INFORMATICS EUROPE

THE 1ST EARLY CAREER RESEARCHERS WORKSHOP COLLOCATED WITH ECCS 2021

Proceedings of Research Statements

The 1^{st} Early Career Researchers Workshop collocated with ECCS 2021

Proceedings of Research Statements

25 October 2021
Madrid, Spain

DOI: 10.5281/zenodo.15863604

Welcome to the Proceedings of the 1st Early Career Researchers Workshop collocated with the European Computer Science Summit (ECCS) 2021, held in Madrid, Spain between 25th and 27th October 2021. The goal of the one-day workshop, organised by Elisabetta Di Nitto (Politecnico di Milano) and Standa Živný (University of Oxford), was to support early career researchers (PhD students and postdocs) in the development of their soft skills related to presentation abilities, networking, developing a research plan, and connections to industry. The workshop was open to early career researchers in all areas of computer science and software engineering. The participants enjoyed invited talks by Geraldine Fitzpatrick (TU Wien), Lynda Hardman (CWI), Justyna Petke (UCL), and Wolfgang Emmerich (UCL, Zuhlke Group) as well as a panel discussion on open science and its impact on career development, held in conjunction with the parallel Leaders Workshop, and an open discussion with industry that has involved as panelists Ruoyi Zhou (IBM Research Europe), Nuria de Lama (Atos and Board of Directors, Big Data Value Association), and Africa Real (HP).

The research statements in this proceedings were submitted by the participants and contain a wide variety of work, either completed or in progress, being undertaken by the current generation of PhD students and early career researchers.

The workshop would have not been possible without the support of many people and organisations. In particular, we would like to thank Informatics Europe and the local organisers in Madrid.

We hope you had a great time in Madrid!

Elisabetta Di Nitto and Standa Živný
Early Career Workshop Organisers, 2021

Table of Contents

Quality Analysis of Mobile Applications with Focus on Security

Kristiina Rahkema[1][0000−0001−7332−2041]

University of Tartu, Estonia kristiina.rahkema@ut.ee

Abstract. Smart phones and mobile applications have become an essential part of our daily lives. It is necessary to ensure the quality of these applications. The purpose of my PhD thesis is to study code quality of mobile applications. Two important aspects of code quality are maintainability and security. My goals are to study code smells, security issues and their evolution in iOS applications and frameworks and to apply the gained knowledge to support developer training an teaching. I have built the tool GraphifyEvolution to analyze source code evolution. The tool is built in a modular manner and can be extended to add support for additional languages and external analysis tools. I have analysed code smells in open source iOS applications and compared them to code smells in Android applications. In the remaining two years of my PhD studies I plan to complete the analysis of code smell and security vulnerability evolution in iOS applications and frameworks. In addition, I will apply GraphifyEvolution in developer training at one industry partner and in university teaching.

Keywords: Cod smells, code evolution, dependencies, mobile apps, security

1 Motivation and Background

With the popularity of smartphones continuously rising, already today more time is spent on smartphones than on desktop computers [10]. In 2019, most of the global market share (76%) was held by Android, while iOS held only 22% [16]. At the same time overall earnings on the iOS App Store are considerably larger than on the Google Play Store [12]. This makes both Android and iOS very attractive targets for developers. It is therefore important to study code quality not only for Android, but also for iOS applications.

There are multiple aspects to consider regarding code quality, for example maintainability and security. One proxy for code quality is code smells. So far there has been very little research on code smells in iOS applications.

Goals: The purpose of my PhD thesis is to study code quality of mobile applications and to help developers build better maintainable and more secure mobile applications. There are three main goals. Since code smells are considered to be obstacles for maintainability, the first goal is to discover the most frequent code smells and how they affect iOS applications. The second goal is to analyse

how security issues are introduced into iOS applications and how these issues can be prevented. The third goal is to build on these analyses to facilitate teaching and training developers to write better maintainable and more secure mobile applications.

Background: The following paragraphs list most important related work. A more thorough description of related work is given in my published articles [14,13,15].

Most of the current research on code smells in mobile applications has been carried out on the Android platform. Mannan et al. [8] analyzed 21 object oriented code smells in open source Android and Java desktop applications. Mateus et al. compared code quality in Android applications written in Java and Kotlin [9]. Hecht [5] developed a tool called PAPRIKA to analyze four object oriented and six Android specific code smells in Android applications. This tool analyses the Android APK, generates a model and saves it in a graph database [7].

Habchi et al. [2] used PAPRIKA to detect code smells in iOS applications. They used ANTLR4 grammars to generate parsers for Swift and Objective-C code. They analysed the AST generated by these parsers to create the applications graphs used by PAPRIKA. They compared code smell occurrences on iOS and Android and concluded that Android applications were more prone to code smells [2]. To the best of our knowledge this has been the only study looking at code smells on iOS applications.

For analysing the code smell evolution of mobile applications, multiple approaches have been used. To track software quality of android applications, Hecht et al. [6] and Mateus et al. [9] both used PAPRIKA to analyse the evolution of Android applications, applying the tool for each commit. Tufano et al. [17] implemented a tool called HistoryMiner that runs DECOR on each commit, for changed files. Habchi et al. [4,3] implemented a tool called SNIFFER that extracts commits from a git repository, detects code smells in each commit using PAPRIKA and outputs the code smell evolution of a project.

Public vulnerability databases (e.g., the OSV[1] database and the NVD[2] database) contain reported vulnerabilities and serve as an important data source for developers to monitor the security of third party applications and frameworks used. Unfortunately not all vulnerabilities are reported publicly and these databases only contain data about very large applications. For some platforms, solutions exist that bring this data closer to developers. For Android applications Nguyen et al. [11] created an Android Studio plugin that detects outdated and vulnerable libraries and checks if a library can be updated without additional code changes. It lacks, however, analyses that check if the vulnerability really affects the application and when suggesting library updates if the library API functionality has stayed the same.

[1] `https://osv.dev/`
[2] `https://nvd.nist.gov/`

2 Analysis of Code Smells in Mobile Applications

Code smells are recurring patterns in code that have been identified as bad practices [1]. They yield technical debt and long-term maintainability problems [1]. Given the significance of the smartphone market it is important to study code smells on both Android and iOS.

Tool - GraphifySwift: Since PAPRIKA has been a successful tool for analyzing a lot of applications at once I decided to build a similar tool for Swift analysis. I decided against the approach taken by Habchi et al. [2], where they needed to first convert iOS applications into a suitable format. I created a tool called GraphifySwift[3] that analysis Swift code, generates a model of this code and enters it into a neo4j graph database. For indexing the source code and generating the structure of the Swift code I used a framework called SourceKittenFramework[4] that acts as a wrapper around Apple's powerful SourceKit. The graph database contains the following nodes: App, Module, Class, Function, Variable and Argument. These nodes are connected through relationships. This model differs slightly from the one used in PAPRIKA. I added the Module node and the following relationships APP_OWNS_MODULE, $MODULE\_OWNS\_CLASS$, IS_TYPE_OF and $DUPLICATES$. Code smells are defined as queries. The query definitions can be found on the tool GitHub page[3]. Running code smell queries using this tool generates CSV files for each code smell and allows us to analyze the results. The tool GraphifySwift is no longer in development, the code and documentation is still available on the tool web page, but all the functionalities are incorporated by a new tool as described in the next section.

Tool - GraphifyEvolution: I extended the tool GraphifySwift and implemented a tool called GraphifyEvolution[5] [15] that can analyse applications written in various languages (Swift, Java and C++), including iOS applications written in Swift. Different to previous tools that output data about the code smell evolution, GraphifyEvolution detects which classes, methods and variables are changed during a commit and records only these changes. This makes it possible to query code smells for all versions of the application at once. Additionally, I implemented the tool in a modular manner making it easy to add support for additional languages and external analysis tools that can be run for each commit. Currently I have implemented preliminary support for Swift, Java and C++. Support for external tools is added for jscpd that finds code duplicates and insider that detects security vulnerabilities.

Empirical Studies on Code Smells in iOS Applications: I identified the most common code smells in open source iOS applications [14]. In terms of percentage of applications affected by a specific code smell the five most common code smells are Lazy Class, Long Method, Message Chain, Ignoring Low Memory Warning and Data Class. Different to what is often seen in iOS developer blogs, Massive View Controller was one of the least common code smells. Under 18% of

[3] `https://github.com/kristiinara/GraphifySwift`
[4] `https://GitHub.com/jpsim/SourceKitten`
[5] `https://github.com/kristiinara/GraphifyEvolution`

applications were affected by this smell. I also compared code smell occurrences in iOS and Android applications [13]. Analysis on Android applications was done by populating the database using PAPRIKA. In total, I identified 19 code smell types that could potentially occur in apps on both platforms. My analysis showed that 18 of the 19 identified code smells occurred in apps on both platforms. Code smell DistortedHierarchy never occurred in iOS apps. It turned out that, contrary to what Habchi et al. [2] expected, the overall density of code smells is higher in iOS apps than in Android apps. The proportions of code smells differ between platforms. Android applications ware more often affected by code smells that seem to corresponds to big and complex classes while iOS applications were more affected by code smells that seem to correspond to smaller and simpler classes. These results can be interesting for developers moving from one platform to the other. It can also be useful for developers of tools for these platforms. The emphasis on which code smells to look at is different depending on the platform.

Future Work: GraphifyEvolution was mainly built to analyse code smell evolution in applications written in Swift. I plan to conduct a large scale study analysing the evolution of code smells in open source iOS applications and iOS frameworks. Similar studies have been done on Android applications [6,4], but so far the evolution of code smells has not been studied in iOS applications.

Visualisations Based on GraphifyEvolution: The analysis produced by GraphifyEvolution can be a useful base for visualisations. It already includes high level information about classes, methods, variables and the relationships between them. Using these aspects in the visualisations make it possible for the user to get an understanding of what the code does and how the functionalities have changed. I have supervised two master theses that built visualisations based on GraphifyEvolution. One of the visualisations, that concentrated on highlighting code smells in each commit, was built in cooperation with a software development company and received positive feedback from their developers. The other visualisation tried to show different aspects of code evolution, such as changes in method calls.

