Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2017

Zvezek G

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY - IS 2017

Volume G

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

Uredil / Edited by
Marjan Heričko

http://is.ijs.si

9.–13. oktober 2017 / 9–13 October 2017
Ljubljana, Slovenia

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek G

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume G

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

Uredil / Edited by

Marjan Heričko

http://is.ijs.si

9. - 13. oktober 2017 / 9th – 13th October 2017
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 2017

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=292477440
ISBN 978-961-264-118-4 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2017

Multikonferenca Informacijska družba (http://is.ijs.si) je z **dvajseto** 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 s **40-letno** tradicijo odlične znanstvene revije. Odlične obletnice!

Multikonferenco Informacijska družba 2017 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Soočanje z demografskimi izzivi
- Kognitivna znanost
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Izkopavanje znanja in podatkovna skladišča
- Vzgoja in izobraževanje v informacijski družbi
- Četrta študentska računalniška konferenca
- Delavnica »EM-zdravje«
- Peta mednarodna konferenca kognitonike
- Mednarodna konferenca za prenos tehnologij - ITTC
- Delavnica »AS-IT-IC«
- Robotika

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 2017 bomo petič 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. Marjan Krisper. Priznanje za dosežek leta bo pripadlo prof. dr. Andreju Brodniku. Že šestič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobilo padanje slovenskih sredstev za akademsko znanost, tako da smo sedaj tretji najslabši po tem kriteriju v Evropi, jagodo pa »e-recept«. Čestitke nagrajencem!

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

FOREWORD - INFORMATION SOCIETY 2017

In its **20th 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 2017 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, and 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 **40 years** of tradition of excellent research publication. These are remarkable achievements.

The Information Society 2017 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Facing Demographic Challenges
- Cognitive Science
- Collaboration, Software and Services in Information Society
- Data Mining and Data Warehouses
- Education in Information Society
- 4th Student Computer Science Research Conference
- Workshop Electronic and Mobile Health
- 5th International Conference on Cognitonics
- International Conference of Transfer of Technologies - ITTC
- Workshop »AC-IT-IC«
- Robotics

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 fifth 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. Marjan Krisper 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. Andrej Brodnik. The information lemon goes to national funding of the academic science, which degrades Slovenia to the third worst position in Europe. The information strawberry is awarded for the medical e-recipe project. 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
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
Mitja Luštrek
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

Invited lecture

AN UPDATE FROM THE AI & MUSIC FRONT

Gerhard Widmer
Institute for Computational Perception
Johannes Kepler University Linz (JKU), and
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna

Abstract
Much of current research in Artificial Intelligence and Music, and particularly in the field of Music Information Retrieval (MIR), focuses on algorithms that interpret musical signals and recognize musically relevant objects and patterns at various levels -- from notes to beats and rhythm, to melodic and harmonic patterns and higher-level segment structure --, with the goal of supporting novel applications in the digital music world. This presentation will give the audience a glimpse of what musically "intelligent" systems can currently do with music, and what this is good for. However, we will also find that while some of these capabilities are quite impressive, they are still far from (and do not require) a deeper "understanding" of music. An ongoing project will be presented that aims to take AI & music research a bit closer to the "essence" of music, going beyond surface features and focusing on the expressive aspects of music, and how these are communicated in music. This raises a number of new research challenges for the field of AI and Music (discussed in much more detail in [Widmer, 2016]). As a first step, we will look at recent work on computational models of expressive music performance, and will show some examples of the state of the art (including the result of a recent musical 'Turing test').

References
Widmer, G. (2016).
Getting Closer to the Essence of Music: The Con Espressione Manifesto.
ACM Transactions on Intelligent Systems and Technology 8(2), Article 19.

KAZALO / TABLE OF CONTENTS

Sodelovanje, programska oprema in storitve v ***1***

informacijski družbi / Collaboration, Software and Services in Information ***1***

Society ***1***

PREDGOVOR / FOREWORD 3

PROGRAMSKI ODBORI / PROGRAMME COMMITTEES 5

Crop Yield Prediction in the Cloud: Machine Learning Approach / Catal Çağatay, Muratli Can 7

Using Cognitive Software to Evaluate Natural Language / Torres Camilo, Tabares S. Marta, Montoya Edwin, Kamišalić Aida 11

An Analysis of BPMN-based Approaches 15

for Process Landscape Design / Polančič Gregor, Huber Jernej, Tabares S. Marta 15

Approach to an alternative value chain modeling / Pavlinek Miha, Heričko Marjan, Pušnik Maja 19

Using Property Graph Model for Semantic Web Services Discovery / Šestak Martina 23

Statecharts representation of program execution flow / Sukur Nataša, Rakić Gordana, Budimac Zordan 27

Code smell detection: A tool comparison / Beranič Tina, Rednjak Zlatko, Heričko Marjan 31

A Qualitative and Quantitative Comparison of PHP and Node.js for Web Development / Heričko Tjaša 35

Skills, Competences and Platforms for a Data Scientist / Podgorelec Vili, Karakatič Sašo 39

Towards a Classification of Educational Tools / Košič Kristjan, Rajšp Alen, Huber Jernej 43

Indeks avtorjev / Author index ***47***

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek G

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume G

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

Uredil / Edited by

Marjan Heričko

http://is.ijs.si

9. oktober 2017 / 9th October 2017
Ljubljana, Slovenia

PREDGOVOR

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

Informatika in informacijske tehnologije so že več desetletij gonilo inoviranja na vseh področjih poslovanja podjetij ter delovanja posameznikov. Odprti standardi in interoperabilnost ter vedno višja odzivnost informatikov vodijo k razvoju inteligentnih digitalnih storitvenih platform in inovativnih poslovnih modelov ter novih ekosistemov, kjer se povezujejo in sodelujejo ne le partnerji, temveč tudi konkurenti. Vse večja in pomembnejša je tudi vključenost končnih uporabnikov naših storitev in rešitev. Napredne informacijske tehnologije in sodobni pristopi k razvoju, vpeljavi in upravljanju omogočajo višjo stopnjo avtomatizacije in integracije doslej ločenih svetov, saj vzpostavljajo zaključeno zanko in zagotavljajo nenehne izboljšave, ki temeljijo na aktivnem sodelovanju in povratnih informacijah vseh vključenih akterjev. Ob vsem tem zagotavljanje kakovosti ostaja eden pomembnejših vidikov razvoja in vpeljave na informacijskih tehnologijah temelječih storitev.

Prispevki, zbrani v tem zborniku, omogočajo vpogled v in rešitve za izzive na področjih kot so npr.:

- modeliranje vrednostih verig storitvenih ekosistemov;
- načrtovanje pokrajin procesov;
- zaznavanje neustreznih načrtovalskih odločitev;
- identifikacija pomanjkljivih programskih komponent;
- odkrivanje semantičnih spletnih storitev;
- vrednotenje naprednih spletnih tehnologij;
- klasifikacija orodij učnega stolpiča;
- identifikacija znanj in kompetenc podatkoslovca;
- učenje in ovrednotenje klasifikatorjev naravnega jezika;
- uporaba algoritmov strojnega učenja v praksi.

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

Marjan Heričko

FOREWORD

This year, the Conference “Collaboration, Software and Services in Information Society” is being organised for the seventeenth 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 as well as collaboration in general.

Information technologies and the field of Informatics have been the driving force of innovation in business, as well as in the everyday activities of individuals for several decades. Open standards, interoperability and the increasing responsiveness of IS/IT experts are leading the way to the development of intelligent digital service platforms, innovative business models and new ecosystems where not only partners, but also competitors are connecting and working together. The involvement and engagement of end users is a necessity. On the other hand, quality assurance remains a vital part of software and ICT-based service development and deployment. The papers in these proceedings provide a better insight and/or propose solutions to challenges related to:

- Modelling large ecosystems value chain;
- Designing process landscape;
- Detecting bad design decision and code smells;
- Discovering semantic web services;
- Evaluation of advanced Web technologies;
- Classification of learning stack tools;
- Identifying skills and competencies of data scientists;
- Training and evaluation of natural language classifiers.
- Applying machine learning algorithms in practice.

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.

Marjan Heričko

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Marjan Heričko

Lorna Uden

Gabriele Gianini

Hannu Jaakkola

Mirjana Ivanović

Zoltán Porkoláb

Vili Podgorelec

Maja Pušnik

Muhamed Turkanović

Boštjan Šumak

Gregor Polančič

Luka Pavlič

Crop Yield Prediction in the Cloud: Machine Learning Approach

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

Can Muratli
Department of Computer Engineering
Istanbul Kültür University
Istanbul, Turkey
o.muratli@iku.edu.tr

ABSTRACT

Crop yield prediction provides critical information for decision makers and directly affects the agricultural policies and trade. Current emerging technologies such as Internet of Things (IoT), big data analytics, cloud computing, and machine learning enabled researchers to design and implement high-performance yield prediction models. In this work, we aimed at investigating several machine learning-based regression techniques such as Boosted Decision Tree Regression and Neural Network Regression for this challenging problem and implementing a wheat yield prediction web service to host on the Azure cloud computing platform. Case studies were performed on the data obtained for south-east region of Turkey and four states in the United States. Experimental results demonstrated that while neural network regression technique provides the best performance for large-scale crop yield prediction datasets, linear regression technique is more appropriate for small-scale datasets.

Categories and Subject Descriptors

I.2.6. [**Computing Methodologies**]: *Learning*

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Crop Yield Prediction; Internet of Things; Sensors; Machine Learning; Regression Techniques; Cloud Computing

1. INTRODUCTION

It is reported that 795 million people in the world are undernourished which means that one in nine people today live without sufficient food [1]. While the current world population is around 7.5 billion people, it's estimated that it will be 9.7 billion people which is 30% higher than the current population [2]. To supply adequate food to this huge population, global food production must improve dramatically. It's estimated that while one farmer now feeds 155 people in the world, by 2050 one farmer will need to feed 250 people which is 61% higher than the current situation [3]. United Nations aims at ending hunger by 2030 and ensuring access to safe and adequate food by all people in the world [4].

Crop yield prediction before harvest can help to manage the agricultural trade policies [5], provide critical data for economic and political stakeholders, and evaluation of climate change impact [6]. Therefore, researchers are still actively involved in the development of crop yield prediction models at national, sub-national, and international levels [6]. Since traditional survey methods are time consuming and error-prone, accurate prediction approaches are currently being developed by different research groups. In addition to the survey method, there are different approaches such as statistical methods, crop simulation models, and remote sensing-based techniques.

In this study, our main objective is to design and implement a wheat yield prediction system based on the data obtained from sensors positioned in stations in south-east region of Turkey. To retrieve and process this data, we collaborated with TARBIL Agro-Informatics Research Centre in Istanbul Technical University which has a terrestrial network called Agricultural Monitoring and Information System (AgriMONIS) with 441 active RoboStations. In addition to the analysis performed on this data, we also developed machine learning-based models for wheat data obtained from four states in the United States. Machine learning-based models were designed and evaluated in Azure Machine Learning Studio platform. The best algorithm in terms of coefficient of determination parameter was transformed into a web service and deployed in Azure cloud computing platform. A client-side web application was implemented using ASP.NET technology to handle the requests of the end users and farmers are being informed via this web application about the yield prediction results.

2. RELATED WORK

There are several studies on crop yield prediction, but we did encounter an end-to-end crop yield prediction system which uses Azure Machine Learning Studio, Azure Cloud platform, and web services technology. Most of the studies in the literature only report experimental results, but do not provide any practical information to build a crop yield prediction system for the real-world scenarios. Also, there is very limited number of studies which applied data in south-east region of Turkey. Çakır et al. [5] built an Artificial Neural Network to estimate the wheat yield prediction in south-east region of Turkey and utilized from meteorological data such as temperature and rainfall records. They used data regarding to the years 2011 and 2012 for training the model, and applied the data regarding the year 2013 to test the prediction model. They reported that results are better than the regression method when Multi-Layer Perceptron (MLP) is applied. The optimal value for the Number of neurons was

reported as 15. Chen and Jing [7] compared two adaptive multivariate Analysis methods based on Landsat-8 images to forecast wheat yield and reported that Artificial Neural Networks (ANN) provides better results than Partial Least Squares Regression (PLSR) technique in terms of coefficient of determination and root mean squared error (RMSE) parameters. Gouache et al. [6] developed wheat yield prediction models in France for 23 departments using yield statistics from 1986 to 2010. They started with 250 variables and reached to 5-7 variables using forward stepwise regression methods to design their prediction models. For 20 departments, acceptable models were implemented. Stas et al. [8] compared Boosted Regression Trees (BST) and Support Vector Machines (SVM) algorithms for the prediction of wheat yields and reported that BST provides better performance than SVM. Our paper is different than these studies as we decided to build a cloud-based prediction system and use state-of-the-art regression algorithms in Azure machine learning platform.

3. METHODOLOGY

While there are many tools available, we preferred Azure Machine Learning Studio due to its cloud computing capabilities and its easy to use nature. The collaboration with TARBIL, which is a focused science center on agriculture that has over 400 stations equipped with various sensors that monitor every phenological state of a field, enabled us to get the precise datasets for our experiments. In addition to those datasets, we also came across with a set of datasets focused on wheat yield in USA [9] which created an opportunity for another case study to evaluate our models.

During our experiments, every regression model available in Azure Machine Learning Studio is tested, however due to some constraints created by the datasets available to us we narrowed our regression options to four which are explained briefly below:

1. Linear Regression: Despite being the most simplistic method amongst other regression models, linear regression is frequently used in many case studies since what it simply does is to attempt to create a linear relationship between one or more features to be used for a prediction of a numeric outcome.
2. Bayesian Linear Regression: It is like linear regression approach however, it uses Bayesian inference that update probability distribution.
3. Boosted Decision Tree Regression: Using an efficient implementation of MART gradient boosting algorithm, Boosted Decision Tree Regression aims to build each regression tree in a step by step fashion, eliminating weaker prediction models.
4. Neural Network Regression: While neural networks are widely used for deep learning and modelling sophisticated problems, they can also be adapted to regression models where more traditional regression models falls short.

We had two different datasets one was from South-East region of Turkey the other one was from four states of United States of America. Having two sets of data, led us to approach this problem in two case studies.

In case study one, we had both phenological data and crop yield information from nine different stations equipped with sensors for the years between 2013 and 2016 which enabled us to use the data from 2013 to 2015 for training and prediction 2016 yield results with given features (Figure 1).

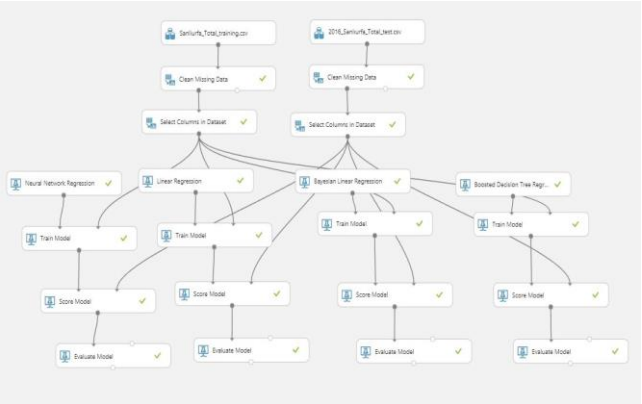


Figure 1. All regression models tested for train and score model South-East Region of Turkey

After combining the two datasets to one both test and train datasets also run with ten-fold cross-validation settings as seen in Figure 2.

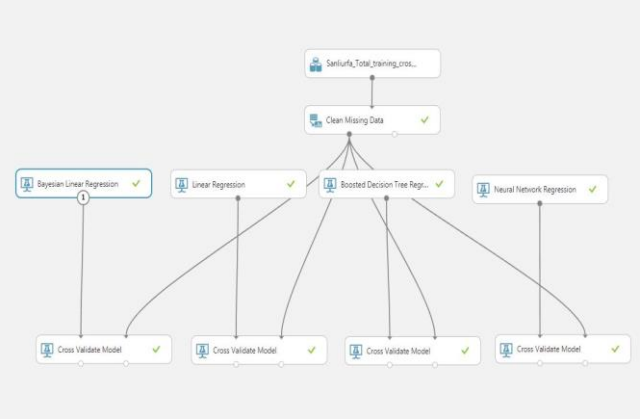


Figure 2. All regression models tested for cross validation model South-East Region of Turkey

Cross-validation evaluation helped us to compare our findings with second case study. In the second case study, we had more than 300.000 records in the dataset. However, since we had only two-years of data, we did not perform a test which uses an external test dataset. Therefore, we made only cross-validation experiment for this large dataset.

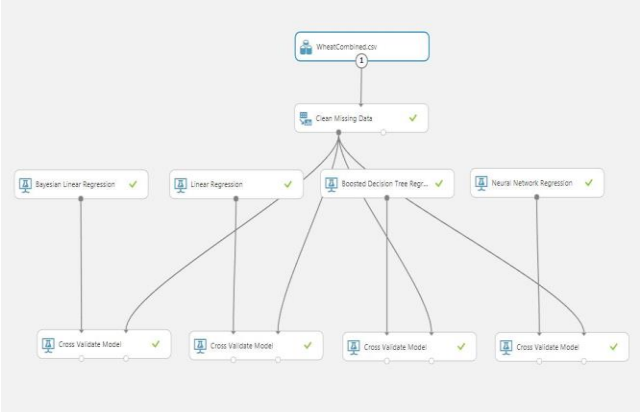


Figure 3 - All regression models tested for cross validation model

For the datasets from Turkey, yield information was in kilograms while in the USA dataset it was percentage based information.

4. EXPERIMENTAL RESULTS

As mentioned earlier, we applied four different regression models to our datasets. We applied 10-fold cross-validation approach for all the case studies and calculated the Coefficient of Determination parameter with the help of Azure Machine Learning Studio. Coefficient of Determination is a value between 0-1 which determines how close the prediction is to the reality. While experimenting, we have seen that both the features and the amount of data affect the results. As seen in Table 1 in our train/score model, the most simplistic approach which is the Linear Regression scored the best results. Neural Network regression failed because of the insufficient number of records. During the 10-fold cross validation experiments after adding the 2016 data to the training dataset which consisted of the data from 2013-2015 we observed significant changes on the results, especially for Bayesian Linear Regression. As the number of records raised in the dataset, Boosted Decision Tree regression had more information to train itself with better results.

In the second case study, we only had two years data with great amount of records for machine learning algorithms to learn from. As in Table 2 all the ten-fold cross validation results increased while Neural Network Regression model, unlike in the first case study, giving a satisfying result. With the lack of the past two year's data, we chose to base our web service on the first case study's dataset and developed further on from there. The web application we developed uses a basic input-output style interface to interact with the user and predict the crop yield when given the input. Input consists of the following features: Region and provenance of the field, current temperature, yearly maximum and minimum temperature, total precipitation, growing day degree, temperature difference parameter, Photo Thermal Unit, Helio Thermal Unit and evapotranspiration parameter.

5. CONCLUSION AND FUTURE WORK

The objective of the crop yield prediction studies is to forecast the crop yield as early as possible during the crop growing season. Weather and climate affect this agricultural production dramatically. In this study, we developed an end-to-end wheat yield prediction system using machine learning algorithms. Case studies were performed on the datasets retrieved from south-east region of Turkey and four states in United States. Linear Regression algorithm provided the best performance in south-east region in terms of coefficient of determination parameter when external test set was used. Neural Network Regression algorithm was the best option for the US dataset when cross-validation analysis was applied. As part of the future work, web application can be replaced with a mobile application and new experiments can be performed when more regions are added to the datasets. Deep learning algorithms might be considered when the dataset becomes very large.

	South East Region of Turkey Wheat Yield ML Results			
	Regression Type	***Relative Absolute Error***	***Relative Squared Error***	***Coefficient of Determination***
	Linear Regression	0.484972	0.300939	0.699061
	Bayesian Linear Regression	0.643469	0.396072	0.603928
	Boosted Decision Tree Regression	0.669278	0.646339	0.353661
	Neural Network Regression	0.969594	1,4013830	-0,4013830
	Regression Type	***Relative Absolute Error***	***Relative Squared Error***	***Coefficient of Determination***
	Linear Regression	0.572134	0.391993	0.608007
	Bayesian Linear Regression	0.325579	0.157755	0.842245
	Boosted Decision Tree Regression	0.512205	0.329844	0.670156
	Neural Network Regression	1,7276340	3,7723160	-2,7723160

Table 1 South East Region of Turkey Wheat Yield ML Results

	United States of America Wheat Yield ML Results			
	Regression Type	***Relative Absolute Error***	***Relative Squared Error***	***Coefficient of Determination***
	Linear Regression	0.376717	0.173927	0.826073
	Bayesian Linear Regression	0.389548	0.180413	0.819587
	Boosted Decision Tree Regression	0.105499	0.01352	0.98648
	Neural Network Regression	0.000736	0.000001	0.999999

Table 2 United States of America Wheat Yield ML Results

6. ACKNOWLEDGMENTS

Data for this project is provided by TARBIL Agro-Informatics Research Centre in İstanbul Technical University. Authors would like to thank to technical and management staff in this research centre who helped us to prepare the dataset.

REFERENCES

[1] J. You, X. Li, M. Low, D. Lobell, and S. Ermon, "Deep gaussian process for crop yield prediction based on remote sensing data", In AAAI, pp. 4559-4566, 2017.

[2] K. B. Newbold, Population Growth. The International Encyclopedia of Geography. 2017.

[3] Cloud Technology Partners, https://www.cloudtp.com/doppler/feeding-10-billion-people/ (2017) (accessed June 17, 2017).

[4] United Nations, zero hunger: why it matters? Sustainable development goal http://www.un.org/sustainabledevelopment/wp-content/uploads/2016/08/2_Why-it-Matters_ZeroHunger_2p.pdf (2015) (accessed June 17, 2017).

[5] Y. Çakır, M. Kırcı, and E. O. Güneş, "Yield prediction of wheat in south-east region of Turkey by using artificial neural networks", In Agro-geoinformatics (Agro-geoinformatics 2014), pp. 1-4, 2014.

[6] D. Gouache, A. S. Bouchon, E. Jouanneau, and X. Le Bris, "Agrometeorological analysis and prediction of wheat yield at the departmental level in France", Agricultural and Forest Meteorology, Vol. 209, pp. 1-10, 2015.

[7] P. Chen, and Q. Jing, "A comparison of two adaptive multivariate analysis methods (PLSR and ANN) for winter wheat yield forecasting using Landsat-8 OLI images", Advances in Space Research, Vol. 59, Issue 4, pp. 987-995, 2017.

[8] M. Stas, J. Van Orshoven, Q. Dong, S. Heremans, and B. Zhang, "A comparison of machine learning algorithms for regional wheat yield prediction using NDVI time series of SPOT-VGT", In Agro-Geoinformatics (Agro-Geoinformatics), pp. 1-5. 2016.

