser, “Protecting Location Privacy Through Path Confusion,” 2005, pp. 194–205.

[18] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, “Preserving User Location Privacy in Mobile Data Management Infrastructures,” in *Privacy Enhancing Technologies*, vol. 4258, G. Danezis and P. Golle, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 393–412.

[19] B. Gedik and Ling Liu, “Location Privacy in Mobile Systems: A Personalized Anonymization Model,” 2005, pp. 620–629.

[20] M. Gruteser and D. Grunwald, “Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking,” 2003, pp. 31–42.

[21] L. Sweeney, “k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 05, pp. 557–570, Oct. 2002.

[22] B. Zhou and J. Pei, “The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks,” *Knowl. Inf. Syst.*, vol. 28, no. 1, pp. 47–77, Jul. 2011.

[23] I. Bilogrevic, M. Jadliwala, V. Joneja, K. Kalkan, J.-P. Hubaux, and I. Aad, “Privacy-Preserving Optimal Meeting Location Determination on Mobile Devices,” *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1141–1156, Jul. 2014.

[24] C.-Y. Chow, M. F. Mokbel, J. Bao, and X. Liu, “Query-aware location anonymization for road networks,” *GeoInformatica*, vol. 15, no. 3, pp. 571–607, Jul. 2011.

[25] C.-Y. Chow, M. F. Mokbel, and X. Liu, “A peer-to-peer spatial cloaking algorithm for anonymous location-based service,” 2006, p. 171.

[26] H. Lee, B.-S. Oh, H. Kim, and J. Chang, “Grid-based cloaking area creation scheme supporting continuous location-based services,” 2012, p. 537.

[27] M. M. E. A. Mahmoud and X. Shen, “A Cloud-Based Scheme for Protecting Source-Location Privacy against Hotspot-Locating Attack in Wireless Sensor Networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1805–1818, Oct. 2012.

[28] M. Y. Mun, D. H. Kim, K. Shilton, D. Estrin, M. Hansen, and R. Govindan, “PDVLoc: A Personal Data Vault for Controlled Location Data Sharing,” *ACM Trans. Sens. Netw.*, vol. 10, no. 4, pp. 1–29, Jun. 2014.

[29] X. Pan, X. Meng, and J. Xu, “Distortion-based anonymity for continuous queries in location-based mobile services,” 2009, p. 256.

[30] F.-Y. Leu, “A novel network mobility handoff scheme using SIP and SCTP for multimedia applications,” *J. Netw. Comput. Appl.*, vol. 32, no. 5, pp. 1073–1091, Sep. 2009.

[31] A. Samanta, F. Zhou, and R. Sundaram, “SamaritanCloud: Secure infrastructure for scalable location-based services,” *Comput. Commun.*, vol. 56, pp. 1–13, Feb. 2015.

[32] G. Myles, A. Friday, and N. Davies, “Preserving privacy in environments with location-based applications,” *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 56–64, Jan. 2003.

[33] J. Meyerowitz and R. Roy Choudhury, “Hiding stars with fireworks: location privacy through camouflage,” 2009, p. 345.

[34] B. Niu, Xiaoyan Zhu, Xiaosan Lei, Weidong Zhang, and Hui Li, “EPS: Encounter-Based Privacy-Preserving Scheme for Location-Based Services,” 2013, pp. 2139–2144.

[35] C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, “An Obfuscation-Based Approach for Protecting Location Privacy,” *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 1, pp. 13–27, Jan. 2011.

[36] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati, “Location Privacy Protection Through Obfuscation-Based Techniques,” in *Data and Applications Security XXI*, vol. 4602, S. Barker and G.-J. Ahn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 47–60.

[37] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: anonymizers are not necessary,” 2008, p. 121.

[38] A. Khoshgozaran and C. Shahabi, “Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy,” in *Advances in Spatial and Temporal Databases*, vol. 4605, D. Papadias, D. Zhang, and G. Kollios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 239–257.

[39] M. L. Yiu, G. Ghinita, C. S. Jensen, and P. Kalnis, “Outsourcing Search Services on Private Spatial Data,” 2009, pp. 1140–1143.

[40] H. Kido, Y. Yanagisawa, and T. Satoh, “An anonymous communication technique using dummies for location-based services,” 2005, pp. 88–97.

[41] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Achieving k-anonymity in privacy-aware location-based services,” 2014, pp. 754–762.

[42] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Enhancing privacy through caching in location-based services,” 2015, pp. 1017–1025.

Analysis of techniques for managing data on mobile devices

Klemen Sagadin
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
+386 51 242 244
klemen.sagadin@gmail.com

Boštjan Šumak
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
+386 2 220 7378
bostjan.sumak@um.si

ABSTRACT

In this paper, we conducted a study of techniques for managing data on mobile devices and defined needs for local data storage and data processing. Based on the results of a preliminary analysis, a set of data management techniques were chosen for a detailed analysis and comparison in terms of their usability, performance and complexity in the software development process. The set of data management techniques that were analysed included the relational database SQLite, the object database Realm and the relational-objective mapper OrmLite. The results of this study showed that there are significant differences among the chosen techniques in their usability for the developer, performance and complexity in the development of software solutions.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *complexity measures and performance measures*

H.2.4 [**Database Management**]: Systems – *Multimedia databases, Object-oriented databases, Query processing, Relational databases, Transaction processing*

General Terms

Algorithms, Measurement, Performance, Design, Reliability, Experimentation, Languages, Theory

Keywords

Techniques of processing and data storage on mobile devices, performance analysis, complexity of development using data storage, functionality of data storage techniques, object database Realm, relational database SQLite, mapper OrmLite.

1. INTRODUCTION

Globally, the number of mobile device users had exceeded the number of desktop computer users in 2014. In 2016, the number of mobile device users has increased to 1,900 million and more than 50% of users spent their time on mobile devices when searching and using digital media. On average, users with Android and iOS mobile devices spend 32% of their time for playing games and 68% on applications that need Internet access [1].

Because an Internet connection is not always available, it is of great importance for devices to work efficiently in off-line mode. For that to be possible, devices need to process data obtained from a network, save it locally and transmit it to the user even when the mobile device has no Internet connection. Because of the increasing use of mobile applications that depend on information gathered from the Internet, applications need to be developed in such a way that they are able to work in off-line mode [2].

For developers of mobile applications that run on systems with limited resources there are various mechanisms for data management. There are techniques for storing data, which are specific to a mobile operating system and corresponding programming environment, and there are techniques that are supported in multiple programming environments and mobile operating systems. A large number of mechanisms and techniques for data storage makes choosing the most suitable mechanism for development of a specific mobile application a challenging task. In this paper, we have conducted an analysis and comparison on mechanisms for managing and storing data on mobile devices with emphasis on the newer tools and concepts, which allow more contemporary approaches. In this study, we have taken into account the importance of presenting information in programming solutions in the form of a domain object oriented programming model, while considering the fact that, for efficient data storage, they often need to be converted properly into a form suitable for permanent storing.

2. TECHNIQUES FOR MANAGING DATA ON MOBILE DEVICES

In this section, the requirements are presented for processing and storing data on mobile devices and groups of techniques for data storage.

2.1 Requirements for Processing and Data Storage on Mobile devices

Applications on mobile devices need information which, in contemporary mobile solutions, can be obtained from various data sources in order to operate and ensure good user experience. We identified two domains where the use of managing and storing data techniques is of crucial importance.

2.1.1 Work in off-line mode

In contemporary IT solutions, embedded and other mobile devices are becoming more connected to the World Wide Web and can access remote data, which are necessary to ensure user experience. Despite better connectivity and regardless of the mobile device location, uninterrupted Internet access is not always possible. This is the reason for an increasing demand for undisturbed functioning of mobile solutions in off-line mode, meaning the mobile solution can work without an Internet connection. This applies particularly for mobile solutions which depend on data obtained from remote sources. In order to ensure functioning of a mobile solution in off-line mode, a suitable data storage technique on mobile device must be used for local managing and storing of information [3][4].

2.1.2 Big data on mobile devices

With the arrival of more advanced smart mobile devices, which use sensors for capturing information from the environment, the amount of data is increasing constantly. Consequently, there is a growing need for managing large data and transmitting the analysed results to the user. With the use of sensors, mobile devices can generate large amounts of data. Furthermore, advancements in multimedia technology (improved cameras, sound recorders etc.) have enabled capturing increased amounts of multimedia information (pictures, sounds and video). This kind of data needs to be processed and stored properly for it to be available for further use. Therefore, contemporary data storage mechanisms must be used for supporting such data [5][6].

2.2 Mobile Data Storage Techniques

Based on an overview of the possibilities for storing data on Android, iOS and Windows Phone mobile operating systems, we can divide data storage techniques into three groups:

- Key-value data storage,
- File data storage, and
- Local database storage.

Key-value data storage presents a database management system which offers a set of basic functions for manipulation with unstructured data objects, where each value has its own unique identifier [7][8]. File data storage presents saving files of specific data format in the mobile device's file system, where information is presented in the form of files [9]. Local database storage is used for saving structured and unstructured information. For storing data on mobile devices we used local databases, which are mostly independent libraries without a server component, without administration need and smaller demands for use of system sources [10][11].

3. ANALAYSIS OF MOBILE DATA STORAGE TECHNIQUES

In this study we focused on techniques for storing complex data structures and, based on a systematic literature review, we have chosen the most researched and used local database for mobile devices, which is SQLite. Because SQLite is a relational database, it needs to map data from a domain programming model to a relational model.

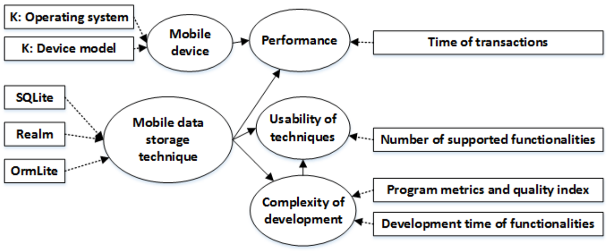

K: Operating system
K: Device model
Mobile device
Performance
Time of transactions
SQLite
Realm
OrmLite
Mobile data storage technique
Usability of techniques
Number of supported functionalities
Complexity of development
Program metrics and quality index
Development time of functionalities

Figure 1. Conceptual research model

Based on the results of a preliminary survey, we have chosen techniques Realm and OrmLite which automate this process and compared them with the relational database SQLite. Realm is an object-relational database that enables direct storing of domain model to the database. OrmLite is an ORM (Object Relational Mapper), which maps objects to the relational database. We have analysed in detail the influence of our chosen storage techniques on performance, usability for developer, and complexity of development by individual use of techniques. Figure 1 shows the conceptual research model.

3.1 Usability of Techniques from the Aspect of the Developer

We have defined functionalities important for a developer of programming solutions and their influence on the final usability of individual techniques. For each technique we observed (1) Its tools support for managing the database, (2) The possibilities of automatic mapping of domain objects to the database, (3) The support for different types of relations between saved data, (4) The support for managing with various data types, (5) The support for advanced data demands, (6) The support for multithread functioning, (7) The support for saving data on physical locations in the memory, (8) The support of transactions and ACID attributes, (9) The support for migrating data with data scheme changes and (10) The support for already built-in data encryption.

We came to the conclusion that the programming interface of OrmLite has no support for database managing tools. Most functionalities for automatic mapping of domain object to database are supported in database Realm and programming interface OrmLite, and are not supported in SQLite database. Support for data managing functionalities and for relations between data is best in the Realm and SQLite databases. All data storage techniques enable advanced data querying. The number of supported functionalities for multithread functioning is biggest in the Realm database and all data storage techniques have the same support for saving data on physical locations in the memory. The Realm database has most supported functionalities for transactions, ACID features and migrating data by data scheme changes. Database encryption is supported in Realm and SQLite databases and is not supported in OrmLite mapper. Most defined functionalities are supported by the Realm database, due to its object orientation and good support for multithread functioning on mobile devices. The SQLite database enables the least defined functionalities because of a lack of support for automatic mapping of objects to the database, which the OrmLite technique is trying to substitute.

Based on the analysis results, a Chi Square statistic test was conducted at a significance level of 1%; we have accepted the alternative hypothesis, stating there are significant differences in the number of supported functionalities between Realm and SQLite techniques and between Realm and OrmLite techniques. We were not able to discard the null hypothesis that there are no significant differences between SQLite and OrmLite, therefore we cannot accept its alternative hypothesis.

3.2 Complexity Analysis of the Development

We have researched complexity of development with use of data storage techniques by an experiment in which we have developed three functionally equivalent software solutions and each technique used one of the analysing data storage techniques. Software solutions are based on a domain object oriented model, whereby the data from entity classes must convert to a form suitable for the local database. We have defined 7 groups of software solution functionalities, which include database configuration (F0), defining software scheme (F1), creating new data inserts (F2), updating data values (F3), deleting already existing data (F4), selecting stored data based on different criteria and aggregate functions (F5) and executing asynchronous transactions (F6).

Regarding the development of software solutions, we measured the time needed for development of individual functionalities. The following Figure 2 presents the average measured times of performed experiments. For development of software solutions by using the Realm database, we needed less than half the time, because of the more demanding configuration and established workability of the OrmLite technique. For software solution development with use of the SQLite database we needed more time for implementation of individual operations on data. Because we had to implement proper methods for data mapping from object model to entity-relational model by ourselves, this resulted in a longer time needed for implementation for each operation on the data.