3 Security Analysis of Mobile Applications

Modern software, including mobile applications, are built using numerous third party frameworks. It is important to keep framework versions up to date to prevent introducing vulnerabilities. In practice however many developers do not update their dependencies.

Envisioned Solution Based on GraphifyEvolution: Analysing the evolution of vulnerabilities also opens up the possibility to analyse vulnerabilities introduced through application dependencies. I am planning on creating a public database that contains analysed open source Swift libraries. GraphifyEvolution will be extended so that it can connect method calls and variable uses from the analysed application to the already analysed open source libraries. This allows a more thorough vulnerability analysis by looking at both the direct and transitive

dependencies of an application and by checking if the vulnerable methods are reachable from the analysed application.

4 Application in Training and Teaching

One reason to look at the evolution of a project is to learn how an application or applications in general evolve. This understanding can help new developers learn how complex applications are built and how they grow. Analysing the evolution of a specific project can also help developers understand why and how problems in their code were introduced which can help prevent such issues in the future.

Developer Training: There is an ongoing cooperation with a Finnish company to use GraphifyEvolution in their developer training. We are planning on analysing code produced by developers in these training sessions to detect patterns and gather feedback if this tool can help facilitate giving feedback.

Teaching at the University: I plan to extend GraphifyEvolution so it can be used by instructors for monitoring and comparing student progress. Visualising changes between code versions would make it easier for instructors to understand which classes, methods and variables were added and how the project has grown since the last version. The extended tool will be used in university courses by instructors and improved based on their feedback.

5 Next Steps

In the first two years of my PhD studies I have implemented a tool called GraphifySwift and its extended version GraphifyEvolution that can analyse the evolution of source code written in Swift, Java and C++. I have used GraphifySwift to analyse code smells in open source iOS apps and compared code smell occurrences in open source iOS and Android apps.

In the next two years of my studies I plan to finish analysing the evolution of code smells and vulnerabilities in open source iOS apps and frameworks. I also plan on extending the tool so that analysis of open source frameworks can be connected to their uses in analysed applications which adds additional possibilities for security analysis. Lastly I plan to apply the tool to developer training and teaching in industry.

Acknowledgements. Funding of this research came from the Estonian Center of Excellence in ICT research (EXCITE), the IT Academy Programme for ICT Research Development,the Austrian ministries BMVIT and BMDW, and the Province of Upper Austria under the COMET (Competence Centers for Excellent Technologies) Programme managed by FFG.

References

1. Fowler, M.: Refactoring: improving the design of existing code. Addison-Wesley Professional (2018)

2. Habchi, S., Hecht, G., Rouvoy, R., Moha, N.: Code smells in ios apps: How do they compare to android? In: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). pp. 110–121. IEEE (2017)
3. Habchi, S., Moha, N., Rouvoy, R.: The rise of android code smells: Who is to blame? In: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). pp. 445–456. IEEE (2019)
4. Habchi, S., Rouvoy, R., Moha, N.: On the survival of android code smells in the wild. In: 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft). pp. 87–98. IEEE (2019)
5. Hecht, G.: An approach to detect android antipatterns. In: Proceedings of the 37th International Conference on Software Engineering-Volume 2. pp. 766–768. IEEE Press (2015)
6. Hecht, G., Benomar, O., Rouvoy, R., Moha, N., Duchien, L.: Tracking the software quality of android applications along their evolution (t). In: 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 236–247. IEEE (2015)
7. Hecht, G., Rouvoy, R., Moha, N., Duchien, L.: Detecting antipatterns in android apps. In: Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems. pp. 148–149. IEEE Press (2015)
8. Mannan, U.A., Ahmed, I., Almurshed, R.A.M., Dig, D., Jensen, C.: Understanding code smells in android applications. In: 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft). pp. 225–236. IEEE (2016)
9. Mateus, B.G., Martinez, M.: An empirical study on quality of android applications written in kotlin language. Empirical Software Engineering pp. 1–38 (2018)
10. Meeker, M.: Internet trends 2018 (2018), `https://www.kleinerperkins.com/perspectives/internet-trends-report-2018/`. Last accessed 20 Sep 2019
11. Nguyen, D.C., Derr, E., Backes, M., Bugiel, S.: Up2dep: Android tool support to fix insecure code dependencies. In: Annual Computer Security Applications Conference. pp. 263–276 (2020)
12. Perez, S.: App store generated 93% more revenue than google play in q3 (2018), `https://techcrunch.com/2018/10/11/app-store-generated-93-more-revenue-than-google-play-in-q3/`. Last accessed 20 Sep 2019
13. Rahkema, K., Pfahl, D.: Comparison of code smells in ios and android applications. In: 8th International Workshop on Quantitative Approaches to Software Quality (QuASoQ 2020) co-located with 27th Asia-Pacific Software Engineering Conference (APSEC 2020). pp. 79–86 (2020)
14. Rahkema, K., Pfahl, D.: Empirical study on code smells in ios applications. In: Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems. pp. 61–65 (2020)
15. Rahkema, K., Pfahl, D.: Empirical study on code smells in ios applications. In: Proceedings of the IEEE/ACM 8th International Conference on Mobile Software Engineering and Systems. pp. 24–27 (2021)
16. StatCounter: Mobile operating system market share worldwide (2019), `https://gs.statcounter.com/os-market-share/mobile/worldwide`. Last accessed 20 Sep 2019
17. Tufano, M., Palomba, F., Bavota, G., Oliveto, R., Di Penta, M., De Lucia, A., Poshyvanyk, D.: When and why your code starts to smell bad (and whether the smells go away). IEEE Transactions on Software Engineering **43**(11), 1063–1088 (2017)

Evolution of Users' Choice Behaviour Under the Influence of Recommender Systems

Naieme Hazrati

University of Bozen-Bolzano
nhazrati@unibz.it

Abstract. The major focus of recommender systems (RSs) research has so far been on improving the precision and quality of the recommendations. However, it is also important to understand whether the recommended items are actually chosen and how they influence users' choice making. Few studies have attempted to analyse the impact of RSs on users' choice making. In this PhD research, we aim at better understanding the impact of RSs on the evolution of the choices made by a collection of users. We propose simulation procedures where users are simulated to make choices over a period of time when they are exposed to alternative RSs. We measure several properties of the users' choices distribution, which capture the RS effect. Our goal is to understand the evolution of the choices of a collection of users as time goes; next choices are influenced by previous choices used by the RS to generate recommendations. Additionally, we propose online experiments to study the effect of RSs on real users' choices. We propose to design web-based platforms where alternative RSs recommend items to the users and study RSs impact by analysing the evolution of the choices.

Keywords: recommender system; simulation; choice model

1 Introduction

Recommender Systems (RSs) are routinely used to influence users' choices by suggesting items that the user will find interesting and of use [6]. The quality of an RS is typically measured by metrics such as precision of recommendations or Click Through Rate (CTR). Even though RSs are used widely, their impact, which can be studied at the level of an individual user or at the level of a community, has not been fully understood yet. Therefore, it is important to fill this research gap by identifying the key conditions and system features, such as the specific RS technique, that affect the users' choices.

A few works that have attempted to understand the impact of RSs have measured aggregated indicators of RSs' impact, i.e., users' choice distribution [5, 3, 7]. They have followed two main approaches: *online experiments* or *simulations*. In online experiments, web platforms are designed where online users select items while they are also exposed to recommendations [5]. Users may select the recommended items or not. In simulations, artificial agents make choices

according to a probabilistic or deterministic choice model, and alternative RSs are simulated to make recommendations to these agents [3, 7]. In these works, an RS's effect on the agents (decision-makers) is simulated by increasing the likelihood of the recommended item(s) to be chosen. In both approaches, choices, real or simulated, are observed for a time period, and then some metrics of collective users' behaviour are measured to capture the RS effect. These approaches have specific limitations. Online experiments have discovered notable results [5], e.g., Matt et. al. [5] discovered that the diversity of the choices can differ depending on the RS's approach. However, since online experiments require the involvement of real users, it is difficult to assess the RSs effect under different circumstances. However, previously conducted simulations have made simplifying assumptions on the environment and their choice models that reduce the significance of their results [7, 3]. Moreover, some have used synthetic data to generate the user preference model and the alternative choice options [3].

Our research goal is to improve both approaches. First, we propose a more realistic simulation of users' choices under the impact of RSs. A reliable simulation design is a prerequisite to understand the true effect of RSs and inform the decision of which RS to use in a real application. Moreover, it is necessary to improve online experiments as the results obtained so far are not sufficient. In fact, online experiments have assessed the qualities only of a few RSs, and also, they have measured only the choice diversity, which is only one of the many dimensions of aggregated users' choice making. We see that it is still necessary to conduct online experiments. Indeed, we aim to improve online experiments to better understand the users' behaviours.

In this PhD research, we first focus on improving the significance of the simulation of users' choices under the effect of an RS, better portraying the reality of such human-computer interactions. We assess the RSs' effect on users' behaviours by measuring metrics that capture the global effect of RSs, such as the Gini index and Shannon entropy, as measures of diversity [2], catalogue coverage of the choices, average quality of the choices, and recommendation acceptance. Moreover, we analyse alternative choice behaviours of users, such as users that tend to choose popular items, recent items or highly-rated items. In this work, we assess the simultaneous impact of alternative choice behaviours and RSs on users' collective choices. Secondly, we focus on conducting online experiments assessing different RSs in terms of their impact. We will design a web platform where products are offered to online users. The users will browse recommendations before making choices. The log of this system can give us information on the behaviour of the users.

2 Simulating Users' Choices

In order to address the aforementioned goals, in our first work, we have proposed a novel simulation procedure of users' choices in the presence of RSs that affect such choices. We have improved previously proposed simulations on the users' iterative choice makings by designing a new simulation [4] that uses real datasets

of choices to correctly define the simulation components: user preferences and choice model. We consider several datasets and RSs types in the simulation. In the following, we briefly describe this work.