[9] USA Wheat Data, https://github.com/prateek47/Wheat_Prediction

Using cognitive software to evaluate Natural Language Classifiers - A Use Case

Camilo Torres
Department of Informatics and Systems
Universidad EAFIT
Medellín, Colombia
ctorres9@eafit.edu.co

Marta S. Tabares
Department of Informatics and Systems
Universidad EAFIT
Medellín, Colombia
mtabares@eafit.edu.co

Edwin Montoya
Department of Informatics and Systems
Universidad EAFIT
Medellín, Colombia
emontoya@eafit.edu.co

Aida Kamišalić
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
aida.kamisalic@um.si

ABSTRACT

The current techniques for natural language processing can be used to identify valuable information such as sentiments or patterns recognized and adjusted for different topics. To apply these techniques, it is required to know how to use and tune prediction models. This requires time, experience and the implementation of different tests to ensure the correct behavior of the models. The aim of this paper is to detect the features to train and evaluate classifiers instances using optimized software, specifically, IBM Bluemix, and its module named Natural Language Classifier. The created classifier was trained with real tweets to classify the texts into three categories: Positive, neutral and negative texts. Afterwards, the classifier was validated with a set of already classified texts. The obtained results indicate how the number of training examples impact the behavior of the classifier and, that the highest accuracy was achieved for positive and negative categories.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Algorithms, Measurement, Experimentation

Keywords

Natural Language, Classifiers, Machine Learning, Bluemix, Watson

1. INTRODUCTION

The daily use of social networks currently results in the large-scale growth in the data and information generated in the world. There is an expected expansion of 40% per year and an estimated size of 50 times by 2020 [1]. Generated data are mostly texts created by users in social networks such as comments or tweets that, in some cases, have free access, while in others, they can be accessed by purchasing on demand packages. This type of information allows companies to carry out market analysis or search for communities of potential clients [2].

The available literature presents several research projects about the algorithms and techniques used for natural language processing [3]. Their results indicate that the time required to implement such techniques and algorithms depends on users' previous mathematical knowledge and on the tuning of the mathematical functions used in the process. Therefore, despite the existing different solutions for text analysis, the implementation of such algorithms may be slow because of different influencing factors such as the tuning required for each process and the tests with different parameters.

To address this problem we evaluate the proficiency of a tool to analyze and classify texts generated from social networks. The texts classifications are labeled with the basic polarity used for sentiment analysis, i.e. positive, negative and neutral labels [4]. The databases for testing and training sets were obtained from the public Twitter API and the Spanish Society for Natural Language Processing (SEPLN).

Among the existing technologies for natural language processing, there are platforms such as IBM Watson, Microsoft LUIS, API.ai, WIT.ai, etc. We decided to use IBM Bluemix, which includes a large variety of Watson services, where the wide catalog of options can be used for intelligent chats, texts' classifications and understanding, as well as the demonstrated results and tests of Watson, such as the Jeopardy game. Here, Watson shows the proficiency to give right answers using natural language processing [5, 6]. We have used the Natural Language Classifier component of the Watson suite that uses convolutional neural networks to do the cognitive process from language [7].

This paper is organized as follows. We present the context and the research questions in Section 2. Section 3 summa-

rizes the related work for tweets' polarity classification. The methodology to address the problem is explained in Section 4. Furthermore, we introduce a detailed explanation of the developed proposal in Section 5. The results of the provided work are shown in Section 6. Finally, Section 7 brings the conclusions of the paper.

2. CONTEXT AND RESEARCH QUESTIONS

During the classifier training, problems such as over-fitting or poor estimation of the models for the training sets, might occur [8, 9]. These factors should be taken into account in order to avoid a bad prediction. Furthermore, adjusting and finding the correct parameters for the adequate behavior of the algorithm is a task that demands time and effort. Technologies, such as IBM Bluemix, already have a set of services used for the natural language processing. These can be solutions that save time consumed for the tuning of classifiers.

The identified problem and the context of using classifiers through the natural language processing for sentiment analysis (polarity) in text, particularly tweets, results in the definition of the following research questions:

- Which are the characteristics under which a classifier should be trained using technologies for the natural language processing and sentiment analysis?

- How effective are classifiers trained using frameworks and automated tools for natural language processing and sentiment analysis?

3. RELATED WORK

One of the most treated problems found in the studied literature is data pre-processing before being used to train classifiers. Elements such as sarcasm, expressions, abbreviations, among others, can generate erroneous predictions. Khan et al. [10, 11] focused on developing classifiers with good preprocessing before training. At the same time, they propose hybrid models based on the classification of emoticons, bags of words, etc. For pre-processing, procedures were proposed such as searching dictionaries to check the existence of terms, replacing abbreviations, completing incomplete words and performing spelling checks.

Mertiya et al. [12] proposed the usage of bayesian classifiers to obtain the polarity of a database of tweets that results in classifications with several false positives, which are submitted to an analysis of adjectives in order to be polarized correctly. The problem with this type of classifiers is that the short texts of the tweets have a characteristic named sparsity, meaning that the data is not very significant, therefore, the classifiers may have errors or bad predictions.

To avoid the problems that happen when this type of short texts are classified, He et al. [13] proposed a different approach using a clustering algorithm called k-means in order to discover related topics, based on the premise that the texts will be more informative if they are grouped into similar topics. Then, the obtained clusters are used to train a bayesian classifier.

Almeida et al. [14] performed an evaluation of supervised algorithms for mining opinions on twitter and emphasized the actions of cleaning and pre-processing the information before it is submitted to a classifier. Furthermore, they proposed a process to make similar classifications according to the process carried out in this study, not only based on polarity and sentiment analysis, but also on the objective classification of the opinions expressed in the texts.

Finally, several of the papers found in the literature identified different problems related to various types of classifiers and, accordingly, there are models that attempt to solve the issues combining different types of classifiers. Lima et al. [4] and Brahimi et al. [15] proposed hybrid solutions to improve classification results using bayesian classifiers, support vector machines, decision trees and k-nearest neighbors. These types of solutions make it possible to increase the accuracy of classifications and to evaluate which learning methods perform best.

The different algorithms and techniques found in this related work are processes that require time in each of the different phases: Pre-processing, extraction, development or algorithms' testing. In this work, we use algorithms already tested in order to speed up the sentiment analysis and polarity detection in tweets. We used IBM Bluemix specifically Watson and its Natural Language Classifier module.

4. METHODOLOGY

We propose an approach for the tool evaluation through the method developed by Wieringa et al. [16]. We try to solve a problem through an engineering cycle, which is carried out by the treatment or planning of solutions, and is validated with questions and answers that we made before and after the treatment. The expected effects are mentioned, and finally, the process is concluded using the results obtained in the treatment. The treatment of this work is described based on the process carried out for the supervised learning introduced by Kotsiantis [17], where the emphasis is on the pre-processing of the data. In order to validate the results, a database with texts of already classified tweets was obtained through the Spanish Society for Natural Language Processing (SEPLN).

5. USING NATURAL LANGUAGE PROCESSING

We based our proposal on the supervised learning process presented by Kotsiantis [17], where we start with the identification of the required data. Figure 1 shows this process's steps, which are modified for the use case described in this study. First, we obtained the training sets from our own tool, and then we made the texts' pre-processing for their correct interpretation in Watson. Furthermore, we present the contribution for the use of Bluemix.

5.1 Data identification

The used data are the different texts from the tweets database. We used Cloudant, a managed NoSQL JSON database service, to perform querying easily through an HTTP API. We imported the data in order to create the training sets.

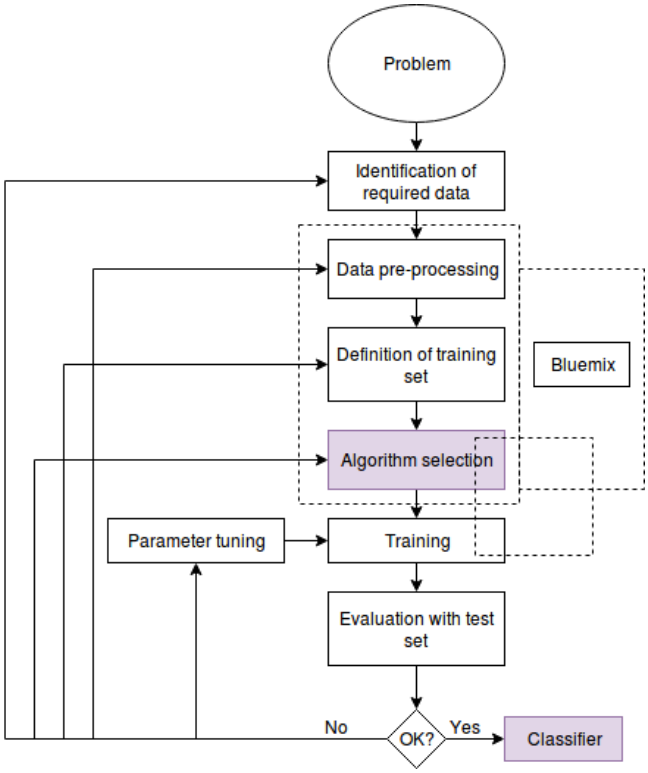


Figure 1: Supervised learning process exposed by Kotsiantis [17] and complemented in this paper.

5.2 Definition of the training sets

To facilitate the selection of the tweets, we developed a web application in Node.js which makes queries to the database and selects a tweet randomly. The selected tweet must be assigned to one of the defined classes (positive, negative or neutral). If it is not possible to label it with one of these polarities, labeling can be omitted or N/A (not applicable) selected. When each class contains approximately 600 tweets, they are exported in CSV format using a script in Node.js which queries the instances through the Cloudant HTTP API.

5.3 Data pre-processing

The texts used as training should go through a cleanup process where special characters like quotes and break lines are replaced as indicated in the Bluemix documentation. For example, each text must be enclosed in quotation marks, if this character is repeated, it must be added twice, i.e. replacing " so as "" to distinguish it from the one that encloses the training text. We perform this process when the CSV file is created. Table 1 shows a file example with two columns, the first one has the training texts and the second the class to which each sentence corresponds.

5.4 Training the classifier

We used the HTTP API of Bluemix, which, through a web service, creates the classifier from the training file separated by commas. At the beginning the classifier is in a training state. The time the classifier needs to be prepared for the consultation varies, depending on the size of the training set.

Table 1: File structure and example

Text	Class
"Let's leave the skin to create a job and our economy grows again. #MensajeGriñan"	positive
"2012 will be a year of titles. Play in a team. Win as a team. Who's with me?#makeitcount http://t.co/Ue7Kh2De"	positive
"#FF @BRmodainfantil moms with children, do not miss it, the best online shop for children's fashion!"	positive
"Impressed by the violence of the media in Morocco. Pushing to photograph Rajoy in Rabat"	negative
"The one who does not want to follow me does not follow me, but the masochist must stop complaining and enjoy"	negative
"These are hard times for everyone! The worst thing will be the staff adjustments, which will not be delayed..."	negative
"You can also follow it in the channel 24 hours of RTVE"	neutral
"A few hours remain to close the last draw of the year. There is still time to sign up"	neutral
"In the Vatican City"	neutral

Finally, the classifier can also be consulted through HTTP requests. The obtained results are the probabilities that an evaluated text could be in each class.

5.5 Evaluation with the test set

To perform the evaluation of the created classifier, we used a test dataset with texts already polarized by the SEPLN. We used 15,000 texts to retrain the classifier in order to make comparisons with the first training set and, at the same time, be evaluated with a test subset taken from the total texts.

After we retrained the classifier, we used a subset from the test database to perform the validation and test the accuracy for the classifier with the respective training sets. We took 300 texts for each class, i.e. in total there were 900 tweets, to test how much the classifiers instances approached their predictions regarding their polarity from the SEPLN. We use these 900 texts in both classifiers instances to compare the results.

6. RESULTS

Following the proposed engineering cycle, and based on the results from the performed tests, some answers can be derived for the raised research questions. Regarding the effectiveness of the classifiers, the results obtained were acceptable in the positive and negative classes for the second training set. The neutral texts class presents results varying in both tests, which leaves evidence of the subjectivity that this type of sentences present. It is important to note that the texts have not been filtered by any process to remove stop words, URLs, hashtags, and other types of words that could affect the classifiers' prediction. Tables 2 and 3 show the results for the first and the second training sets, respectively and table 4 shows the accuracy for each training set.

We observe that the classifier with the second training set presents better results than the classifier with the first training set. The second training set was created with 15,000 records, which is the maximum number of records supported

Table 2: Results of the test set for the first training set

	Total	Right predictions	Right predictions' rate
Positive	300	105	35%
Negative	300	158	52.6%
Neutral	300	191	63.6%

Table 3: Results of the test set for the second training set

	Total	Right predictions	Right predictions' rate
Positive	300	235	78.3%
Negative	300	254	84.7%
Neutral	300	147	49%

Table 4: Accuracy for each training set

	Training set 1	Training set 2
Accuracy	50.4%	70.7%

by the Bluemix Natural Language Classifier module. It is probable that the large set of texts and the type of texts used by the SEPLN, made the classifier with the second training set get a better prediction and closer to the original polarity of the test database. It is also probable that the class for neutral texts will be more subjective and, therefore, could be the reason for obtaining different results in the two tests. We conclude that the classifier obtained better results with the second training set because of the large number of examples.

7. CONCLUSION

We proposed the usage of natural language classifiers, using IBM Bluemix and its services for text analysis, in order to speed up the process of parameterization and algorithms' tuning. We conclude that the classifiers created in this manner have a good effectiveness according to the texts' cleaning process. The neutral classification is the most subjective and prone to bad predictions. It is important to emphasize that the cleaning process has a great influence on the classification results, in addition to the subjectivity in the creation of the training sets.

8. ACKNOWLEDGEMENT

We acknowledge the support of the Colombian Center of Excellence and Appropriation on Big Data and Data Analytics - Alianza CAOBA (http://alianzacaoba.co/), under which the project is developed. We sincerely thank the researchers and students who participated in tweets' classification.

9. REFERENCES

[1] Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Abdullah Gani, Salimah Mokhtar, Ejaz Ahmed, Nor Badrul Anuar, and Athanasios V. Vasilakos. Big data: From beginning to future. *International Journal of Information Management*, 36(6):1231–1247, dec 2016.

[2] Francesco Piccialli and Jai E. Jung. Understanding Customer Experience Diffusion on Social Networking Services by Big Data Analytics. *Mobile Networks and Applications*, pages 1–8, dec 2016.

[3] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, mar 2002.

[4] Ana Carolina E S Lima, Leandro Nunes De Castro, and Juan M. Corchado. A polarity analysis framework for Twitter messages. *Applied Mathematics and Computation*, 270:756–767, nov 2015.

[5] Grady Booch. The soul of a new watson, jul 2011.

[6] D. A. Ferrucci. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15, may 2012.

[7] Carmine DiMascio. Create a natural language classifier that identifies spam. *https://www.ibm.com/developerworks/library/cc-spam-classification-service-watson-nlc-bluemix-trs/index.html*, 2015.

[8] Alex A. Freitas and Alex A. Understanding the crucial differences between classification and discovery of association rules. *ACM SIGKDD Explorations Newsletter*, 2(1):65–69, jun 2000.

[9] Douglas M. Hawkins. The Problem of Overfitting, 2004.

[10] Farhan Hassan Khan, Saba Bashir, and Usman Qamar. TOM: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems*, 57(1):245–257, jan 2014.

[11] Farhan Hassan Khan, Usman Qamar, and M. Younus Javed. SentiView: A visual sentiment analysis framework. In *International Conference on Information Society, i-Society 2014*, pages 291–296. IEEE, nov 2015.

[12] Mohit Mertiya and Ashima Singh. Combining Naive Bayes and Adjective Analysis for Sentiment Detection on Twitter. *2016 International Conference on Inventive Computation Technologies (ICICT)*, pages 1–6, aug 2016.

[13] Yunchao He, Chin Sheng Yang, Liang Chih Yu, K. Robert Lai, and Weiyi Liu. Sentiment classification of short texts based on semantic clustering. In *Proceedings of 2015 International Conference on Orange Technologies, ICOT 2015*, pages 54–57. IEEE, dec 2016.

[14] Yudivian Almeida and Velarde Suilaan. Evaluacion de Algoritmos de Clasificacion Supervisada Para El Minado De Opinion en twitter. *Investigación Operacional*, 36(3):194–205, 2015.

[15] Belgacem Brahimi, Mohamed Touahria, and Abdelkamel Tari. Data and text mining techniques for classifying Arabic tweet polarity. *Journal of Digital Information Management*, 14(1):15–25, 2016.

[16] Roel J. Wieringa and Ayse Morali. Technical Action Research as a Validation Method in Information Systems Design Science. *Design Science Research in Information Systems. Advances in Theory and Practice*, 7286:220–238, 2012.

[17] S B Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica, An International Journal of Computing and Informatics*, 31(31):249–268, 2007.

An Analysis of BPMN-based Approaches for Process Landscape Design

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

Jernej Huber
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
jernej.huber@um.si

Marta S. Tabares
Universidad EAFIT,
Department of Informatics and Systems,
Antioquia, Colombia
mtabares@eafit.edu.co

ABSTRACT
Process landscapes represent the top part of an organizational process architecture. As such, they define the scope and relationships between its processes. Process landscapes diagrams simplify process-related communication by leveraging the benefits of visual notations. However, in contrast to business process diagrams, where nowadays BPMN is the prevalent notation, process landscape diagrams lack of standardization. In the article, we review and analyze notations used for modeling process landscapes, as well as non-normative BPMN-based approaches applicable for their representation. Based on analyzed approaches, we evaluate the applicability of BPMN for process landscape design.

Categories and Subject Descriptors
C.0 **[Computer Systems Organization]**: General – *Modeling of computer architecture*; D.2.9 **[Software engineering]**: Management – *Software process models (e.g., CMM, ISO, PSP)*

General Terms
Management, Documentation, Standardization, Verification

Keywords
Process landscape, process map, BPMN, analysis

1. INTRODUCTION
A common starting point for process design and all activities related to BPM is to identify and structure organization's processes (i.e. process identification phase) [1]. Regularly, users tend to represent identified processes in a visual manner in the form of a process landscape (i.e. process map) diagram.

The main purpose of process landscapes is to specify organizational processes on a bird's-eye view. With process landscapes, an organization can more easily gain an overview of its main processes and their major interdependencies. Therefore, the usage of process landscapes simplifies process-related communication and represent a starting point for detailed process discovery (i.e. AS-IS process modeling). Besides, process landscapes are a common way to represent processes-based reference models for the operation (e.g. ITIL, CMMI) and the management (e.g. COBIT) of organizational IT infrastructure and services.

There are no standardized languages for creating process landscapes. Consequently, modelers most commonly define their own 'overviews of processes' by imitating existing diagrams (e.g. value chains) or proposing their own more or less intuitive representations. A common approach for BPMN experts is to represent process landscapes with a subset of BPMN elements. However, despite BPMN is an ISO and de-facto standard for process modeling, landscapes are out of its scope.

While the non-normative BPMN-based process landscape diagrams appear in practice, this article reviews and analyses related approaches to identify their strengths and weaknesses. Based on analyzed approaches, we evaluate the applicability of BPMN for such modeling purpose.

2. PROCESS ARCHITECTURES AND LANDSCAPES
Process landscapes represent the top part of a process architecture - a conceptual model that organizes processes of a company and makes their relationships explicit (Figure 1).

Figure 1: A conceptual representation of a process architecture

A process architecture usually defines two types of relationships: horizontal and vertical. Horizontal relationships define 'output/input' relationships between processes, i.e. the outcome of a process represents an input for the next process, (e.g. 'consumer-producer' or 'order-to-cash' relationship). Vertical relationships between processes define different levels of details of a process, i.e. a process diagram on a lower level represents a more detailed view of the same process on the level above.

The top-level of a process architecture is commonly reserved for process landscape diagrams. A single process landscape diagram shows the main processes of an organization as well as the dependencies between them, which is shown in the Figure 2 and Figure 3. Those two figures represent the two examples of process landscape diagrams with processes as 'black-boxes' and arrows representing the flow of deliverables between different processes. Rectangles represent the stakeholders, external to an organization.

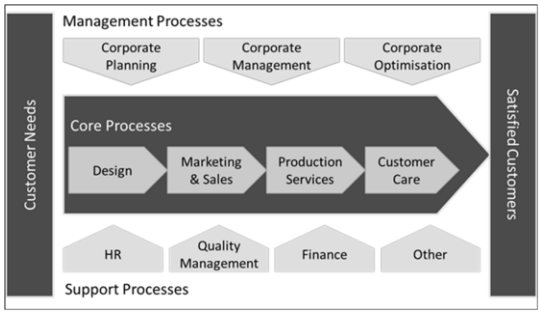


Figure 2: An example landscape diagram (ISO 9001)

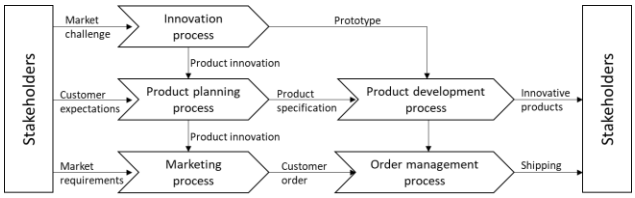


Figure 3: An example landscape diagram [2]

A process landscape diagram serves as a framework for defining the priorities and the scope of process modeling and redesign projects. Each element of a process landscape model may point to a more concrete business processes on the lower levels.

2.1 Process landscape notation

A visual notation (i.e. visual language, graphical notation, or diagramming notation) consists of a set of graphical symbols (visual vocabulary), a set of compositional rules (visual grammar) and definitions of the meaning of each symbol (visual semantics). A common denominator of process landscape diagrams (Figure 2 and Figure 3) are the following elements:

a. **Business process.** Although not explicitly defined, the landscape diagrams clearly highlight the concept of a business process. Visually, a business process is frequently represented with an arrow, where there are also alternative representations, e.g. a rectangle and a rectangle with rounded corners (Figure 4).

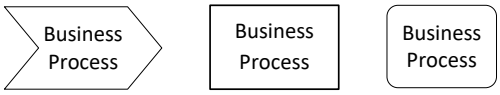


Figure 4: Business process symbols

b. Process groups / types. On a process landscape diagram, the business processes are commonly distinguished by their purpose (e.g. core processes, management processes and supportive processes), which is visualized either by (1) encircling and labelling a set of processes (Figure 5, left) or (2) specializing the process symbol for individual types of processes (Figure 5, right).

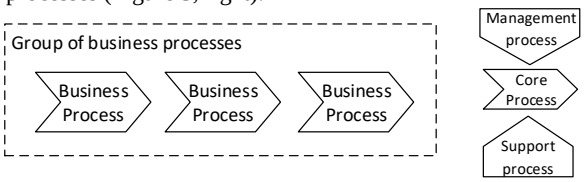


Figure 5: Representation of a group (left) and/or type of processes (right)

Besides manipulating the shapes of symbols, the planar visual variables and symbol orientation might imply the type of a process. E.g., supportive processes are usually positioned below the core processes, with arrows pointing up, whereas management processes are positioned above them, with arrows pointing down (Figure 5).

c. Parent / child process relationships. Processes may be hierarchically organized which is represented either by (1) visualizing sub-processes by using (visual) sub-sets or (2) by using non-directed solid lines between processes as common in 'organizational charts' (Figure 6).

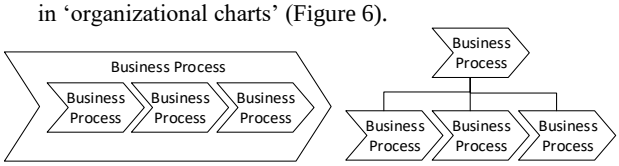


Figure 6: Hierarchical relationships between processes – subsets (left), organigram-based (right)

d. **Process sequence**. The sequence which defines the order on how processed are performed is mainly represented implicitly with a horizontal sequence of process symbols (Figure 7, the left process is performed prior to the right one).

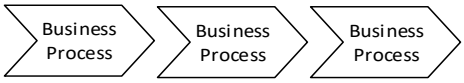


Figure 7: Implicit representation of a sequential relationship between processes

However, since this implicit representation of a sequential relationship enables only a simple linear relationship between processes, explicit representations of processes orderings are visualized with solid directed lines (Figure 8). Another drawback of implicitly ordering the processes is that a diagram reader could misinterpret a set of non-sequentially performed processes, put in a line, as being performed sequentially.