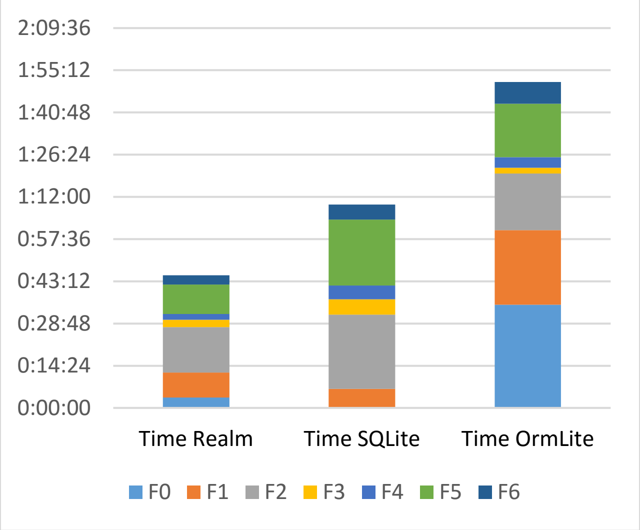


Figure 2. Time needed for development with the use of data storage techniques

We have analysed the developed software solutions with tools for calculation of software code metrics. After reviewing an article researched by (*Gerlec, C., and Heričko, M. (2010). Evaluating refactoring with a quality index. World Academy of Science*) we have chosen the set of metrics WMC, DIT, CBO and LCOM, based on which we have calculated an index of software code quality. The chosen metrics are non-complementary and non-correlational between each other. The results of code analysis are shown in the following Table 1 [12].

Table 1. Results of software code metrics and quality index

Software solution	WMC	DIT	CBO	LCOM	Qi
Realm	1	4	3,8	4	3,5
SQLite	1	2	3,75	3,38	3
OrmLite	3	2	5,67	3,22	3,25

Due to better results in DIT metric, which assesses the depth of inheritance in classes, the Realm database has reached a higher index because of its degradation and inheritance hierarchy tree, when using database software constructs. OrmLite mapper has achieved a higher result in comparison with Realm database in WMC metric, which assesses the sum of class methods` complexity, because of a higher number of classes with smaller amounts of methods. The results of OrmLite mapper in DIT and CBO metrics are less successful, because classes which use their own libraries are not well degraded and are coupled more between each other. Consequently, they present a more complex software solution. The software solution with the use of the software database constructs of SQLite reached lower results due to bad results in the WMC and DIT metrics. There are multiple classes needed for use of the SQLite database that helps us with easier SQL command writing and with creating a database; therefore, classes are bigger and poorly built. All three software solutions achieved comparable results in the LCOM metric, because it presents minor software solutions we divided well, based on dependencies between individual class attributes.

Based on the time needed for software solution development, use of OrmLite mapper was the most complex. However, we must take into consideration the less complex final implemented software code in comparison with the SQLite database. The fastest development and highest quality software code was reached by use of the Realm database and this is the reason we consider it the least complex of the compared techniques for a developer's use.

Based on the gained results of average times needed for solution development we ran a Leven's test of homogeneity variances and, based on the results, we came to the conclusion that homogeneity variances are being violated. Therefore, we decided to use Welch's statistic test for hypothesis testing with a significance level of 5%. A significant difference was established between the average times measured for the Realm and SQLite techniques and for the Realm and OrmLite techniques. We were not able to discard the null hypothesis, which states that there are no significant differences between the SQLite and OrmLite techniques; therefore, we cannot confirm that significant differences exist in the measured times between the SQLite and OrmLite techniques.

Based on the calculated indexes of quality, we can confirm the hypothesis for the existing differences in code with use of each of the data storage techniques.

3.3 Performance Analysis

To perform an analysis of performance of data storage techniques, we developed a mobile application which conducts transactions on data storage operations, with which we can measure the times needed for each individual transaction's execution. The mobile application, developed for analysing the complexity of development and explained in the previous Chapter, tests existing software code. During analysis of each technique, we were changing the capacity of processed data in individual transactions, with the purpose of trying to understand the impact of processed data on the performance of the functioning of individual techniques. We performed multiple tests for performance analysis and divided them into several groups, based on the chosen data storage techniques, as well as the functionality of testing and group of processed data. We tested the performance of inserting certain entities and relations, updating data, deleting data and obtaining data based on data relations, obtaining data based on arithmetic operator and calculating based on values of aggregate functions. Based on a systematic overview of the literature, we chose a metric for measuring performance and the time needed for execution of individual transactions. Times were measured with nanoseconds and built-in methods of the Java programming language. We have implemented each tested operation by DAO class, where data storage techniques use the same software interface; therefore, we can ensure equivalency and comparability of the performed tests. Tests were run on an LG G3 mobile device with installed Android 6.0 operating system. With each data storage technique we conducted 8 groups of tests and for each test we increased the capacity of data storage by the logarithmic scale with the base 10 that ranges from 1 to 10,000.

We came to the conclusion that the data storage technique influences the performance when executing individual transactions. We performed different types of tests which showed that the Realm database was the most powerful, confirmed by statistical tests, proving that there are significant differences between operational capabilities in comparison with the SQLite database and OrmLite mapper. The OrmLite and SQlite techniques achieved comparable results, confirming that it is not possible to prove significant differences between them. In certain tests with a smaller number of data, the techniques reached extremely comparable results, although the difference in operational capability increased with increasing the number of data.

Based on the results gained from the One-Way Anova statistical test at a significance level of 5%, we confirmed that there are significant differences in performance between the OrmLite and Realm techniques and between the Realm and SQLite techniques. We cannot discard the null hypothesis; therefore, we cannot confirm there are significant differences in performance between the OrmLite and SQLite techniques.

4. CONCLUSION AND FUTURE WORK

We have analysed an area of data storage techniques on mobile devices and came to the conclusion that data storage techniques can be devided into three groups, based on their characteristics. Based on preliminary research, we chose the SQLite, OrmLite and Realm techniques and compared them in terms of their usability, complexity of development and operational performance. The results provided proof that data storage techniques have impact on analysed concepts. Based on the performed comparative analysis and experiments we found that, for development of mobile solutions, the use of the Realm data storage technique is more efficient in comparison with the SQLite and OrmLite data storage techniques, because the Realm technique supports most analysed functionalities. Consequently, Realm`s execution of technique is more efficient, its implemented software code less complex and there is less time needed for development. We were not able to provide proof for significant differences between the SQLite and OrmLite techniques in operational capabilities and times needed for development. However, we did confirm that, OrmLite mapper in comparison with the SQLite database, supports more functionalities and its implemented software solutions are less complex. We have confirmed that picking the right data storage technique has impact on the efficiency of software solution development. Techniques which enable automatised mapping from domain model to data storage have proven to be more effective and in the Realm object database even more capable.

In future work research we will expand existing research with analysis of techniques for energy waste and other sources on mobile devices. This concept could not be analysed in detail due to limitations in this research. However, it does have an impact on the experience of the final user and mobile solution developer.

5. REFERENCES

[1] Bosomworth, D. 2016. Mobile marketing statistics 2016. http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/.

[2] Whitney, L. 2012. Offline Capabilities: Native Mobile Apps vs. Mobile Web Apps. http://www.sitepoint.com/offline-capabilities-native-mobile-apps-vs-mobile-web-apps/.

[3] Elgan, M. 2014. The hottest trend in mobile: going offline! | Computerworld. http://www.computerworld.com/article/2489829/mobile-wireless/the-hottest-trend-in-mobile--going-offline-.html.

[4] Mahemoff, M. 2013. ‘Offline’: What does it mean and why should I care? http://www.html5rocks.com/en/tutorials/offline/whats-offline/.

[5] Liebowitz, J. 2016. Big Data and Business Analytics. CRC Press.

[6] Walls, T. A. and Schafer, J. L. Models for Intensive Longitudinal Data. Illustrate.

[7] Basescu, C., Cachin, C., Eyal, I., Haas, R., Sorniotti, A., Vukolic, M. and Zachevsky, I. 2012. Robust data sharing with key-value stores. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), 1–12.

[8] Aerospike Inc. What is a Key-Value Store? http://www.aerospike.com/what-is-a-key-value-store/.

[9] Sadaqat, J., Maozhen, L., Ghaidaa, A. and Hamed, A. 2010. File annotation and sharing on low-end mobile devices. Seventh International Conference on Fuzzy Systems and Knowledge Discovery, 6, Fskd, 2973–2977.

[10] Roukounaki, K. 2014. Five popular databases for mobile - Developer Economics. http://www.developereconomics.com/five-popular-databases-for-mobile/.

[11] Ouarnoughi, H., Boukhobza, Olivier, J., P., Plassart, L. and Bellatreche, L. 2013. Performance analysis and modeling of SQLite embedded databases on flash file systems, Des. Autom. Embed. Syst., 17, 3–4, 507–542.

[12] Gerlec, C. and Hericko, M. 2010. Evaluating refactoring with a quality index, World Acad. Sci. Eng. Technol., 63, 3, 76–80.

Can we predict software vulnerability with deep neural network?

Cagatay Catal
Department of Computer Engineering
Istanbul Kültür University
Istanbul, Turkey
c.catal@iku.edu.tr

Akhan Akbulut
Department of Computer Engineering
Istanbul Kültür University
Istanbul, Turkey
a.akbulut@iku.edu.tr

Sašo Karakatič
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
saso.karakatic@um.si

Miha Pavlinek
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
miha.pavlinek@um.si

Vili Podgorelec
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
vili.podgorelec@um.si

ABSTRACT

In this paper, we present an alternative approach to software vulnerability prediction with modern machine learning methods – with deep learning methods. Deep learning methods are techniques where features in our case software metrics) are processed and sent through multiple layers where transformations and computations are done in sequence to form a prediction model.

The deep learning methods have not been used for software vulnerability prediction so far and could provide a new and potentially competitive alternative to the existing techniques. In the paper we make an overview of existing solutions on the subject and compare them to the proposed system with deep learning. The deep learning techniques are presented in details and a proposition for the prediction system is made.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.5 [**Testing and Debugging**]: Testing tools, Code inspections; D.2.8 [**Software Engineering**]: Metrics—complexity measures, performance measures

General Terms

Algorithms, Measurement, Reliability, Theory

Keywords

Software vulnerability prediction, machine learning, deep learning

1. INTRODUCTION

Software security vulnerabilities are still very common and new alerts and reports from several agencies are published every day. One such incident was published on May 13, 2015 when US Food and Drug Administration (FDA) reported an alert about computerized infusion pumps which can be programmed remotely and malicious Internet users can modify the dosage of therapeutic drugs. FDA suggested several actions for the hospitals which are using these systems to secure them. As we see in this recent incident, software security vulnerabilities are quite dangerous for software-intensive systems.

If vulnerable components of software systems can be detected prior to the deployment of the software, verification resources can be assigned effectively. This research area is known as *software security vulnerability prediction* and researchers developed several prediction models so far. Although some of the researchers showed the benefit of those models, we need much better models in terms of prediction accuracy, precision, and recall. Some of the companies do not adopt these models yet due to their inefficient prediction performance [1].

Software developers apply static code analysis tools [2] and code reviews [3] to avoid security vulnerabilities. For large scale software systems and systems of systems, it is not practical to review all the code against possible threats. Therefore, good vulnerability prediction model is inevitable.

Although there are many research papers on this topic, companies like Microsoft still do not adopt Vulnerability Prediction Models (VPM) [4]. The reason is related with the low prediction performance on the source code level in terms of recall and precision evaluation parameters. In that study, Morrison et al. (2015) reported that *state-of-the-art* models do not provide accurate prediction and security-specific metrics can be utilized in the later studies to achieve an acceptable performance.

According to our literature survey, we did not encounter any research paper which applied deep learning. Deep learning is being used by many high-tech companies such as Facebook, Microsoft, and Google to solve challenging problems such as facial recognition, real-time translation, and speech recognition respectively. We aim to advanced machine learning techniques such as deep learning to predict vulnerable components accurately.

In this paper we make an initial review of the field and propose a new outlook on the problem. In the next chapter, the review of the related work is presented. Then, we follow up with the presentation of the modern machine learning technique – *deep learning* or *deep neural networks*. We continue with the proposed novel approach to the software vulnerability prediction technique with the deep neural networks.

2. RELATED WORK

Shin and Williams (2008) [5], [6] reported that complexity metrics have correlation with security vulnerabilities. They worked on Mozilla JavaScript Engine. Shin et al. (2011) applied logistic regression technique and analyzed the relationship of developer activity, complexity, and code churn with software security vulnerabilities [7]. Chowdhury and Zulkernine (2011) used decision trees to predict the vulnerabilities by using complexity, cohesion, and coupling metrics [8]. Mean accuracy was 72.85% on Mozilla Firefox data. Zimmermann et al. (2010) reported that traditional metrics such as complexity, code churn, and organizational measures have a weak correlation between vulnerabilities for Windows Vista [9]. Although complexity, cohesion, and coupling metrics have been studied in previous studies in detail, security-specific metrics should be determined and applied in the models. Shin and Williams (2013) investigated whether fault prediction models can be used for vulnerability prediction or not [10]. They built both fault prediction and vulnerability prediction models and concluded that fault prediction models provide similar results as vulnerability prediction models, but both of them must be improved to reduce the number of false positives.

Recent studies on vulnerability prediction started to focus on machine learning techniques. Scandariato et al. (2014) presented a model based on machine learning to predict the vulnerabilities [11]. Terms in the source code are taken into account and their associated frequencies are noted. Twenty Android applications were used for the validation of the prediction approach. During the experiments, they analyzed the performance of Naive Bayes and Random Forest algorithms on this problem. They reported that Random Forest provides better performance than Naive Bayes algorithm. Walden et al. (2014) prepared a vulnerability dataset which has 223 vulnerabilities [12]. They used Drupal, Moodle, and PHPMyAdmin projects to analyze vulnerabilities. As the machine learning algorithm, they applied Random Forests algorithm and reported that models using text mining is better than models using metrics in terms of recall parameter. They used 3-fold cross-validation and experiments were performed 10 times.