We use four datasets *MovieLens* 100*k* [1] and 3 *Amazon* datasets: *Kindle*, *Apps* and *Games* [2]. We simulate the choices of the users considering the presence of alternative RSs. Our simulation leverages the knowledge of the users' preferences (derived from the dataset) and simulates repeated choices. The choices are made within monthly time intervals according to a probabilistic multinomial choice model [3]. The higher the estimated utility of a user for an item, the higher is the probability that the item will be chosen. Additionally, we assume that users are not aware of the entire catalogue of the items and can only choose items in a set called "Awareness Set" [3]. We build an initial awareness set for each user. Moreover, if an item is recommended, it enters the user's awareness set. Plus, the utility of the recommended item is increased, so it's more likely that this item is chosen. We compared five RSs, including personalized and non-personalized ones. We train the RSs by using the choices actually present in the dataset, up to a certain date, and incrementally, month by month, using the simulated users' choices. For each month of simulated choices, we calculated the Gini index [2] over the choices made by the users up to that month. Additionally, we consider other metrics such as catalogues coverage, chosen items' average utility, Shannon entropy, popularity of the chosen items and the ratio of choices for the recommended items.

We have discovered that RSs have different effects on users' choice diversity. In fact, personalized RSs lead to a lower Gini index (equivalent to higher diversity) than non-personalized ones. Moreover, even among the personalized or non-personalized RSs, the effect on the Gini index varies with the specific RS approach. Additionally, we discovered that with personalized RSs, the users select items with higher utility, which means that users should be more satisfied with personalized RSs. We observed that users choices popularity differs with the RS. Finally, we discovered that the size of the awareness set of the users is very important to modulate the acceptance of the recommendations.

The obtained results are important because they show that, depending on the system goal, a specific RS can be selected for a target context, e.g., if a company aims at increasing the sales of less popular items, it is possible to apply a personalized RS that has shown to increase the diversity in the simulations.

In a second analysis, we first assessed (data analysis) that users' choices are influenced by distinguished properties of the items, s.a., their popularity or age (time from their first introduction). Then, we have measured global properties of users' choices (e.g., their diversity) when simulated choices are made among recommended items, but also assuming that users are influenced in their choices by the above-mentioned properties of the items; for instance, they tend to prefer more recent items. In order words, we have tested alternative assumptions related to the choice model. We have found some interesting results showing

[1] https://grouplens.org/datasets/movielens/100k
[2] http://jmcauley.ucsd.edu/data/amazon/

the importance of analysing the RS's joint effect and the user choice model on alternative scenarios (dataset). We discovered, for instance, that the diversity of the choices seems not to be influenced by the choice model. But, the choice model and the RS may have a big impact on the average rating and age of the chosen items. Moreover, a non-personalised RS that recommends items with the highest average rating may produce, under a choice model that prioritize more recent items, in one dataset (Kindle), choices distributed similarly to the actual choices, while not in another (Movielens).

3 Future Work

Firstly, we plan to improve the choice model. So far, the adopted choice model is a multinomial logit choice model. This has also been used in previous studies [3, 1]. However, a true user's behaviour is more complex and diverse, e.g., in the music domain, a user's choices are strongly dependent on the previous ones, which is not captured by the multinomial logit model very well.

Additionally, it is worth emphasising that the aim of simulating the users' choices is to predict future choice behaviours. Therefore, it is necessary to use only the choices observed up to a certain point in time to predict the future ones, i.e., not using any information derived from the knowledge of the future real choices, as we still did (e.g., to determine how many choices a user is doing in a simulated month). Instead, we should be able to fully predict the number and type of choices the users will make in the future. Applying the simulation in different domains is another critical facet in this PhD research. Different domains have different characteristics, e.g., in the music domain, the users may listen to a song multiple times, while in the book domain, they are less likely to read a book even twice.

Finally, we aim to conduct online experiments studying users' choice behaviours in the presence of an RS. We plan to build a platform where the users are asked to rate/like items, and then a range of RSs are used to generate relevant suggestions for them. Thereafter, we will ask users to select other items while we offer recommendations to them. We will also ask users to rate the chosen items after consuming them. The choices and the feedback can help us understand the RSs and their impact more thoroughly.

References

1. Brock, W.A., Durlauf, S.N.: A multinomial-choice model of neighborhood effects. American Economic Review **92**(2), 298–303 (2002)
2. Dorfman, R.: A formula for the gini coefficient. The review of economics and statistics pp. 146–149 (1979)
3. Fleder, D., Hosanagar, K.: Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management science **55**(5), 697–712 (2009)
4. Hazrati, N., Elahi, M., Ricci, F.: Simulating the impact of recommender systems on the evolution of collective users' choices. In: Proceedings of the 31st ACM Conference on Hypertext and Social Media. pp. 207–212 (2020)

5. Matt, C., Hess, T., Weiß, C.: The differences between recommender technologies in their impact on sales diversity. aisel.aisnet.org (2013)
6. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Recommender systems handbook, pp. 1–34. Springer (2015)
7. Szlávik, Z., Kowalczyk, W., Schut, M.: Diversity measurement of recommender systems under different user choice models. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)

Machine Learning applied to manage effort estimation and requirements in software projects*

Víctor Pérez-Piqueras[0000−0002−2305−5755]

Departamento de Sistemas Informáticos, Escuela Superior de Ingeniería Informática, Universidad de Castilla-La Mancha, Spain
victor.perezpiqueras@uclm.es

Abstract. Incremental complexity in software project management makes planning and decision making increasingly slow and difficult. Advances in Artificial Intelligence are shown as a possible solution to these management problems. In this way, it is intended to use Machine Learning algorithms applied to the field of software project management, in order to automate tasks and processes and support decision-making.

Keywords: Project management · machine learning · effort estimation · next release problem

1 Introduction to the research

As the years pass, software products and developments get more and more complex, requiring not only more effort in design and functionality, but also in planning and management. This implies that as the software gets more complex, so does its management. Thus, it has become necessary to create tools that help and support the software project management, in order to reduce its complexity, speed up the development process and support decision making. This line of research appears to solve many of the most common problems that arise in the field of software project management. In this research, two main problems have been tackled using Machine Learning algorithms:

1.1 Requirements prioritization

This problem consists of deciding the most important requirements to work on next, and it is also known as the "Next Release Problem" (NRP) [2]. It is a difficult and manual task that is performed many times throughout a project. To solve this problem satisfactorily, a certain balance must be found between the set of selected requirements, customer and stakeholder demands, and available

* This research statement and its initial results have been partially funded by the Junta de Comunidades de Castilla-La Mancha through the project SBPLY/17/180501/000493.

resources.

Some of the algorithms that have been used to solve this problems are evolutionary algorithms (genetic, NSGA-II) [6], greedy algorithms (GRASP) [3] and Ant Colony [7, 8].

1.2 Effort estimation of requirements

Requirements are created to define specific needs that must be addressed in the software product being developed. To ease the project planning, requirements are often given an estimation of the effort that must be done to complete them. In plan-driven projects, this estimation is made once at the beginning of the project. But in projects applying value-driven methodologies, estimation of requirements is done iteratively many times during the projects life. Moreover, the requirements estimation is made by the development team, giving each requirement an expert value relative to the rest of the requirements. This repetitive, manual and experience requiring task is feasible to be solved by Machine Learning algorithms.

Machine learning models [5, 4, 1] have been applied to estimate the size of user stories. Neural networks offer good results compared to the rest; in all cases, the relevant information used is the text of the product backlog items. When working with textual information, the development of rule-based algorithms, descriptive or based on probabilistic graphic models could offer a great capacity to explain the prediction returned.

2 Overview of the Research Design and General Methodological Approach

This research will focus on two specific aims that will tackle the problems described previously:

- **A1. Search-Based Software Engineering applied to Next Release Problem.** Resolution of the NRP problem, using evolutionary and greedy algorithms. Special emphasis on the development of single and multivariate EDAs (Estimation of Distribution Algorithms), since they explicitly deal with the relationships between the variables. In addition, some of the models that can be created contain a structure that allows relationships between the elements of the problem to be identified. This is of special interest since, up to the author's knowledge, no EDA has been applied to the NRP problem yet. New forms of representation of the problem will be developed in addition to the classic one of requirements vectors, associated cost and list of stakeholders: Dependencies between logical and physical requirements, return on investment, forced order, uncertainty in importance of requirements, quality of requirements, etc.

– **A2. Supervised classification applied to effort estimation of requirements.** Neural networks will be used to generate automatic estimation of the effort of requirements. These neural networks will use textual information, such as requirement title and description to predict the estimation.

3 Expected Deliverables

– Reports for all experiments regarding A1 and A2.
– User interface and code shared with research community for algorithms which provided best results.
– Publication in journals from the JCR ranking. Candidates are shown in Table 1.

Table 1: Candidates for publications.

Journal	Category	Impact Factor
Knowledge-Based Systems	Computer Science, Artificial Intelligence	5.921
Applied Soft Computing	Computer Science, Artificial Intelligence	5.472
Automated Software Engineering	Computer Science, Software Engineering	1.857
Journal of Systems and Software	Computer Science, Information Systems and Software	2.450
Empirical Software Engineering	Computer Science, Software	3,156
Expert Systems with Applications	Computer Science, Artificial Intelligence	5,452
Information and Software Technology	Computer Science, Computer Science Applications, Information Systems, Software	2,726
Symposium on Search Based Software Engineering	Conference	
International Symposium on Empirical Software Engineering and Measurement	Conference	
International Conference on Requirements Engineering	Conference	

4 Schedule of Activities

The schedule of activities is shown in Figure 1.

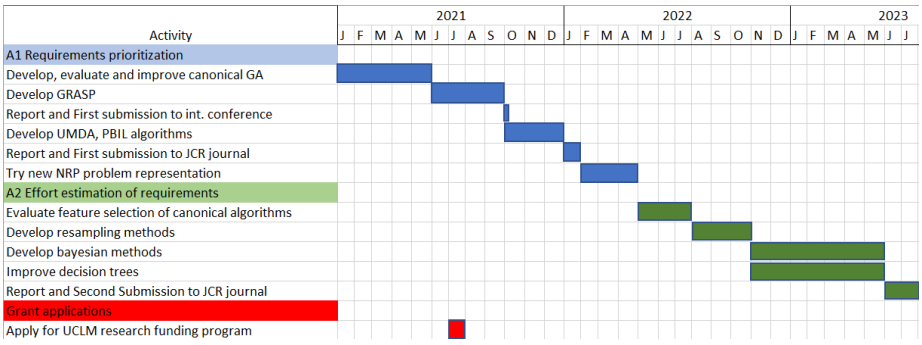


Fig. 1: GANTT diagram of activities.