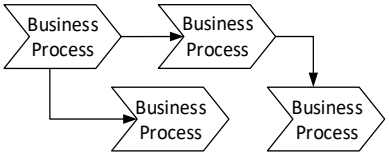


Figure 8: Explicit representation of a sequential relationship between processes

Arrows-based representation of process ordering enables more complex ordering relationships (e.g. when a process ends, two processes are initialized). Sequential relationships might be labelled, representing artefacts or data being transferred between processes (i.e. process outputs – process inputs as presented in the Figure 3).

e. **Participant.** A participant, usually visualized with a rectangle (Figure 9), presents someone who is involved (i.e. internal participant) or interacts (i.e. external participant) with a business process. Most commonly, process landscapes visualize external participants (e.g. suppliers and customers), which are related to processes, either by providing inputs or receiving outputs. This corresponds to the concept of a 'value system' which consists of following value chains: supplier, the focal enterprise and consumer [2]. The relationships to participants are represented either implicitly (e.g. with leveraging visual planar variables) or explicitly (with solid arrows).

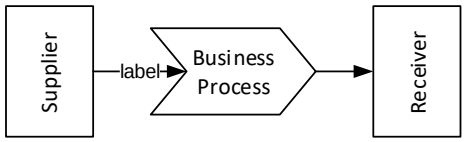


Figure 9: Representation of (external) process participants and their (explicit) relations

3. BPMN-based approaches

Business Process Model and Notation (BPMN) is a well-established standard for process modeling and automation [3]. From the modeling aspect, it defines a vocabulary, grammar and semantics for creating different types of process diagrams, namely: process diagrams, collaboration diagrams, choreography diagrams and conversation diagrams. In light of process diagrams, BPMN states that [4] *"processes can be defined at any level from enterprise-wide Processes to Processes performed by a single person."*. Although this could be understood as BPMN supports modeling of process landscapes, they are not mentioned in any version of the specification, nor recommended by researchers [5].

Nevertheless, since BPMN is widely adapted by industry, modelers frequently use BPMN for visualizing systems of black-box processes (i.e. some kind of process landscapes) by applying the approaches, presented in the next sub-chapters.

3.1.1 Abstract collaboration diagrams

A common and syntactically valid BPMN representation of process landscapes is to use black-box Pools and Message flows, i.e. collaboration diagrams with hidden details (Figure 10). A BPMN Pool is a visual representation of a Participant, which may reference at most one business process. A Message flow represents exchange of messages between two 'message aware' process elements (e.g. activities, message events and black-box Pools).

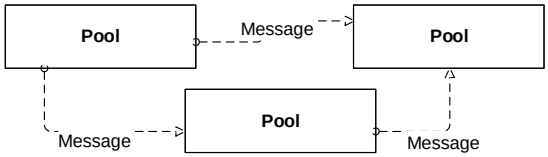


Figure 10: BPMN Pools and Message flows

The strength of such representation of a process landscape is compliance with BPMN specification and simplicity. On the other hand, there are several drawbacks. First, the visual appearance of this approach is unconventional for process landscapes (i.e. processes being represented with rectangles). Second, the relationships between processes represent information exchange, where process landscape diagrams most commonly visualize sequential relationships between processes and processes clustering. Third, there is a lack of concepts, which may be regularly used for landscape modeling, namely, sequential relationships, process hierarchy and process types, whereas there is a symbol deficit in case of representing a participant and a process (rectangle symbol is used in both cases).

3.1.2 Conversation diagrams

Conversation diagrams. Another valid way for representing process landscapes in BPMN is by using Conversation diagrams (Figure 11), which were introduced in the second major revision of BPMN. Formally, they are not a standalone type of BPMN diagrams but merely an abstract view of BPMN collaboration diagrams.

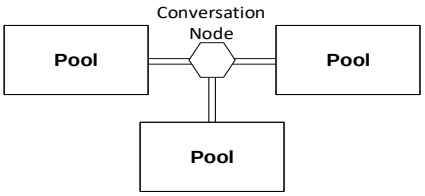


Figure 11: BPMN Conversation diagram

Conversation diagrams are an effective way for representing interactions between processes; however, similar to previous approach, they are based on a small set of elements, which are inappropriate for modeling of conventional process landscapes (i.e. conversation nodes, representing correlated messages and pools, representing participants or processes).

3.1.3 Enterprise-wide process diagrams

As stated in the specification [4], BPMN can be used for business process modeling on any level of granularity. In accordance to this, the system of an organization's processes may be modeled as a single process consisting of individual processes being modeled as activities, i.e. sub-processes (Figure 12).

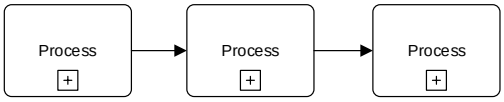


Figure 12: BPMN Sub-processes representing processes

By using this approach, one is able to present the majority of process landscape constructs, namely processes (i.e. with BPMN sub-processes), sequential interactions (i.e. BPMN sequence flows), groups or types of processes (i.e. BPMN group element) and participants (i.e. BPMN lanes). However, there are several major drawbacks of this approach. First, such diagrams are visually inconsistent with process landscape diagrams (e.g. processes being represented with rounded rectangles and participants with horizontal lines). Second, these diagrams are inconsistent with BPMN syntax and semantics, making them invalid (e.g., BPMN Process and BPMN Sub-process are two distinct BPMN meta-model elements). Third, this approach is also impractical, since the majority of processes are discovered on a lower level of granularity (e.g. based on the services or products a business process delivers) and afterwards interrelated into a process landscape diagram.

4. DISCUSSION

Table 1 summarizes a comparison of BPMN-based approaches for landscape design in respect to common process landscape concepts.

In respect to abstract syntax comparison, we can conclude that none of aforementioned BPMN approaches supports all of the concepts common in process landscapes modeling. Besides, the following inconsistencies exist. The first and second approach uses the same element for representing a participant and a process – BPMN Pool (i.e. symbol overload), whereas the third approach uses element BPMN Activity in contrast to its definition (i.e. semantics).

Table 1: Comparison of BPMN-based approaches for landscape design

Process landscape concept	Common visualization	BPMN approach for landscapes modeling		
		1 - Abstract collaboration diagrams	2 - Conversation diagrams	3 - Enterprise-wide process diagrams
Business process	See Figure 4	BPMN Pool	BPMN Pool	BPMN Activity [+]
Process group / cluster	See Figure 5, left	BPMN Group	BPMN Group	BPMN Group
Process type	See Figure 5, right	No standardized BPMN element	No standardized BPMN element	No standardized BPMN element
Hierarchical relationship between processes	See Figure 6	No standardized BPMN element	No standardized BPMN element	Parent activity – child activity relationship
Sequential relationship between processes	See Figure 7 and Figure 8	No standardized BPMN element	No standardized BPMN element	BPMN Sequence flow
Information flows	See Figure 8	BPMN Message flow	BPMN Message flow Conversation Node	Directed association
Internal and external participant	See Figure 9	BPMN Pool	BPMN Pool	BPMN Pool

In respect to the concrete syntax comparison (i.e. notation), Table 1 demonstrates that none of BPMN approaches result in diagrams with a graphical similarity to common landscape diagrams.

According to above, we can conclude that BPMN is inappropriate for modeling the process landscapes. This finding is also supported by Freund and Rücker [6], who state that *'even when we've already modeled one or more process landscapes using BPMN at a customer's request, primarily with the collapsed pools and message flows described we cannot recommend doing this'.*

Analytically, this was confirmed by Malinova [7], who performed a semantical mapping between BPMN and 'Process maps'. Her results show that BPMN is not appropriate for process landscape design.

According to the benefits and weaknesses of existing approaches for (BPMN-based) process landscape design, following research directions are feasible. First, a standardized language for process landscapes may be designed by considering the best practices of non-formal process landscape notations. The focal risk of this research direction is to develop a solution, which has to gain standardization and industry adoption. Second, BPMN structure and the notation may be extended for effective support of process landscapes. In this case, the major risk is the intervention into the structure and notation of a well-adopted and standardized language.

5. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0057).

6. REFERENCES

[1] L. Fischer, R. Shapiro, B. Silver, and Workflow Management Coalition, *BPMN 2.0 handbook second edition: methods, concepts, case studies and standards in business process management notation*. Lighthouse Point, FLa.: Future Strategies, 2012.

[2] M. Weske, *Business process management concepts, languages, architectures*. Berlin; New York: Springer, 2012.

[3] M. Kocbek, G. Jost, M. Hericko, and G. Polancic, "Business process model and notation: The current state of affairs," *Comput. Sci. Inf. Syst.*, vol. 12, no. 2, pp. 509–539, 2015.

[4] OMG, "Business Process Model and Notation version 2.0," 03-Jan-2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/. [Accessed: 15-Mar-2011].

[5] J. Freund and B. Rücker, *Real-Life BPMN: Using BPMN 2.0 to Analyze, Improve, and Automate Processes in Your Company*, 2 edition. CreateSpace Independent Publishing Platform, 2014.

[6] J. Freund and B. Rücker, *Real-Life BPMN: With introductions to CMMN and DMN*, 3 edition. CreateSpace Independent Publishing Platform, 2016.

[7] M. Malinova and J. Mendling, "Why is BPMN not appropriate for Process Maps?," *ICIS 2015 Proc.*, Dec. 2015.

Approach to an alternative value chain modeling

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

Marjan Heričko
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
marjan.hericko@um.si

Maja Pušnik
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
maja.pusnik@um.si

ABSTRACT
This paper is focused on describing an alternative approach to modeling value chains, which are an important part of presenting business activities, and the value each activity delivers. They document translation of data and services into business value, essential in times of ever growing productivity and competition. There are several possible notations for value chain modeling, each holding specific characteristics. Since each domain has its own demands, the goal of this paper is focused on finding the most suitable approach to modeling based on one or more notations, addressing a representative domain within smart cities (a case study of the health domain is included). An approach to value chain modeling is supported by existing notations and documentation techniques.

Categories and Subject Descriptors
H.5.3 [**Information interfaces and presentation**]: Group and Organization Interfaces - *collaborative computing*, organizational design.

General Terms
Documentation. Design.

Keywords
Value chains, use cases, smart city.

1. INTRODUCTION
The paradigm shift in business practices is going from the "product-driven orientation" of the past to today's "customer-driven orientation", which is characterized by increased demand of variability, product variety, amounts of customer-specific products, and shortening product life cycles [1]. Therefore, it is beneficial to the business to identify the key activities and capabilities that flow through the business and define a value chain [2]. A value chain is a high-level model intended to describe the process by which businesses receive raw materials, add value to the raw materials through various processes to create a finished product, and then sell that end product to customers. The concept was suggested by Michael Porter in 1985 [4]. The raw material and product concept can also be transferred to business services and different, intangible business goals. For achieving business goals, companies have to cooperate with or within each other, and their value chains are connected in so-called value systems. Each value system consists of a number of value chains, each of which is associated with one enterprise.

A value chain simplifies complex value systems, since it breaks down the activities a company performs, and analyzes their contribution to the commercial success. In a way it organizes the activities of an enterprise. For example, value chains are a well-known approach in business administration to organize the work that a company conducts to achieve its business goals. Value chains are often used in business modeling for different areas, e.g. medicine, lists of online services, etc. In this paper, we propose an adapted approach to model value chains in the smart city domain, where the value chain describes the transformation cycle of data into value for the benefits of citizens and the community [3].

After the Introduction section, the history of value chain notations is presented in Section 2. The proposed approach is described in Section 3, supported by an example of modeling value chains in Section 4. Lastly, conclusions and future work are presented.

2. VALUE CHAINS
The idea of value chains is to represent an organization as a system divided into subsystems with inputs, transformations and outputs. The process of turning inputs into value-added outputs usually consists of various activities, where some of them are primary, and others are supporting activities [5]. The most common notation for a value chain is by Porter; often used within EU projects like "Project BLUENE", "European Big Data Value Partnership Strategic Research and Innovation Agenda", "European Data Portal". Porter's value chains are, basically, used to identify activities conducted by specific companies, with the purpose of providing a product or a service. They can be applied in different fields, such as the definition of B2B and B2C segments in any field. An example of a classic Porter's value chain is presented in Figure 1.

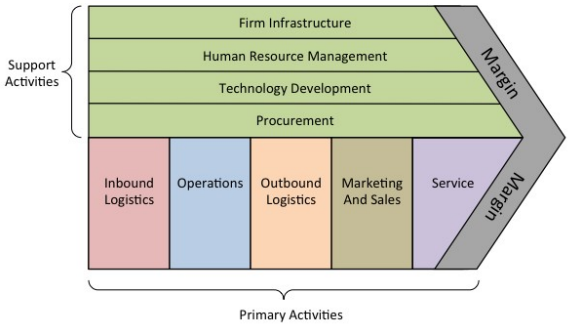


Figure 1. Classic value chain by Porter [4]

With Porter, notation support is provided to describe not only classic supply chain processes, but also services and collaborations among companies that use them. Despite their

popularity, classic value chains are useful only in a limited range of domains. The main disadvantage is their manufacturing oriented format. Therefore, other forms of value chains appeared, especially in the field of ICT, where there are several alternatives to value chain modeling. The first who applied value chains to Information Systems were Rayport and Sviokla within their work on virtual value chains [6]. Some other relevant value chains in the ICT domain are the following:

- **Service value chain** – the relative value of the activities required to establish the product or service,
- **Service value network** – a dynamic way of providing services based on a coordinated value chain of companies,
- **Value stream mapping** – a method for analyzing the current state and planning a future sequence of events that lead the product or service from beginning to client,
- **Data value chain** – information flow is described as a series of steps needed to generate value and useful insights from data [7],
- **Big data value chain** –modeling the high-level activities that comprise an information system, and
- **Digital value chain** –a set of processes designed to transform raw data into actionable information that can drive better decisions and insights [8].

Every value chain begins with inputs. In manufacturing, these are raw materials like steel or wood, while in the field of ICT raw materials can be considered as raw data. In this step, heterogeneous data can be gathered from mobile internet devices, sensor-devices, or extracted from existing sources in structured or unstructured format. Applying new technologies to existing products, practices and processes can be best described with digital value chains, the activities of which are depicted in Figure 2 and described in the continuation.

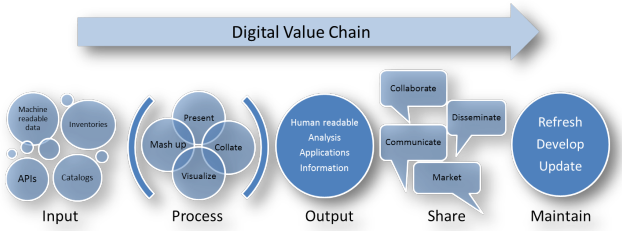


Figure 2. Digital Value Chain [8]

Raw data does not have any value until it is processed. Therefore, in the second step, collected data is processed and, if necessary, mashed up and/or visualized. In the processing, activity data is transformed and mapped from raw format to determined format trough actions such as parsing, joining, standardizing, augmenting, cleansing, consolidating and filtering. Processed data can be combined and exposed through the web APIs, which are analogous to components in manufacturing. More details on this activity are presented within data value chain [9]. As an output, a refined data is provided, new information, or even enhanced functionality, which can be input into the next value chain or used by application developers, leaders or other end users. Application developers can take aggregated data streams and combine them in any number of ways to create information components. Leaders can help with new visualizations to improve decision- making, and others can use information and services to improve their effectiveness [9]. Sharing outcomes is important for promotional purposes, to inform not just end users about it, but also especially developers, who can use it as inputs to develop new innovative solutions. The final step is actually repeating. With new data, technology updates and a new audience, a digital value chain is changing constantly.

3. PROPOSED APPROACH

Despite several notations which were listed and described briefly, none of them fulfilled completely the needs of a complex smart city. In this paper, a proposal of an alternative approach is given, aggregating existing practices, adjusted to the needs of a smart city domain. Usually the aim of the value chain is to increase profits by creating a value, but value chains can also be used to identify opportunities where end users benefit from the final outcome. In the ICT domain, identification of value chains needs detailed consideration of existing problems, obstacles, potential improvements based on ICT, and the inclusion of various stakeholders. The approach described in this paper is based on digital value chains and on documenting existing data, services and processes. It is designed to address and solve issues in several smart city fields using ICT tools and techniques. The most important activity is the documentation of **key challenges**, **target groups** and actors, existing **data sources**, web services and **potential scenarios**.

In the process of identification and documentation of challenges, each challenge was described clearly with all specifics and details, so that everyone could understand it. References to a service, which presents a potential solution were also provided. An example of current challenges in the context of health would be long waiting hours, unnecessary visits to the doctor and improved control of a patient's progress.

A list of target groups describes key actors who are involved in scenarios. Some of the actors are providers, and others are end users such as citizens.

By documenting available data sources and services, various information needs to be provided. Besides the title and description, each end user must understand the data or service purpose and benefits, know who are the owners and potential consumers, and has to be informed about accessibility, privacy restrictions and price. In the case of web services, input and output parameters are important as well.

Based on the inventory of existing data and services, potential scenarios can be defined. Each scenario is described through flow of events with alternatives, and the entire concept is presented with use cases. Some additional information regarding initiating events, participants, included services, inputs/outputs and execution is also provided.

4. AN EXAMPLE OF MODELING A VALUE CHAIN OF A HEALTH CARE SCENARIO

In this chapter, a real example is presented of an applied approach to value chain documentation in the smart city domain. Customized value chains have already been used to define the role and impact of ICT in developing smart cities within other related works [10].

Value chains were designed in accordance with our approach within the EkoSMART program, the purpose of which is to

develop a smart city ecosystem with all the supporting mechanisms necessary for efficient, optimized and gradual integration of individual areas into a unified and coherent system of value chains [11]. One of the most important objectives of the program is to integrate solutions from different sectors into a common ecosystem. The resulting value chains, based on technologies like electronic and mobile devices, related software solutions and intelligent data processing, are enhancing the quality of current services. Moreover, sectoral value chains will be inputs for the cross-sectoral value chains.

4.1 Smart cities and their characteristics

Cities are marked with locations that have a high level of accumulation and concentration of economic activities; they are spatially complex and connected with transport systems. The larger the city, the greater the complexity and the challenges and the risks of disturbances. The fundamental paradigm of the present world is the continuous technological advancement, which, on the one hand, represents a certain proportion of new problems, but, on the other hand, technology is precisely the one thing where key solutions to this problem can be found. Since the world cannot be "reversed", it is necessary to look for suitable solutions that would facilitate modern pressures to focus on the core of new life, which is represented largely by Information and Communication Technologies (ICT). The quality services provided by ICT can relieve people greatly, help them with time optimization, organization, and, last but not least, motivate them.

The purpose of the field is to develop approaches and prototypes, which provide the basic conditions for effective transformation of the healthcare, traffic, energy, waste and other systems, focusing on the following main fields:

- Smart economy,
- Smart people,
- Smart governance,
- Smart mobility,
- Smart environment,
- Smart living.

In the context of a smart city, a value chain is defined as connected activities within a particular sector with various stakeholders, which collaborate with the aim to provide quality services to enhance the life quality and/or strengthen economic growth in an environmentally friendly manner. Designed value chains are intended for data owners, service providers, application developers, city leaders, citizens and others.

4.2 Designing a value chain for the Health sector

In the Health sector, value chains were identified that should be considered within the context of the introduction of smart healthcare services, like telemedicine and telecare. The main goal is the preparation of quality and comprehensive healthcare services using ICT tools and techniques, where value chains are designed to identify and upgrade the occasionally problematic quality of today's treatment and care of these groups, primarily through the use of electronic and mobile devices and related software solutions, in particular artificial intelligence in the cloud, or locally, for example, on a mobile device or with customized sensors and carrying devices. A connection of existing solutions is planned with new smart city solutions. By documenting health processes based on meetings with representatives from the field, the following problems were detected:

- Multiple treatments
- Distribution of services
- Inflexible working time and poor ordering system
- The burden on healthcare personnel and the long waiting period
- Patient monitoring disabled
- Inaccessibility of data
- Deficient legislation, missing Standards and protocols

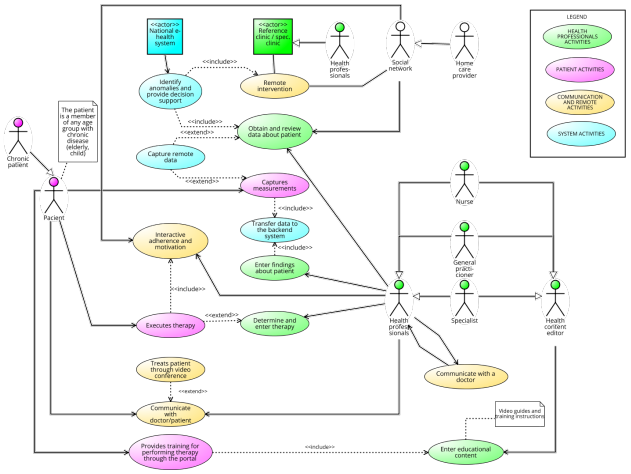


Figure 3: Use case diagram for health vertical.

In order to address these problems effectively, target groups were identified in addition to data and services which are needed to enhance existing processes. All the parts were described and connected in a comprehensive diagram of use cases, which includes a list of activities of identified participants (Figure 3). The common use case diagram includes several possible application scenarios.

Individual usage scenarios were also presented in more detail with a detailed description, characteristics, flow of events and separate use case diagram. The *Establishing treatment* scenario is explained as an alternative value chain presentation, where activities are as follows: The patient has a problem, (1) Enters the system, (2) The system is assigned a medical treatment, and (3) The patient follows this treatment, trying to achieve the set criteria. Characteristics of the scenario are categorized in the following groups: Basic information, People and IT, Inputs / outputs and Implementation. Table I represents actual characteristics for the *Establishing treatment* scenario.

A high-level representation of a final value chain with participants, inputs, outputs and intermediate assets can be seen in Figure 4. The value chain has four pillars: Participants, Input Data, Information/Services and Output results. The participants are the providers as well as users of the service, followed by all data necessary and, further, more services, designed by and for the participants, based on accumulated data. Finally, based on all previous pillars, final services with added value for the city (or company) in the form of different outputs and results are presented.

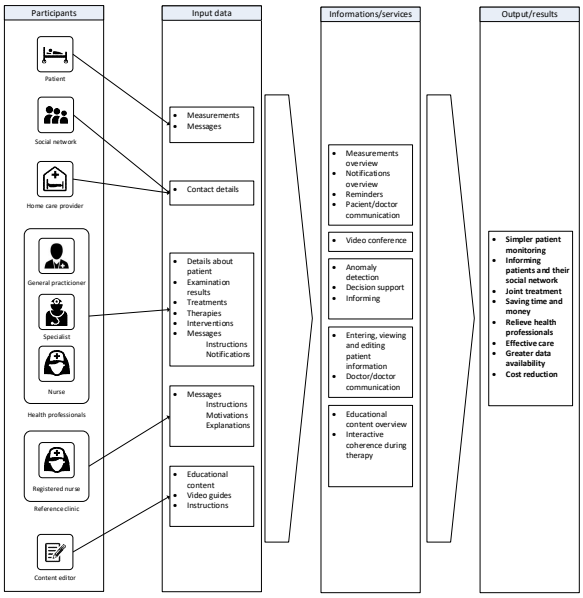


Figure 4. High-level representation of a value chain in the Healthcare domain.