Mokhov et al. (2015) showed that machine learning approach is effective to detect vulnerabilities and implemented a tool called MARFCAT for fast code analysis [13]. Tool works on source code level, binary level, and bytecode level. Shar et al. (2015) applied static and dynamic code attributes to detect vulnerabilities in web applications [14]. They used not only supervised machine learners but also semi-supervised algorithms to analyze the performance of prediction. They reported that semi-supervised learning is preferable when vulnerability data is limited. Last (2016) explained the research on Vulnerability Discovery Models development to forecast the zero-day vulnerability [15]. He stated that the research created two approaches based on machine learning and one approach based on regression technique.

Grieco et al. (2016) implemented a tool called VDiscover which applies machine learning techniques for the prediction of vulnerabilities in test cases [16]. Experimental result showed that the proposed approach predicts the programs which contain dangerous memory corruptions effectively. Medeiros et al. (2015) used taint analysis in conjunction with data mining [17]. Candidate vulnerabilities are detected with taint analysis and false positives are identified by using data mining technique. In addition to detection of vulnerabilities, automatic corrections of vulnerabilities are performed by adding fixes to the source code automatically. The approach has been validated on a large set of PHP applications and compared to well-known PHP tools for static code analysis. The performance was 5% better than PhpMinerII and 45% better than Pixy's performance in terms of accuracy and precision.

3. DEEP LEARNING

Deep learning is a term that combines together techniques of machine learning that result in complex models where each model is composed of multiple processing layers. For the sake of simplicity and understandability, we will focus our research on the deep neural networks, which are a subset of deep learning method. Deep learning approaches have dramatically improved the state-of-the-art results in several fields, which have traditionally been dominated by ensemble machine learning techniques or other approaches. These field, with their state-of-the-art solutions are shown in the Table 1.

Table 1. Applied deep learning on different problems

Image recognition	[18]–[21]
Speech recognition	[22]–[24]
Prediction of the drug molecule activity	[25]
Analyzing the particle accelerator data	[26], [27]
Natural language processing and understanding	[28]
Language translation	[29], [30]

Deep neural networks are composed of several layers, where first layers have a goal of representation learning. With representation learning we can feed in raw data and the method discovery proper problem representation on its own. Each layer in network represents a non-linear module that transforms and represents the data in different way. [31]

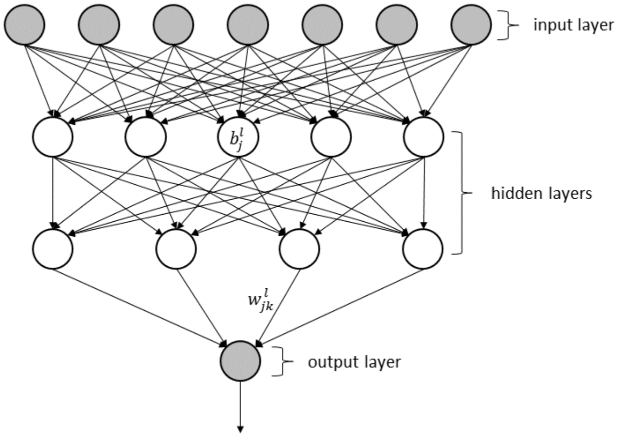


Figure 1. Example of deep neural network with two hidden layers

Deep neural networks always start with the first layer of inputs – raw data. For data of an image, the input layer can be different intensity levels for each pixel on each of the color levels. The following layer can transform the raw data in such way, that only edges in different angles and orientations are highlighted. Next layer can detect round shapes, corners or other intensity transitions on the image. Following layer usually combine the outputs of the edge, corner and round detection layers and detect motifs and shapes which are combined by edges, corners and other shapes. Next layer can combine the output of motif and shape detection layers in even higher dimension figure, where familiar shapes are starting to form: rectangles, triangles, circles and other shapes or parts of these shapes. Then the output of this layer is fed into next layers which can detect even lees abstract shapes. This process can be repeated as long as necessary and each next layer searches for less abstractions and moves onto real shapes and figures. [32]

The main thing to note here is that these layers are not designed by hand, but are usually learned through the process of backpropagation on the whole neural net through all of the layers. Instead of the backpropagation process, some heuristic approaches can be used, such as genetic algorithms and simulated annealing, but this is out of the scope of this paper.

As in other machine learning techniques, we can use deep neural networks for different types of problems. Different kinds of deep neural networks are used for different kind of problem. Following is the list divided by machine learning problem with specific neural network designed used for that problem.

- **Supervised deep learning**: Deep convolutional networks, Recurrent neural networks
- **Unsupervised deep learning**: Auto encoders, Restricted Boltzmann machines, Deep Belief Networks
- **Semi-supervised deep learning**: Ladder networks

4. DNNs FOR VULNERABILITY PREDICTION SYSTEM

We propose a system that utilizes the deep neural network machine learning technique for prediction of the software vulnerabilities. This can be done in number of ways. If there is previous vulnerability data, supervised learning models can be applied. If there is no previous data, unsupervised deep learning algorithms can be used. If there is very limited vulnerability data, semi-supervised deep learning techniques can be investigated. We will analyze all of these three problems in the project to solve them efficiently.

There are number of implementation of deep neural networks that can be used in the proposed system. Following is the list of such libraries, packages and software that are mainly used in the industry or in the research.

- **TensorFlow** [33] – an open source library developed by Google and written in Python and C++ language, which can be used with other Python and C++ software through the API provided.
- **Theano** [34] **–** an open source Python library for DNN developed by University of Montreal.
- **Torch** [35] – an open source machine learning library written in C, maintained by Facebook and Google engineers and used by Google DeepMind and Facebook AI research teams.
- **Deeplearning4j** – an open source C and C++ implementation of deep neural networks developed by Skymind that provide Java API.
- **Caffe** [36] – implemented in C++ and Python and pdovides APIs for C++, Python and Matlab.
- **Keras** [37] – an Python library which utilizes TensorFlow or Theano and provides an easy to use API.

One or multiple libraries can be used in vulnerability prediction software – it depends on the programming language used. All of the above deep neural network libraries contain basic convolutional and recurrent layers. The performance of specific type of neural network will have to be determined with the experiment.

5. CONCLUSION

During our literature review we recognized the lack of usage of modern machine learning technique of deep neural networks for software vulnerability prediction. Deep neural networks represent the *state-of-the-art* on multiple optimization, prediction and pattern recognition problems, so there is a surprising lack of application with them in software engineering topic

Our paper serves to persuade researchers, that this problem is worth to tackle while this topic still remains under-researched. Multiple deep neural network types could be used with this kind of problem, but the performance of each on vulnerability prediction is yet to be determined.

6. REFERENCES

[1] C. F. Kemerer and M. C. Paulk, “The impact of design and code reviews on software quality: An empirical study based on PSP data,” *IEEE Trans. Softw. Eng.*, vol. 35, no. 4, pp. 534–550, 2009.

[2] A. G. Bardas and others, “Static Code Analysis,” *J. Inf. Syst. Oper. Manag.*, vol. 4, no. 2, pp. 99–107, 2010.

[3] M. V Mäntylä and C. Lassenius, “What types of defects are really discovered in code reviews?,” *IEEE Trans. Softw. Eng.*, vol. 35, no. 3, pp. 430–448, 2009.

[4] P. Morrison, K. Herzig, B. Murphy, and L. Williams, “Challenges with applying vulnerability prediction models,” in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, 2015, p. 4.

[5] Y. Shin and L. Williams, “An empirical model to predict security vulnerabilities using code complexity metrics,” in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 2008, pp. 315–317.

[6] Y. Shin and L. Williams, “Is complexity really the enemy of software security?,” in *Proceedings of the 4th ACM workshop on Quality of protection*, 2008, pp. 47–50.

[7] Y. Shin, A. Meneely, L. Williams, and J. A. Osborne, “Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities,” *IEEE Trans. Softw. Eng.*, vol. 37, no. 6, pp. 772–787, 2011.

[8] I. Chowdhury and M. Zulkernine, “Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities,” *J. Syst. Archit.*, vol. 57, no. 3, pp. 294–313, 2011.

[9] T. Zimmermann, N. Nagappan, and L. Williams, “Searching for a needle in a haystack: Predicting security vulnerabilities for windows vista,” in *2010 Third International Conference on Software Testing, Verification and Validation*, 2010, pp. 421–428.

[10] Y. Shin and L. Williams, "Can traditional fault prediction models be used for vulnerability prediction?," *Empir. Softw. Eng.*, vol. 18, no. 1, pp. 25–59, 2013.

[11] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, "Predicting vulnerable software components via text mining," *IEEE Trans. Softw. Eng.*, vol. 40, no. 10, pp. 993–1006, 2014.

[12] J. Walden, J. Stuckman, and R. Scandariato, "Predicting vulnerable components: Software metrics vs text mining," in *2014 IEEE 25th International Symposium on Software Reliability Engineering*, 2014, pp. 23–33.

[13] S. A. Mokhov, J. Paquet, and M. Debbabi, "MARFCAT: Fast code analysis for defects and vulnerabilities," in *Software Analytics (SWAN), 2015 IEEE 1st International Workshop on*, 2015, pp. 35–38.

[14] L. K. Shar, L. C. Briand, and H. B. K. Tan, "Web application vulnerability prediction using hybrid program analysis and machine learning," *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 6, pp. 688–707, 2015.

[15] D. Last, "Forecasting Zero-Day Vulnerabilities," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, 2016, p. 13.

[16] G. Grieco, G. L. Grinblat, L. Uzal, S. Rawat, J. Feist, and L. Mounier, "Toward Large-Scale Vulnerability Discovery using Machine Learning," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, 2016, pp. 85–96.

[17] I. Medeiros, N. Neves, and M. Correia, "Detecting and removing web application vulnerabilities with static analysis and data mining," *IEEE Trans. Reliab.*, vol. 65, no. 1, pp. 54–69, 2016.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013.

[20] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[22] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černock\\`y, "Strategies for training large scale neural network language models," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, 2011, pp. 196–201.

[23] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and others, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[24] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8614–8618.

[25] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure--activity relationships," *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, 2015.

[26] T. Ciodaro, D. Deva, J. M. De Seixas, and D. Damazio, "Online particle detection with neural networks based on topological calorimetry information," in *Journal of Physics: Conference Series*, 2012, vol. 368, no. 1, p. 12030.

[27] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kegl, and D. Rousseau, "Learning to discover: the Higgs boson machine learning challenge," *URL http//higgsml. lal. in2p3. fr/documentation*, 2014.

[28] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," *arXiv Prepr. arXiv1406.3676*, 2014.

[29] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[30] S. J. K. Cho, R. Memisevic, and Y. Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation," 2015.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and others, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv Prepr. arXiv1603.04467*, 2016.

[34] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, G. Desjardins, D. Warde-Farley, I. Goodfellow, A. Bergeron, and others, "Theano: Deep learning on gpus with python," in *NIPS 2011, BigLearning Workshop, Granada, Spain*, 2011.

[35] N. Léonard, S. Waghmare, and Y. Wang, "RNN: Recurrent library for torch," *arXiv Prepr. arXiv1511.07889*, 2015.

[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[37] F. Chollet, "Keras: Deep learning library for theano and tensorflow." 2015.

Exhaustive key search of DES using cloud computing

Aleks Drevenšek
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
aleks.drevensek@gmail.com

Marko Hölbl
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
marko.holbl@um.si

ABSTRACT

In this paper we present the time complexity of exhaustive key search for the DES algorithm using modern cloud computing. We demonstrate that it is possible to perform a brute force attack on a known encryption algorithm in practice using commercially available cloud computing services. We also discuss a previous attempt of exhaustive key searches, and explain the methods and preparations for the experiment. The time complexity is still very high, but the time needed for finding a key can be improved using cloud computing, but not with the available free resources.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Data Encryption Standard (DES)

General Terms

Algorithms, Measurements, Performance, Experimentation, Security

Keywords

Cloud computing, exhaustive key search, DES, Microsoft Azure

1. INTRODUCTION

One of the goals of modern cryptography is the assurance of confidentiality which is achieved through the use of encryption. Algorithms of encryption, referred to as ciphers, are classified into two types: Asymmetrical and symmetrical. Symmetrical ciphers use one key for both encryption and decryption [1]. This paper focuses on these types of ciphers. Additionally, these symmetric ciphers are classified into block and stream ciphers[2].

Exhaustive key search, or brute force attack, on a modern symmetric cipher is a method of trying every single key in known key space to identify the key used to encrypt selected plain text [3]. Modern cloud services are perfect options to execute such heavy tasks [4].

The purpose of our research was to demonstrate that it is possible to find an encryption key using cloud computing in reasonable time, which could open new questions about the security of modern algorithms.

In this paper we will answer two questions: Is it possible to execute an exhaustive key search for an algorithm successfully using cloud computing? How long would such an exhaustive key search take?

The chosen block cipher for the experiment was DES [5]. It is a symmetrical block cipher. Due to its short key length, only 56-bits, it is prone to brute force attacks [6].

2. PREVIOUS DES CHALLENGES

In the past, several competitions were carried out by RSA Security with the intention of finding a key for DES. The data provided to the competitors was: A known algorithm, part of a plain text and the full cipher text. The competitors were provided with 192 bits of the plain text, the methods of converting plain text to a hexadecimal value and separate it into blocks of 64 bits, and the methods of padding to the full block, since DES is a block cipher and operates with 64-bit blocks of data. The encryption mode was CBC [5].