5 Preliminary results

The specific aim A1 of the research has already started being tackled. To solve the requirements prioritization problem or NRP, an experimentation model has been designed. The experiments launched to evaluate each algorithm execute multiple times different combinations of the parameters, calculating final metrics that will be used to evaluate the performance of each one of the algorithms.

In the research, three algorithms have been designed to solve the NRP: a genetic algorithm designed by the authors, that stores a set of non dominated solution and uses a mono-objective metric as fitness function; the NSGA-II (Non Sorted Genetic Algorithm), a multi-objective algorithm frequently used to solve the NRP; and GRASP (Greedy Randomized Adaptive Search Procedure) algorithms.

The results obtained by the first two algorithms have already been analyzed, selecting the best configurations and comparing them to find their weaknesses and strengths (see Figures 2 and 3). The results were generated by applying the algorithms to two datasets [3] that represent instances of the NRP.

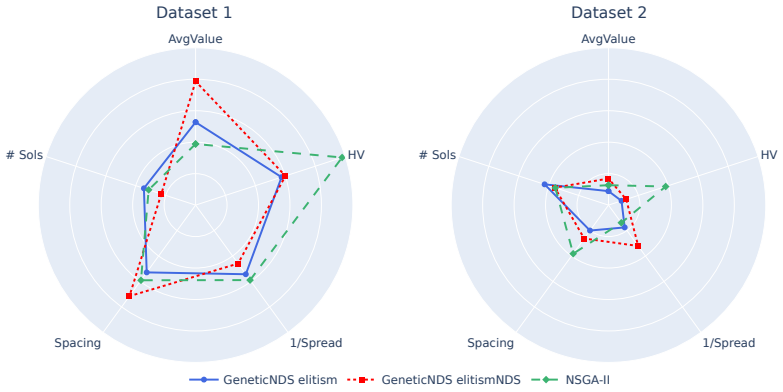
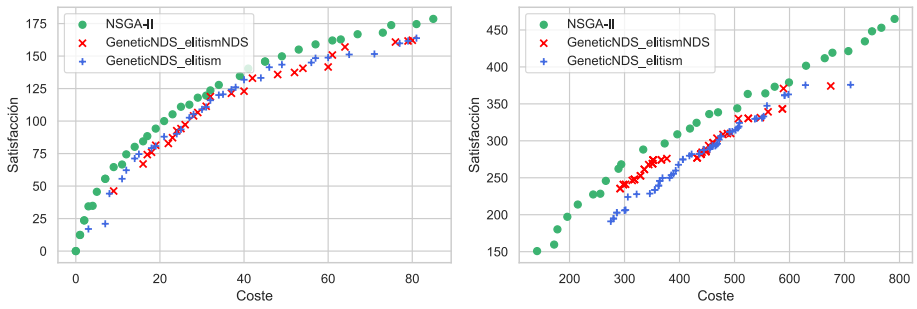


Fig. 2: Kiviat graph of metric comparison between algorithms.



(a) Results for Dataset 1

(b) Results for Dataset 2

Fig. 3: Optimal Pareto fronts of the best configurations of the algorithms analyzed.

References

1. Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J., Pedrycz, W.: Predicting development effort from user stories. In: International Symposium on Empirical Software Engineering and Measurement. pp. 400–403 (2011)
2. Bagnall, A.J., Rayward-Smith, V.J., Whittley, I.M.: The next release problem. Information and Software Technology **43**(14), 883–890 (dec 2001)
3. Chaves-González, J.M., Pérez-Toledano, M.A., Navasa, A.: Software requirement optimization using a multiobjective swarm intelligence evolutionary algorithm. Knowledge-Based Systems **83**(1), 105–115 (jul 2015)
4. Choetkiertikul, M., Dam, H.K., Tran, T., Pham, T., Ghose, A., Menzies, T.: A Deep Learning Model for Estimating Story Points. IEEE Transactions on Software Engineering **45**(7), 637–656 (2019)
5. Dantas, Emanuel; Perkusich, Mirko; Dilorenzo, Ednaldo; Perkusich, A.: Effort Estimation in Agile Software Development: an Updated Review. In: International Conference on Software Engineering and Knowledge Engineering (2018)
6. Durillo, J.J., Zhang, Y., Alba, E., Harman, M., Nebro, A.J.: A study of the bi-objective next release problem. Empirical Software Engineering **16**(1), 29–60 (feb 2011), https://link.springer.com/article/10.1007/s10664-010-9147-3
7. Sagrado, J., Águila, I., Orellana, F.: Multi-objective ant colony optimization for requirements selection. Empirical Software Engineering **20**, 577–610 (2015)
8. de Souza, J.T., Maia, C.L.B., Ferreira, T.d.N., do Carmo, R.A.F., Brasil, M.M.A.: An Ant Colony Optimization Approach to the Software Release Planning with Dependent Requirements. In: Cohen, M.B., Ó Cinnéide, M. (eds.) Search Based Software Engineering. pp. 142–157. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

Auditing Machine Learning Models for Online Educational Platforms

Roberta Galici[1][0000−0002−1825−0097]

Department of Mathematics and Computer Science,
University of Cagliari,
V. Ospedale 72, 09124 Cagliari, Italy
roberta.galici@unica.it

Abstract. Machine learning (ML) is transforming education and fundamentally changing teaching, learning, and research. In response to this situation, it is important to ensure that the machine learning models used in this context are accountable. However, the literature has generally focused on improving the accuracy of these models as much as possible,and only recently has started to investigate aspects such as fairness, transparency, privacy. A process of auditing is fundamental to ensure that existing models take care of these aspects, according to the perceptions of the users and the impact these models could have on them. In our work, we plan to create algorithms and web-based tools that can allow stakeholders to assess the accountability of their own machine learning models from multiple perspectives, enriching the evaluation framework and going beyond accuracy only evaluations. Our contributions can help to create and deploy responsible machine learning models for education.

Keywords: Data mining · Education · Educational Data Mining · Higher Education · Machine Learning.

1 Motivation

Machine learning is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems [4]. During the last decades the use of data mining is increasing a lot in many fields such as bioinformatics, marketing campaigns, cyber security, education and many more. In particular, educational data mining (EDM) is a research area that deals with the development of methods to explore data originating in an educational context [19]. EDM is defined by The Educational Data Mining community website[1], as "an emerging discipline, concerned with developing methods for exploring the unique types of data that come from the educational setting, and using those methods to better understand students, and the settings which they learn in". There are papers that cover the main aspects of data mining in education such as [1, 3, 17, 18].

[1] www.educationaldatamining.org

Educational machine learning is very important because it can be used for classifying and predicting students' performance [16], dropouts [21], score predictions [15] and also teachers' performance [2]. It can help educators to track academic progress to improve the teaching process, it can help students in course selection and educational management to be more efficient and effective. For instance, knowledge tracing [13] is the task of modelling student knowledge over time so that we can accurately predict how students will perform on future interactions. Dropout is an event that occurs when a student or more than one, decide to leave school or university. So it is very important to avoid this situation and possibly to inform the teacher of what is happening so that he can contact the student and try to solve the problems. Student success is a crucial component of higher education institutions because it is considered as an essential criterion for assessing the quality of educational institutions. Predicting such success is particularly important because it can help decision makers to provide the needed actions at the right moment, and to plan the appropriate training in order to improve the student's success rate. Another important way in which machine learning is used is with course recommendation system. In fact, based on students choices during the time, the system can suggest them courses similar to those they have already attended.

In my research project I aim to analyze the perceptions of stakeholders with respect to aspects such as fairness, transparency, security and privacy in the context of machine learning models for education. Based on the outcomes of these analyses, I than aim to improve these models such that they account for the above aspects. Specifically, I am interested in answering the following research questions:

1. How are educational machine learning models perceived from the point of view of fairness, security, privacy, transparency and safety?
2. How can we improve accountability of machine learning models in education with respect to the above perspectives?

2 Background

2.1 How is machine learning applied in education

Machine learning and education processes are closely interconnected. ML algorithms analyze how the students perceive and explore the information they are given. It helps the system to either draw the user back and go through some learning points again or let them step further. ML also helps the teachers to monitor and trace the learning process individually. As compared to traditional methods in classrooms, where the goal is to deliver the course, but not ensure that everyone got it, ML gives an advantage of more profound information perception. ML technologies analyze the content of online courses and help to figure out whether the quality of the offered information meets the applicable standards on one hand, and on the other, it shows how the users perceive the data and do they understand what is taught. There is another application of machine

learning in education that deals with scores and grading. As each online course reflects the learning abilities of a large number of students, they have a lot of experience in grading them.

The authors in [6] introduced EdNed, which is the dataset of all student-system interactions collected over 2 years by Santa, a multi-platform AI tutoring service with more than 780K users in Korea available through Android, iOS and web. The recorded behavior is limited to question-solving activities, so it is important to expand it. The next year they proposed Saint+ [20], a successor of the previous one. They planned to model not only students' problem-solving records, but also various learning activities, such as watching lectures and studying explanations for each exercise. In addition, they planned to explore architectures for knowledge tracing models other than transformer based encoder-decoder model that separately processes exercise information and student response information. Others developed DAS3H [5], a student model that explicitly accounts for memory decay and the benefits of practice when items can involve multiple skills at the same time. In particular, they need to improve the scalability of DAS3H model, design models that do not rely on specifically designed time windows, using for example self-exciting processes such as Hawkes processes.