Table I Characteristics for the scenario "Establishing treatment"

Scenario name	Establishing medical treatment
Description of the scenario	The purpose of the scenario is to describe the initialization of the treatment of a patient with a chronic disease. The scenario involves ordering a patient for a review where the doctor gives them a treatment, and the nurse introduces treatment information and informs the patient of the use of the assigned equipment and the implementation of the activity.
Variants	If the patient has several treatments, the doctor will obtain further findings and, on the basis of communication with other doctors, will form a joint therapy
The trigger of the scenario.	The process is passed by a patient who comes to the check due to the problem.
Participants	Patient, Health personnel
Included services	A service for entering data processing and Editing educational content
Scenario input.	Patient information, Data processing
Scenario output	Program, Schedule, Therapies, Educational content
Activities	Obtaining / entering patient information, Data entry information, Entering therapy, Establishing patient / doctor and doctor / doctor communication, Editing educational content

5. CONCLUSION AND FUTURE WORK

Graphical presentation value within any company is an important part of understanding the focus of business processes, their strengths and weaknesses. Several techniques can be used to present the value flow. However, a combination of notations was used for the purpose of presenting the smart city complex system of users, data, services and scenarios. A use case diagram was used to present the behavior and set of actions of several participants. Within a use case diagram, several scenarios can be derived; each scenario defined in the form of a Table (characteristics of a scenario). Lastly, a high-level representation of a value chain is presented (including the four value pillars). In the future work, refinement of the approach will be performed.

6. ACKNOWLEDGMENTS

This joint work is enabled by the program "Eko Sistem Pametnega Mesta", supported by the European Union, European Regional Development Fund and Ministry of Education, Science and Sport.

7. REFERENCES

[1] Martínez-Olvera, C., Davizon-Castillo, Y. A. 2015. Modeling the Supply Chain Management Creation of Value — A Literature Review of Relevant Concepts. *Business, Management and Economics » "Applications of Contemporary Management Approaches in Supply Chains"* (Apr- 2015).

[2] Business Modelling. https://www.enterprise-architecture.org/business-architecture-tutorials/79-business-value-chain. Accessed: 2017-09-15.

[3] *Smart City Value Chain*. White Paper e-madina. November 2016. http://www.e-madina.org/wp-content/uploads/2016/11/White-Paper-e-Madina-3.0-Value-Chain-of-Smart-cities.pdf.

[4] Porter, M. E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York.: Simon and Schuster. Retrieved 9 September 2013.

[5] "*Decision Support Tools: Porter's Value Chain*". Cambridge University: Institute for Manufacturing (IfM). Retrieved 9 September 2013

[6] Rayport, J. F., & Sviokla, J. J. 1995. *Exploiting the virtual value chain*. Harvard Business Review, 73, 75–85.

[7] Curry, E. 2016. *The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches.* New Horizons for a Data-Driven Economy. Springer International Publishing, 2016. 29-37.

[8] Data Tip #1 – Your Digital Value Chain: 2013. http://captricity.com/blog/data-tip-1-your-digital-value-chain/. Accessed: 2017-09-15.

[9] Understanding the Data Value Chain. 2014. [http://www.ibmbigdatahub.com/blog/understanding-data-value-chain]. Accessed: 2017-09-15.

[10] Webb, M., Finighan, R., Buscher, V., Doody, L. and Cosgrave, E. 2011. *Information marketplaces- The new economics of cities.* The Climate Group, Arup, Accenture. (2011).

[11] EkoSmart – Ekosistem pametnega mesta. 2017. http://ekosmart.net/. Accessed: 2017-09-1

Using Property Graph Model for Semantic Web Services Discovery

Martina Šestak
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
+385 42 390 847
msestak2@foi.hr

ABSTRACT
Web services have significantly contributed to the integration of different businesses. Service-oriented computing (SOC) paradigm still represents an implementation challenge for developers. Several approaches have been developed over the years for different processes related to Web services. Nowadays, traditional Web services are often supplemented with semantics to achieve higher levels of automation and interoperability. In this paper, a new approach for semantic Web services discovery based on property graphs is proposed. The proposed model proves that the semantic Web service model specified in OWL-S language can be represented as a property graph, which can be queried to discover Web services based on query parameters.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Query formulation, Retrieval models, Search process, Selection process.*

H.3.5 [**Information Storage and Retrieval**]: Online Information Services - *Web-based services.*

General Terms
Design, Languages, Standardization, Theory.

Keywords
Labeled property graph model, web services discovery, PGQL.

1. INTRODUCTION
Application integration is an important challenge in the modern business environment. Over the years, many concepts and solutions have been developed to address this challenge (e.g., middleware or Enterprise Application Integration solutions). The most recent solution for integrating multiple applications are Web services. Their compliance with the existing Web technologies and standards and platform independency represent a significant advantage.

Web services can be defined as a "software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."[17].

Nowadays, due to the information overload of SOAP messages, RESTful Web services are used more often. Since their focus is on resources[11], the messages exchanged between applications have a simpler format, which makes REST services a simpler alternative to SOAP-based services, which is more applicable in many situations.

Through both technologies, the client can access and retrieve the required data from a specific Web service by invoking the correct interface method of that service.

According to [16], each Web service should be capable of being defined, described, and discovered. Web service description process can be divided into three layers[14]:

1. service invocation
2. service publication and discovery description
3. composite web services description

Recently, the concept of SOA has been supplemented with semantic Web concepts, which resulted in semantic Web services (SWS) technology. SWS enables the Web services to be automated and carried out by intelligent software agents [4]. In SWS, additional meaning is added to the basic Web service information. Thus, the main motivation behind this technology is to increase the level of automation of information processing, and to improve the interoperability of Web services.

There are several languages developed for semantic Web services as well. OWL-S is the most popular Web service ontology used for SWS description. In OWL-S, a semantic Web service description consists of three elements [8]:

1. service profile – contains general information about the service (name, description, inputs, outputs, preconditions, results)
2. service model – contains information about how the service works (by using structures like loops, sequences, etc.)
3. service grounding – contains information about how to use the service

In this paper, the focus will be on the **service model** element. The process of semantic Web services discovery will be discussed by analyzing several models proposed in the literature. Based on this work, a new approach will be proposed and explained.

The rest of the paper is organized as follows: in Section 2, different approaches for semantic Web services discovery process will be discussed. In Section 3 and 4, labeled property graph model and the property graph query language (PGQL) properties will be explained. In Section 5, the new approach will be introduced and described. Finally, a conclusion will be made to summarize the characteristics of the proposed approach and challenges, which will be further analyzed in future work.

2. SEMANTIC WEB SERVICES DISCOVERY APPROACHES
Web services discovery in general is "the act of locating a machine-processable description of a Web service that may have been

previously unknown and that meets certain functional criteria" [15]. The goal of the process is to find an appropriate service within the Web service directory which meets some predefined criteria. It is worth mentioning that in recent years the importance of other nonfunctional criteria (e.g., reliability, response time, availability, etc.)[12] has also been recognized, which led to the development of different Quality of Service (QoS) modeling approaches in the (semantic) Web services description, discovery and composition processes.

As already mentioned, OWL-S is one of the ontology languages, which can be used in the SWS discovery process. OWL-S service model contains Web services viewed as a collection of processes, which represent the specification of how the client interacts with the service [9]. If the process receives information and returns some new information based on its input, the information production is described only by specifying inputs and outputs of that process. Otherwise, if the process makes more complex transformations and changes, then the production is described by the process preconditions and results [9]. A process may require some information to be executed, i.e., it can have any number of inputs, and it can also produce any number of outputs for Web service requestors. Thus, process inputs and outputs specify the data transformation, which takes place during the process execution.

A sample OWL-S service model is shown in Fig. 1. A Web service called "BorrowedBooks" is shown as a process, which returns the Transaction ID, client name, date when a requested book was borrowed, and whether the book was returned for a given title of the book and its author. Input information is shown as an incoming edge, and output information as the outgoing edge from the process.

Many different approaches have been proposed in the literature for the discovery (selection) process.

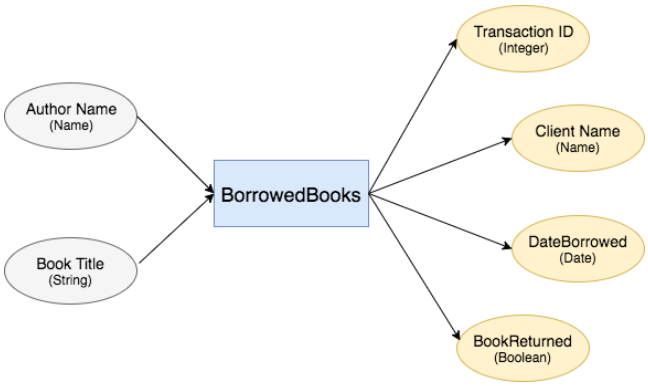


Figure 1. Sample OWL-S service model

3. RELATED WORK

In [2], authors have divided different SWS discovery approaches into three categories: algebraic, deductive and hybrid approaches. The algebraic approach includes approaches based on graph theory (e.g., iMatcher, AASDU[1], etc.), the deductive approach uses logic-based reasoning (e.g., Inputs and Outputs, Preconditions and Effects matching, etc.), while the hybrid approach combines both algebraic and deductive approach.

Sachan et al. [12] proposed a new modeling approach for QoS-based semantic WS model and formalization of several QoS attributes. The proposed QoS approach is agent-based, i.e., introduces the additional mediator Agent, which selects the most appropriate Web service available based on different QoS parameters set by the clients (users). The authors built an ontology of selected QoS parameters, and used that ontology on the model built in OWL-S Editor available in Protégé.

Klusch et al. [5] introduced a hybrid SWS matching approach with a mechanism called OWLS-MX, which they applied to services specified in OWL-S. The authors managed to prove that logical reasoning is not sufficient for semantic Web services discovery, so they combined logical reasoning with information retrieval (IR) similarity metrics.

Srinivisan et al. [13] presented an OWL-S/UDDI matchmaker architecture. The authors used OWL-S Integrated Development Environment (IDE) to build and discover OWL-S based Web services. OWL-S IDE supports various processes of the SWS lifecycle (service description, publication, discovery and execution). The Web service description can be generated based on code or model within the OWL-S Editor. Services descriptions are stored inside OWL-S registry. Web service discovery is performed through executing a query to the registry by using a specific Application Programming Interface (API). The registry performs a matching process, and returns OWL-S descriptions of the matched services.

4. PROPERTY GRAPH MODEL

The property graph data model is nowadays the base data model for many graph databases (e.g., Neo4j, Titan, etc.). This model is an easy-to-understand representation of the way data is stored in graph databases. Since it represents an extension to graphs in mathematics, it can be formalized in the following way [1]:

A property graph G is a tuple (V,E, ρ, λ, σ), where:

a. *V is a finite set of vertices (or nodes).*
b. *E is a finite set of edges (or relationships) such that V and E have no elements in common.*
c. *$\rho : E \rightarrow (V \times V)$ is a total function. Intuitively, $\rho(e) = (v1, v2)$ indicates that e is a directed edge from node v1 to node v2 in G.*
d. *$\lambda : (V \cup E) \rightarrow Lab$ is a total function with Lab a set of labels. Intuitively, if $v \in V$ (resp., $e \in E$) and $\rho(v) = \iota$ (resp., $\rho(e) = \iota$), then ι is the label of node v (resp., edge e) in G.*
e. *$\sigma : (V \cup E) \times Prop \rightarrow Val$ is a partial function with Prop a finite set of properties and Val a set of values. Intuitively, if $v \in V$ (resp., $e \in E$), $p \in Prop$ and $\sigma(v, p) = s$ (resp., $\sigma(e, p) = s$), then s is the value of property p for node v (resp., edge e) in the property*

(Labeled) property graph data model consists of the following elements [6]:

- Nodes - different entities with attributes and unique identifier
- Labels - semantical description of the role of each entity, where a single node or relationship can have multiple labels at the same time
- Relationships - connections between nodes, where each connection has a start and an end node

[1] AASDU (Agent Approach for Service Discovery and Utilization)

- Properties - key-value pairs, which represent node and relationship attributes

A simple property graph model shown in Fig. 2 contains 3 nodes. Node labeled "Group" has a property "Name", and the other two nodes with no labels have properties "Name" and "Age". These two nodes are connected with a relationship labeled "knows", which has a property "Since" indicating for since when the two persons know each other.

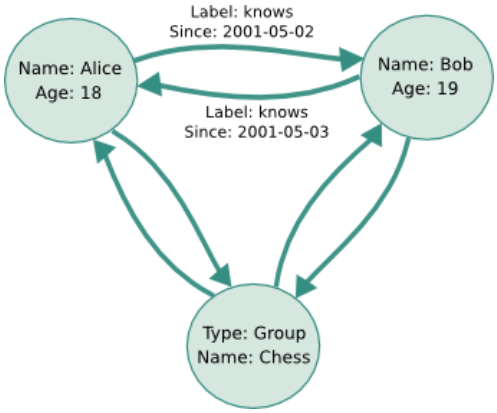


Figure 2. Sample property graph model [3]

Property graphs represent an expressive and simple mechanism for describing the richness of data [7], where a connection between two nodes is easily represented, and both nodes and relationships can have various attributes of different complexity. The property graph model is an easy-to-understand representation of the property graph, which is why it can be used for modeling semantic information about Web services. As shown in the previous section, the OWL-S service model is also a directed graph. Thus, in the proposed approach the described concepts of the property graph data model will be applied to the OWL-S service model.

However, in order to efficiently query the property graph model, several query languages have been developed. In the following section, the characteristics of the Property Graph Query Language (PGQL) will be discussed.

5. PROPERTY GRAPH QUERY LANGUAGE (PGQL)

PGQL is a new SQL-like query language for property graphs developed by Oracle [10]. The language offers a wide collection of statements to be executed in order to query the property graph and find the required data.

PGQL is based on graph pattern matching algorithm, i.e., when executing a PGQL query, the query engine finds all subgraphs within the original graph, that match the specified query pattern.

To query a property graph, the SELECT clause is used, which specifies the data entities to be returned in the query result. In the example property graph shown in Fig. 3, to return the name of the persons who know each other, the following PGQL would be executed:

SELECT n.name, m.name

WHERE

(n WITH type='Person')-[e:knows]->(m WITH type='Person')

The pattern *(n)-[e]->(m)* defined in the WHERE clause represents a topology constraint, which is a description of a connectivity relationship between vertices and edges in the pattern [10].

For n-step hops between nodes it is possible to specify path expressions, which are then used in the WHERE clause of the query.

6. PROPERTY GRAPH-BASED APPROACH FOR SEMANTIC WEB SERVICES DISCOVERY

Since the OWL-S representation of the service model is a graph, the characteristics of property graph models can be applied to the OWL-S service model. This graph-based service model can then be queried using the PGQL clauses during the semantic Web services discovery process.

The proposed approach will be explained on the sample Web service model shown in Fig. 3. The Web service called "MovieService" contains the following three processes:

1. *GetMovieGenres* – for a given movie name and year returns the genre name of that movie
2. *GetMoviePersonnel* – for a given movie name and year return the list of all actors' and directors' names of that movie
3. *GetGenreDirectors* – for a given genre name returns the list of directors' names which produced movies in that genre

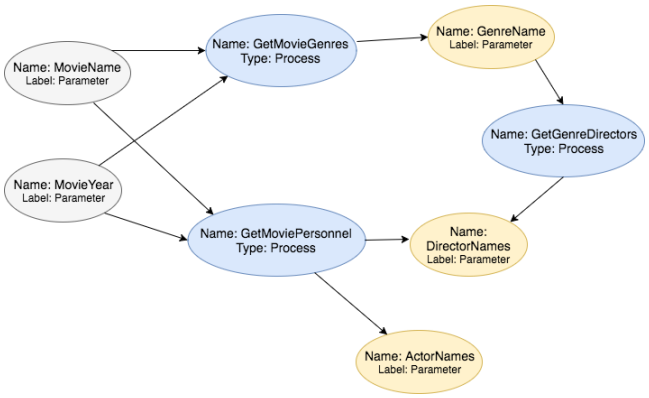


Figure 3. Sample service model of the proposed approach

The sample service model contains three processes with different number of input and output parameters. Since this is a property graph, both nodes and relationships can be supplemented with additional labels and properties. The processes shown in this model are simple, so they are described only with their input and output parameters.

In the example, the nodes representing the processes have a label (type) "Process", which distinguishes them from parameter nodes labeled "Parameter". A parameter node can represent both an input and an output parameter of the process (e.g., parameter "Genre Name" is the output parameter of the "GetMovieGenres" process, but the input parameter of the "GetGenreDirectors" process).

The defined property graph service model can be queries by using PGQL.

In order to discover (find) Web services, which use movie name as an input parameter, the following PGQL query would be executed:

SELECT s.name

WHERE (p1 WITH name = 'MovieName')-[e1]->(s)

It is also possible to discover which Web services can be called to find director names for a given movie name with the following PGQL query:

PATH get_directors := ()-[]->(s WITH type = 'Process')-[]->()

SELECT s.name

WHERE

(p1 WITH name = 'MovieName')-/:get_directors/->*

(p2 WITH name = 'DirectorNames')

The proposed property graph model representing the service model can be easily implemented in a graph database (e.g., Neo4j). By using different libraries developed for connecting with graph databases, a developer can create Web services input and output parameters as node instances (classes), store them in a graph database instance, and include them in a specific Web service class definition. The PGQL queries can then be executed on the database instance to find the necessary web service and other information.

Therefore, the property graph model represents a new approach, which combined with the PGQL query language could be used for semantic Web services discovery. At this moment, the model can be used to represent a simplified OWL-S service model without including QoS parameters mentioned in the previous sections.

7. CONCLUSION AND FUTURE WORK

In this paper, the characteristics of semantic Web services have been discussed with a special focus on SWS discovery approaches. Based on the OWL-S ontology language and its graph representation of semantic Web service model, a new approach has been proposed. The approach includes using property graphs to model semantic Web services, and discovering the required services by executing PGQL queries on that property graph. The built property graph can be implemented in graph databases to build a graph database of existing Web services and used for SWS composition process. Future work includes extending the model by adding Web services methods (operations), and by including and verifying different QoS parameters against the proposed model.

8. REFERENCES

[1] Angles, R. A Foundations of Modern Graph Query Languages.

[2] Bitar, I. El, Belouadha, F.-Z. and Roudies, O. 2014. Semantic web service discovery approaches: overview and limitations. *arXiv preprint arXiv:1409.3021*. (2014).

[3] Getting started with SylvaDB: *http://sylvadb.com/get-started/*. Accessed: 2017-09-14.

[4] Klusch, M. 2008. Semantic web service description. *CASCOM: intelligent service coordination in the semantic web*. Springer. 31–57.

[5] Klusch, M., Fries, B. and Sycara, K. 2006. Automated Semantic Web Service Discovery with OWLS-MX. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2006), 915–922.

[6] Lal, M. 2015. *Neo4j Graph Data Modeling*. Packt Publishing Ltd.

[7] Malak, M. and East, R. 2016. *Spark GraphX in action*. Manning Publications Co.

[8] Martin, D., Paolucci, M., McIlraith, S., Burnstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T.R., Sabou, M., Solanki, M. and others 2004. Bringing semantics to web services: The OWL-S approach. (2004).

[9] OWL-S: Semantic Markup for Web Services: *https://www.w3.org/Submission/OWL-S/*.

[10] PGQL 1.0 Specification: 2017. *http://pgql-lang.org/spec/1.0/*. Accessed: 2017-09-15.

[11] Rodriguez, A. 2008. Restful web services: The basics. *Online article in IBM DeveloperWorks Technical Library*. November (2008), 1–11.

[12] Sachan, D., Dixit, S.K. and Kumar, S. 2014. QoS aware formalized model for semantic Web service selection. *International Journal of Web & Semantic Technology*. 5, 4 (2014), 83.

[13] Srinivasan, N., Paolucci, M. and Sycara, K. 2006. Semantic web service discovery in the OWL-S IDE. *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on* (2006), 109b--109b.

[14] Varga, L.Z. and Sztaki, Á.H. 2005. Semantic Web Services Description Based on Web Services Description. *W3C Workshop on Frameworks for Semantics in Web Services* (2005).

[15] Web Services Architecture: 2004. *https://www.w3.org/TR/ws-arch/*.

[16] Web Services Architecture Requirements: 2002. *https://www.w3.org/TR/2002/WD-wsa-reqs-20021011*.

[17] Word Wide Web Consortium 2004. Web Services Architecture.

Statecharts representation of program execution flow

Nataša Sukur
Faculty of Sciences
University of Novi Sad
Trg Dositeja Obradovića 3
21000 Novi Sad, Serbia
nts@dmi.uns.ac.rs

Gordana Rakić
Faculty of Sciences
University of Novi Sad
Trg Dositeja Obradovića 3
21000 Novi Sad, Serbia
goca@dmi.uns.ac.rs

Zoran Budimac
Faculty of Sciences
University of Novi Sad
Trg Dositeja Obradovića 3
21000 Novi Sad, Serbia
zjb@dmi.uns.ac.rs

ABSTRACT

Source code and software in general is prone to errors. This is due to bad design decisions, lack of experience of developers and constant need to change the existing software. Because the code changes rapidly and in strict time limitations, it is not always possible to preserve the quality and reliability of the source code. It is very important to detect errors in time and to fix or remove them in an automated or semi-automated manner. Automation of error detection and removal can be accomplished by various tools or platforms. The platform which is used for software quality analysis in this case is a static analysis oriented, language independent, SSQSA (Set of Software Quality Static Analyzers), which is based on an universal intermediate source code representation, eCST (enriched Concrete Syntax Tree). The tree structure is useful for code representation and comprehension, however its structure is not immediately suitable for representing program control flow. That is why the control flow graphs were introduced to SSQSA and were represented later visually in the form of higher level automata, statecharts. Apart from introducing formal representation of control flow graphs, statecharts introduced additional functionalities to SSQSA. Some of them are hierarchical control flow graph representation, possibility of simulation in one or more parallel work flows, also planned to be expanded to interprocedural level, and performing various kinds of estimation.

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification—*Formal methods*

General Terms

Measurement, Languages

Keywords

software quality, static analysis, formal methods, control flow graphs, statecharts

1. INTRODUCTION

Formal methods are getting more and more attention in the world of software. In the beginning, using formal representation was more usual for hardware than for software. Software is more complex in terms of state components and the process to produce abstract models is more difficult.[1, 3] Formal verification of software systems is seldom automated. Usually, an abstract model is manually created in order to perform formal verification of a large system. This requires investing a considerable amount of time and expertise. However, because the model was manually made, analysis of this model cannot be considered reliable. Systems are also usually large and change quickly. This means that it is very complex to continuously create new and update existing abstract models, as well as expensive, prone to errors and difficult to optimize. For all those reasons, an automated solution to abstract model generation is needed.[2, 5] Also, there has been a lot of foundational work on defining safe abstractions, but research for model reduction has not been explored enough.[3]

There are many approaches to software analysis and software quality measurement. Algorithms that are implemented for this purpose usually operate on some internal representations of this code, such as trees, graphs or some meta models.[5] SSQSA platform [9] uses its own structure, called enriched Concrete Syntax Tree and derived graph-based representations. Some of the algorithms for software quality measurement in SSQSA use this intermediate structure to perform their calculations and analysis, such as software metrics analysis, timing analysis and code clone detection.