2.1 Competition DES-I

The first DES competition was held in 1997 and was won by a group of three called DESCHALL. They tackled the problem by building a distributed network of applications for executing the exhaustive key search [7]. They used a client-server architecture, where the server determined which key space was to be searched next and which keys were already checked [7]. Their clients were physical computers owned by volunteers. The fastest computer they used was a Power PC 604e with processor speed of 250 MHz and search speed of 1.5 million keys per second [7]. An improvement of searching algorithm was conducted while the search was being performed. Near the end of the competition, a team developed a new technique called bit slicing that allowed it to search 32 or 64 keys simultaneously, depending on the CPU architecture – a 32-bit CPUs was able to calculate 32 keys simultaneously and 64-bit CPUs 64 keys. With this improvement the fastest speed calculated on a 167MHz UltraSPARC computer was 2.4 million keys per second [7].

The first competition was completed successfully in 96 days with 51.8% of key space searched. They recorded more than 78,000 unique IP addresses on the server and had around 14 thousand highest concurrent computers searchers.

2.2 Competition DES-II-1 and DES-II-2

The second DES competition was held in the beginning of 1998. The wining organisation was distributed.net, which used a similar infrastructure as DESCHALL. The key was found after 39 days. The highest search speed in this competition was 32,430 million keys per second. 90% of all key space has to be searched. The organisation distributed.net estimated that their computing power was equivalent to 22 thousand computers with Intel Pentium II at 333 MHz That was about double the power of the best DES-I competitors' resources [8].

The third competition was announced in the same year. The EFF [9], with a dedicated super computer created specifically for this purpose, named Deep Crack won. This super computer was using advanced hardware implementation of DES, which was faster than the equivalent software implementations. The average speed was 88,804 million keys per second. The total time of the search was 56.05 hours. The size of the key space that was needed to be searched to find the key was around 24.8% [10].

2.3 Competition DES-III

The last competition was in January, 1999. In this competition the highest prize money was given if the search would be completed within 24 hours and if the searches would take more than 56 hours, no reward would be given[11].

The winner of the competition was a team consisting of distributed.net [12] and EFF [12]. The search was finished in 22 hours and 15 minutes. The average search speed was 199,000 million keys per second which is more than double the speed of Deep Crack (88,804 million keys per second). They also needed to check around 22.2% of the key space, which was the lowest number of keys needed to be searched in all competitions [13].

3. CLOUD COMPUTING

Cloud computing simplifies the access to ready to use computer resources. The main feature is the availability of computing power which is necessary for exhaustive key search [14].

We identified the necessary resources to execute an exhaustive key search. We focused on cloud services that were offering a cloud computing service. The first resource that was considered was computing power, the number of CPU cores and the amount of available memory. Computing power is defined by the type of virtual machine. The second resource was storage, for storing searching application and results. In contrast to CPU we did not need a huge amount of storage [15].

While most cloud services offered a similar type and amount of resources, only Microsoft Azure offered a dedicated service for high performance computing, which is referred to as Azure Batch. With that in mind, we decided to use the Microsoft Azure cloud service. Their Batch service is designed to execute computing that requires up to 10 thousand processor cores [4].

The computers used in the Azure Batch service are of the same type as Microsoft Azure virtual machines. They are divided into three groups: A, D and D version 2, with Dv2 being the fastest regarding CPU resources. Virtual machines of type A use the Intel Xeon E5-2670 processor with speed of 2.6 GHz and type Dv2 use Intel Xeon E5-2673 v3 CPUs with speed of 2.4 GHz that can be boosted up to 3.2 GHz. The instances that we used in our experiment are shown in Table 1 [16].

Table 1: Microsoft Azure type of virtual machines

Instance	Number of cores	Memory (GB)
A1	1	1.75
A2	2	3.5
A3	4	7
A4	8	14
A5	2	14
A6	4	28
D1v2	1	3.5
D2v2	2	7
D11v2	2	14

4. USING CLOUD COMPUTING TO PERFORM AN EXHAUSTIVE KEY SEARCH OF DES

Our experiment was conducted in an on-line environment. Research context was specific.

4.1 Experimental variables

Independent variables were connected to the chosen cloud service. They were all combined in packets called virtual machines. In this way we could not change them separately. Our independent variables were: CPU speed, number of CPU cores and memory size.

The CPU speed was a continuous variable with a value range of 0 GHz to 3.3 GHz. The number of cores was a discrete variable with a value range of 1, 2, 4, 8, 16 or 20. The memory size was a continuous variable with a range from 0.75 GB to 140 GB.

We defined two dependent variables: The search speed and the time required to find the key. The first was defined by dividing the number of all searched keys with the required search time. The second variable was continuous as well and was calculated from the search speed and the number of all keys.

4.2 Experimental plan

First we needed to prepare the environment. This step included the log-in procedure with a valid Azure Batch account. For our experiment we used a trial account. Then we created a pool of virtual machines - up to 20 cores. Finally, we uploaded the program for exhaustive key search.

The following step was performing the exhaustive key search. We used speed evaluation mode to be able to measure speed and calculate results. In this mode the search program checked 2^{30} keys and then finished (exited).

We measured the time for each instance of a virtual machine separately. The time needed to prepare instances, transfer files and other auxiliary tasks was ignored. For the measured time we considered only the time needed to execute a key search.

We repeated all steps for each different type of virtual machine, iterating through all the available types. The procedure of the experiments is shown in Figure 1. We could conduct the same procedure on different variables and we would expect different results.

$$\begin{array}{l} X\ O_1 \\ X\ O_2 \\ X\ O_3 \\ \ \ \vdots \\ X\ O_n \end{array}$$

Figure 1: Representation of experiment process

5. KEY SEARCH IMPLEMENTATION

The task of exhaustive key search is highly time and resource consuming. Keeping that in mind, we were forced to use performance improvements. Our software was written in the C++ programming language and we used updated versions of tools that were used in past competitions.

The used software employs the method of bit slicing and is intended to be used on 64 bit processors for best performance results. Since we wanted to execute the search on multiple keys simultaneously over one search cycle we had to transform the input data. We had to convert the starting data into hexadecimal values. The next operation was to transform data into a bit slicing compatible format, where each bit that was marked as 1 was transformed into the highest possible value of data type unsigned __int64.

After the data was prepared, we executed the search using the method keySearch(). The task of the method was to prepare candidate keys, variable for multithreaded mode if that would be optimal and execute another method deseval() for the current set of keys. If a key was returned we found the correct key, otherwise the deseval() was repeated with a new set of keys.

Before starting the search, the deseval() method would load first using the first set of keys. The purpose of that was to load it into memory and save on time while actually executing a search. The measuring process started just before the first execution of deseval() and stopped after the last execution was finished.

This main method, deseval(), uses modified S-boxes for deciphering with multiple keys at once. The method runs 14 rounds of the algorithm before it is possible to check if all keys are incorrect. In 1.6% of all executions, the method continued and proceeded to do multiple checks over the last 2 rounds. Compared to the normal process of the DES deciphering procedure, we were able to check the correctness of the key after only 14 rounds compared to the 16 that the process would normally take. Another improvement that was included was the possibility to check 64 keys at once instead of just one.

6. RESULTS

During the execution of the experiment we noticed that not all Microsoft Azure virtual machines were available. This may be due to the fact that Microsoft Azure Batch is a new service and may still have some imperfections. We were able to perform searches on the following 9 instances: A1-A6, D1v2, D2v2 and D11v2. For each instance we normalized the speed to one core.

Table 2: Search speed of available instances

Instance	Cores	Speed (keys per second)	Speed per core (keys per second)
D2v2	2	36,616,485	18,308,242
D1v2	1	17,724,361	17,724,361
D11v2	2	33,172,943	16,586,471
A1	1	14,573,834	14,573,834
A5	2	27,506,451	13,753,225
A3	4	52,123,389	13,030,847
A2	2	22,510,310	11,255,155
A6	4	41,688,997	10,422,249
A4	8	82,671,837	10,333,979

The instances of type D2 version 2 were faster than all of the instances of type A, which was expected since they use newer CPUs. We observed that instances with less cores were mostly faster than those with more.

The fastest instance was D2v2 with 2 cores of Intel Xeon E5-2673 v3 at a speed of 2.4 GHz with 7GB of memory. The search speed per core was about 18.3 million keys per second. The second fastest instance was D1v2 which performed about 500 thousand keys per second slower. The third instance type D11v2 was slower than the first by over 1.7 million keys per second. Since they all use the same hardware, we assumed the cause could lie in the overhead of the virtualization.

We compared instance type A versus type Dv2. The average calculated speed of type A was around 12,228,215 keys per second while the average speed of type Dv2 was 17,539,682 keys per second, which means that instance type Dv2 were, on average, faster by 43.4%. The fastest instance D2v2 was faster by 25.6% than the fastest type A instance - A1.

6.1 Time Complexity

We calculated the time required to finish successfully an exhaustive key search of a random generated DES key for the fastest instances. According to the rules of previous DES competitions we generated a key randomly and used it to encrypt an arbitrary plain text. To perform an exhaustive key search for this key successfully we would have to search 34.26% of all keys. Based on this, we calculated the different times required by the search.

Table 3: Time required to find a random key with D2v2

Number of cores	Total search speed (keys per second)	Required time
20	366,164,856	26.01 months
400	7,323,297,120	39 days
6,700	122,665,226,760	56 hours
10,000	183,082,428,000	37.45 hours
17,000	311,240,127,600	22 hours

The results for the fastest instance D2 version 2 indicate that, with the limited number of cores available in the trial version of Azure, the time complexity of the search would be more than 26 months. To achieve the winning time of the DES-II challenge, 39 days, we would need 400 cores. To lower the time to 56 hours as of the next DES competition 6,700 cores would be needed. With the maximal number of cores allowed by Microsoft, 10,000, the search would take 37.45 hours. To get faster results than the ones from the competitions, we would have to use more cores than are allowed by the cloud, that is 17,000.

6.2 Worst case scenario

The worst case scenario for an exhaustive key search would be if the random generated key would be the last key in the sets of keys which needed to be searched – the entire key space would need to be searched. We recalculated our results to fit the worst case scenario. Instances used in this calculation were D2v2.

Table 4: Time required to find the last key with D2v2

Number of cores	Total search speed (keys per second)	Required time
20	366,164,856	75.9 months
1168	21,384,027,590	39 days
19,523	357,431,824,184	56 hours
10,000	183,082,428,000	4.5 days
49,695	909,828,125,946	22 hours

Using instance D2v2, in the case of searching through every key the time required to finish with 20 cores would be almost 76 months. To beat the best time of 39 days, we would need 1,168 cores. To beat the 56 hours of the winner of the second competition, the required number of cores would be 19,523. This is already over the maximum number of cores allowed by the Microsoft Azure cloud service. Using the maximum number of cores, we would need

4.5 days. To find a key faster than in all previous competitions we would have to use almost 50,000 cores.

6.3 Virtualization overhead

Since modern cloud computing is powered by virtualization technology we also investigated this aspect. While virtualization may have numerous advantages, it has also drawbacks. One is performance loss. To measure how much of performance is lost when using virtualization, we ran our searching algorithm on a normal computer (Intel i5-3570k CPU 3.4GHz). The personal computer was able to search 30 million keys per second per single core, that is around 64% faster than the speed of cloud instance D2v2. If we subtract the difference in speed of the CPUSs we could assume that the loss in performance is around 22%.

7. CONCLUSION

In this paper we presented the use of Microsoft Azure cloud services with the new Azure Batch service for high performance application computing, namely for exhaustive key search for the DES algorithm. We used a brute force attack approach and estimated the computing power needed for a successful attack.

We used different variants of the Microsoft Azure cloud service platform (A1, A2, A3, A4, A5, A6, D1v2, D2v2 and D11v2). According to our measurements the fastest instance was type D version 2. The maximal number of cores of virtual machines that could be run per account was 10,000. Since cloud computing is based on virtualization, there is some loss of performance - we calculated it to be around 22%. The documentation of the cloud provider estimated the loss to be around 15-20% [17].

It can be concluded that exhaustive key search can be performed successfully. The required time depends on the number of activated cores. In the worst case scenario, using the maximum number of cores, 10,000, it is possible to find the key in a time of 4.5 days. If we wanted to improve this time, we would need more cores, which would require multiple accounts.

Another aspect of speeding up the process would include the optimization of the software used for searching. In Key feature the S-boxes were outdated and by updating them to the newest version we could lower the time complexity.

Finally, it has to be noted that those results are based on the Microsoft Azure cloud and could differ if other cloud provider would have been used.

For the future we could research the search speed of other encryption algorithms, mostly those which are still considered as secure.

8. REFERENCES

[1] G. J. Simmons, ‘Symmetric and asymmetric encryption’, *ACM Comput. Surv. CSUR*, vol. 11, no. 4, pp. 305–330, 1979.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. Boca Raton: CRC Press, 1997.

[3] F. Rubin, ‘Foiling an Exhaustive Key-Search Attack’, *Cryptologia*, vol. 11, no. 2, pp. 102–107, Apr. 1987.

[4] ‘Azure Batch feature overview | Microsoft Azure’. [Online]. Available: https://azure.microsoft.com/en-us/documentation/articles/batch-api-basics/. [Accessed: 17-May-2016].

[5] ‘RSA Laboratories - Contest Rules’. [Online]. Available: http://www.emc.com/emc-plus/rsa-labs/historical/contest-rules.htm. [Accessed: 12-Apr-2016].

[6] F. PUB, *Data Encryption Standard (DES)*. 1999.