In general, the above machine learning models are increasingly used in the real world and education is one of the many application domains where these models are becoming more and more pervasive. However, studies on machine learning models are those that measure the accuracy or how effective they are from a learning point of view, tend not to consider what stakeholders' perceptions in terms of perspectives like fairness and transparency. Some studies have focused their attention on fairness, which means ensuring that personal and social circumstances do not prevent students from achieving their academic potential. Nevertheless, machine learning models needs to be transparent, in other words, this means to teach while making obvious the intellectual practices involved in completing and evaluating a learning task. The goal of transparent teaching is to promote students' conscious understanding of how they learn.

2.2 Beyond accuracy aspects

Artificial Intelligence (AI) is becoming more and more important in the last decades and for this reason, many studies have been conducted. In [14] they introduce a framework for algorithmic auditing that supports artificial intelligence system development end-to-end, to be applied throughout the internal organization development lifecycle. The authors in [9] designed a survey to assess users' perceptions of search engine biases, with the goal of diagnosing the reasoning underneath their preferences of the real search pages or the synthesized pages. They also investigated the effects of bias-mitigation on users' satisfactions. They discover that participants prefer the real search pages over the synthesized ones with a significant higher ratio. The first systematic investigation of industry teams' challenges and needs for support in developing fairer ML systems was conducted by [10]. Even when practitioners are motivated to improve fairness in their products, they often face various technical and organizational barriers. One

of the most prominent issues in education policy today, accountability is a key element in the success of education improvement systems. In fact, accountability means holding everyone with responsibilities to high standards of performance while audits are tools for interrogating complex processes, often to determine whether they comply with company policy, industry standards or regulations.

3 Research plan

Many studies have been conducted in machine learning and in particular in education. The majority of researchers have studied fairness, privacy, security and all the others, but these perspectives have been under-explored on machine-learning models applied to education. In fact, it is important to know what users think and try to help them with the difficulties that they may have. I assumed to spend three months on research and one month for experiments with students. The process of auditing can be made for a specific field or more generic, so it is possible to include many fields. The purpose of my research is to fill this gap in the literature and therefore I plan to follow these three main steps.

Identification of case studies for ML models in education. The first step concerns on the identification of which machine learning models can be interesting to analyze the accountability. We have identified dropout prediction and educational recommender systems as the most interesting ones, but we are also open to add other models. At the beginning, it would be useful to analyze the dropping out of students for universities courses, then the probability of leaving courses. In particular, dropping out of school refers to abandoning one's schooling before getting an initial diploma. In our specific case, it means that a student decide to leave the course even before doing the exam. Dropping out is a complex and multifaceted phenomenon. Students are constantly influenced by various factors and when these have a negative influence, they are called risk factors. Risk factors increase the likelihood that a student will struggle at school or university, which can lead to dropping out. On the other hand, a recommender system is used to generate meaningful recommendations to a collection of users for items or products that might interest them. In education, it would be useful to recommend a specific course to a student, based on his or her previews choices. For example, if a student is studying Physics, the system will try to suggest something which is strictly related to that faculty. First we want to start with university data, e.g., from Moodle platforms, for dropout and then we plan to use large-scale datasets, such as COCO [7]. to create algorithms (e.g., recommender systems) and to make auditing with them. Moodle provides log data at the site and course level. This data can be accessed and downloaded as activity reports. You can see what pages the students accessed, the time and date they accessed it, the IP address they came from, and their actions (view, add, update, delete). On the other hand, COCO includes information collected from a leading global marketplaces for online learning and teaching at scale. In particular, Udemy enables experts in various areas to offer courses at no charge or for tuition fees.

Study of the ML model accountability. For each case study, identified in a previous step (dropout prediction or course recommendation), the second step consists in investigating which aspects we want to account for and why it is important to perform an accountability study on these aspects. The idea is to bring students together, so it can be possible to present them examples of machine learning models based on case studies and ask them to answer questions contained in questionnaires based on certain cases in which these models can be used in the real world. For example, considering the dropout prediction, we can have a ranking of students organized in two tables that may be at risk of failing or giving up the course. It might be useful to ask the teacher if there are problems with the two ranking. The teacher can notice that there are too male students, so the model is unfair, or maybe he can not understand why these students are at risk and can leave the course, so this is a lack of transparency. This step may include also interviews with developers, teachers and other stakeholders in education. The data will be collected through the questionnaires that will be distributed to the interested stakeholders. We will produce an article that describe each case of study, in particular, how to do accountability and what we found from our study. This is the most exploratory part. In a second time, given a case study, using state-of-the-art models we are going to make the accountability based on the guidelines developed in the previous step. The output will be conference papers and journal articles describing the accountability of existing systems for specific case studies such as the following ones [11, 12, 8].

Development of accountable ML models. Last, for the third step, we plan to make the considered machine learning models more responsible, based on the findings and guidelines defined in the previews step. Our goal is to provide definitions, algorithms and tools that allow educations stakeholders to assess the accountability of their systems in a simple way. It would be useful also to create a web-based interface own machine learning model and obtain an evaluation form covering different aspects important for accountability (related to fairness, transparency, social impact and others). One of the outputs could be a web-based tool, possibly disseminated as a demo paper.

References

1. Baker, R.S., Yacef, K.: The state of educational data mining in 2009: A review and future visions. JEDM— Journal of Educational Data Mining **1**(1), 3–17 (2009)
2. Bienkowski, M., Feng, M., Means, B.: Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. Office of Educational Technology, US Department of Education (2012)
3. Castro, F., Vellido, A., Nebot, A., Mugica, F.: Applying data mining techniques to e-learning problems. In: Evolution of teaching and learning paradigms in intelligent environment, pp. 183–221. Springer (2007)
4. Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., Wang, W.: Data mining curriculum: A proposal (version 1.0). Intensive Working Group of ACM SIGKDD Curriculum Committee **140**, 1–10 (2006)

5. Choffin, B., Popineau, F., Bourda, Y., Vie, J.J.: Evaluating das3h on the ednet dataset. In: AAAI 2021-The 35th Conference on Artificial Intelligence/Imagining Post-COVID Education with AI (2021)
6. Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., Baek, J., Bae, C., Kim, B., Heo, J.: Ednet: A large-scale hierarchical dataset in education. In: International Conference on Artificial Intelligence in Education. pp. 69–73. Springer (2020)
7. Dessì, D., Fenu, G., Marras, M., Recupero, D.R.: Coco: Semantic-enriched collection of online courses at scale with experimental use cases. In: World Conference on Information Systems and Technologies. pp. 1386–1396. Springer (2018)
8. Edge, K.: Educational assessment, evaluation and accountability. Educational Assessment, Evaluation and Accountability **24**(3), 173 (2012)
9. Han, B., Shah, C., Saelid, D.: Users' perception of search engine biases and satisfaction. arXiv preprint arXiv:2105.02898 (2021)
10. Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M., Wallach, H.: Improving fairness in machine learning systems: What do industry practitioners need? In: Proceedings of the 2019 CHI conference on human factors in computing systems. pp. 1–16 (2019)
11. Leithwood, K., Earl, L.: Educational accountability effects: An international perspective. Peabody Journal of Education **75**(4), 1–18 (2000)
12. Maroy, C., Pons, X.: Accountability policies in education. A Comparative and Multilevel Analysis in France and Quebec. Cham: Springer (2019)
13. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
14. Raji, I.D., Smart, A., White, R.N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., Barnes, P.: Closing the ai accountability gap: defining an end-to-end framework for internal algorithmic auditing. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 33–44 (2020)
15. Ramaswami, M., Bhaskaran, R.: A chaid based performance prediction model in educational data mining. arXiv preprint arXiv:1002.1144 (2010)
16. Rastrollo-Guerrero, J.L., Gomez-Pulido, J.A., Duran-Dominguez, A.: Analyzing and predicting students' performance by means of machine learning: a review. Applied sciences **10**(3), 1042 (2020)
17. Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert systems with applications **33**(1), 135–146 (2007)
18. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **40**(6), 601–618 (2010)
19. SHARMA, G.: Tendency of educational data mining in digital learning platform
20. Shin, D., Shim, Y., Yu, H., Lee, S., Kim, B., Choi, Y.: Saint+: Integrating temporal features for ednet correctness prediction. In: LAK21: 11th International Learning Analytics and Knowledge Conference. pp. 490–496 (2021)
21. Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., Ragos, O.: Implementing automl in educational data mining for prediction tasks. Applied Sciences **10**(1), 90 (2020)

Decision Support Systems for Business Model Development*

Sebastian Gottschalk

Software Innovation Lab, Paderborn University, Paderborn,Germany
sebastian.gottschalk@uni-paderborn.de

Abstract. The development of effective business models is an essential task for a company to stay successful. While in the past, much research has been focused on the conceptualizing of the parts of business models of a company (e.g., business aspects), less focus has been spent on the actual modeling of the business models (e.g., models aspects). Here, taking the modeling part more into account can support software tools to transfer from simple design tools for business modeling to decision-support systems to develop effective business models. In my thesis, I perform a design science study to develop an approach for the situation-specific development of business models. Based on the concept of situational method engineering, which is used to construct software development method based on the actual situation of the company, and feature models, which are used to store different product configurations in a single model, I develop an approach to construct and enact a business model development method based on the situation of the company. Moreover, I allow the provision of it-support for different development steps of the method. I apply my research to developing business models for mobile applications and evaluating it with a case and a user study.

Keywords: Business Model Development · Situational Method Engineering · Model Engineering · Service Provider · Software Ecosystem

1 Introduction

The development of effective business models, defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value" [16], is an essential task for a company to stay competitive. This is one of the results of the GE Innovation Barometer 2018 [7], a study with over 2000 business executives. In this study, 64% of these executives have the "difficulty to define an effective business model to support new ideas and make them profitable" [7]. An important reason for this is that customers want solutions for perceived needs rather than just product which affects that the business model can often

* My research is partially supported by the German Research Foundation (DFG) within the CRC "On-The-Fly Computing" (Project Number: 160364472SFB901) and the German Federal Ministry of Education and Research (BMBF) through "Software Campus" grant (Project Number: 01IS17046).

be more important than the latest technology of the product [5]. The development of those business models can be supported with so-called Business Model Development Tools (BMDTs). However, there existing lack of research in those tools [20] and also existing tools in practice have several limitations regarding their support in developing business models [19]. To break those limitations, a shift from design support to decision support of those systems has to be done [17], which also raises the questions of what parts of business model development could be automated with those tools [4].