Enriched Concrete Syntax Tree is the key of language independence of SSQSA. It is a syntax tree which contains all concrete tokens from the original source code, including comments, and enriched by a predefined set of universal nodes. These nodes are created in order to generalize different structures of various programming languages whose purpose is the same. The set of universal nodes is minimal, and the nodes were carefully selected, so that they are applicable to the structure of all languages supported by SSQSA. They are inserted as imaginary nodes in this tree together with the original source code tokens. For example, if we represented a semicolon and a comma with a universal node SEPARATOR, we would also have the data that precisely represents the SEPARATOR universal node. The whole source code is represented like this and it is possible to completely reconstruct the original source code from eCST.

Sometimes it is not necessary to have a structure which contains the amount of data such as eCST and it is not optimal for all kinds of analysis. That is why it was necessary to introduce Control Flow Graphs [6] to this platform. They were derived from eCST by extracting only the nodes which were of importance for control flow analysis.

Although this was an upgrade to SSQSA features and some additional algorithms were implemented on this structure, there was still a problem in the case of larger pieces of code, where the resulting graphs were quite complex [1]. Representing control flow graphs in a hierarchical manner seemed like a good solution for reduction of complexity. That is one of the reasons why statecharts were introduced to SSQSA. Apart from solving the complexity issue, they give us the possibility of simulation, parallel execution and estimation.

The rest of the paper is organized as follows: Section 2 reflects on some of the related papers and tools. Section 3 describes the introduction of Control Flow Graphs to SSQSA. Section 4 explains the meaning and purpose of statecharts in SSQSA. In Section 5, an example of a statechart is shown and described in more detail. In Section 6, future work is presented and finally, we conclude our paper in Section 7.

2. RELATED WORK

There are many tools and papers that deal with visual representations and simulation of the source code. However, most of them lack some features. Usually, they focus on some specific programming language. Some of them are able to represent the control flow or the structure of programs in a visual way, but most of them do not have the possibility of simulation and testing. If there are testable representations, then another problem emerges: the state explosion problem [1]. Also, many of them are not formal representations and not all of them are platform independent.

There have already been some attempts at creating formal representations of source code. In papers, such as [2] it is explained how the authors extracted finite state models from Java source code. Their approach also addresses the state explosion problem and tries to solve it. However, this solution is only applicable to Java code.

Work by [7] deals with interprocedural graphs. The intention is to check if fixing errors in code actually eliminate them and whether fixing errors means introducing some new ones. The whole approach is based on static analysis and it generates the graph based on all existing versions of program and tries to discover and fix faults and propagate that to newer versions.

A PhD thesis [10] performs static analysis of programs, based on their dependence graphs. The idea is that a single statement can affect some other statements and parts of code. This approach is formal and language independent. It focuses on sequential, imperative programs.

2.1 Related tools

Visustin v7 Flow chart generator[1] is a visualization tool that can represent code in the form of program flow diagram and activity diagram. It has support for 43 languages, such as C, COBOL, Fortran, Java, JavaScript, Pascal, PHP and Ruby. It has also support for simulation, but it has a problem when it is presented with large pieces of code - it becomes hard to analyze and has no hierarchical representation.

Graphviz (Graph Vizualization Software)[2] is open source and it is used for graph visualization. Graphviz has wide use in networking, bioinformatics, software engineering, database design, machine learning... This tool performs graph drawing, based on specifications in DOT[3] language. The downside is that the graphs have to be first specified in DOT language in order to be represented by Graphviz. It is also not able to simulate control flow behavior.

Apart from these, there are also other tools that have some similar features, such as MOOSE[4] and RefactorErl[5]. However, some of them are not oriented towards formal representation and some cannot simulate the represented code.

3. CONTROL FLOW GRAPHS IN SSQSA

The control flow graph can be represented as a set

$$G = (N, A, S, E) \tag{1}$$

where N are the nodes of the graph, A is a binary relation N x N, which represents the graph transitions, S and E are start and end nodes of the graph [6]. The purpose of control flow graphs is to track all possible paths of program execution for important reasons, such as to detect dead code or infinite loops. In order to generate them, it was necessary to extract them from eCST. The subset of universal nodes was selected and it was enough to represent the program flow accurately.

Some nodes of importance for the control flow representation are statement nodes, such as assignment statements and function calls. Apart from them, nodes which were also included in the graph are branch statements, branches and loop statements, as well as their corresponding condition nodes. Some pieces of information were included because of their importance for generating statecharts. Statecharts are highly structured and it is important for us to preserve that structure by saving information about nodes such as compilation unit, block scope and function declaration in the control flow graph.

The control flow graph that was first created in SSQSA focused only on one function or procedure. The starting node was the entry point of this function/procedure. The rest of the control flow graph was created by extracting previously defined nodes of interest from eCST and connecting them in a way that they represent control flow of the original source code. This graph is directed and it has cycles, which exist due to the nature of source code. If the language independent condition evaluation is successfully implemented (the number of repetitions of some cycles is calculated), it will be possible to perform calculations, such as worst case execution time estimation. Currently, we are trying to create interprocedural graphs [8], described in detail in Section 6.

[1] http://www.aivosto.com/visustin.html

[2] http://www.gnu.org/softwarhttp://www.graphviz.org/

[3] http://www.graphviz.org/doc/info/lang.html

[4] The MOOSE book, http://www.themoosebook.org/book

[5] http://plc.inf.elte.hu/erlang/

4. STATECHARTS

Statecharts are defined as a visual formalism for complex systems [4]. They were later included in UML diagrams. Another name for this diagram is Harel's automaton[6]. The main benefit of using statecharts reflects in the ability of representing parallel states, tracking history within complex states and tracking values of variables throughout the flow. They are highly expressive - they are able to show a very detailed preview of the system which is to be created, but they can also be very compact due to their hierarchical nature and show only the system on higher levels of abstraction.

Statecharts take care of modelling hierarchy, concurrency and communication and that makes them important for tracking complex real-time systems. Reactive systems are event driven and they constantly have to react on various kinds of internal and external events. It was very difficult to represent them in a way that was realistic, but also formal and precise. Statecharts are a solution to that problem, since they make the process of specifying and designing these complex systems a lot easier and more natural. All possible behavior of reactive systems was easily represented by a set of allowed in and out events, conditions and actions and some time limitations.

Dynamic behavior of a complex system is easily represented by using states and events. The system is always in at least one state and if some event occurs, it transitions to another state under some conditions. A transition can occur from one state to another, which can all happen inside a complex state. Transitions can also be recursive if they have the same state as origin and target. Some transitions can cause to exit or enter a complex state. They can also trigger events, which affect the simulation. If standard finite automata were used for this purpose, they would be very difficult to understand due to very large complexity because of the generated number of states. Statecharts use hierarchy, they provide us modularity and good structure and make it easy to represent independent parallel execution.[4]

4.1 The importance of statecharts for SSQSA

Statecharts introduced graphical representation of Control Flow Graphs to SSQSA and created a more compact version of them. Also, because statecharts are a formal representation, the system is represented in a non-ambiguous way and it becomes a very trivial problem to detect program paths which are possible and the ones which are not. This component is useful for testing parts of code in early stages of development. It has some features of a debugger - it can show us if parts of code show odd behavior, why the control flow unexpectedly changes, and it includes much more visualization in representing what is currently happening.

4.2 Implementing statecharts in SSQSA

Statecharts consist of two kinds of states, complex (which can also be orthogonal, with parallel regions) and simple ones. Based on the nature of the universal nodes that were used (if they represented complex or simple program structures), they were transformed into suitable kind of states. Universal nodes that stand for complex program constructs (i.e. that contain other program elements) were represented as complex states. For example, these are compilation units which can consist of elements such as functions, or function declarations which can contain statements. Some statements are also represented as complex states because they contain other statements, such as branch statements (and their branches) and loop statements. These complex states can contain other simple or complex states. Universal nodes that were represented as simple states can only trigger some events or manipulate the variable values. Statecharts also have some additional states related to entering and exiting the whole statechart or its complex parts.

For the purpose of testing statecharts, Yakindu SCT[7] was used. Parts of this tool are open source. Some of the limitations are directly related to this tool, such as lack of complex data types. For now, only integer, real, boolean, string and void are supported. These data types are enough for the proof of concept phase, but it is necessary to additionally implement these features or replace this tool with one that can also represent nontrivial pieces of code.

5. EXAMPLE

In Figure 1, we present a statechart generated based on a piece of code written in programming language Modula-2. It represents a part of an algorithm which counts factoriel of a given number. The purpose of this figure is to show how the part of code which is a loop is represented in a simplified way. In Figure 2, we present how this loop statement looks like when it is expanded and what is really happening inside this complex state.

The same algorithm was implemented in Java programming language. The resulting statecharts are identical to the ones in Figure 1 and 2. A more detailed preview and comparison of different algorithms can be found in [11].

6. FUTURE WORK

Although statecharts are currently focused on representing the control flow of one function or procedure, the idea is to make an interprocedural representation, first on the level of a compilation unit and then to expand it to represent the

[6]After David Harel, the creator of statecharts

[7]https://www.itemis.com/en/yakindu/state-machine/

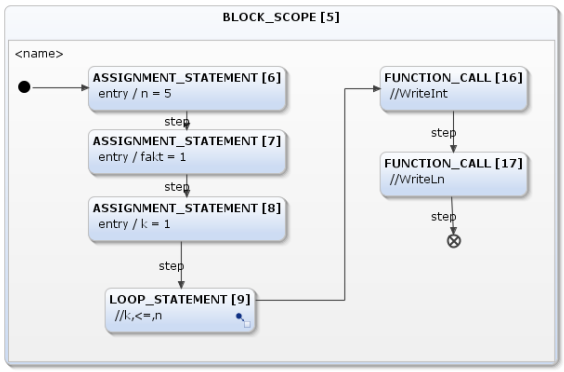


Figure 1: Statechart based on sample code in Modula-2, which shows how factoriel of a number is calculated. The loop statement is collapsed.

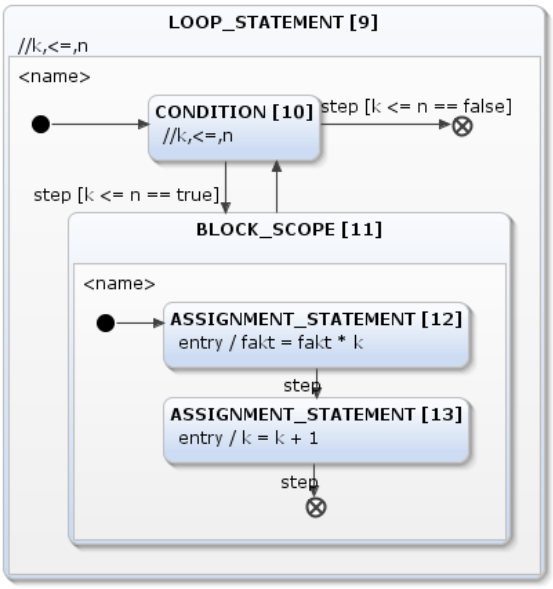


Figure 2: Statechart based on sample code in Modula-2, which shows the expanded loop statement in factoriel calculation.

complete software. This will be done using the graph dependency networks. Once the control flow graphs for all procedures and functions are constructed, function call nodes will be detected and these graphs will be connected into one, which represents the whole system. By implementing this, it will be possible to improve statecharts, also in an interprocedural way. So far, statecharts were tested mostly on one object-oriented language (Java) and one procedural Pascal-like language (Modula-2). Therefore, there is also room for improvement in terms of testing how statecharts are generated based on source code written in other languages.

Another idea is that, if we succeeded in refining statecharts to the lowest level and if we introduced the environment variable, we would greatly improve simulation of the source code in the evaluator. That would mean having the most realistic representation of how whole or a part of source code would execute in reality.

7. CONCLUSIONS

The eCST provides us with complete information about the source code. Therefore, possible limitations will not be related to lack of information about the source code. The true challenge will be to represent everything that is important in a manner suitable to the nature of statecharts. Our approach has proven feasible so far, but that will be put under further inspection once more complex pieces of code are introduced.

Once we have a component that is possible to visualize and simulate the complete code under analysis and test existing systems or the ones that are still under development, detecting obvious flaws in source code design and execution will be trivial. The tool will be able to simulate the execution of the code without the need for setting up the environment and running the code. Statecharts could also be used to introduce new people to various projects. One will be able to view the system on different levels of abstraction and take a step by step approach in getting familiar with it. A statechart is more dynamic than a simple diagram which represents project structure. Statecharts in SSQSA would mean the ability of viewing systems, simulating them, and not having to worry if parts of this system are written in different languages.

It is also important to note that statecharts are not introduced to SSQSA only to visualize and simulate the system. They are important for predicting different outcomes if some parts of code are executed and to evaluate qualities, such as correctness and reliability.[4]

8. REFERENCES

[1] E. M. Clarke, W. Klieber, M. Nováček, and P. Zuliani. Model checking and the state explosion problem. In *Tools for Practical Software Verification*, pages 1–30. Springer, 2012.

[2] J. C. Corbett, M. B. Dwyer, J. Hatcliff, S. Laubach, C. S. Pasareanu, H. Zheng, et al. Bandera: Extracting finite-state models from java source code. In *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 439–448. IEEE, 2000.

[3] M. B. Dwyer, J. Hatcliff, R. Joehanes, S. Laubach, C. S. Păsăreanu, H. Zheng, and W. Visser. Tool-supported program abstraction for finite-state verification. In *Proceedings of the 23rd international conference on software engineering*, pages 177–187. IEEE Computer Society, 2001.

[4] D. Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[5] G. J. Holzmann and M. H Smith. Software model checking: Extracting verification models from source code. *Software Testing, Verification and Reliability*, 11(2):65–79, 2001.

[6] J. Laski and W. Stanley. *Software verification and analysis: An integrated, hands-on approach*. Springer Science & Business Media, 2009.

[7] W. Le and S. D. Pattison. Patch verification via multiversion interprocedural control flow graphs. In *Proceedings of the 36th International Conference on Software Engineering*, pages 1047–1058. ACM, 2014.

[8] F. Nielson and H. R. Nielson. Interprocedural control flow analysis. In *ESOP*, volume 99, pages 20–39. Springer, 1999.

[9] G. Rakić. *Extendable and Adaptable Framework for Input Language Independent Static Analysis, Novi Sad Faculty of Sciences, University of Novi Sad*. PhD thesis, doctoral dissertation, 2015.

[10] J. A. Stafford and A. L. Wolf. *A formal, language-independent, and compositional approach to interprocedural control dependence analysis*. PhD thesis, University of Colorado, 2000.

[11] N. Sukur. Reprezentacija toka izvrsavanja programa dijagramom stanja, nezavisna od ulaznog jezika. Master's thesis, Faculty of Sciences, University of Novi Sad, Serbia, 9 2016. in Serbian.

Code smell detection: A tool comparison

Tina Beranič
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
tina.beranic@um.si

Zlatko Rednjak
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
zlatko.rednjak@gmail.com

Marjan Heričko
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Maribor, Slovenia
marjan.hericko@um.si

ABSTRACT
Technical debt can be identified with different techniques, including code smell detection. Furthermore, different approaches are available to detect code smells and some of those approaches are implemented among different tools. In this article, different tools for code smell detection were selected with the goal of comparing their outputs. The fact is that compared tools detect different code smells with varying degrees of success and that the intersection of detected code smells within tools is very small. Because of this, the connection between detected code smells and the value of technical debt is hard to define. The results are supported with empirical analysis from 32 software projects, different code smells and 3 code smell detection tools.

Categories and Subject Descriptors
D.2.8 [**Software Engineering**]: *Metrics*

D.2.9 [**Software Engineering**]: *Management-Software quality assurance (SQA)*

General Terms
Measurement

Keywords
Code smell, technical debt, intersection of detected code smells

1. INTRODUCTION
Since there is a growing need for rapid changes, some decisions have to be accepted quickly. Those decisions, especially the inadequate ones, affect the whole software development life cycle and can have a significant impact on code quality. We have to be especially careful and pay attention to those program entities that contain irregularities or deficiencies.

The technical debt helps evaluate resulting problems and one of the most recognized techniques for identifying technical debt is the detection of code smells. There are different types of code smells, divided into different groups aimed at achieving a better understanding. A lot of tools can be found in the literature that help detect code smells, but each of them detect code smells with the help of different techniques.

The goal of our research was to look into available code smell detection tools and, using selected ones, compare the detected code smells while also analyzing the intersection of the results. Since code smell is a technique of identifying technical debt, the second part of the research was aimed at finding a connection between detected code smells and the value of calculated technical debt.

The article is organized as follows. First, the theoretical background about technical debt and code smell is presented. Section 4 presents the case study we carried out while section 5 and 6 contain a discussion of obtained results. The article is concluded with section 7.

2. TECHNICAL DEBT
The technical debt was first named in 1992 by Ward Cunningham as "not-quite-right code" [1]. Through the years, technical debt has won more recognition and its basic meaning has been upgraded. Since it represents a metaphor that can be subjectively understood [2], [3], a single generally accepted definition of technical debt still cannot be found.

Nevertheless, authors describe technical debt as a set of decisions taken within a project. Those decisions usually bring short-term success, but in the future they can cause problems, which are resolved with more effort than would be needed in the beginning [4]–[9].

Through the time, different types of technical debt were introduced. For example design debt, architecture debt, documentation debt, test debt, code debt and environment debt [5]–[7], [10], [11]. Soon, based on research and needs, some other types of technical debt appeared in the literature. For example, build debt, defect debt, requirement debt, test automation debt, service debt, versioning debt, people debt, process debt and usability debt [5]–[7].

2.1 Technical debt identification
One of the technical debt identification methods is source code analysis, where techniques can be divided between a static and dynamic analysis [8], [12]. Each one focuses on a specific field in a source code and does not provide for the detection of a variety in code smell. In the literature, the most represented ones are the following techniques:

- Modularity violations [12], [13].
- Design patterns and grime buildup [12], [13].
- Code smells [8], [12]–[14].
- Automatic static analysis [8], [12], [13].
- Source code comments [15].

3. CODE SMELL
One of the techniques for identifying technical debt is code smell detection. It refers to indicators within source code that point to

deeper problems in a software product [16]. Code smell means writing code in a way that violates the principles of best programing practices. The removal of code smells is usually done with source code refactoring, where the need for refactoring increases the likelihood of the existence of code smells in software [17].

Tufano et al. [18] presents an analysis of occurrence of code smells in software products. Usually code smell emerges when adding new functionalities and changing existing ones. This interesting finding was also that by refactoring existing source code new types of code smells can enter.

3.1 Code smell groups and types

Many different types of code smells are defined in the literature. For a better understanding of different types of code smells, several groups were defined, each containing different code smell types [19]: (1) *Bloaters* represent something in the code that is so big it cannot be effectively handled; (2) *Object-Orientation Abusers* contain examples where the possibilities of OO design is not fully exploited; (3) *Change Preventers* contain code smells that refers to code structures that considerably hinder software modification; (4) *Dispensables* present parts that are unnecessary and should be removed from a source code; (5) *Encapsulators* joins code smells connected to data communication mechanisms or encapsulations; (6) *Couplers* express the code, which is tightly coupled; (7) *Other* code smells.

3.2 Code smell detection

Code smell detection can be done with the help of software metrics. Different authors connected selected software metrics with detection of specific types of code smells [20], [21]. Based on the correlation between software metrics and code smell types, tools that identify some of these types have been developed [16], [22]–[24]. When identifying code smells with a software metric it is important to use reliable software metric threshold values. If thresholds are not set properly, a variety of false positive values can be detected [25]. Based on this, some other code smell detection techniques have been developed, for example, a technique based on the combination of machine learning algorithms, which achieved more than 96% accuracy when detecting different code smells [26].

4. CASE STUDY

As part of the case study, two research questions were formed:

- Are different open source tools for detecting code smells providing different results?
- What is the connection between detected code smells in selected tools and the value of technical debt calculated in the SonarQube tool?

To answer these questions, different software projects were analyzed to gather empirical data. We analyzed the projects gathered in "Qualitas Corpus." But since we needed the projects compiled to byte code to make an analysis, we chose a compiled version of the Qualitas Corpus [27]. So we can provide analysis without major changes in source code. In the end, based on criteria, 32 software projects were analyzed. Also, the criteria was set to select appropriate code smell types and tools for code smell detection to be used within a case study. The criteria were inspired by data gathered in a preliminary literature review.

4.1 Selected tools and code smells

Based on criteria, two tools for code smell detection were selected and corresponding code smell types. The information is presented in Table 1. Both selected tools are Eclipse extensions. To be able to answer the second question, we have to prepare the SonarQube tool, which detects code smells based on predefined rules. Among 255 rules that indicate the existence of code smells, 12 that follow Fowler definition [28], were selected. Because SonarQube does not enable code smell classification in different groups, this step was done by hand. The rules that follow before mentioned definitions was combined in a profile and classified into groups.

Table 1. Selected tools and code smells

Tool	Code smell	Tool version
JSpIRIT	God class, Feature envy, Brain method, Brain class	4.3.2
JDeodorant	God class, Feature envy, Long Method	5.0.64
SonarQube	God class, Feature envy, Brain method, Long method, Brain class	5.6.4 LTS

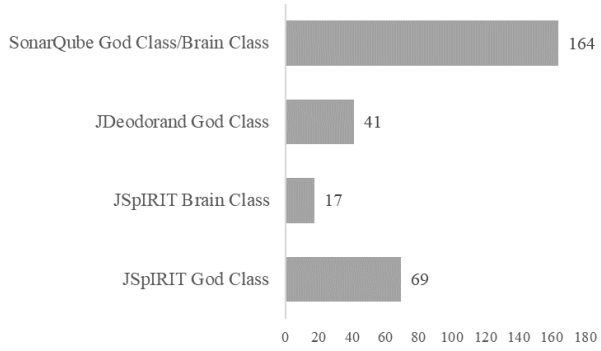
4.2 Analysis of empirical data

In the first step, selected projects were analyzed using the selected tools. The aim was to detect and count different code smell occurrence within different projects. Also, the distribution of activated rules in SonarQube was done. With this, we gain an insight into the appropriateness of the mapping rules for forming different groups in SonarQube and different types of code smells in JSpIRIT and JDeodorant. The secondary appropriateness of mapping rules was done with the intersection of detected code smells among the tools.

All gathered data present a starting point for finding correlations between code smell and technical debt and comparison of different tools for code smell detection.

5. COMPARISON OF DETECTED CODE SMELLS

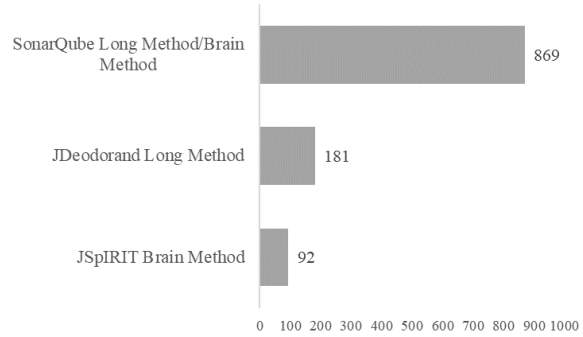
Used tools and considered code smells within the case study are presented in Table 1.