[7] M. Curtin and J. Dolske, ‘A Brute Force Search of DES Keyspace’.

[8] D. McNett, ‘[RC5] [ADMIN] The secret message is...’, 24-Feb-1998.

[9] Electronic Frontier Foundation, Ed., *Cracking DES: secrets of encryption research, wiretap politics & chip design*, 1st ed. San Francisco, CA: Electronic Frontier Foundation, 1998.

[10] ‘EFF DES Cracker Press Release, July 17, 1998’. [Online]. Available: https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_descracker_pressrel.html. [Accessed: 12-Apr-2016].

[11] ‘distributed.net: Project DES’. [Online]. Available: http://www.distributed.net/DES. [Accessed: 12-Apr-2016].

[12] ‘RSA Laboratories - DES Challenge III’. [Online]. Available: http://www.emc.com/emc-plus/rsa-labs/historical/des-challenge-iii.htm. [Accessed: 12-Apr-2016].

[13] ‘Brute force attacks on cryptographic keys’. [Online]. Available: http://www.cl.cam.ac.uk/~rnc1/brute.html. [Accessed: 12-Apr-2016].

[14] S. Srinivasan, *Cloud Computing Basics*. Springer, 2014.

[15] ‘Azure infrastructure services implementation guidelines’. [Online]. Available: https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-infrastructure-service-guidelines/. [Accessed: 13-Apr-2016].

[16] ‘Pricing - Virtual Machines (VMs) | Microsoft Azure’. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/virtual-machines/. [Accessed: 13-Apr-2016].

[17] ‘Optimizing Performance on Hyper-V’. [Online]. Available: https://msdn.microsoft.com/en-us/library/cc768529(v=bts.10).aspx. [Accessed: 23-May-2016].

From a New Paradigm to Consistent Representation

Gordana Rakić
University of Novi Sad
Trg D. Obradovića 4
21000 Novi Sad, Serbia
goca@dmi.uns.ac.rs

Jozef Kolek
University of Novi Sad
Trg D. Obradovića 4
21000 Novi Sad, Serbia
jkolek@gmail.com

Zoran Budimac
University of Novi Sad
Trg D. Obradovića 4
21000 Novi Sad, Serbia
zjb@dmi.uns.ac.rs

ABSTRACT

In this paper, a method for mapping between language constructs that belong to different programming paradigms is provided. The method is based on an universal source code representation used by Set of Software Quality Static Analyzers (SSQSA) platform, and motivated by need to consistently support different paradigms by static analysis. The method is illustrated by an example of integration of support for functional paradigm.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Language Constructs and Features

General Terms

Languages

Keywords

SSQSA, eCST, Functional Languages, Scheme

1. INTRODUCTION

Static analysis of computer pqrograms is the analysis that is performed without actual execution of programs. Static analysis is mostly performed on source code of computer program or on some intermediate representation of it (e.g. an intermediate code, a tree, a graph, a combination of previous ones, or even some complex meta-model). Systematic and consistent application of static analysis techniques can significantly improve the quality of a software product (to find weak points, discover bad design, bad maintainability, etc.). Static analysis is usually done by specialized tools. However, in practice they suffer from several weaknesses (e.g. limitations regarding supported languages) [6]. Furthermore, it is shown that different tools give differqent results for the same metrics applied on the same source code [4],[5].

Characteristics of contemporary software projects are that they sometimes last for decades, while during decades they become complex and heterogeneous with respect to technologies and languages. A characteristic example are software products where business logic is developed in some dynamic multi-paradigm language, where functional paradigm is always very popular. These business components are often hidden behind modern user interfaces developed in, lanuages desingned for that. Even if functional paradigm is not well supported by static analysis tools, there is an interest coming from practice for improvements in this area. Some language-specific tools are already in mature phase of development [2].

Described conditions bring us to the very difficult task of reconciliation of opposed objectives - heterogeneous projects are to be consistently supported by static analysis. This support has to involve multiple tools because of limitations of available ones, but we cannot rely on consistency of analysis results among tools. Solution is to achieve consistency of static analysis by involving only one universal tool that will support all languages, technologies and platforms. SSQSA platform [6] is on a good way to meet these goals.

2. SSQSA AND ECST

Set of Software Quality Static Analyzers (SSQSA) is a platform for building and integration of a set of software tools for static analysis. Starting aim of the framework is consistent software quality analysis for projects developed in multiple languages, paradigms, and technologies. Essential characteristics of SSQSA platform are its:
(1)**Extendibility by new analysis.** All implementations of analysis algorithms are independent of the input programming language and each of the integrated analyzers can be uniformly applied to software systems that are written in different programming languages. Furthermore, after integration of a new analysis it is applicable to all languages supported by SSQSA.
(2)**Adaptability to a new language.** Support for a new language can be integrated. After adding a new language, whole set of implemented analyses is immediately applicable to it. Introducing the support for a new input language into SSQSA framework is a straightforward semi-automated procedure [6]. For these purposes we need an appropriate (formal) specification of the programming language. Afterwards, we should only follow the steps of the established procedure.
These characteristics of SSQSA platform are based on universality of enriched Concrete Syntax Tree (eCST)[6]. eCST is based on concepts of syntax trees. It contains full source

code without abstractions enriched with universal nodes. Universal nodes are predefined so-called imaginary nodes with language independent meanings which denote semantic concepts expressed by specific constructs of a language (e.g. LOOP_STATEMENT is used to denote any loop expressed by *for, while, do, repeats*, etc. depending on the language).

Currently, SSQSA supports representative set input languages, while support for functional languages is still weak. Namely, integration of functional languages such is Erlang or Scala is in the testing phase. However, support for a clean functional language such is Scheme is still not introduced. In this paper we describe a motivation and an approach to integration of functional language. Focus of the paper is on mapping of functional constructs to eCST universal nodes. We map some of the most characteristic functional constructs written in Scheme to illustrate the approach.

3. MAPPING APPROACH

As it was mentioned earlier the translation of programming language Scheme into the eCST is done by adding universal nodes into generated syntax tree by the specification. Here, we are describing how the particular syntax elements are marked by the corresponding universal nodes. We focus on some constructs characteristic for functional languages to demonstrate the approach, while the rest of constructs will be just mentioned.

Our approach is based on previous experience. When considering a specific construct in Scheme, concrete mapping method consists of following steps: (1) Determine the construct which is to be mapped (e.qg. code fragment) (2) Determine its semantic (e.g. definition of the function) (3) Determine all factors participating in it (e.g. argument declarations body or statements) (4) Compare the role of all factors with other supported languages in order to find an equivalence (5) Define mapping which is consistent with supported languages

4. CASE STUDY: SCHEME

The programming language Scheme is a functional, and dynamically typed programming language. It is based on mathematical concept called lambda calculus, introduced by Alonzo Church [1]. Although the lambda calculus is very powerful concept which can be used to write any program, it is not the most practical approach. Therefore, Scheme brings some minor modifications of it. Unlike basic lambda calculus, in the Scheme lambda expression can bind several variables at once. It also contains constants, numbers, data structures, and so on. Also, Scheme contains various programming language constructs, assignment operation, environment of defined names, libraries for input/output operations, etc. The language Scheme adds to basic lambda calculus all the necessary features that a practical programming language needs. Thus, Scheme becomes a simple but very expressive programming language. It finds its place in education and also in software industry.

There are two assumptions before mapping Scheme to eCST. (1) Scheme symbol can be redefined in any way, without any restriction. For instance, the expression if can be easily redefined using the function *(define (if x y) (+ x y))*, where if becomes a function that makes a sum of two numbers, and returns the result. We assume that redefinitions of important syntax constructs are not performed. (2) Scheme support macros. It is supposed that, all Scheme code that is used as input to SSQSA is a Scheme code with already expanded macros. Therefore, the macro free Scheme code is expected, and macros are not considered in this paper.

In this section we are passing through characteristic constructs of Scheme languagqe, level by level.

4.1 High-level entities

The largest entity that has to be marked is a compilation unit. Compilation unit is marked using the universal node COMPILATION_UNIT which is always the root node of a single source unit that has to be compiled or interpreted separately. To make a parallel to other languages, compilation unit is a single compilable unit (e.g. class or module), usually determined by an input file. Scheme compilation unit consists of expression sequence where in the most Scheme interpreter implementations the last expression is evaluated when unit is loaded.

Scheme entity can be Scheme programs and library. Scheme libraries can import and export functions. The universal node PACKAGE_DECL which is used for marking a program packages must be child of the COMPILATION_UNIT. Even the Scheme does not have packages in the real sense, each compilation unit is marked with this node. Names that are imported from libraries are marked with the universal node IMPORT_DECL, whose direct child must be NAME which marks identifiers.

At the next level of hierarchy in Scheme entity we can find variable or function definition. Another important construct in any program is block.

4.2 Block

Scheme defines sequences or special expressions that are used for grouping of other expressions. These sequences are defined using keyword begin. The last expression in the body of *begin* block returns its value as a value of the block. Sequences are nothing more than a scopes without locally declared variables. It can be compared to block between *BEGIN* and *END* in Modula-2 or between { and } in Java. Sequences, starting with expression *begin*, are placed in sub-tree of the universal node BLOCK_SCOPE.

Let expressions in Scheme represents expressions with the scope and locally defined variables. There are four different let expressions: *let, let*, letrec*, and named *let.* Example of basic *let* expression would be:

```
(let ((x 10) (y 20)) (+ x y))
```

This is a block with two declared variables and one operation on them, i.e. statement. Variables x and y are bound to the numbers 10 and 20 respectively and the whole let expression returns the sum of these two values. When translating Scheme into eCST the expression *let, let**, and *letrec* are treated equally and they are marked by universal node BLOCK_SCOPE, The named *let* is treated as a function, because it can recursively call itself (Section 4.4.3).

4.3 Variables

In Scheme variables are declared and defined using *define* or *let* expression. Following examples of using *define* and *let* to define a variable are equivalent, while *let* is usually used only inside the function body.

```
(define x 10) or (let x 10)
```

These are corresponding constructs to variable declaration with initialization in any other language, e.g. *int* $i = 10$ in Java. In both cases, a variable declaration is marked with universal node VAR_DECL. VAR_DECL has the universal node TYPE as a direct child. In Scheme a type of newly declared variable is determined implicitly, thus TYPE subtree stays empty until we determine types. This is a task for eCST Manipulator [6]. Consistent post-processing of dynamic types is planned for future work (Section 6). Initialization is observed as an assignment statement and inside it variable name is marked with the NAME and value by node VALUE. The x is the name of variable and 10 is value that variable x is bound to.

4.4 Functions

A Scheme functions are defined using define and let expressions, as well. There are several approaches to define functions. In all cases it is marked using the universal node FUNCTION_DECL, list of parameters is marked using FORMAL_PARAM_LIST, and each parameters marked by PARAMETER_DECL, which is similar as for the first approach. The node NAME marks the function name. Similarly, like variables, parameters have their name and type. Inside function body we can find different expressions (i.e. statements).

4.4.1 Define

The first approach to declare function is mostly used in practice. For example:

```
(define (sum x y) (+ x y))
```

This is an equivalent case as definition of, for example, procedure in Modula-2 or method in Java. Thus, this function declaration is marked by universal nodes FUNCTION_DECL, FORMAL_PARAM_LIST, and PARAMETER_DECL, as described. The node NAME marks the function name, which is *sum* in this particular case, and TYPE remains empty. Parameters also have their name and type. Names are x and y, while type is temporarily empty.

4.4.2 Define lambda

The second approach to function definition is by using keyword lambda. Lambda function is treated as an anonymous function bounded to a variable. Analogy which can be used when observing these variables whose type is anonymous function are procedural types in the programming language Modula-2. Example of the function defined by this approach is:

```
(define sum (lambda (x y) (+ x y)))
```

This can be observed as a variable whose type is the lambda function. Therefore, the root node is VAR_DECL, with two children nodes NAME (*sum*) and TYPE with whole lambda function in the subtree. Lambda function is again marked by FUNCTION_DECL, FORMAL_PARAM_LIST, and PARAMETER_DECL, as described, while the node NAME of FUNCTION_DECL remains empty in this case.

4.4.3 Let

A special case of the *Let* block is named let. It is used to express tail-recursion. It can be observed as a function that can be called only from its body. Therefore it is a function with certain restrictions on syntactical level. However, once when this function is defined according to language rules it has all characteristics of recursive function. For example:

```
(define (factorial x)
  (let loop ((x x) acc 1))
    (if (zero? x) acc
          (loop (sub1 x) (* x acc)))))
```

It is obvious that this is equivalent to recursive function definition in any other language. The main difference is that other languages usually do not require explicit syntax constructs for recursive function. Named *let* is marked by universal nodes used for other function definitions, where name of the *let* block is a name of the function.

4.5 Statements

Blocks and function bodies are built from statements. Statements in Scheme vary from simple expressions to complex ones such are branch statements, loop statements and continuation statements.

4.5.1 Function calls

Scheme comes with the two possible ways that functions can be called. For example:

```
(sum a b c), or (apply sum a b '(1 2 3))
```

The first way one is the mostly used. The second way is explicit call of function by using command apply. The main difference is in the way they are executed, while the meaning is the same. Both types of function calls are marked by the universal node FUNCTION_CALL, whose direct children nodes are NAME and ARGUMENT_LIST. The node ARGUMENT_LIST is used to mark a list of actual parameters, and ARGUMENT is used to mark each argument in the list.