In my research, I focus on one aspect that I called the situation-specific development of business models. To support the business model development, different domain experts propose various methods to develop such business models in the form of development processes (e.g. [15]) and method repositories (e.g. [3]). Moreover, these experts provide knowledge in the form of taxonomies of possible (e.g. [1]) and patterns of successful (e.g. [6]) companies. However, both the methods and the knowledge need to fit the situation of the company to support the development of effective business models. Otherwise, the development of an ineffective business model can lead to poor market penetration of the services or even a company bankruptcy [18]. Although various business model development approaches have been proposed, none of them provides fully-fledged tailoring to the service provider's situation. In my thesis, I address this problem by introducing a holistic tool-supported approach to support the situation-specific development of business models. For this, I use the concept of situational method engineering [13], which is used in software development to create development methods that fit the situation of the software project, to create the method repository and the development process. Moreover, I use the concept of feature models [2], which is used in product line engineering to model different variants of a product, to store knowledge about existing and patterns of successful business models. With both, I can support the business model development on the process (e.g., development method) and the product (i.e., business model) level.

The approach introduces the role of the *Meta-Method Engineer*, the *Method Engineer*, the *Domain Expert*, the *Business Developer* and other *Stakeholder*. In contrast to other business model development approaches, my approach points out the importance of the *Method Engineer* who formalizes the methods and knowledge to make them useable for the *Business Developer*. In my approach, the *Meta-Method Engineer* needs to create the meta-model to handle methods and knowledge once. Based on that, the *Method-Engineer* models existing methods and knowledge in repositories based on the experience of the *Domain Expert*. After that, the *Method-Engineer* constructs the development methodology out of both repositories based on the described situation of the *Business Developer*. Finally, the *Business Developer* enacts the method to develop his business model together with the *Stakeholder*. Here, different steps of the method can be supported with software modules. I implement the whole approach as a web-based open-source tool and evaluate it with a case study of creating a business model for a local event platform's mobile app and a user study of a lean development seminar at Paderborn University.

1.1 Research Approach

This thesis uses a design science research (DSR) process to build an approach for the situation-specific development of business models. We use DSR because it focuses on developing an artifact with the intention of a stepwise improvement using cycles. Here, the output in the form of evaluated results of a cycle is used as the next cycle's input. As a concrete method, we choose the DSR cycle of Kuchler and Vaishnavi [14]. The process is shown in Fig. 1 and consists of three cycles with the five steps of taking *Awareness of [the] Problem*, making *Suggestion* for the solution, the *Development* of a corresponding artifact, the *Evaluation* of our solution, and the drawing of a *Conclusion*.

In the *First Cycle* (2019-2020), we analyzed the current literature and software tools to understand the problem of software-based business model development. Based on that, we created conceptional parts for the situation-specific development of business models, implement them in software fragments, and evaluate the technical feasibility. Moreover, we have provided a tool review of decision support within the so-called Business Model Development Tools (BMDTs).

Currently, in the *Second Cycle* (2020-2021), we take the lessons learned from the last cycle and the tool review to create an integrated concept for the situation-specific development of business models. For this, we implement a software tool and evaluate it with a case study on creating the business model for the mobile app of a local event platform. Moreover, we use our existing tool review to create a reference architecture for BMDTs that researchers can use.

In the *Third Cycle* (2021-2022), we will take the lessons learned from the second cycle to create a modularized concept of situation-specific development of business models. The modularization will consist of a core architecture that different modules can extend to provide decision-support in various enacted pro-

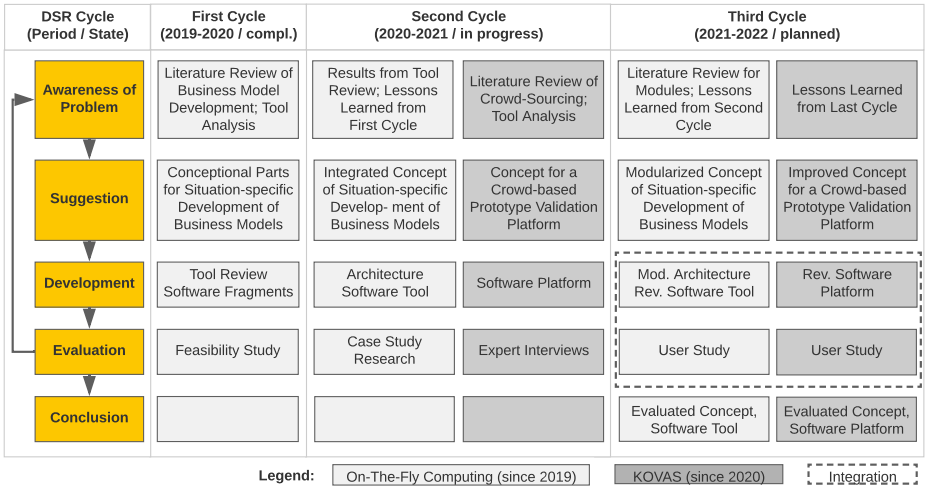


Fig. 1. Design Science Research Process based on Kuechler and Vaishnavi [14]

cess parts. Here, we will also review additional literature for modules we are developing. After implementing the core architecture and the modules, we will evaluate the approach with a user study. Finally, we conclude with an evaluated concept, a modularized architecture, and a software tool.

2 Current Research of First and Second Cycle

My current research can be divided into the following two parts: First, I am a researcher in the collaborative research center "On-The-Fly Computing"[1]. The research center is funded by the German Research Foundation (DFG) and has the goal to develop future it-markets in which software is composed out of essential services that are traded over worldwide accessible markets. Here, I research the development of business models for the essential services and the composing software. Second, I am a project manager in the Software Campus project "KOVAS"[2]. The project is funded by the German Federal Ministry of Education and Research (BMBF) and aims to allow researchers to manage their first project with their own budget. Here, I manage a project to allow the validation of service prototypes by using crowd-sourcing techniques. Based on both and with respect to DSR, I propose the situation-specific development of business models. Based on my research project, I first analyzed the business models of various software ecosystems to create a variability model of the ecosystem provider's business model, and its dependencies to the service providers and the users [9]. At this point, I also decided to focus my thesis on service providers and mobile app providers in particular.

In the *First Cycle* of DSR and with the focus of mobile app developers, I transferred the concept of software product lines to the topic of business model development. For that, I used feature models to store the business model information of various business models for mobile applications. Out of this, a concrete business model for a single business model can be derived as an instance of the feature model [10]. Moreover, I created a process to create and adapt those business models based on the conduction of experiments [11]. Here, I evaluated the feasibility of the approach with a tool implementation and the application of a usage scenario.

In the *Second Cycle*, I worked on the extensibility of the approach by concerning knowledge about methods and models from different domain experts. For that, I have used situational method engineering to derive a method repository with various methods to develop and validate business models for mobile applications [12]. Moreover, I have worked on an approach to consolidate the knowledge about business models from different domain experts [8]. For both, I have applied the case study of developing business models for a local event platform. For that, I have developed the BMDL Method Modeler[3] and the BMDL

[1] On-The-Fly Computing Project: https://sfb901.uni-paderborn.de
[2] KOVAS Project: https://www.sicp.de/en/projekte/kovas
[3] Method Modeler: https://sebastiangtts.github.io/bmdl-method-modeler/

Feature Modeler[4]. I currently integrate both solutions into each other to enact the development method and conduct the case study. For the *Crowd-based Service Prototype Validation*, I organized the development of a first set of design principles for a crowd-based prototype validation platform. For that, I lead an analysis of literature and existing tools in the areas of rapid prototyping, crowdsourcing, and business model development. Based on that, the design requirements for the validation were extracted and mapping to the design requirements of the platform. Moreover, I organized the conceptualizing and development of the platform together with its evaluation.

3 Future Research and Expected Contributions

In the *Third Cycle*, I will work on the extensibility of the approach to different mobile applications and the modularization of the concept. For that, I will use the concept of service-oriented architectures to provide additional it-support to specific steps of the development method. This could be, for example, the calculation of outcomes of different business models or the semi-automated competitor analyses of other services in the same store. To validate the applicability of the approach, I will conduct a user study within a lean development seminar at Paderborn University. Here, the students will use lean development techniques to develop a mobile application and use the developed tool for developing a corresponding business model. For the *Crowd-based Prototype Validation Platform*, I will organize the improvement of the design principles of the platform and the corresponding development. For that, I will focus on the deeper integration of experiments into the presented prototypes so that split-tests of different service prototypes could be performed effectively. Moreover, I will lead the integration of the platform as a module of the developed tool and an evaluation in the lean-development seminar.

By using DSR, I expected the evaluated concept, the modularized architecture, and the software tool as an output of my thesis. With this output, I aim to contribute to research and practice in the following way: First, the evaluated concept provides a new view of how SME could be applied to the business modeling domain. Second, the modularized architecture supports researchers with a reference model to build new BMDTs. Third, the software tool supports practitioners in developing effective business models for their services. Moreover, the crowd-based prototype validation platform should support practitioners in validating their services before the development to save development resources.

Acknowledgements

I want to thank Prof. Dr. Gregor Engels from the database and information systems group of Paderborn University for the guidance of my research and the supervision of my thesis.