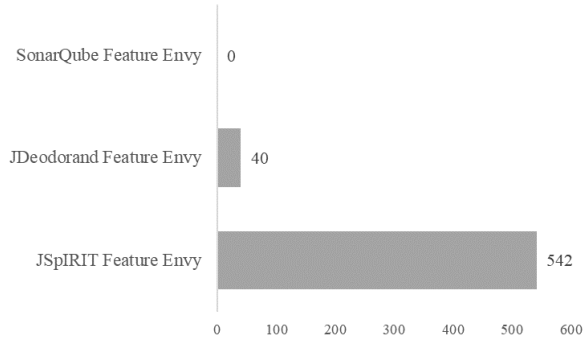
Graph 1 – Identification of God class and Brain class code smells within tools

In this analysis, we combine results for God class and Brain class (Graph 1), Brain method and Long Method (Graph 2), while with Feature envy (Graph 3) code smell was analyzed independently. As can be seen in Graph 1 and Graph 2, the SonarQube detected more God class/Brain class and Long method/Brain method code

smells than the tools JSpIRIT and JDeodorand did together. However, SonarQube has a problem with detecting Feature envy code smell, since it was not able to detect it within software project analysis (Graph 3). For this purpose we look again at the rules in SonarQube, but it cannot be stated that one of those rules can reliablly detect Feature envy code smell. The rule that we select is, in our opinion, the one that has the highest probability of detecting the mentioned code smell. On the other hand, when detecting Feature envy (Graph 3), JSpIRIT prevails, which detected 500 more occurrence than JDeodorand did. The latter has the tendency of detecting God class and Brain method. Overall, the most code smells are detected in SonarQube.



Graph 2 - Identification of Long method and Brain method code smells within tools



Graph 3 - Identification of Feature envy code smells within tools

Based on the results, we identified the intersection of detected code smells among the tools, presented in Figure 1. This was done by analyzing a project that was not selected for analysis, but is still a part of Qualitas Corpus. When identifying God class/Brain class code smell, the intersection between all the tools is 2.7% and when detecting Long method/Brain method it is 6.08%. To identify the causes for low intersection, we again look into the SonarQube tools. We looked into rules that are activated when detecting code smell common to other two tools.

6. THE CONNECTION BETWEEN CODE SMELLS AND TECHNICAL DEBT

For establishing a correlation between code smell and technical debt, the data about time contributed by each of the 12 used rules in SonarQube was acquired. The technical debt, when all 255 rules were used, was 5,460 days. When we activate just 12 selected rules, the technical debt was 1,982 days, which is 27% of all technical debt.

Since these 12 rules are used for code smell detection, it can be seen how much they contribute to the overall technical debt of a project. But the problem lies in the low intersection between tools when detecting code smells. An even more detailed analysis about activated rules within selected code smells does not bring any clearer results.

7. CONCLUSION

The case study was made to compare detected code smells among three different tools: JSpIRIT, JDeodorand and SonarQube. Since the first two define code smells which are not part of SonarQube, the rules within the SonarQube were mapped to a variety of groups, representing selected code smells.

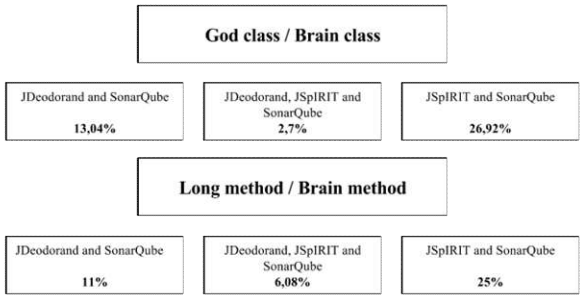


Figure 1 – Intersection between tools for code smell detection

The detected code smells by tools were compared within three identified categories, and the intersection of detected code smells was presented. The intersection between the used tool is very small (Figure 1). This can be attributed to the use of different detection techniques in different tools. JDeodorand proved to be the best at detecting Long method code smell, JSpIRIT at detecting Feature envy code smell and SonarQube at detecting God class/Brain class and Long method/Brain method code smells.

The second part of the case study was aimed at finding a connection between detected code smell and technical debt in SonarQube. The intersection of detected code smells between different tools and SonarQube is very small. We can also add the fact that the classified rules are not activated proportionally. Since rules do not contribute equality to a technical debt calculation, the impact of code smell detection to technical debt cannot be defined.

There are many research opportunities in this area. The rules for code smell detection among tools could be compared in details for the purpose of unification. In addition, the future work can be oriented in an attempt to provide the generally accepted definition of technical debt. At last, selected tools, JSpIRIT and JDeodorand could be upgraded for technical debt calculation.

8. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency under The Young Researchers Programme (SICRIS/SRA code 35512, RO 0796, Programme P2-0057).

9. REFERENCES

[1] W. Cunningham, “The WyCash Portfolio Management System,” *SIGPLAN OOPS Mess.*, vol. 4, no. 2, pp. 29–30, Dec. 1992.

[2] P. Kruchten, R. L. Nord, and I. Ozkaya, “Technical Debt: From Metaphor to Theory and Practice,” *IEEE Software*, vol. 29, no. 6. pp. 18–21, 2012.

[3] N. A. Ernst, S. Bellomo, I. Ozkaya, R. L. Nord, and I. Gorton, “Measure It? Manage It? Ignore It? Software Practitioners and Technical Debt,” in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 2015, pp. 50–60.

[4] J. Yli-Huumo, A. Maglyas, and K. Smolander, “How do software development teams manage technical debt? – An empirical study,” *J. Syst. Softw.*, vol. 120, no. Supplement C, pp. 195–218, 2016.

[5] Z. Li, P. Avgeriou, and P. Liang, “A systematic mapping study on technical debt and its management,” *J. Syst. Softw.*, vol. 101, no. Supplement C, pp. 193–220, 2015.

[6] N. S. R. Alves, T. S. Mendes, M. G. de Mendonça, R. O. Spínola, F. Shull, and C. Seaman, “Identification and management of technical debt: A systematic mapping study,” *Inf. Softw. Technol.*, vol. 70, pp. 100–121, Feb. 2016.

[7] N. S. R. Alves, L. F. Ribeiro, V. Caires, T. S. Mendes, and R. O. Spínola, “Towards an Ontology of Terms on Technical Debt,” *2014 Sixth International Workshop on Managing Technical Debt*. pp. 1–7, 2014.

[8] N. Zazworka, R. O. Sp’\inola, A. Vetro’, F. Shull, and C. Seaman, “A Case Study on Effectively Identifying Technical Debt,” in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 2013, pp. 42–47.

[9] M. Fowler, “TechnicalDebt,” 2003. [Online]. Available: https://martinfowler.com/bliki/TechnicalDebt.html. [Accessed: 14-Sep-2017].

[10] E. Tom, A. Aurum, and R. Vidgen, “An exploration of technical debt,” *J. Syst. Softw.*, vol. 86, no. 6, pp. 1498–1516, 2013.

[11] C. Fernández-Sánchez, J. Garbajosa, C. Vidal, and A. Yagüe, “An Analysis of Techniques and Methods for Technical Debt Management: A Reflection from the Architecture Perspective,” *2015 IEEE/ACM 2nd International Workshop on Software Architecture and Metrics*. pp. 22–28, 2015.

[12] N. Zazworka, A. Vetro’, C. Izurieta, S. Wong, Y. Cai, C. Seaman, and F. Shull, “Comparing Four Approaches for Technical Debt Identification,” *Softw. Qual. J.*, vol. 22, no. 3, pp. 403–426, Sep. 2014.

[13] C. Izurieta, A. Vetrò, N. Zazworka, Y. Cai, C. Seaman, and F. Shull, “Organizing the Technical Debt Landscape,” in *Proceedings of the Third International Workshop on Managing Technical Debt*, 2012, pp. 23–26.

[14] N. Zazworka, M. A. Shaw, F. Shull, and C. Seaman, “Investigating the Impact of Design Debt on Software Quality,” in *Proceedings of the 2Nd Workshop on Managing Technical Debt*, 2011, pp. 17–23.

[15] E. d. S. Maldonado and E. Shihab, “Detecting and quantifying different types of self-admitted technical Debt,” *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*. pp. 9–15, 2015.

[16] E. Fernandes, J. Oliveira, G. Vale, T. Paiva, and E. Figueiredo, “A Review-based Comparative Study of Bad Smell Detection Tools,” in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, 2016, pp. 18:1–18:12.

[17] F. A. Fontana, M. Mangiacavalli, D. Pochiero, and M. Zanoni, “On Experimenting Refactoring Tools to Remove Code Smells,” in *Scientific Workshop Proceedings of the XP2015*, 2015, pp. 7:1–7:8.

[18] M. Tufano, F. Palomba, G. Bavota, R. Oliveto, M. Di Penta, A. De Lucia, and D. Poshyvanyk, “When and Why Your Code Starts to Smell Bad,” in *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, 2015, pp. 403–414.

[19] M. Mantyla, J. Vanhanen, and C. Lassenius, “A taxonomy and an initial empirical study of bad smells in code,” *International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings.* pp. 381–384, 2003.

[20] F. A. Fontana, V. Ferme, A. Marino, B. Walter, and P. Martenka, “Investigating the Impact of Code Smells on System’s Quality: An Empirical Study on Systems of Different Application Domains,” *2013 IEEE International Conference on Software Maintenance*. pp. 260–269, 2013.

[21] F. A. Fontana and S. Spinelli, “Impact of Refactoring on Quality Code Evaluation,” in *Proceedings of the 4th Workshop on Refactoring Tools*, 2011, pp. 37–40.

[22] A. Hamid, M. Ilyas, M. Hummayun, and A. Nawaz, “A Comparative Study on Code Smell Detection Tools,” *Int. J. Adv. Sci. Technol.*, vol. 60, pp. 25–32, 2013.

[23] A. Chatzigeorgiou and A. Manakos, “Investigating the Evolution of Bad Smells in Object-Oriented Code,” *2010 Seventh International Conference on the Quality of Information and Communications Technology*. pp. 106–115, 2010.

[24] F. A. Fontana, P. Braione, and M. Zanoni, “Automatic detection of bad smells in code: An experimental assessment.,” *J. Object Technol.*, vol. 11, no. 2, p. 5: 1–38, 2012.

[25] F. A. Fontana, V. Ferme, M. Zanoni, and A. Yamashita, “Automatic Metric Thresholds Derivation for Code Smell Detection,” *2015 IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics*. pp. 44–53, 2015.

[26] F. Arcelli Fontana, M. V Mäntylä, M. Zanoni, and A. Marino, “Comparing and experimenting machine learning techniques for code smell detection,” *Empir. Softw. Eng.*, vol. 21, no. 3, pp. 1143–1191, 2016.

[27] R. Terra, L. F. Miranda, M. T. Valente, and R. S. Bigonha, “Qualitas.class Corpus: A compiled version of the Qualitas Corpus,” *Softw. Eng. Notes*, vol. 38, pp. 1–4, 2013.

[28] M. Fowler and K. Beck, *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.

A Qualitative and Quantitative Comparison of PHP and Node.js for Web Development

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

ABSTRACT

In this paper, the possibilities and uses of PHP and Node.js for web development were studied and presented. Based on the results of a literature overview, a model for analysis and comparison of technologies was defined, where experiences gained from the development of an equivalent web solution using both evaluated server-side technologies were also taken into consideration. We found out that technologies, apart from maturity and popularity, differ the most in available support and aspects related to performance.

Categories and Subject Descriptors

D.3.2 [**Programming Languages**]: Language Classifications – *JavaScript.*

General Terms

Measurement, Documentation, Performance, Economics, Security, Languages

Keywords

Web Development, Web Technologies, Server-side Programming, PHP, JavaScript, Node.js

1. INTRODUCTION

With the ongoing and rapid development of web technologies, web developers have to make a difficult decision. With so many available web technologies, choosing one or a few of them for developing web sites and web applications is often a very challenging task. Therefore, we compared PHP with Node.js to evaluate the main similarities and differences between these technologies.

The dynamic scripting language PHP is widely used in the field of web development and has become one of the most used and dominant web technologies for server-side programming over the last few years [25, 35]. On the contrary, the dynamic scripting language JavaScript has become enormously popular for front-end development. However, with the birth of Node.js, JavaScript offered a new and promising server-side technology based on JavaScript [32].

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 describes the proposed model for analysis and comparison of technologies PHP and Node.js. Section 4 presents the result of our analysis and a comparison. Section 5 summarizes the main results of our work.

2. RELATED WORK

There are a lot of studies evaluating and analyzing different Web technologies. In the literature overview, we came across only two works that compared PHP and Node.js. Both evaluated these technologies based on performance.

Lei et al. [12] compared the performance of Node.js, Python-Web and PHP with objective systematic tests and realistic user behavior tests. The results yielded some valuable performance data, showing that Node.js performs much better than the traditional PHP in a high concurrency situation, especially in input/output intensive situations. PHP handles small requests well, but struggles with large requests. Both Node.js and PHP are not suitable for computation-intensive situations [12].

Chaniotis et al. [10] conducted a series of stress tests on popular server-side technologies. Once again PHP proved to be less effective in input/output operations and resource utilization with an increasing number of concurrent users than Node.js.

The rest of the studies in the field of PHP and Node.js comparison were not scientifically based. Rather, they are founded on expert judgement. Some of them are stated below.

C. Buckler [28] evaluated PHP and Node.js based on ten rounds. Each of them considered a general development challenge which could be applied to any web technology. According to his study, PHP had better help and support, better integration, and easier deployment, and it was simpler to build a basic web page, whereas Node.js had an easier and more logical language syntax, better development tools, more supported environments, better performance, and higher programmer passion. The author could not decide which one had a brighter future ahead of it.

P. Wayner [9] also compared PHP with Node.js and found out that PHP was better when mixing code with content, had a better deep code base, was simpler, had better support, did not need a client app, had better integration for SQL, enabled a higher speed of coding and was more popular among developers. Node.js was better with separating concerns, had a modern syntax and new modern features, had dozens of language options, thinner service calls, better integration for JSON, was faster and was better based on solidarity.

3. PROPOSED MODEL FOR ANALYSIS AND COMPARISON OF TECHNOLOGIES

When analyzing and comparing web technologies, many aspects should be considered. To the best of our knowledge, there are only two scientific papers evaluating PHP and Node.js, both only by

performance. Thus, we examined the literature that compared web technologies in general [1, 5, 11, 13, 14, 16, 17, 18, 26, 27] and based on the results we defined a model for analysis and comparison of technologies with the following criteria: development, supported platforms, costs, community and support, performance, and security. For every criterion we defined various variables, as seen in Table 1.

Table 1. Proposed model for analysis and comparison of technologies.

Criterion	Variable
Development	Syntax
	Libraries
	Learning curve
	IDE
Supported Platforms	Operating system
	Web servers
	Databases
Costs	Technology
	Operating system
	Webservers
	Databases
Community and Support	Quality of Documentation
	Release frequency
	Maturity
	Popularity among developers
	Usage for web sites
Performance	Simple web site
	Input/output intensive web site
	Computation-intensive web site
Security	SQL injection
	Cross-site scripting
	Cross-site request forgery
	Denial of service

Results were retrieved based on a documentation analysis, official web sites of PHP and Node.js, literature, performance tests and experiences, gained from developing equivalent web applications with both technologies. More details have been given in our previous work [34].

4. ANALYSIS AND COMPARISON OF PHP AND NODE.JS

4.1 Development

PHP is known for having a large variety of functions, however the problem of inconsistent naming between versions exists. It does not promote good structure of code; however, it enables it. Node.js has brought JavaScript to the back-end and therefore can be used both client and server side. Its syntax is quite modern and therefore Node.js offers a modern development approach. However, it only has a few core statements and functions, everything else must be included with extra modules, which are very simple to install, include, and use with the help of the Node Package Manager. PHP also has a lot of libraries available. However, searching for them, installing them, and using them is a bit more difficult. PHP is quite simple and provides a quick and easy introduction for new PHP developers, whereas Node.js is more complex and harder to learn. Development with both technologies can be done in various IDEs.

4.2 Supported Platforms

PHP and Node.js are quite similar based on the (in)dependence of platforms. They are both compatible with the most popular operating systems, web servers and databases. Both technologies can therefore be used with Microsoft Windows, macOS and Linux [22, 25] operating systems, which together have a 97.7 % market share [31]. Both are compatible with Apache, nginx and IIS web servers [15, 25], which together have a 94.7 % market share [36]. PHP and Node.js can also be used with various databases both relational (Oracle, MySQL, Microsoft SQL Server, PostgreSQL, DB2, SQLite) and non-relational (MongoDB, Cassandra, Redis) [8]. The main difference between the two technologies is that Node.js can configure its own web server [22].

4.3 Costs

Both technologies are free and open source [22, 25]. Because of the extensive support of the most popular platforms, both have free and payable versions available. Microsoft Windows has licensing costs [20], macOS is free for users of Mac computers [4], while Linux is free for everybody [33]. The Apache web server is free to use [3]. Nginx has both a free and payable option available [21]. IIS can be used without licensing costs on Windows operating systems [19]. Also, the HTTP module, which offers Node.js the ability to configure a web server, is free to use because it is included in Node.js by default [22]. Databases MySQL, MongoDB, PostgreSQL, Cassandra, SQLite and Redis are free, whereas Oracle, Microsoft SQL Server and DB2 are payable with limited free versions [8].

4.4 Community and Support

Based on help and support that is available for both technologies, we discovered that PHP [25] has better official documentation than Node.js [22], especially for beginners, whereas documentation for Node.js includes more complex examples. PHP was created in 1995 [25] while Node.js appeared in 2009 [2]. However, even though PHP has existed for nearly 22 years, it has 239 known releases [25] which is less releases than the relatively younger Node.js, which has only existed for 8 years and has 371 known releases [22]. Node.js has also had more frequent releases in the most recent period (from 1st January 2016 until 31st December in 2016). In this one-year period, Node.js had 68 releases [22], whereas PHP had 33 [25]. Therefore, although both technologies are actively developing, Node.js is developing faster. However, PHP remains far more popular among developers and has more experienced developers, according to results from an analysis of available help and support on two popular web forums: Stack Overflow and Code Project. Also, according to statistics of usage of technologies for web sites. There are 1,101,785 asked questions tagged with PHP [30] on Stack Overflow and 183,444 tagged with Node.js [29]. On Code Project there is 33,761 hits for PHP [7] and 1,825 for Node.js [6]. Statistics from the 1st of September show that PHP is used by 82.8% of all web sites whose server-side programming language is known, whereas Node.js is used by only 0.4 % [35].

4.5 Performance

For assessing performance, we designed, developed, and conducted three test scenarios with each of the evaluated technologies. With all tests, we kept requests at 5000 with 10, 50, 100, 150 and 200 concurrent users. We used requests per second as metrics data. Each test was repeated ten times. The lowest and the highest results from each test were excluded and an average from the rest of the data was calculated.

For our tests we used PHP 7.1.7, Apache 2.4.27 and MySQL 5.7.18 to test the performance of PHP. To test the performance of Node.js, we used Node.js v8.2.1 with a mysql module and MySQL 5.7.18. The client/server machine in our test ran a Windows 10, 64-bit

operating system with an Intel i5 processor and 4 GB of RAM. As a testing tool we used ApacheBench 2.3.

To test the performance of a basic web site developed in each technology we used a web site that only outputs some text. The results of the first basic performance comparison showed that the performance of Node.js is better than PHP with the same number of concurrent users, especially when the number of users increases (Figure 1.).

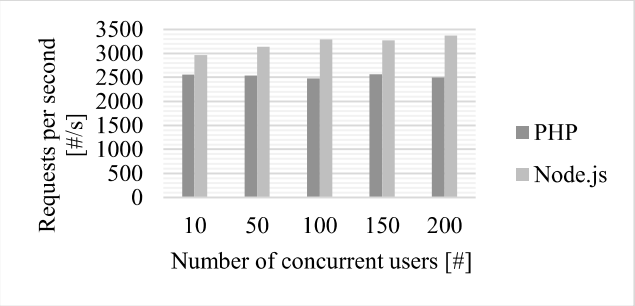


Figure 1. Performance of simple web site.

The test scenario for evaluating the performance of an input/output intensive web site was based on a select operation of the database. The retrieved results distinguished the difference between the two technologies. Node.js is far more adapted at input/output intensive operations, even when the number of concurrent users increases.

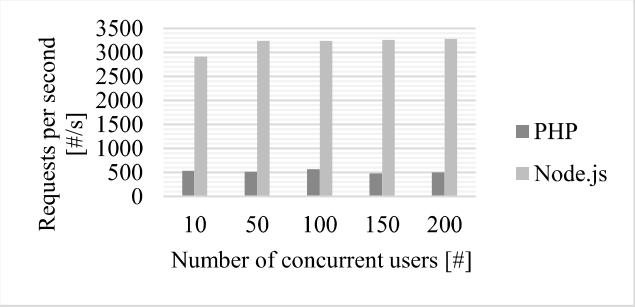


Figure 2. Performance of input/output intensive web site.

We used a calculation of the fifteenth value of Fibonacci for assessing performance of a computation-intensive web site. Neither PHP nor Node.js are suitable for computation-intensive situations. However, Node.js performs better among the two (Figure 3).

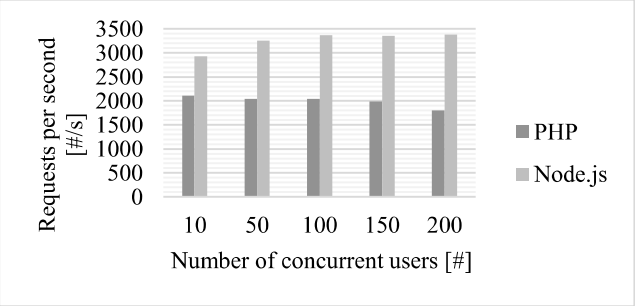


Figure 3. Performance of compute-intensive web site.

The results of our performance tests are in accordance with studies from other authors [9, 12] and were also expected. Node.js uses an event-driven, asynchronous model, which enables it to respond rapidly to an input/output operation. Therefore, it is suitable for data intensive, fast responding, real time web applications and applications with many users. However, it is not adapted to computation-intensive situations. On the other hand, PHP, although efficient on the server-side, is more suitable for middle and small-scale websites.

4.6 Security

Both PHP and Node.js offer solutions against popular security risks, such as SQL injection, cross-site scripting, cross-site request forgery and denial of service. PHP has a mostly built-in solution, e.g. prepared statement and validation functions for protection against SQL injections [25]. Some of the solutions are not included and must be added with libraries, e. g. when preventing cross-site scripting attacks [24]. Node.js does not include any safety solutions by default. All of them must be added with the help of modules, which can be installed by the Node package manager, e. g. the mysql module offers an escape function for preventing SQL injection [23].

5. CONCLUSION

We analyzed and compared two web technologies – PHP as the most used technology for server-side development, and Node.js as a relatively young technology with much potential. PHP is known for being quite easy to learn and easy to develop with and offers the possibility of simply mixing PHP and HTML code. Node.js, on the other hand, has more modern syntax and features, with the main feature being a non-blocking, evet-driven, asynchronous input/output model. One of its primary characteristics is also the movement of the usage of JavaScript from front-end to back-end and the possibility of developing from front-end to back-end in one language – JavaScript.

Based on the analysis and comparison of technologies, we found out that technologies are very similar based on the basic characteristics of supported platforms and costs. The only difference between them is that Node.js can create its own server. When evaluating security, both technologies offer solutions against the most common web attacks. PHP is older and simpler to learn when compared with the younger and more modern Node.js, with better support and help from both official documentation and the community of developers. The main difference between technologies is in performance, where PHP is more suitable for sites for managing content and displaying data and Node.js is more for complex solutions with many concurrent users and more difficult processing. Therefore, the technologies, apart from their maturity and popularity, differ the most in available support and aspects related to performance.

6. REFERENCES

[1] Alok Ranjan, Rajeev Kumar, and Joydip Dhar. 2010. A Comparative Study between Dynamic Web Scripting Languages. In *ICDEM 2010*. Springer, Tiruchirappalli, 288–295. DOI10.1007/978-3-642-27872-3_43.

[2] Amos Q. Haviv, Adrian Mejia, and Robert Onodi. 2016. *Web Application Development with MEAN*. Packt Publishing, Birmingham.