4.5.2 Branch statements

In Scheme there are many of conditional expressions: if, not, and, or, cond, when, unless, and case. The if expression is equivalent to conditional expression in Java-like languages. For example, following expressions are equivalent.

```
(if (< x y) \#t \#f), and
(condition ? consequent : alternative)
```

A conditional expression is marked using the universal node BRANCH_STATEMENT, condition is marked using CONDITION, while consequent and alternative are marked using BRANCH as a direct child of the BRANCH_STATEMENT. The conditional expressions *not, and* and *or* are marked by universal node LOGICAL_OPERATOR.

Comparison of Agile Methods: Scrum, Kanban, and Scrumban

Lucija Brezočnik
Faculty of Electrical Engineering
and Computer Science,
University of Maribor,
Maribor, Slovenia
lucija.brezocnik@um.si

Črtomir Majer
Faculty of Electrical Engineering
and Computer Science,
University of Maribor,
Maribor, Slovenia
crtomir.majer1@um.si

ABSTRACT

In software development, companies are forced to introduce changes in the way they manage a project's development because of ever-shorter cycles and ongoing changing requirements. The changes in development projects are frequently conducted by the introduction of agile methods, which have in recent decades sharply increased in popularity. But a major question remains: "Which agile method is optimal for our company?" In order to answer this question, we compared the three most prevalent among them: Scrum, Kanban and Scrumban.

Categories and Subject Descriptors

D.2.9. [**Software Engineering**] – Management
D.2.10. [**Software Engineering**] - Design

General Terms

Management, Design, Theory.

Keywords

agile software development, agile methods, Scrum, Kanban, Scrumban

1. INTRODUCTION

Software companies switch to agile development mostly due to the desire to accelerate product delivery, enhance the ability to manage fast-changing priorities, to increase productivity, and to improve software quality [12]. Interestingly, the cost of the project and maintenance of software has no significant impact in making the transition [5, 12]. From this, we can conclude that the biggest problem in the traditional approach is a period of software development and a decreased ability to manage changing priorities. But that is precisely what is most important for customers [3].

In this paper, we focus on three agile methods – Scrum, Kanban, and Scrumban. Research [5] has shown that about half of businesses still use the waterfall model, while the other half uses agile and iterative approaches. Companies using agile methods, according to data from the tenth annual survey VersionOne [12], most often opt for Scrum and Scrum + XP (70%), Scrumban (7%) and Kanban (5%). From our selection of agile methods, we removed Extreme Programming (XP), because its principles are often used in combination with other methods (Scrum, Kanban).

2. AGILE METHODS

The main point of agile methods is the constant embrace of changes, which is in contrast with traditional methods. Changes are a natural part of development projects and as such should be adequately addressed [8].

2.1 Scrum

Scrum [8, 9, 10] is an agile framework that comprises principles and practices that help teams deliver new products as soon as possible with continual improvements and with rapid adaptation to changes. Scrum has three roles: Product Owner (the voice of the customer who is responsible for the ROI and should not be mistaken with the product manager), the Scrum Master (who observes the team, ensures that there are no violations in the Scrum rules, and removes any impediments that the team may have), and the Team (cross-functional team that is responsible for delivering shippable increments of product at the end of each Sprint).

The Sprint is a fixed-length iteration and represents the basic unit of development. Before each Sprint, the Sprint Planning event takes place in which the Sprint Backlog is defined. All Sprints end with a Sprint Review and a Sprint Retrospective. In the Sprint Review, the Team and Product Owner are involved and seek opportunities for improvement. The Sprint Retrospective convenes a Scrum Master and tries to optimize the development process itself.

2.2 Kanban

Kanban is a process management method developed at Toyota and builds on the experience of other agile methods. The main objective of Kanban is the elimination of delays and waste, which has a positive effect on workflow optimization. It is based on the Just-In-Time technique for task scheduling, which requires the precise definition and implementation of a task as late as possible in the workflow, to get rid of unnecessary re-planning [4, 6]. The three basic guidelines for the Kanban method are:

- Visualize workflow. That is typically done with the Kanban board, which clearly defines all the required steps (board columns) of the development process. Tasks are prioritised and put in the board column that best defines the current state of the task. Tasks are moved between states until they get into the Done state – the goal is to finish tasks that are already in the flow as soon as possible instead of starting new tasks [6].
- Limit Work in Progress (WIP). Each step in the process must have a WIP limit, optimized according to complexity, in terms of the number of workers and other parameters. The WIP limit forces us to focus on one task at a time instead of doing multiple things concurrently. It

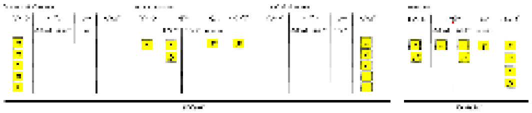


Figure 1: Board comparison in Scrum and Kanban

follows the “achieve more by doing less” principle, which has repeatedly been proven true [4, 6].

- The “pull” principle. When moving tasks between stages, we must obey the pull rule, which states that we can only take some new task in a certain stage if the WIP limit has not already been reached. This helps us with the early identification of delays and impediments the workflow, thus encouraging teamwork.
- Minimalize, measure and improve. Kanban maintains existing teams, processes, roles and responsibilities of the team – it introduces minimal changes for its adoption. It establishes some control over process flow, but keeps the existing approaches that work well in place. Kanban encourages the usage of agile metrics to measure performance, monitor the progress and improve workflow efficiency [4, 6].

2.3 Scrumban

Scrumban is a composite of Scrum and Kanban methods, as it contains the basic properties of Scrum and the flexibility of Kanban. Long-term development goals, in Scrumban, are defined via bucket size planning. Each bucket contains a development plan, that needs to be realised within a given time, for example: three months for the nearest bucket. This bucket holds fine-grain definitions of tasks, while buckets that represent long-term plans, for example, a 1-year bucket, hold only a draft – those buckets are deficient [7]. That is due to the Just-In-Time principle taken from Kanban, which urges us to make fine-grained plans as late as possible. Just like Kanban, Scrumban also limits the Work-in-Progress and enforces the “pull” principle for moving tasks between stages [1]. Scrumban does not require any new roles (like Scrum), however, it encourages short daily meetings and kaizen events that are meant for the resolution of everyday impediments [2, 11]. Scrumban stipulates that iterations should not be longer than two weeks, but unlike Scrum, it allows for long-running tasks which can extend across several iterations. This can lead to an incomplete product at the end of the iteration, which is why Scrumban has introduced a Feature Freeze (FF). When the team is approaching the end of the current iteration, it stops working on new features, and instead focuses on finishing those already in process. Features that are still incomplete need to be disabled or removed from the final product, so the incremental release can be made [2, 7, 11].

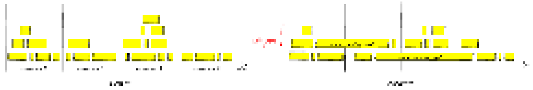


Figure 2: Tasks in iterations in Scrum (left) and in Kanban (right)

3. COMPARISON OF AGILE METHODS

Hereafter, we will compare the methods according to the 12 main perspectives.

3.1 Board

The board is used in virtually all methods but it differs in terms of how we use it. The Scrumban board is reset with each Sprint, which means that all tasks are put into the ToDo column. When using Kanban, resets do not occur, because there are no iterations – new tasks are provided in a constant flow. The Scrumban board typically looks like a Kanban board, but we can experience some resets when finishing the current bucket and moving to the next one.

3.2 Artifacts

Scrum requires a clearly defined product backlog, sprint backlog and burndown chart, thus requiring more effort from the team to keep artefacts up-to-date, compared to Scrumban and Kanban. While Kanban does not demand any specific artefact, Scrumban requires an iteration backlog and bucket plans.

3.3 Iterations

Scrum defines iterations (called Sprints) as part of the Scrum lifecycle. They can last from one to three weeks. At the end of every Sprint, we expect a totally functional product with new features or other upgrades that are accepted by the product owner. Scrumban also has iterations, which are not strictly defined in terms of tasks and length; however, their duration should not be longer than two weeks since shorter iterations allow for a more rapid adaptation to change. Kanban does not define iterations, as new tasks are defined on demand as late as possible.

3.4 Tasks

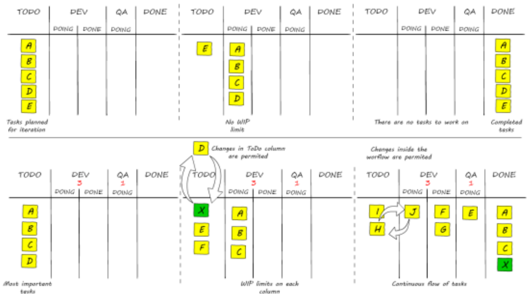

TODO
DEV
QA
DONE
DOING
DONE
DOING
A
B
C
D
E
Tasks planned for iteration
No WIP limit
There are no tasks to work on
Completed tasks
Changes in ToDo column are permited
Changes inside the worflow are permited
3
1
X
F
G
H
I
J
Most important tasks
WIP limits on each column
Continuous flow of tasks

Figure 3: Changes in work plan in Scrum (above) and in Kanban (below)

The time span of each task in Scrum is limited to the duration of Sprint. In any case, we try to break such long tasks into smaller ones, so that no task is longer than one day. Kanban and Scrumban do not limit the time span of tasks. Even though Scrumban has iterations, it allows long-running tasks (Figure 1).

3.5 Priority

With Scrum, task prioritisation is made when planning the Sprint, while with Kanban task prioritisation is made on a daily basis with just-in-time planning and the pull principle. Whenever a new task is pulled into the workflow, it must have the highest priority for the team. Scrumban first prioritises work with bucket size planning, after which tasks are defined and prioritised for each iteration and, lastly, on a daily basis as is the case with Kanban.

3.6 Work estimation

Scrum prescribes task estimation before each Sprint, while Kanban and Scrumban do not require estimation. Some teams prefer to define tasks in such a manner that all tasks have similar complexities, thus requiring approximately the same time for completion.

3.7 Team

Scrum teams must be cross-functional, which means they are able to provide product increment entirely on their own (from planning to deployment). Kanban and Scrumban allow cross-functional and specialised teams, depending on the product type and what works best for a given scenario.

3.8 Roles

Scrum prescribes the following roles: product owner, development team and Scrum master. The product owner is responsible for TODO, and the Scrum Master is responsible for daily meetings and solving the non-technical problems of the team. Kanban and Scrumban do not define any special roles, so that tasks for maintaining the agile method are divided among team members.

3.9 Changes in work plan

Scrum does not allow any changes in the work plan when Sprint is running – that is why we make detailed plans and estimations before Sprint, and do not make any changes (Figure 3, above). Scrumban and Kanban provide no rules that forbid changes in the work plan at any given time (Figure 3, below). The tasks in the ToDo state can be easily replaced with new ones. Also, tasks that are already in process can be taken back to ToDo and more important tasks can be pulled in.

3.10 Bug fixing

There are two types of software faults, those that appear at the time of development (often called defects) and ones that appear after software is released and running in a real-time environment, called bugs. Kanban and Scrumban allow unplanned bug-fixing right away – if fixing a bug has a higher priority than current tasks in ToDo, then this task is put on the board. With Scrum, bug-fixing is by-the-book planned for the next Sprint – it would be unreasonable to change the current Sprint plan due to all the preparations and estimations that are done before the Sprint. In reality, we know that critical bugs must be fixed as soon as possible, so Scrum teams take different approaches to tackling this problem. Some teams define one day of a week (or part of a day) as a "bug fixing day," other teams reduce the number of story points for Sprint, so that there is still some time left for unexpected things, like fixing bugs.

3.11 Stress

Research show that stress is highly correlated to the amount of work that a team member is responsible for. The ideal workload per person would be evenly distributed to their optimal level. A person must not feel too much of a burden on their shoulders, thus leading to exhaustion, nor too free, which leads to poor progress (Figure 4). Team members must see a constant improvement in the product, which keeps them motivated and dedicated. With Kanban, we can achieve a mostly evenly distributed workload because there are no iterations, and thus tasks are continuously added to the workflow. Sprints in Scrum are time-limited (typically from 2 to 4 weeks), so there is often more work done at the end of the Sprint than in the beginning (Figure 5). Scrumban is somewhere in the middle of those two, because it allows long-running tasks, so team members are not so stressed if some tasks are not completed. For highly motivated and self-initiative teams, Kanban can be a good fit. Teams that do not have such properties need time limits in which some progress is expected, thus Scrum and Scrumban provide a better match.

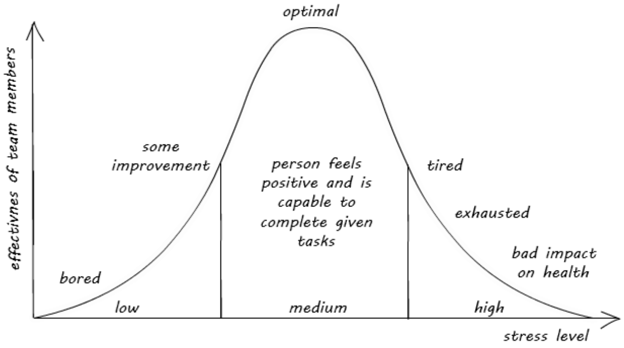


Figure 4: Graph of stress levels, depending on the work done

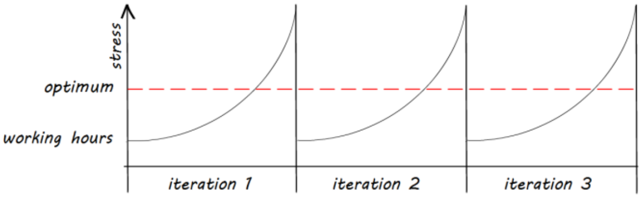


Figure 5: Stress level through iterations

3.12 Activities to maintain the agile method

Scrum activities to keep the method alive, consist of an up-to-date backlog, a Sprint backlog, daily meetings, a board and retrospective. Kanban requires visualisation of a workflow (typically a board) and demands the respect of Work-in-Progress limits for each stage of the process. Scrumban extends Kanban activities and adds bucket size planning, daily events (standups) and iteration planning.