[4] Feature Modeler: https://sebastiangtts.github.io/bmdl-feature-modeler/

References

1. Altmann, J., Ion, M., Bany Mohammed, A.A.: Taxonomy of Grid Business Models. In: Grid Economics and Business Models, vol. 4685, pp. 29–43. Springer (2007)
2. Apel, S., Batory, D., Kästner, C., Saake, G.: Feature-Oriented Software Product Lines. Springer, Heidelberg (2013)
3. Bland, D.J., Osterwalder, A.: Testing business ideas. John Wiley & Sons, Hoboken (2020)
4. Bouwman, H., de Reuver, M., Heikkilä, M., Fielt, E.: Business model tooling: where research and practice meet. Electronic Markets **30**(3), 413–419 (2020)
5. Chesbrough, H.: Business model innovation: it's not just about technology anymore. Strategy & Leadership **35**(6), 12–17 (2007)
6. Gassmann, O., Frankenberger, K., Csik, M.: The business model navigator: 55 models that will revolutionise your business. Pearson, Harlow (2014)
7. General Electric Inc: GE Global Innovation Barometer 2018, https://www.ge.com/reports/innovation-barometer-2018/
8. Gottschalk, S., Kirchhoff, J., Engels, G.: Extending Business Model Development Tools with Consolidated Expert Knowledge. In: Business Modeling and Software Design, vol. 422, pp. 3–21. Springer (2021)
9. Gottschalk, S., Rittmeier, F., Engels, G.: Business Models of Store-Oriented Software Ecosystems: A Variability Modeling Approach. In: Business Modeling and Software Design, vol. 356, pp. 153–169. Springer (2019)
10. Gottschalk, S., Rittmeier, F., Engels, G.: Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach. In: Software Business, pp. 192–207. Springer (2019)
11. Gottschalk, S., Rittmeier, F., Engels, G.: Hypothesis-driven Adaptation of Business Models based on Product Line Engineering. In: International Conference on Business Informatics (CBI). IEEE (2020)
12. Gottschalk, S., Yigitbas, E., Nowosad, A., Engels, G.: Situation-Specific Business Model Development Methods for Mobile App Developers. In: Enterprise, Business-Process and Information Systems Modeling, vol. 421, pp. 262–276. Springer (2021)
13. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: Situational Method Engineering. Springer, Heidelberg (2014)
14. Kuechler, B., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. Eur. J. Inf. Syst. **17**(5), 489–504 (2008)
15. McGrath, R.G.: Business Models: A Discovery Driven Approach. Long Range Planning (43), 247–261 (2010)
16. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
17. Osterwalder, A., Pigneur, Y.: Designing Business Models and Similar Strategic Objects: The Contribution of IS. Journal of the Association for Information Systems **14**(5), 237–244 (2013)
18. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business, USA (2014)
19. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. Electronic Markets **60**(11), 2794 (2019)
20. Veit, D., Clemons, E., Benlian, A., Buxmann, P., Hess, T., Kundisch, D., Leimeister, J.M., Loos, P., Spann, M.: Business Models: An Information Systems Research Agenda. Business & Information Systems Engineering **6**(1), 45–53 (2014)

The Power of Convex Relaxations in Combinatorial Optimisation

Caterina Viola

Department of Computer Science, University of Oxford, UK
caterina.viola@cs.ox.ac.uk

Abstract. Valued constraint satisfaction problems form a large and expressive class of combinatorial optimisation problems. The aim of my research project is the systematic study of the applicability of convex relaxations to the study of the computational complexity of the exact and approximated solvability of valued constraint satisfaction problems

Keywords: convex relaxations, combinatorial optimisation, constraint satisfaction, algorithms, computational complexity

1 Introduction

Convex relaxations are a powerful tool to design polynomial-time algorithms for exact and approximate solutions of combinatorial optimisation problems [7]. Convex relaxations have been studied in theoretical computer science and operational research [3] and they have been also successfully applied to several topics from artificial intelligence such as, for example, local clustering, community detection [6], segmentation [10], and 3D reconstruction [11]. To solve an instance of a minimisation problem (or a maximization problem), the instance is formulated as an integer program that is then relaxed to a convex program which can be solved in polynomial time, e.g., to a linear program or to a semidefinite program. The ratio $\frac{O}{F}$ of the optimum O of the original problem to the optimum F of the relaxed problem is called integrality gap (in a maximization problem the integrality gap is $\frac{F}{O}$). If the integrality gap is 1 then the solution to the relaxed problem is an exact solution to the original problem. Otherwise, an approximate solution to the problem can be obtained by designing a suitable polynomial-time algorithm that converts the solution to the relaxed problem into an integer one.

The better-known and most used convex relaxations are linear programming (LP) relaxations and semidefinite programming (SDP) relaxations. In LP relaxations, the original integer problem is relaxed to a linear program, i.e., an instance of an optimisation problem with variables that range over the reals and which can be interpreted as probability distributions on the possible solutions of the integers problem; while in SDP relaxations, the original integer problem is relaxed to a semidefinite program, i.e., an instance of an optimisation problem where the variables are vectors ranging over the reals and whose norms can be

interpreted as probability distributions on the possible solutions of the integers problem. SDP relaxations were pioneered by a result of Goemans and Williamson [8] who obtained the best possible approximation algorithm for Max-Cut. The importance of LP and SDP relaxations to obtain both theoretically interesting results and practical application relies upon the fact that LP and SDP can be solved in polynomial-time by interior-point methods (see, [9] and [15, **?**,**?**]), which also provide practically efficient algorithms. A convex relaxation can be gradually strengthened by adding local constraints which are satisfied by integer solutions, thus obtaining a hierarchy of convex relaxations. The most famous hierarchies of LP relaxations are those proposed by Sherali and Adams [16] and by Lovász and Schrijver [14], while a hierarchy of SDP relaxations was proposed by Lassere [13].

The objective of the research project is the systematic study of the power of convex relaxations. More precisely, the objective is to provide a precise characterization of the computational combinatorial optimisation problems which are exactly and approximately solved by a specific convex relaxation (e.g., an LP relaxation, an SDP relaxation, or a certain level of one of the mentioned hierarchies). A further goal of the project is to investigate the consequences of the power of specific convex relaxations on the computational complexity of valued constraint satisfaction problems.

Valued constraint satisfaction problems (VCSPs) are a wide class of computational optimization problems. The input of a VCSP instance consists of finitely many variables, an objective function, that is, a finite sum of cost functions defined on a fixed set called the domain and applied to some of the given variables, and a rational threshold. The computational task is to decide whether there exists an assignment of values from the domain to the variables whose cost, i.e., the value of the objective function corresponding to the assignment, is at most the given threshold.

Convex relaxations have been already applied to study the computational complexity of VCSPs both in the finite-domain and infinite-domain setting. In the finite-domain setting, the applicability of a basic LP relaxation and of a constant level of the Sherali-Adams relaxations for exact solvability of VCSPs was completely classified in [12] and [17], respectively. For infinite-domain VCSPs, a sufficient condition for the exact solvability via a basic linear programming relaxation was given in [2]. This result settled the polynomial-time tractability of VCSPs for specific classes of piecewise linear cost functions (e.g., for submodular piecewise linear homogeneous cost functions) whose complexity was not known before. Most recently, this result was generalised by a sufficient condition for the exact solvability via a combination of the basic linear programming relaxation and the affine integer programming (AIP) relaxation [18]. The AIP relaxation [4, 1] is a variation of the LP relaxation where the variables are allowed to take values in the set of all integers. In fact, the sufficient condition for the applicability of the combined basic LP and AIP relaxation also applies to a generalisation of VCSPs called promise valued constraint satisfaction problems (PVCSPs), and extended a result from [5]. For infinite-domain VCSPs, the combined basic LP

and AIP relaxation is strictly more powerful than the basic LP relaxation alone, in the sense that the combined relaxation exactly solves all the VCSPs that are exactly solved by the basic LP relaxation; and, furthermore, it settles the polynomial time tractability for a class of infinite-domain VCSPs that do not satisfies the sufficient condition for the applicability of the basic LP relaxation. The result on the combined basic LP and AIP relaxations, paves the way to the study of different combinations of two, or more, convex relaxations.

The aim of this project is to turn the study of exact solvability of VCSPs via a certain convex relaxation into a systematic approach and to extend it to the study of approximate solvability of VCSPs.

References

1. Barto, L., Bulín, J., Krokhin, A., Opršal, J.: Algebraic approach to promise constraint satisfaction. Journal of the ACM **68**(4) (Jul 2021), https://doi.org/10.1145/3457606
2. Bodirsky, M., Mamino, M., Viola, C.: Piecewise linear valued CSPs solvable by linear programming relaxation. to appear in ACM Trans. on Computational Logic (2021), preprint available at arxiv.org/abs/1912.09298. An extended abstract appeared in Proceedings of the 27th EACSL Annual Conference on Computer Science Logic (CSL) with the title Submodular Functions and Valued Constraint Satisfaction Problems over Infinite Domains
3. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
4. Brakensiek, J., Guruswami, V.: An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19). pp. 436–455. SIAM (2019), https://doi.org/10.1137/1.9781611975482.28
5. Brakensiek, J., Guruswami, V., Wrochna, M., Živný, S.: The power of the combined basic LP and affine relaxation for promise CSPs. SIAM Journal on Computing **49**, 1232–1248 (2020), https://doi.org/10.1137/20M1312745
6. Bühler, T., Rangapuram, S.S., Setzer, S., Hein, M.: Constrained fractional set programs and their application in local clustering and community detection. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 624–632. PMLR, Atlanta, Georgia, USA (17–19 Jun 2013), https://proceedings.mlr.press/v28/buhler13.html
7. Chlamtac, E., Tulsiani, M.: Convex relaxations and integrality gaps. In: Handbook on semidefinite, conic and polynomial optimization, pp. 139–169. Springer (2012)
8. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM **42**(6), 1115–1145 (1995), https://doi.org/10.1145/227683.227684
9. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4), 373–395 (1984), http://dx.doi.org/10.1007/BF02579150
10. Klodt, M., Steinbruecker, F., Cremers, D.: Moment constraints in convex optimization for segmentation and tracking. In: Advanced Topics in Computer Vision. Springer (2013), https://doi.org/10.1007/978-1-4471-5520-1$_8$