[3] Apache Software Foundation. 2017. Retrieved July 20, 2017 from https://www.apache.org/free/.

[4] Apple. 2017 Mac. Retrieved July 20, 2017 from https://support.apple.com/explore/new-to-mac.

[5] Bernard Kohan. 2010. PHP vs ASP.net Comparison. Retrieved July 7, 2017 from http://www.comentum.com/php-vs-asp.net-comparison.html.

[6] Code Project. 2017. Node.js. Retrieved July 23, 2017 from https://www.codeproject.com/search.aspx?q=node.js.

[7] Code Project. 2017. PHP. Retrieved July 23, 2017 from https://www.codeproject.com/search.aspx?q=php.

[8] DB-Engines. 2017. Ranking. Retrieved July 19, 2017 from https://db-engines.com/en/ranking.

[9] InfoWorld. 2017. PHP vs Node.js: An epic battle for developer mind share. Retrieved July 21, 2017 from https://www.infoworld.com/article/3166109/application-development/php-vs-nodejs-an-epic-battle-for-developer-mind-share.html.

[10] Ioannis K. Chaniotis, Kyriakos-Ioannis D. Kyriakou, and Nikolaos D. Tselikas. 2015. Is Node.js a viable option for building modern web applications? A performance evaluation study. *Computing*, 97, 10 (Oct. 2015), 1023–1044. DOI: https://doi.org/10.1007/s00607-014-0394-9.

[11] Manisha Jailia, Ashok Kumar, Manisha Agarwal, and Isha Sinha. 2016. Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework. In *ICTBIG 2016*. IEEE Computer Society, Indore, 1–5. DOI: https://doi.org/10.1109/ICTBIG.2016.7892651.

[12] Kai Lei, Yning Ma, and Zhi Tan. 2014. Performance Comparison and Evaluation of Web Development Technologies in PHP, Python and Node.js. In *CSE*. IEEE Computer Society, Chengdu, 661–668. DOI: http://dx.doi.org/10.1109/CSE.2014.142.

[13] Kais Samkari and Ammar Joukhadar, 2008. A Comparison Matrix for Web HCI. In *ICTTA 2008*. IEEE Computer Society, Damascus, 1–5. DOI: http://dx.doi.org/10.1109/ICTTA.2008.4530335.

[14] Khampheth Bounnady, Khampaseuth Phanthavong, Somsanouk Pathoumvanh, and Keokanlaya Sihalath. 2016. Comparison the processing speed between PHP and ASP.NET. In *ECTI-CON.* IEEE Computer Society, Chiang Mai, 1–5. DOI: https://doi.org/10.1109/ECTICon.2016.7561484.

[15] Kyle Schutt and Osman Balci. 2016. Cloud software development platforms: A comparative overview. In *SERA.* IEEE Computer Society, Balitmore, 3–13. DOI: http://dx.doi.org/10.1109/SERA.2016.7516122.

[16] Lance Titchkosky, Martin Arlitt, and Carey Williamson. 2003. A performance comparison of dynamic web technologies. *ACM SIGMETRICS Performance Evaluation Review*, 31 (2003), 2–11. Retrieved July 15, 2017 from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.5875.

[17] Manya Sharma. 2015. Web Development Technology-PHP. How It Is Related To Web Development Technology ASP.NET. *International Journal of Scientific & Technology Research*, 4, 1 (2015), 23–24. Retrieved July 14, 2017 from https://doaj.org/article/28a3ca4f9a66496aa6dc8d49951c054f.

[18] Matt Raible. 2010. How I Calculated Ratings for My JVM Web Frameworks Comparison. Retrieved July 15, 2017 from https://raibledesigns.com/rd/entry/how_i_calculated_ratings_for.

[19] Microsoft. 2017. Web Server. Retrieved July 20, 2017 from https://www.microsoft.com/web/platform/server.aspx.

[20] Microsoft. 2017. Windows. Retrieved July 20, 2017 from https://www.microsoft.com/sl-si/windows/get-windows-10.

[21] Nginx. 2017. Retrieved July 20, 2017 from https://www.nginx.com/products/pricing/.

[22] Node.js. 2017. Retrieved June 20, 2017 from https://nodejs.org.

[23] NPM. 2017. Module mysql. Retrieved July 21, 2017 from https://www.npmjs.com/package/mysql.

[24] OWASP Foundation. 2017. PHP Security Cheat Sheet. Retrieved July 21, 2017 from https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet.

[25] PHP. 2017. Retrieved June 20, 2017 from http://php.net/.

[26] S. P. Ahuja and R. Clark. 2005. Comparison of Web Services Technologies from a Developer's Perspective. In *ITCC*. IEEE Computer Society, Los Alamitos, 2, 791–791. http://dx.doi.org/10.1109/ITCC.2005.106.

[27] Scott Trent, Michiaki Tatsubori, Toyotaro Suzumura, Akihiko Tozawa, and Tamiya Onodera. 2008. Performance Comparison of PHP and JSP as Server-Side Scripting Languages. In *ACM/IFIP/USENIX*. Springer, Berlin, 164–182. DOI: https://doi.org/10.1007/978-3-540-89856-6_9.

[28] Site Point. 2017. Smackdown: PHP vs Node.js. Retrieved July 21, 2017 from https://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js/https://www.sitepoint.com/sitepoint-smackdown-php-vs-node-js/.

[29] Stack Overflow. 2017. Tagged Questions: Node.js. Retrieved July 23, 2017 from https://stackoverflow.com/questions/tagged/node.js.

[30] Stack Overflow. 2017. Tagged Questions: PHP. Retrieved July 23, 2017 from https://stackoverflow.com/questions/tagged/php.

[31] StatCounter GlobalStats. 2017. Desktop Operating System Market Share Worldwide. Retrieved July 19, 2017 from http://gs.statcounter.com/os-market-share/desktop/worldwide/#daily-20170601-20170601-bar.

[32] Stefan Tilkov and Steve Vinoski. 2010. Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing* 14, 6 (Nov. –Dec. 2010), 80–83, 6. DOI: http://dx.doi.org/10.1109/MIC.2010.145.

[33] The Linux Foundation. 2017. What is Linux? Retrieved July 20, 2017 from https://www.linux.com/what-is-linux.

[34] Tjaša Heričko. 2017. Analysis and Comparison of PHP and Node.js for Web Development. Diploma's thesis. Retrieved September 1, 2017 from https://dk.um.si/IzpisGradiva.php?id=67222&lang=slv.

[35] W3 Techs. 2017. Historical yearly trends in the usage of server-side programming languages for websites. Retrieved September 1, 2017 from https://w3techs.com/technologies/history_overview/programming_language/ms/y.

[36] W3 Techs. 2017. Usage of web servers for websites. Retrieved June 20, 2017 from https://w3techs.com/technologies/overview/web_server/all.

Skills, Competences and Platforms for a Data Scientist

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

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

ABSTRACT
In this paper, we identify the core competences and skills of a Data Scientist, where we build on already existing research about the already practicing Data Scientists and on existing frameworks. We complement this research with the practitioners' survey about popular Data Science platforms and our own research on the search term trends and job posting trends.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous

I.2.m [**Artificial Intelligence**]: Miscellaneous

General Terms
Data Science, Framework, Platforms.

Keywords
Data science, data scientist, skills, competences, platforms.

1. INTRODUCTION
In the last few years, Data Science has become one of the most rapidly growing interdisciplinary fields, where it combines different aspects of computer engineering, mathematics, and other managerial skills. The employer and employee review website Glassdoor even rates Data Scientist as the number one best job in America in 2017 (regarding the job satisfaction, number of a job openings and median base salary) [1].

The precise skill set of a Data Scientist is not so well defined yet, as it gets mixed up with other job roles, such as Data Analyst, Machine Learning Engineer, Statistician, Data Engineer, Business Analyst, Data Architect and others. The differences between these titles are not always clear and are used interchangeably, especially among people outside the domain of Data Science. The purpose of this paper is not to make a clear differential line between these different job roles, but to define what core skills of a Data Scientist are. This could help any employers to identify if the Data Scientist is the one they need in their organization. Also, the clear list of definitions, skill sets and most common platforms used by Data Scientists could be used by people striving to become a Data Scientist and work on each skill of the broad spectrum of competences and skills needed and expected by a Data Scientist.

2. DATA SCIENCE COMPETENCES AND SKILLS FRAMEWORKS
With the growing demand for staff with knowledge and skills of Data Science, several more or less commonly accepted frameworks that have been used for defining Data Science (alongside with general computer science, ICT and similar) competences, skills and subject domain classifications, have emerged. These frameworks can be, with some alignment, built upon and re-used for better acceptance from research and industrial communities. One of the most elaborate is EDISON Data Science Framework (ESDF), developed within the scope of European project "Edison – building the data science profession" [2]. The ESDF provides a collection of documents that define the Data Science profession, which have been developed to guide educators and trainers, employers and managers, and Data Scientists themselves, collectively breaking down the complexity of the skills and competences need to define Data Science as a professional practice.

The ESDF itself, however, builds on existing standard and commonly accepted frameworks, as is the Big Data Interoperability Framework, published by the NIST Big Data Working in September 2015 [3]. It provides various definitions, among them for Data Science, Data Scientist and Data Life Cycle, which can be used as a starting point for further analysis.

"**Data Science** is the extraction of actionable knowledge directly from data through a process of discovery, or hypothesis formulation and hypothesis testing. Data Science can be understood as the activities happening in the processing layer of the system architecture, against data stored in the data layer, in order to extract knowledge from the raw data.

Data Science across the entire data life cycle incorporates principles, techniques, and methods from many disciplines and domains including data cleansing, data management, analytics, visualization, engineering, and in the context of Big Data, now also includes Big Data Engineering. Data Science applications implement data transformation processes from the data life cycle in the context of Big Data Engineering." [3]

"A **Data Scientist** is a practitioner who has sufficient knowledge in the overlapping regimes of business needs, domain knowledge, analytical skills, and software and systems engineering to manage the end-to-end data processes in the data life cycle.

Data Scientists and Data Science teams solve complex data problems by employing deep expertise in one or more of these disciplines, in the context of business strategy, and under the guidance of domain knowledge. Personal skills in communication, presentation, and inquisitiveness are also very important given the complexity of interactions within Big Data systems." [3]

The main focus of a data scientist is thus to discover meaningful patterns in data and synthesize useful knowledge by performing all the necessary steps throughout the whole data life cycle – the

collection of raw data, (pre-)processing of data and transforming it into useful information, performing data analysis via various data analytics algorithms and tools, interpreting and evaluating the discovered patterns in order to produce useful knowledge, and validating the induced knowledge models to produce value. Analytics is used to refer to the methods, their implementations in tools, and the results of the use of the tools as interpreted by the practitioner [4]. The analytics process is the synthesis of knowledge from information.

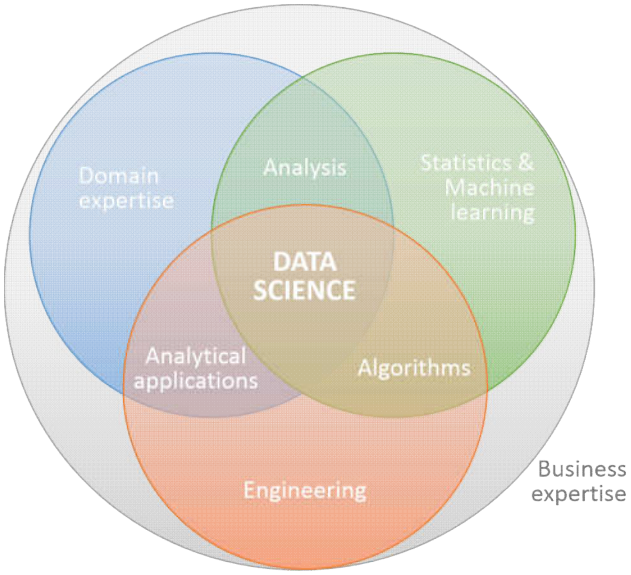


Figure 1. Data Science definition by NIST BD-WG [3].

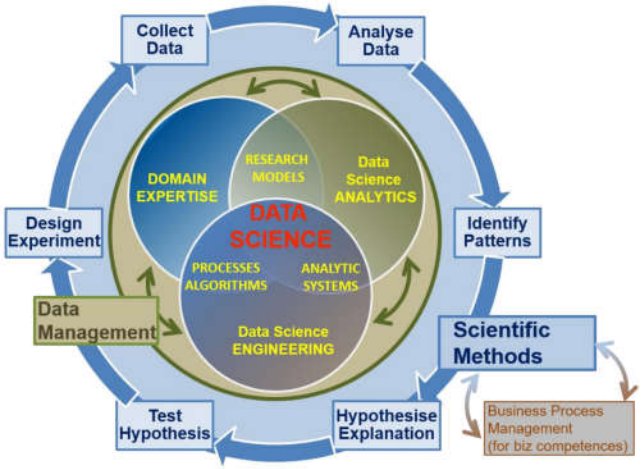
In order to cover the competence, required from a Data Scientist, a good knowledge of data analytics is needed (the two most important fields of analytics are statistics and machine learning), a good understanding of engineering (programming, software engineering and data management in order to provide analytical applications), as well as a fair amount of domain expertise. Figure 1 provides graphical presentation of the multi-factor/multi-domain Data Science definition.

2.1 General/research vs. business profile

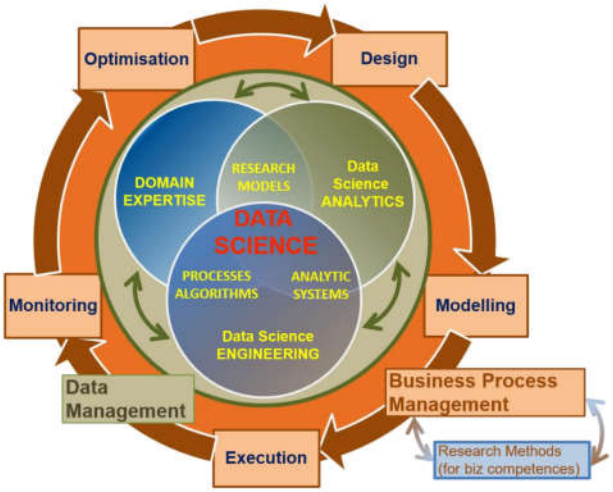
As the Data Science covers a lot of topics, many different competences and skills are required from a data scientist. In this manner, data scientists tend to focus on some specialization within the whole data science scope. In general, two major profiles can be identified – a general, research oriented profile, and a business oriented profile (see Figure 2). For both profiles a fair amount of analytics and engineering knowledge as well as the domain expertise is required. Besides that, the research oriented profile concentrates primarily on the use of scientific methods – formulation of test hypothesis, experiment design, data collection and analysis, pattern discovery and explanation of discovered knowledge. On the other hand, the business oriented profile focuses on business process management – monitoring the important data and designing, modelling, optimizing, and executing the data-driven business processes.

3. DATA SCIENTISTS' SKILL SETS

The existing standard and commonly accepted frameworks for defining Data Science competences are very good aligned with several reports and scientific papers which provide research results of what skills a data scientist should have.



(a) Data Science competence groups for general or research oriented profiles



(b) Data Science competence groups for business oriented profiles

Figure 2. Relations between the Data Science competence groups for (a) general or research oriented and (b) business oriented professions/profiles [4].

In [5] the authors present the findings compiled from 50 different reports of research in articles, journals, and books, and conducted via experts' views using the Delphi technique, regarding data scientist skills required by the industry. They provided a list of 41 data scientists' skills and categorize them into five major categories adapted from [6] – computer science, analytics, data management, decision management, and entrepreneurship:

- **Computer Science** includes programming, where R and Python are predominant programming languages, as well as privacy, security and systems architecture.
- **Analytics** focuses primarily on statistics and machine learning, and includes natural language processing, probability, simulation.
- **Data management** covers all data handling skills and puts emphasis on databases, data modelling and visualization, data mining, business intelligence and general data processing.

- **Decision management** focuses on decision making, while encompassing communication and ethics.
- Finally, **Entrepreneurship** includes business and economics.

On the other hand, the EDSF also categorizes all the skills required for a data scientist into five major categories, namely analytics, engineering, data management, research methods and project management, and business analytics [4]:

- **Analytics** focuses on the use of machine learning, data mining and text mining techniques, the application of predictive and prescriptive analytics, the use of statistics, operations research, optimization and simulations, and the assessment, evaluation and validation of results.
- **Engineering** includes the use of ICT systems and software engineering, cloud computing and big data technologies, databases, data security, privacy and intellectual property rights protection, as well as algorithms design.
- **Data management** put emphasis on specifying, developing and implementing enterprise data management and data governance strategy and infrastructure and includes data storage systems, data modeling and design, data lifecycle support, data quality, integration, and digital libraries and open data.
- **Research methods and project management** encompasses the use of research methods principles in developing data driven applications and implementing the whole cycle of data handling, development and implementation of data collection processes, and consistent application of project management workflow.
- **Business analytics** focuses on the use business intelligence, business process management, econometrics for data analysis and applications, user experience design, data warehouses for data integration, and data driven marketing.

4. PRACTITIONER PLATFORM SURVEY AND TRENDS

After defining what the required skills for a Data Scientist are, in this section we look at what the current state of the skills is among active practitioners of Data Science. So far, no thorough analysis of all skills of Data Scientists was done, but there is a good survey about the frameworks they use in their line of work.

In August of 2017 KDnuggets, one of the most popular websites about data science based on independent ranking [7], had a poll for their readers [8]. The poll asked the following question: "Did you use R, Python (along with their packages), both, or other tools for Analytics, Data Science, Machine Learning work in 2016 and 2017?". The poll was completed by 954 people and it showed the following results.

The results of the poll clearly indicate that there is a shift from R programming language to Python programming language in respect of Data Science, Analytics and Machine Learning (see Figure 3). The usage of R programming language fell by 6 percentage points, while usage of Python rose from 34% to 41% (the increase of 7 percentage points) of readers that finished the poll. The poll indicates that use of both, R and Python for Data Science, Analytics and Machine learning also rose from 8.5% to 12% (the increase of 3.5 percentage points), which can be contributed to practitioners slowly switching from R to Python but still using R for some specific part of work.

Next, the poll results also show the transitions from one to another platform for Analytics, Data Science, and Machine learning (see Figure 4). The chart in Figure 4 clearly shows the following. Python users are more loyal than R users, as 91% of readers stuck with Python from 2016 to 2017, and only 74% of readers stuck with R from 2016 to 2017. Also, only 60% of readers that use other platform and languages stuck to those from 2016 to 2017.

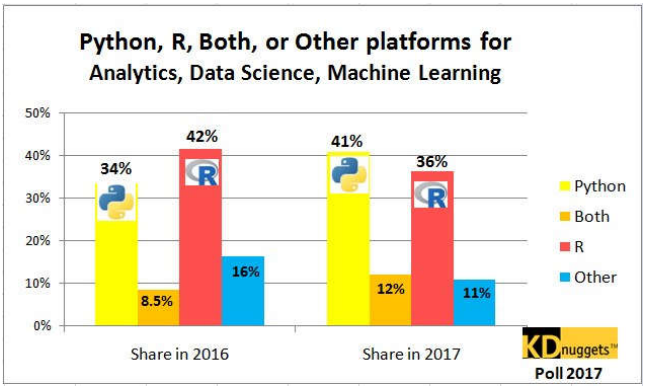


Figure 3. Share of R, Python, both R and Python, or other platforms usage for Analytics, Data Science or Machine Learning for 2016 and 2017 [8].

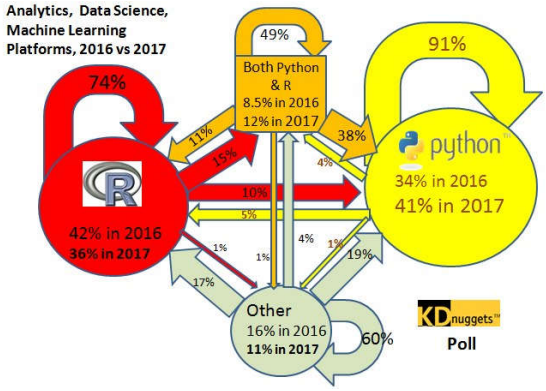


Figure 4. The transition between different programming languages for Data Science, Analytics and Machine Learning from 2016 to 2017 [8].

As the chart shows, only 5% of Python users switched to R exclusively, while 10% of R users switched to using Python exclusively. There is a clear flow of R users (15%) that switched to using both, R and Python, but users of both platforms in 2016 had a major switch to using Python exclusively (38%). There was only 4% of switch by Python users to using both platforms, and only 11% of readers that used both platforms in 2016 that switched to using only R. There is also a clear flow of users that are using either R or Python for Analytics, Data Science and Machine Learning from other platforms - 17% to using only R, 19% to using only Python and 4% to using both, R and Python.

KDnuggets performed a similar poll in 2015 [9], which served as a basis for trend recognition of platform usage. Figure 5 shows these trends of using different programming languages/platforms for Analytics, Data Science, and Machine Learning. The Figure 5 clearly shows that the use of other platforms (not R or Python) is dropping and it will probably continue to drop in the following years. The usage of R peaked in 2015 but had a somehow sharp

drop in 2017, while the usage of Python programming language is steadily rising and should continue to grow if this trend continues.

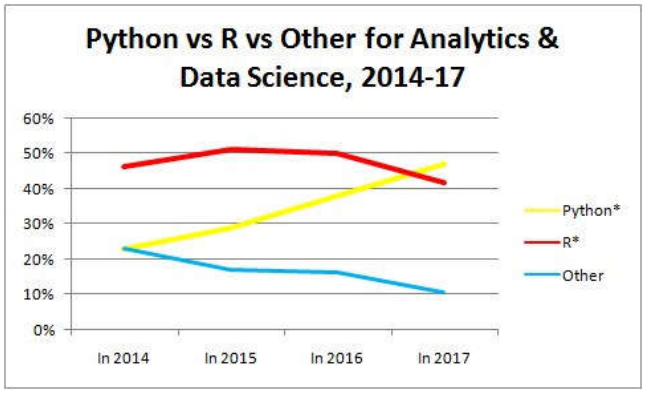


Figure 5. Platform usage for Analytics, Data Science and Machine Learning from 2014 to 2017 [8].

We made a quick glance at the popularity of R and Python platforms for Data Science, ourselves. Figure 6 shows the Google Trend chart, where it shows search term popularity on the timeline. We compared two search terms: "Python Data Science" (blue trend line), and "R Data Science" (red trend line).

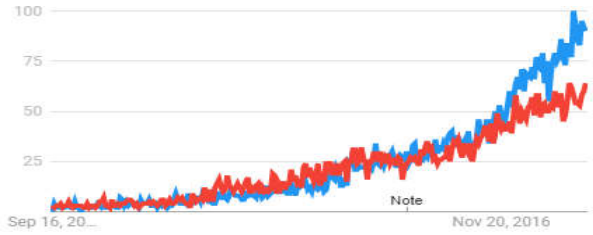


Figure 6. Google Trends search term popularity for last five years for terms "Python Data Science" for blue trend line and "R Data Science" for red trend line (September 9th, 2017).

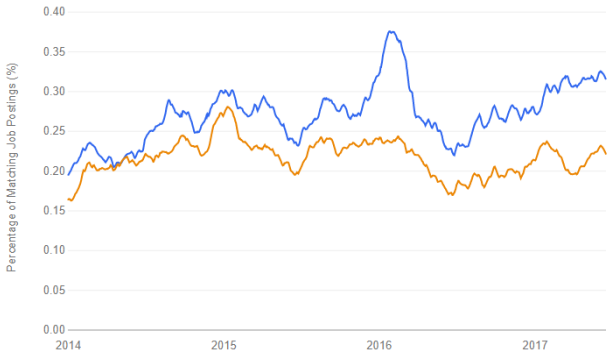


Figure 7. Job posting trends on Indeed.com for last five years for terms "Python Data Science" for blue trend line and "R Data Science" for orange trend line (September 9th, 2017).