4. CONCLUSION

In this paper, we presented the most widespread agile methods: Scrum, Kanban and Scrumban. Each method has its own advantages and disadvantages, but it is necessary to bear in mind that none of them will benefit businesses, if not used in the right way. It is, therefore, important to choose the one agile method that best meets the requirements and wishes of the company.

Scrum certainly works best in mature companies that have experienced teams who have been working on the product or project for more than one year. For companies with continuous production that need a rapid response to changes and product teams that are working in support and maintenance of the product, we recommend using Kanban. Scrumban is best for young, small companies since it boasts the flexibility of Kanban and the basic characteristics of Scrum.

Agile methods definitely include a strong component of flexibility. Teams could, regardless of the method chosen, adapt it in a way that would serve their purpose – i.e. an effective work organisation and the development of quality products.

5. REFERENCES

[1] Baleviciute, G. 2014. *Whitepaper – Scrum vs Kanban vs. Scrumban*. Retrieved September 10, 2016, from http://goo.gl/dkrbGE.

[2] Bieliūnas, E. 2014. *Scrum-ban for Project Management*. Retrieved September 10, 2016, from http://goo.gl/JgfaaA.

[3] Bittner K., Lo Giudice D., DeMartine A., Mines C., Hammond J. S., Turrisi T., and Izzi M. 2016. Forrester Research – *Boost Application Delivery Speed And Quality With Agile DevOps Practices*.

[4] Brechner, E. 2015. *Agile Project Management with Kanban*, Microsoft Press.

[5] Gartner. 2015. Holz B. presentation “Agile in the Enterprise”.

[6] Klipp P. 2014. *Getting Started with Kanban*, Amazon Digital Services LLC.

[7] Misevičiūtė, D. 2014. *Scrumban: on demand vs. long-term planning*. Retrieved September 10, 2016, from http://www.eylean.com/blog/2014/11/scrumban-on-demand-vs-long-term-planning/.

[8] Pichler, R. 2010. *Agile Product Management with Scrum: Creating Products That Customers Love*. Addison Wesley.

[9] Swisher, W. P. 2014. *Implementing Scrumban*. Retrieved September 10, 2016, from https://switchingtoscrum.files.wordpress.com/2013/12/implementing-scrumban_v1-32.pdf.

[10] VersionOne. 2016. *VersionOne 10th Annual State of Agile Report*.

[11] Sutherland J., and Schwaber, K. 2013. *The Scrum Guide*. Retrieved September 10, 2016, from http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf.

[12] Sutherland J. 2010. *Scrum Handbook*, The Scrum Training Institute.

4.5.3 Loop statement

Scheme has a loop expression do which can be compared to for statement in programming language Java.

```
(do ((v (make-vector 5))(i 0 (+ i 1)))
        ((= i 5) v) (vector-set! v i i))
```

Loop statements are marked by using the universal node LOOP_STATEMENT. However, characteristic approach for dealing with repetitions in functional languages is recursion. In eCST recursive functions are marked as regular functions (Section 4.4.3, while semantic transformations are are planed for future work.

4.5.4 Continuations

First class continuations of computer programs are constructions that are representing program state which can be saved as value of variable, to be used at a later point in the program. Programming language Scheme implements these first class continuations by an operator *call-with-current-continuation*. When translating Scheme into eCST the continuation call is marked using JUMP_STATEMENT, since continuations are changing the control flow of program. An operator *call-with-current-continuation* is marked by using OPERATOR.

5. RELATED WORK

Before definition of this method some languages were mapped to eCST and integrated into SSQSA [6]. These were mainly imperative languages, and mapping among their constructs was more logical. However, Erlang as a functional language was integrated up to prototype level [7], while some issues remained unsolved.

Authors of [3] tried to cross the gap between imperative and functional programming by refactoring. They were motivated by integration of functional paradigm in Java programming language, and the goal was to provide Java developers with refactoring techniques that will lead them to functional code. Basically, this is a kind of mapping between two paradigms and can be useful in our research for a comparison of approaches. However, they provide only two refactoring methods, focused on two new Java features, while other constructs are not covered.

6. CONCLUSIONS AND FUTURE WORK

In this paper we describe a method for mapping constructs, that belong to a new paradigm, to the eCST in SSQSA platform. We illustrate it by introducing functional paradigm and Scheme as a clean functional language. In that way we provide a double contribution of this paper: (1) determined rules for mapping functional language to eCST to be followed while integrating language which includes functional paradigm, and (2) determined a method to be applied while introducing support for any new paradigm. The method recommends first to choose a language which is clean represent of the paradigm to be integrated. Afterward, we are passing through all paradigm-specific constructs, analysing them and comparing with similar constructs from other, already supported paradigms, determining equivalent ones, and specifying concrete mapping. Finally, the mapping defined on a clean language is to be applied whenever we need mapping of this paradigm to eCST. In case that we find some new construct that belongs to already supported paradigm, we can apply the same procedure to meet the consistency of mapping.

Furthermore, this method provides SSQSA with consistency among languages and paradigms. Namely, when we are integrating a multi-paradigm language, we are determining paradigms included in that language, recognise which construct belongs to which paradigm, and map each paradigm separately according to the defined method. This is applied to each new language and each new paradigm.

There are still some open questions to be addressed in future work. They are related to more general issues. One of these question is: How to map implicitly defined types is dynamically typed languages? Next question is related to similarities and differences between iterations and recursions. This topic especially rises with control-flow analysis where two kinds of repetitions should be consistently analyses. Nevertheless, these are not problems related only to functional languages, while all aspects of these issues are consistently mapped to eCST among integrated languages. Therefore, they are not subject of this paper, but will be subject of improvements of SSQSA platform. The future work directly related to integration of Scheme and functional paradigm into SSQSA is testing the analysers over new datasets that will contain code written in functional languages.

7. REFERENCES

[1] H. P. Barendregt and E. Barendsen. Introduction to lambda calculus. *Nieuw archief voor wisenkunde*, 4(2):337–372, 1984.

[2] I. Bozó, D. Horpácsi, Z. Horváth, R. Kitlei, J. Kőszegi, M. Tejfel, and M. Tóth. RefactorErl – Source Code Analysis and Refactoring in Erlang. In *Proc. of the 12th Symposium on Programming Languages and Software Tools*, pages 138–148, Tallin, Estonia, October 2011.

[3] A. Gyori, L. Franklin, D. Dig, and J. Lahoda. Crossing the gap from imperative to functional programming through refactoring. In *Proc. of the 2013 9th Joint Meeting on Foundations of Software Engineering*, pages 543–553. ACM, 2013.

[4] R. Lincke, J. Lundberg, and W. Löwe. Comparing software metrics tools. In *Proc. of the International Symposium on Software Testing and Analysis*, ISSTA '08, pages 131–142, Seattle, WA, USA, 2008. ACM, New York, NY, USA.

[5] J. Novak and G. Rakić. Comparison of software metrics tools for: net. In *Proc. of 13th International Multiconference Information Society (IS'10)*, pages 231–234, Ljubljana, Slovenia, 2010.

[6] G. Rakić. Extendable and adaptable framework for input language independent static analysis, 2015.

[7] M. Tóth, A. Páter-Részeg, and G. Rakić. Introducing support for erlang into ssqsa framework. In *Proc. Of The International Conference On Numerical Analysis And Applied Mathematics 2014 (ICNAAM-2014)*, volume 1648, page 310012. AIP Publishing, 2015.

Introduction to Case Management Model and Notation

Mateja Kocbek
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
mateja.kocbek@um.si

Gregor Polančič
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
gregor.polancic@um.si

ABSTRACT

A case is presented as a proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome. Cases are used in many areas of human operations. The most common example of a case is from medicine, where every patient represents its own case. Every case requires its own operations and functions whereas sometimes humans, who are involved, can use their knowledge from previous cases. This article presents a new standard, called CMMN (Case Management Model and Notation), which has recently been published by OMG. It covers the whole process of case management. The presentation of standard CMMN includes abstract and concrete syntax as well the semantics and diagram interchange specifications.

Categories and Subject Descriptors

I.6.5 [**Simulation and modelling**]: Model Development - *Modelling methodologies*.

General Terms

Management, Documentation, Performance, Standardization, Design, Languages, Theory.

Keywords

CMMN, Case Management Model and Notation, Case Management, BPMN.

1. INTRODUCTION

In everyday life, many different cases can be found. A case is a very common term and can represent a variety of different things or concepts. Its common definition is *"a particular situation or example of something"* [11], whereas in CMMN specification [5], a case is presented as *"a proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome"*.

An illustrative example of listed definitions of cases, can be found in medicine, where a case involves the care of a patient, together with his/her medical history as well as current situation. Other examples of cases are: a law case, social security case, employment case, etc. A project-related case definition states that *"a case is a project, transaction, service or response that has different states (for example: opened, doing, closed) over a period of time to achieve resolution of a problem, claim, request, proposal, development or other complex activity"* [13].

A case always contains some kind of a subject which may be a person, a legal action, a business transaction, or some other focal point around which actions are taken to achieve an objective [5]. Besides, resolving a specific case usually requires a lot of information [5], whereas new cases, with no previous experience of involved individuals, can be resolved intuitively [5].

As mentioned above, resolving a case includes information, actions, human resources, knowledge, etc., which can be united in case management. Case management is usually driven by a team of case/knowledge workers, who make decisions or perform certain tasks [5].

One of the most important characteristics of case management is planning. Every case requires a high degree of flexibility, which is essential for the success of human activities. Flexibility is needed with selection of tasks for a case, run-time ordering of the sequence in which the tasks are executed, and ad-hoc collaboration with other knowledge workers on the tasks [5]. Case or knowledge workers are those, who have to determine which tasks are applicable, or which follow-up tasks are required to perform [5]. Decisions may be triggered by events or new facts that continuously emerge during the course of the case, i.e. the receipt of a new document, completion of certain tasks, or achievement of certain milestones [5].

In 2014, Case Management Model and Notation (or CMMN) was introduced, by OMG (Object Management Group), as a standard for case management [4]. This article focuses on CMMN, with the following structure. Chapter 2 gives an overview of CMMN. In chapter 3, the actual use of standard is presented. We will conclude our article with discussion and conclusion.

2. RATIONALE FOR INTRODUCING CMMN

CMMN is in general a graphical representation for expressing a case [10]. It provides an efficient notation for capturing less repeatable, dynamic, information-rich contexts. CMMN was introduced to document the ad-hoc scenarios faced by knowledge workers in which they need to respond to a continuous flow of business events, data and documents. The CMMN specification defines abstract elements, notation, execution semantics and exchange formats [5]. A consortium of 11 companies contributed to the development of CMMN, which is being maintained by the OMG. Version 1.0 of CMMN was released in May 2014 [5], currently CMMN version is 1.1 – Beta [4].

2.1 CMMN versus BPMN

The focal rationale for CMMN introduction was a need for more flexibility for knowledge workers when modelling business processes. Flexibility is needed, because some tasks can be done independently of the time and the sequence of tasks is not important. So, workers can decide which work to do and what order is the best in a particular case. This is the main difference compared to the well accepted business process standard - BPMN. Within BPMN models, an exact order of activities is defined (i.e. structured process), e.g. activity A has to finish before activity B starts. However, the exact order is not always the best way to solve specific instances or cases. A good example is a health case, where knowledge workers (i.e. medical stuff, administration, etc.) do not know precisely in which direction the specific case will evolve. Another illustrative example is also exception handling, where flexibility is welcome. But it is also reasonable to stress that to some level, processes have to be defined. For example, a nurse has to know exactly which steps need to be taken, when a patient comes to a hospital.

Above we discussed the differences between CMMN and BPMN. BPMN is well known, used and accepted standard, but CMMN can fill out the existing weaknesses of BPMN. Currently, CMMN and BPMN are used separately [12].

2.2 CMMN structure

Beside a modelling notation, CMMN defines a meta-model, a XML–based model for Interchange (XMI) and XML-Schema for exchanging Case models among different environments and tools [5].

The meta-model can be used by case management definition tools to define functions and features that a business analyst might use when defining a case model for a particular type of case. The notation is intended to express the model graphically [5].

This specification enables portability of case models, so that users can take a model defined in one CMMN implementation and use it in another one. The CMMN XMI and/or XML-Schema are intended for importing and exporting case models among different CMMN implementers [5].

A case model is intended to be used by a run-time case management product to guide and assist a knowledge worker in the handling of a particular instance of a case, for example a particular invoice discrepancy. The meta-model and notation are used to express a case model in a common notation for a particular type of case, and the resulting model can subsequently be instantiated for the handling of a particular instance of a case [5].

2.3 CMMN Notation

The outermost element that defines a case, is *Case Plan Model* (Figure 1). The various elements of a Case Plan Model are depicted within the boundary of the Case Plan Model shape. The Case Plan Model comprises: all elements that represent the initial plan of the case and, all elements that support the further evolution of the plan through run-time planning by case workers.