As chart shows, there was the almost even popularity of both search terms until the end of 2016, there was just a slight lead of R in the year 2015. In the beginning of 2016, the Python search term took over the lead and its search term popularity gained more and more lead as the time progressed. Although popularity for both terms increased, the search term for Python platform has a clear advantage. After that, we also did a trend analysis on the job posting site Indeed.com. Figure 7 shows two trends for the same search terms as before ("Python Data Science" and "R Data Science") for last five years. Even in the job posting aspect, the Python platform has a clear advantage in comparison to R.

5. CONCLUSION

In this paper, we present the definition of a Data Scientist and some frameworks of its required skill set and competences. We presented existing research in the field of identifying the core skills and competences and survey the current state of needed and popular skills among practicing Data Scientists. We may conclude that a Data Scientist requires a diverse set of skills and has to adapt to new platforms as their popularities change throughout the time. It is yet to be seen how these skills and popular frameworks used in the work of a Data Scientist will change in the future, but for now we can conclude that skills of analytics, engineering, data management, research methods, project management and business analytics using Python and R platforms present a core of skills every Data Scientist needs.

6. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0057).

7. REFERENCES

[1] —, 50 Best Jobs in America, Glassdoor [online] https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm Accessed on 2017-09-07

[2] EDISON Data Science Framework (EDSF), http://edison-project.eu/edison/edison-data-science-framework-edsf Accessed on 2017-09-07

[3] NIST SP 1500-1 NIST Big Data interoperability Framework (NBDIF): Volume 1: Definitions, September 2015. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-1.pdf Accessed on 2017-09-07

[4] EDISON Data Science Framework: EDSF Part 1: Data Science Competences Framework (CF-DS) Release 2, July 2017. http://edison-project.eu/data-science-competence-framework-cf-ds Accessed on 2017-09-07

[5] Abidin, W.Z., Ismail, N.A., Maarop, N., Alias, R.A.: Skills Sets Towards Becoming Effective Data Scientists, In: Proceedings of the 12th International Conference, KMO 2017, Beijing, China, August 2017, Communications in Computer and Information Science, vol. 731, Springer, 2017.

[6] Stadelmann, T., Stockinger, K., Braschler, M., Cieliebak, M., Baudinot, G., Ruckstuhl, G. Applied data science in Europe – challenges for academia in keeping up with a highly demanded topic. European Computer Science Summit (2013)

[7] —, Top 75 Data Science Blogs And Websites For Data Scientists. http://blog.feedspot.com/data_science_blogs/ Accessed on 2017-09-08

[8] Piatetsky, G. Python overtakes R, becomes the leader in Data Science, Machine Learning platforms. KDnuggets, 2017. http://www.kdnuggets.com/2017/08/python-overtakes-r-leader-analytics-data-science.html Accessed on 2017-09-08

[9] Piatetsky, G. R, Python users show surprising stability, but strong regional differences. KDnuggets, 2015. http://www.kdnuggets.com/2015/07/poll-primary-analytics-language-r-python.html Accessed on 2017-09-08

Towards a Classification of Educational Tools

Kristjan Košič
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
kristjan.kosic@um.si

Alen Rajšp
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
alen.rajsp@um.si

Jernej Huber
Faculty of Electrical Engineering and Computer Science, University of Maribor
Maribor, Slovenia
jernej.huber@um.si

ABSTRACT
As part of Didakt.UM project, which aims to exchange of experience and create a platform that would enable the efficient search and selection of suitable ICT solutions, used for educational purposes within the University of Maribor, an analysis and classification of such ICT solutions were made. Out of 82 entries, 63 tools were classified into the broad classification, which intends to cover the widest range of ICT solutions, used by the students and staff of University of Maribor.

Categories and Subject Descriptors
K.3 **[Computers and education]**: General; H.4 **[Information systems applications]**: General; K.6 **[Management of computing and information systems]**: General

General Terms
Management, Documentation, Performance, Economics, Human Factors, Standardization, Legal Aspects.

Keywords
Software classification, Software taxonomy, Educational software, Software usage, Learning stack

1. INTRODUCTION
The preparation of an exhaustive list of ICT solutions presents a complex challenge due to the extremely high number and variety of solutions and their respective domains of use. The need to place these solutions within the different levels of the classification makes the challenge even more complex [1].

The purpose and objective of our research was two folded. Firstly, we provided an analysis of the existing situation regarding the usage of ICT solutions, used for educational purposes by the students and staff of University of Maribor (from hereinafter referred to as UM). Secondly, the classification of aforementioned ICT solutions was prepared to give a foundation for establishing the learning stack [2], which represents a collection of applications, cloud services, content repositories and data sources that can be accessed through a content platform. Such platform would enable the pedagogical staff to search, comment and rate the suitable ICT tools within the repository and to exchange the didactic experience and good practices within the UM environment.

The document consists of the following sections. We provide a short overview of related work that covers different approaches in classifying the ICT resources in section 2.

Secondly, we present our classification proposal with a mind map of the classification with the first level categories in section 3.1. We continue with statistical analysis of the survey of the usage of ICT tools within the UM in section 3.2.

Lastly, we provide a sneak-peak into a project deliverable in the form of a two-level classification table, which offers a more detailed overview of the resulting classification.

2. RELATED WORK
At the highest level, technological infrastructure can be divided into hardware and software resources [3]. Hardware refers to the mechanical, visible and tangible part of the information technology, while software presents a set of instructions prepared to obtain an adequate final result [4]. With the rapid development of smartphones and embedded systems, the hardware-dependent software has been recently gaining unprecedented use within a wide range of domains such as medicine, telecommunications, automotive industry and others [5].

A global classification of educational tool was not found. Various sub-classifications, related to specific use-cases or niche domains were included in the analysis. In general, software is usually divided into application and system solutions [4]. The latter offer an infrastructure environment for running application software and include operating systems, drivers, system utilities, and servers. The category of application solutions can include software that enables information management, education, business infrastructure, simulation, media processing, software development and solutions that contain the concepts of gamification [6], [7]. Multiple taxonomies for classifying software already exist, one of most known being the ACM Computing Classification System, which was lastly revised in 2012 [8]. Main categories from the ACM taxonomy are: General and reference; Hardware, Computer systems organization; Networks; Software and its engineering; Theory of computation; Mathematics of computing; Information systems; Security and privacy; Human-centered computing; Computing methodologies; Applied computing; Social and professional topics; Proper nouns: People, technologies and companies. The purpose of ACM taxonomy is to provide a categorization of technology-related topics. From application domain standpoint it provides relatively poor coverage for some application types, such as information display and consumer-oriented software [9]. On the other hand, open-source community, with sites as SourceForge and Google Code, provide a good overview of most types of software, developed by such communities. Google's approach for defining application domains avoids the hierarchical structure and relies on tagging [9]. Additionally, many authors have developed their own more or less up-to-date taxonomies, which divide software into categories based on the purpose of use (e.g. data-dominant, systems, control dominant and computation-dominant software, the categories that are furtherly divided into domains of use) [9] or directly by the domain of use [10].

3. CLASSIFICATION PROPOSAL

3.1 Classification attributes

We aimed to avoid classifying software only by purpose and domain of use and strived to provide more comprehensive and holistic approach for classifying software in education sector. Thus, we included a multitude of other factors (attributes). Examples of such factors include the type of the usage [11] (e.g. web, mobile and desktop), usage domain [12] (e.g. general-purpose or specific, such as Mathematics or Medicine), group work support (on a scale of a team, community, organization) [13], time aspect of collaboration [14] (synchronous and asynchronous) and the purpose of the use, which was further divided to functionalities. This was done to cover the widest possible range of application software, which will, of course, continue to expand in the future with the development of the ICT field. Regarding the purpose of use, we placed an emphasis on solutions in the field of education, where we divided the purpose of using such tools into three sections: learning content management [15], knowledge testing and evaluation [15] and learning analytics [16].

Among the other purposes of use, that are furtherly divided into more specific functionalities, are polling capabilities, group work support (collaboration, communication and coordination), media processing, statistical data processing, data storage, software development, software deployment, enterprise resource planning, modeling, project management, virtualization and simulation.

The initial part of the classification is a general description of tools, with data regarding the manufacturer, type of license, price of tools, solution provider within the UM and support/service level, examples of usage both in general and within the UM and a corresponding contact person. General description is followed by positioning the ICT solutions according to the Klasius P [17] classification.

The following diagram shows the classification of ICT solutions used for educational purposes within the UM. For reasons of greater transparency, we only show the first level nodes of classification.

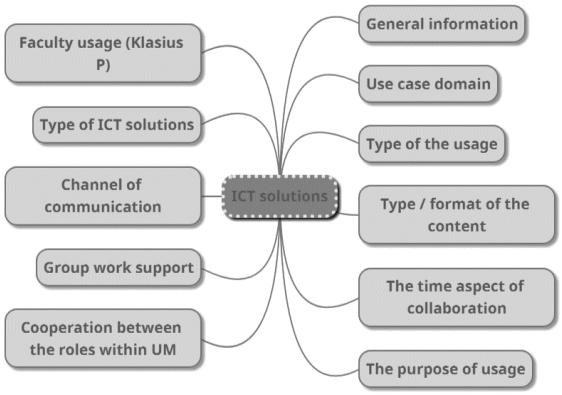


Figure 1. Top level attributes of the proposed classification.

3.2 Statistical analysis of ICT tools usage

Based on data from the survey on the usage of ICT solutions within the UM, we identified 82 entries, of which 19 entries were defective, with missing data regarding the type of solution, manufacturer etc.

Altogether, we classified the following 63 ICT solutions, namely:

Moodle, Geogebra, Sony Virtuoso in Soloist, Expression Studio, CyberLink PowerDirector, Articulate, iSpring, Hype, Sibelius, Adobe Photoshop, Photofiltre Studio X, Audacity, Windows Movie Maker, HandBrake, MKVTToolNix, Subtitle Edit, Hot Potatoes, Google Docs, Sheets, Slides, Forms, Poll Maker, Skype, The Jupyter Notebook, matplotlib, WinMIPS64, XAMPP, Usb Web Server, UwAmp, WampServer, SonarQube, Java Web Start, ERPSim, Vox Armes, BIM server, Xerte, Oracle database server, Adonis CE, Pantheon X, Bizagi Business Process Modeler, Microsoft Visio, Microsoft Dynamics Nav, Microsoft Project, Aris Architect & Designer, Aris Express, JDeveloper, Eclipse, SQL Developer, SQL Developer Data Modeler, Greenfoot, Tableau, Orange, SAP Lumira, SAP-ERP, Oracle VM VirtualBox, VMware Workstation Player, Linux Ubuntu, Kali Linux, SPSS, AnyLogic, Turning Point, Kahoot, Padlet, Anatomy 4D, Virtual Patient MedU, ThinkDesign Suite.

The column chart in Figure 2 shows the number of solutions according to the domains of the usage. It is important to stress that one tool can belong to more than one domain. Majority of tools represented the computer science and informatics domain (28), while 19 tools were general-purpose tools (such as Skype, Google Docs) that can be used within any domain.

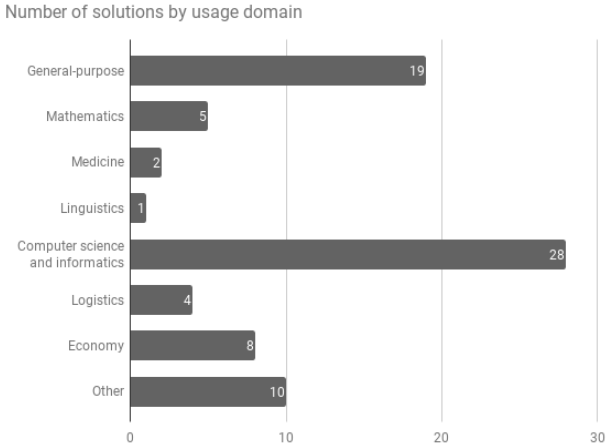


Figure 2. Number of solutions by usage domain.

The pie chart in Figure 3 shows the ratio between the open source and proprietary solutions. Most of the documented equipment (39 out of 63) were proprietary.

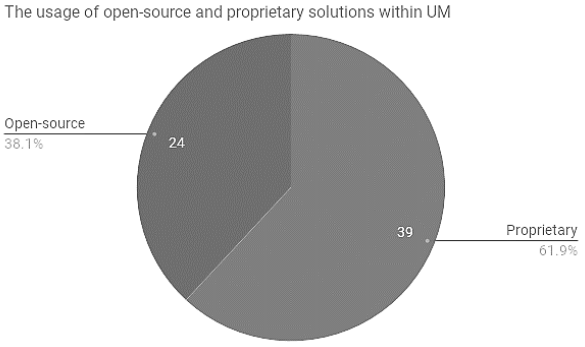


Figure 3. Ratio between open-source and proprietary tools

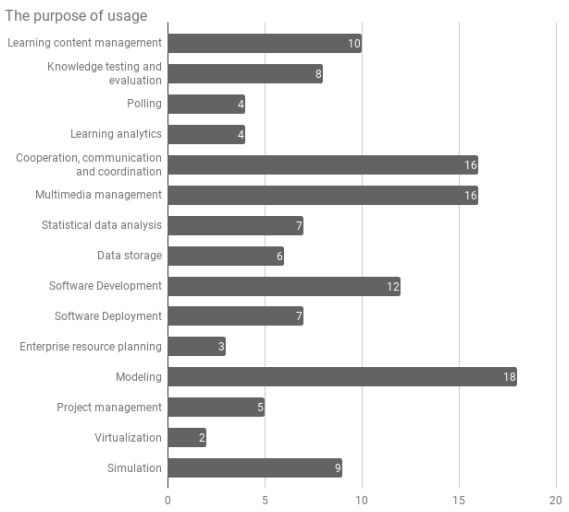


Figure 4. Number of solutions by the purpose of usage.

The column diagram in Figure 4 shows the number of tools that support at least one functionality from the categories, which describe purpose of use. Most often, the tool was intended for modeling (18); cooperation, communication and coordination (16); multimedia management (16), software development (12) and learning content management (10).

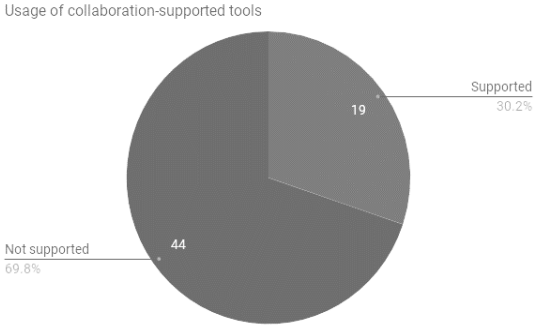


Figure 5. Ratio of collaboration-supported tools.

The pie chart in Figure 5 shows the proportion of solutions in terms of collaboration support. The 19 tools from the survey allow groups to work together, while the remaining 44 solutions do not have such support.

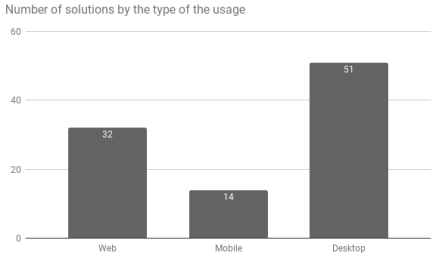


Figure 6. Number of solutions by the type of the usage.

The column chart in Figure 6 shows the number of solutions based on the type of client, with 32 tools permitting online use within the browser. The 14 tools can be accessed with mobile smartphones and 51 tools are developed as desktop applications. Again, it is important to stress that each of the tool can have more than type of a client.

The column chart in Figure 7 shows the number of solutions according to the types of ICT solutions as proposed in our classification. Most tools meet the following types of ICT solutions: information management (30), software development (17), and education (16).

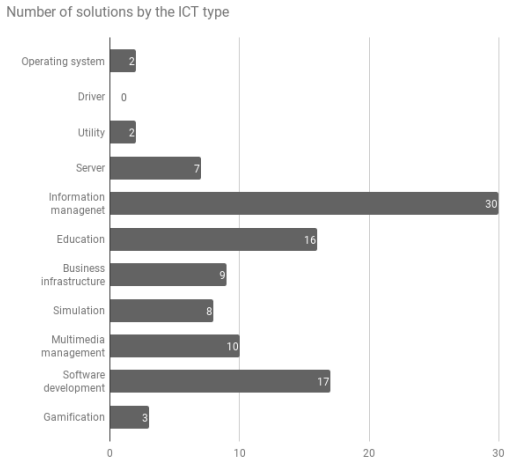


Figure 7. Number of solutions by the ICT type.

3.3 Proposed two-level classification

The Table 1 presents the more detailed introspection into our classification proposal. Within this article, we limited the number of attribute hierarchy level into two levels. The actual classification was divided into the three-level hierarchy of classification attributes, hence being even more comprehensive.

Table 1. Classification of used solutions (to the second level)

1st level of classification	2nd level of classification
General information	Name of the solution; Description; Manufacturer; Manufacturer's URL; License type; Price; Provider; Provider's URL; Support/service level; Minimum system requirements; General use case; UM use case; UM contact person
Faculty usage (Klasius P)	1 - Teacher training and education science; 2 - Humanities and arts; 3 - Social sciences, business and law; 4 - Science, Mathematics and Computer Science; 5 - Engineering, manufacturing and construction; 6 - Agriculture, Forestry, Fisheries, Veterinary; 7 - Health and welfare; 8 - Services
Use case domain	General-purpose; Specific
Type of ICT solutions	System software; Application software
Type of the usage	Web; Mobile; Desktop
Channel of communication	Video; Sound; Text

Type / format of the content	Video material; Graphical material; Sound; Text; Spreadsheet; Presentation; Any file
Group work support	Among the members of the organization; Among the members of the community; Among the team members
The time aspect of collaboration	Asynchronous; Synchronous
Cooperation between the roles within UM	Student; Teacher; Domain expert; Administrator
The purpose of usage	Learning content management; Knowledge testing and evaluation; Polling; Learning analytics; Cooperation, communication and coordination; Multimedia management; Statistical data analysis; Data storage; Software Development; Software Deployment; Enterprise resource planning; Modeling; Project management; Virtualization; Simulation

The result of our in-depth analysis was a report in which we provided the three-level classification of ICT solutions, a brief description of each attribute of the classification and the actual placement of 63 identified ICT solutions within our proposed classification.

4. CONCLUSION

Classification of educational tools is a broad topic that still has a lot of room for improvement and research. In the future, we suggest additional classification and categorization of tools combined with pedagogical learning approaches related to the specific needs of the instructor. Moreover, the framework could be expanded with pedagogical classifications and requirements related to regional/local pedagogical classifications (i.e. related to the specific country).

5. ACKNOWLEDGEMENTS

This research was carried out within the project *Didakt.UM*, which is financed by the Slovenian *Ministry of Education, Science and Sport* and European Union from *European Social Fund*.

6. REFERENCES

[1] A. Saito, K. Umemoto, and M. Ikeda, "A strategy-based ontology of knowledge management technologies," *Journal of Knowledge Management*, vol. 11, no. 1, pp. 97–114, Feb. 2007. DOI=https://doi.org/10.1108/13673270710728268.

[2] Jan-Martin Lowendahl, "Hype Cycle for Education," *Gartner*, 2016. [Online]. Available: https://www.gartner.com/doc/3364119/hype-cycle-education-. [Accessed: 07-Sep-2017].

[3] M. Afshari, K. A. Bakar, W. S. Luan, B. A. Samah, and F. S. Fooi, "Factors Affecting Teachers' Use of Information and Communication Technology," *International Journal of Instruction*, pp. 77–104, 2009.

[4] I. Masic *et al.*, "Information Technologies (ITs) in Medical Education," *Acta Informatica Medica*, vol. 19, no. 3, p. 161, 2011. DOI=https://doi.org/10.5455/aim.2011.19.161-167.

[5] W. Ecker, W. Müller, and R. Dömer, "Hardware-dependent Software," in *Hardware-dependent Software*, Dordrecht: Springer Netherlands, 2009, pp. 1–13.

[6] A. Saito, K. Umemoto, and M. Ikeda, "Journal of Knowledge Management A strategy-based ontology of knowledge management technologies," *Journal of Knowledge Management Journal of Knowledge Management Journal of Knowledge Management*, vol. 11, no. 6, pp. 97–114, 2007.

[7] TechTarget, "What is software?," 2017. [Online]. Available: http://searchmicroservices.techtarget.com/definition/software. [Accessed: 31-Aug-2017].

[8] ACM, "The 2012 ACM Computing Classification System," 2012. [Online]. Available: https://www.acm.org/publications/class-2012. [Accessed: 05-Sep-2017].

[9] A. Forward and T. C. Lethbridge, "A taxonomy of software types to facilitate search and evidence-based software engineering," in *Proceedings of the 2008 conference of the center for advanced studies on collaborative research meeting of minds - CASCON '08*, 2008, p. 179.

[10] R. L. Glass and I. Vessey, "Contemporary application-domain taxonomies," *IEEE Software*, vol. 12, no. 4, pp. 63–76, Jul. 1995. DOI=https://doi.org/10.1109/52.391837.

[11] SD Times, "Web, desktop, mobile: What's the difference?," 2017. [Online]. Available: http://sdtimes.com/web-desktop-mobile-whats-the-difference/. [Accessed: 31-Aug-2017].

[12] eduCBA, "What is application software & its types," 2017. [Online]. Available: https://www.educba.com/what-is-application-software-its-types/. [Accessed: 31-Aug-2017].

[13] H. Fuks *et al.*, "The 3C Collaboration Model," in *Encyclopedia of E-Collaboration*, IGI Global, pp. 637–644.

[14] TechTarget, "Synchronous vs. asynchronous communication: The differences," 2017. [Online]. Available: http://searchmicroservices.techtarget.com/tip/Synchronous-vs-asynchronous-communication-The-differences. [Accessed: 31-Aug-2017].

[15] S. R. Malikowski, M. E. Thompson, and J. G. Theis, "A Model for Research into Course Management Systems: Bridging Technology and Learning Theory," *Journal of Educational Computing Research*, vol. 36, no. 2, pp. 149–173, Mar. 2007. DOI=https://doi.org/10.2190/1002-1T50-27G2-H3V7.

[16] R. Scapin, "Learning Analytics in Education: Using Student's Big Data to Improve Teaching," 2015.

[17] Statistični urad Republike Slovenije, "Klasius-P," 2017. [Online]. Available: http://www.stat.si/Klasius/Default.aspx?id=5. [Accessed: 05-Sep-2017].

Indeks avtorjev / Author index

Beranič Tina 31
Budimac Zordan 27
Catal Çağatay 7
Heričko Marjan 19, 31
Heričko Tjaša 35
Huber Jernej 15, 43
Kamišalić Aida 11
Karakatič Sašo 39
Košič Kristjan 43
Montoya Edwin 11
Muratli Can 7
Pavlinek Miha 19
Podgorelec Vili 39
Polančič Gregor 15
Pušnik Maja 19
Rajšp Alen 43
Rakić Gordana 27
Rednjak Zlatko 31
Šestak Martina 23
Sukur Nataša 27
Tabares S. Marta 11, 15
Torres Camilo 11

Konferenca / Conference

Uredil / Edited by

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

Marjan Heričko