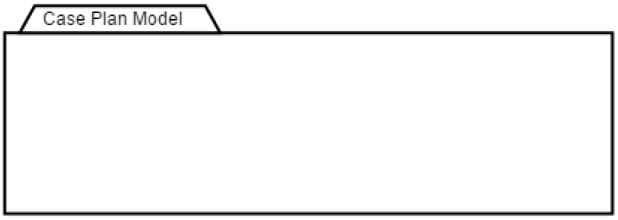


Figure 1: Case Plan Model

All information, or references to information, that is required as context for managing a Case, is defined by exactly one *Case File*. A Case File is meant as a logical model. It does not imply any assumptions about physical storage of information. A Case File contains *Case File Items* (Figure 2) that can be anything from a folder or document stored, an entire folder hierarchy referring or containing other Case File Items.

Case management planning is typically concerned with determination of which tasks are applicable, or which follow-up tasks are required. Case workers execute the plan, particularly by performing tasks as planned and adding *Discretionary Tasks* (Figure 3) to the plan of a case instance.

In CMMN planning is a run-time effort. Users (i.e. case workers) are said to "plan" (in run-time), when they select Discretionary Items from a Planning Table, and move them into the plan of the case (instance). A Planning Table defines the scope of planning: *Collapsed Planning Table* (discretionary elements are not visible) and, *Expanded Planning Table* (discretionary elements are visible).

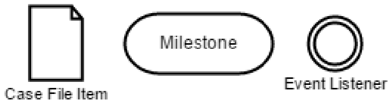


Figure 2: Elements

CMMN defines the following Plan Model Elements: *Stage* – considered as episodes of a Case (shown in Figure 3), *Task* – atomic unit of work during a case (also shown in Figure 3), *Event Listener* – something that happens during the course of a case (shown in Figure 2), *Milestone* – an achievable target defined to enable evaluation of progress of the case (also shown in Figure 2).

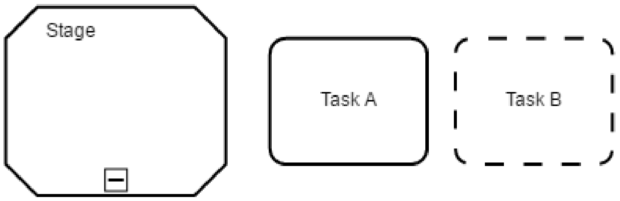


Figure 3: Element Stage, Task and Discretionary Tasks

In CMMN, an event is something that "happens" during the course of a case. Event may trigger the enabling, activation and termination of Stages and Tasks, or the achievement of Milestones. *Standard events* are: Case File Items lifecycle transitions, and Stages, Tasks and Milestones lifecycle transitions. In CMMN there are also *Event Listeners*, that are used to influence the proceeding of the Case directly, instead of indirectly via impacting information in the Case File. There are also two special Event Listeners: *Timer Event Listener*, which is used to catch predefined elapses of time, and *User Event Listener* enables direct interaction of a user with the case.

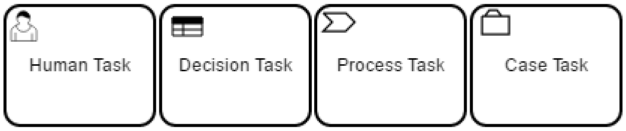


Figure 4: Tasks

CMMN also defines a variety of Tasks (Figure 4): *Human Task* – a non-blocking task, that is not waiting for the work to complete, but it completes immediately upon installation, *Decision Task* – a blocking task, that is waiting until the work associated with the Task is completed, *Process Task* – can be used in the case to initiate a business process, and *Case Task* – can be used to initiate another case.

A Sentry "watches out" for important situations to occur, which influence the further proceedings in a case. A Sentry is a combination of an Event and/or Condition. A Sentry can be used as an entry criterion or as an exit criterion and may consist of two parts: an On-Part specifies the event that serves as trigger, and an If-Part specifies a condition, as Expression that evaluates over the Case File [1,5,7] (Figure 5).

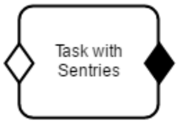


Figure 5: Task with Sentries

Besides, various Decorators can be added to CMMN shapes. Table 1 presents Decorators (Planning Table, Entry Criterion, Exit Criterion, Auto Complete, Manual Activation, Required, Repetition) applicability to CMMN shapes (Case Plan Model, Stage, Task, Milestone, Event Listener, Case File Item, Plan Fragment). Symbol "+" means that a certain shape accepts associated Decorator [5].

Table 1: Decorators Applicability Summary Table [5]

	Planning Table	Entry Criterion	Exit Criterion	Auto Complete	Manual Activation	Required	Repetition
Case Plan Model	+		+	+			
Stage	+	+	+	+	+	+	+
Task	+*	+	+		+	+	+
Milestone		+				+	+
Event Listener							
Case File Item							
Plan Fragment							

*Human Task only.

A case can be considered in ad-hoc manner, which is some kind of equivalent to ad-hoc processes in BPMN, because there is no specific order or sequence of the completion of the tasks. It is also permitted to perform tasks in any frequency [3]. Usually all ad-hoc activities are conducted by human resources, who determinate the sequence, time and frequency of the performance of each activity in ad-hoc process [3].

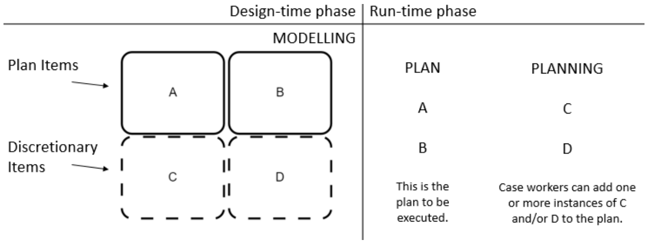

Design-time phase
Run-time phase
MODELLING
Plan Items
Discretionary Items
A
B
C
D
PLAN
A
B
This is the plan to be executed.
PLANNING
C
D
Case workers can add one or more instances of C and/or D to the plan.

Figure 6: Phases of a Case

Besides, a case may be in one of the two phases: *design-time* and *run-time* (Figure 6). During the design-time phase, business analysts engage in modelling, which includes defining: (1) tasks that are always part of pre-defined segments in a case model, and (2) "discretionary" tasks that are available to the case worker, to be applied in addition, to his/her discretion. In the run-time phase, case workers execute the plan, particularly by (1) performing tasks as planned, and (2) adding discretionary tasks to the case plan instance in run-time [3].

As we already mentioned, a very important part in case management is reference to data about the subject of the case. The collection of data about the case is often described as a Case File. Case workers use structured and unstructured data when decision-making [3].

Cases are directed not just by explicit knowledge about the particular Case and its context represented in the Case File, but also by explicit knowledge encoded as rules by business analysts, the tacit knowledge of human participants, and tacit knowledge from the organization or community in which participants are members [3].

3. CURRENT CMMN ACCEPTANCE

The use of standard CMMN is not widespread. It was designed to be used when planning activities, that do not require an exact order. Every group of tasks has to be performed, but the time and sequence are not important. In the following paragraphs, some aspects of use of standard CMMN are discussed.

Table 2 is showing Operating Models used in companies. Models for Coordination, Diversification, Unification and Replication have its own degree of Process Integration and Process Standardization [6] [1]. Table shows that CMMN has low degree of Process Standardization for Coordination and Diversification.

Table 2: Operating models

Process Integration	High	**Coordination**	**Unification**
	Low	**Diversification**	**Replication**
		Low	High
		CMMN	BPMN
		Process Standardization	

According to the fact that CMMN was introduced in 2014 and also that version 1.1 is in its Beta phase, it makes sense that there is not a great number of tools that support standard CMMN. At the time of the survey we detected only two adequate tools. The first tool for modelling with standard CMMN is Camunda [2], an open source platform for Business Process Management. It is suitable for development and provides business-IT-alignment based on BPMN for structured workflows, CMMN for less structured Cases and DMN for business rules [2]. The other tool for standard CMMN is CMMN Modeler (Trisotech) [9]. It is a payable tool.

CMMN was primary designed for business analysts, which are the anticipated users of Case management tools for capturing and formalizing repeatable patterns of common Tasks, Event Listeners, and Milestones into a Case model [5].

3.1 Illustrative Example

In this section, a simple example is presented, in which we collected few common elements of the CMMN, which were also introduced in previous chapter. The example briefly defines a process of *writing a document*, with its basic components.

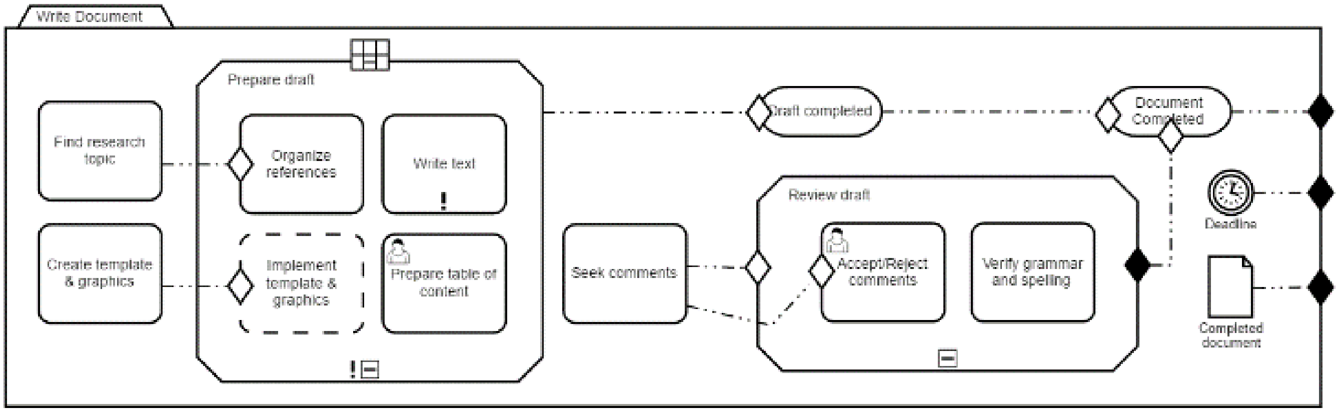


Figure 7: Model of CMMN [8]

Figure 7 represent a CMMN model that encompass the whole process of writing a document. First, in the model, we have two tasks: “Find research topic” and “Create template & graphics”. Initially, any of those tasks can be performed. The next, more extensive element is a Stage, with the name “Prepare draft”. It contains four tasks. The task “Organize references”, is a Tasks with entry criterion (see the symbol in Figure 5). It is mandatory that Tasks, related to this Sentry, perform earlier. The next task “Write Text”, is special because it contains exclamation mark at the bottom of the shape, which means that the performance of this tasks is required. The same symbol (exclamation mark) is positioned on the level of Stage “Prepare draft”. The task “Prepare table of content” is a Human Task, marked with a small human symbol in the left upper corner of the shape. The last task in this Stage is “Implement template & graphics”. It also has a Criterion and it is a Discretionary Tasks, which is symbolized with dotted line.

Later on we can see task “Seek comments” and also Stage “Review draft” with two constituting tasks. The speciality of this part is an exit Criterion (see the symbol in Figure 5). Both used Stages “Prepare draft” and “Review draft” are later on connected to element Milestones with entry Criterion. Two additional elements are also used, namely: Event Listener (Timer) and Case File Item. The first one defines a deadline for completing a document, and the second one contains an actual document. The last important concept, we need to highlight is the Case Model. It is symbolized with a folder and covers the whole described process (also shown in Figure 1). Case model “Write document” includes three exit Criterions.

4. DISCUSSION

In our article, we presented a novel standard for Case Management, - CMMN, which also includes a notation for modelling business processes and graphically expressing a Case. CMMN has some similarities with well-known and accepted standard BPMN. There are some similar elements, like Tasks, Events, Sub process, etc., but there is also very important difference between CMMN and BPMN. BPMN requires accurate knowledge of a business process that is intended to be used when modelling. There is actually no space for flexible execution of business processes. On the opposite, CMMN offers flexibility, which is very welcome (or also required) in many business process cases. As we already mentioned, the CMMN is in its beginnings, but it has a great potential for at least to be used in combination with BPMN. Our intentions for future research are to perform a survey, to recognize the actual acceptance and potential use of CMMN.

5. REFERENCES

[1] Gagne, D. Case Management Model and Notation (CMMN): An Introduction. 2016. https://prezi.com/yu3lbxamg09v/case-management-model-and-notation-cmmn-an-introduction/.

[2] GmbH, C.S. Camunda Tool. 2016. https://camunda.org.

[3] Hinkelmann, K. Case Management Model and Notation - CMMN. 2014. http://knut.hinkelmann.ch/lectures/bpm2013-14/06_CMMN.pdf.

[4] OMG. OMG CMMN. 2014. http://www.omg.org/spec/CMMN/.

[5] OMG (Object Management Group). Case Management Model and Notation 1.0. May (2014), 82.

[6] Ross, J.W., Weill, P., and Robertson, D.C. Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. 2006.

[7] Rucker, B. Camunda BPM 7.2: CMMN Case Management (English). 2015.

[8] Torsten Winterberg. Oracle - CMMN. https://blogs.oracle.com/soacommunity/entry/case_management_model_and_notation.

[9] Trisotech. Trisotech - CMMN Modeler. http://www.trisotech.com/cmmn-modeler.

[10] Wikipedia. Wikipedia - CMMN. https://en.wikipedia.org/wiki/CMMN.

[11] Cambridge Dictionary. https://dictionary.cambridge.org/dictionary/english/case.

[12] BPMN and CMMN Compared. 2014. http://brsilver.com/bpmn-cmmn-compared/.

[13] AIIM - What is Case Management? 2016. http://www.aiim.org/What-is-Case-Management.

Indeks avtorjev / Author index

Konferenca / Conference

Uredili / Edited by

Sodelovanje, programska oprema in storitve v informacijski družbi / Collaboration, Software and Services in Information Society

Marjan Heričko