11. Kolev, K., Brox, T., Cremers, D.: Fast joint estimation of silhouettes and dense 3D geometry from multiple images. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(3), 493–505 (2012), https://doi.org/10.1109/TPAMI.2011.150
12. Kolmogorov, V., Thapper, J., Živný, S.: The power of linear programming for general-valued CSPs. SIAM Journal on Computing **44**(1), 1–36 (2015), http://dx.doi.org/10.1137/130945648
13. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM Journal on Optimization **11**(3), 796–817 (2001), https://doi.org/10.1137/S105262340036680
14. Lovász, L., Schrijver, A.: Cones of matrices and set-functions and 0-1 optimization. SIAM Journal on Optimization **1**, 166–190 (1989), https://epubs.siam.org/doi/pdf/10.1137/0801013
15. Nesterov, Y., Nemirovskii, A.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, Philadelphia (1994), https://epubs.siam.org/doi/abs/10.1137/1.9781611970791
16. Sherali, H.D., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM Journal on Discrete Mathematics **3**(3), 411–430 (1990), https://doi.org/10.1137/0403036
17. Thapper, J., Živný, S.: The power of Sherali–Adams relaxations for general-valued CSPs. SIAM Journal on Computing **46**(4), 1241–1279 (2017), http://dx.doi.org/10.1137/16M1079245
18. Viola, C., Živný, S.: The Combined Basic LP and Affine IP Relaxation for Promise VCSPs on Infinite Domains. ACM Trans. Algorithms **17**(3) (Jul 2021), https://doi.org/10.1145/3458041

Providing Decision Makers with Tailored Decision Support Systems

Jonas Kirchhoff *

Software Innovation Lab, Institute of Computer Science, Paderborn University, Germany
jonas.kirchhoff@upb.de

Abstract. Decision makers show an increased need for decision support due to the rising volatility, uncertainty, complexity and ambiguity in business environments. In this research statement, I demonstrate that decision support provided by decision support systems must be tailored to individual decision makers for optimal decision making. This includes adapting the decision support system to the goal of a decision maker and resources available to them such as datasets, software tools or time to identify an optimal decision. Since this individualization is not yet sufficiently addressed by existing systems, I present a solution concept that enables the creation of decision support systems which are tailored to individual decision makers. The concept utilizes ideas from Situational Method Engineering and Data Ecosystems. I furthermore explain my research approach to realize the solution concept.

Keywords: Adaptive Decision Support System · DSS Generator · Situational Method Engineering · Data Ecosystem · Energy Distribution Network Planning

1 Motivation

I am currently working as a researcher in the research project *FlexiEnergy*[1], in which PhD students from three disciplines and five industry partners collaborate to design and develop a decision support system for the cross-sectoral planning of energy distribution networks. Energy distribution networks are regional networks responsible for transporting generated energy to residential or industrial energy consumers. Distribution Network Operators (DNOs) must decide on when, where and how to reinforce or expand their network over the next decades. When deciding for or against investments, DNOs – like many decision makers – must consider frequent, unpredictable change in many influencing factors with unknown cause-effect relationships [7,1,8]. In order to make investment decisions despite these challenging circumstances, DNOs heavily rely on assistance provided in the form of decision support systems (DSS).

* Funding provided by European Regional Development Fund as part of the research project *FlexiEnergy* (grant number: EFRE-0801186).

[1] https://flexi-energy.de/en

During the design and development of a DSS for energy distribution network planning in the context of our research project, we encountered difficulties when trying to define a holistic planning process to be supported by the DSS. We could attribute these difficulties to the fact that the optimal planning process varies between DNOs due to situational factors. For instance, a DNO might require redundancy in their network to support the failure of a single asset like a transformer or a cable, while another DNO does not require such extra reliability. If the reliability requirements of DNOs do not align with the planning process implemented in the DSS, the proposed network investments are either too costly due to the included redundancy, or the resulting network does not fulfill reliability requirements. Similarly, we learned that load forecasts considering consumer technologies such as PV-systems adopted by individual buildings are more accurate than other approaches. However, a forecast on the level of individual buildings requires many socio-economic information about the buildings which are not readily available for each DNO. Consequently, the holistic planning process supported by the DSS likely needs to be extended with activities for additional data acquisition or activities need to be replaced with approaches whose data requirements are fulfilled.

While exchanging experiences with colleagues working in other research projects, I found that a "one-size-fits-all" decision support approach can also be suboptimal in other domains, e.g., during supply chain planning [14] or when deciding on a business model [4]. Instead, a decision support system must consider the context of decision makers, i.e., their overall goal and available resources such as data sets, software tools or time to identify an optimal decision. I therefore want to address the following research question in my dissertation: *How can individualized decision support systems be provided that are tailored according to the context of individual decision makers, i.e., their individual goals and available resources?*

2 Related Work and Research Gap

After identifying the need for decision support systems which are tailored to individual decision makers, I identified existing approaches in the decision support domain which seemed to address a related problem.

Adaptive DSS (ADSS) are decision support systems which "support human decision making judgements by adapting support to the high-level cognitive needs of the users, task characteristics, and decision contexts" [3]. However, these systems only seem to consider the selection of a single model across a (fixed) set of models, which renders them unsuitable for complex planning problems like distribution network planning where multiple models need to be combined, e.g., for load forecasting, network simulation and network optimization. Furthermore, the availability of resources like data sets or time to identify a decision is not included in the decision context and therefore not considered during adaptation.

DSS Generators are "a set of tools to support the design and construction of a [...] DSS" [9]. However, I did not find any general-purpose DSS generators

which can be utilized in any application domain, in particular energy distribution network planning. Furthermore, the generators are usually not meant to be directly used by decision makers but by trained "implementors", which can add a significant delay between the decision maker requiring support and an actual DSS being available.

In the domain of computer science, there are two particular approaches which are partially related to tailored decision support systems:

Situational Method Engineering aims to "design, construct and adapt methods, techniques and tools for systems development [...] for a specific situation" [5]. The output however is a method for developing a system which is enacted by the development team, not an actual DSS which can be used by a decision maker [2].

Data Ecosystems are "a set of networks composed by autonomous actors that directly or indirectly consume, produce or provide data and other related resources (e.g., software, services and infrastructure)" [12]. However, contrary to the definition, many data ecosystems (or even "decision support ecosystems" [13]) only focus on providing and exchanging data and ignore related resources, which nullifies the opportunity to create a holistic DSS.

In conclusion, I did not find any existing approach which sufficiently addresses decision makers' demands for tailored decision support systems.

3 Solution Concept

Although none of the approaches presented in the previous section fully addresses the demand of decision makers for tailored decision support systems, all approaches include concepts which contribute towards this goal. I therefore derived a solution concept from these approaches with the aim to combine their advantages while improving on their downsides. The solution concept is presented in Fig. 1 and subsequently explained in detail following the depicted three phases.

The initial phase is the *Context Identification* phase. During this phase, the *Decision Maker* interacts with the *Context Module* to document the decision context, i.e., the decision to be made and the available resources. For the *Context Module* to know which aspects to prompt the *Decision Maker* for, a *Domain Expert* must provide a *Domain Ontology* describing decision problems in the relevant domain. This documentation of the domain's characteristics may also include information about frequently analysed decision problems. Based on the feedback provided by the *Decision Maker*, the *Context Module* produces a machine-readable *Decision Context Formalization* of the decision problem and resources.

The *Decision Context Formalization* is subsequently used in the *Composition Module* of the *Decision Support Composition* phase. The goal of this phase is to assemble multiple *Decision Services* into a *Decision Support Composition* which represents an ideal planning process for the decision context of the *Decision Maker*. For this purpose, multiple *Decision Service Provider* provide a *Decision*

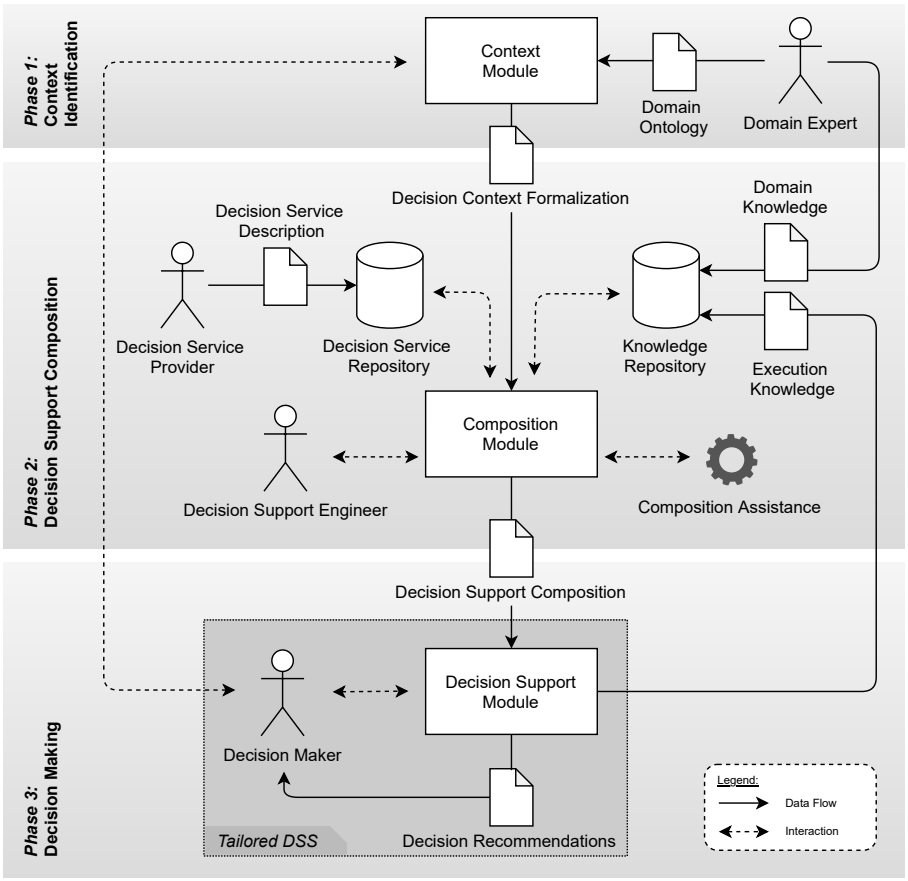


Fig. 1. Derived solution concept for providing tailored decision support systems

Service Description of their available decision services (e.g., software, data or compute infrastructure) in a *Decision Service Repository*. The composition of decision services can be performed manually by a *Decision Support Engineer* or partially/fully automated by a *Composition Assistance* (see [11] for the relevance of such assistance). A *Knowledge Repository* provides information to identify ideal compositions based on *Domain Knowledge* provided by *Domain Experts* or experience gathered during previous service executions (*Execution Knowledge*). It is important to note that a single person can optionally act in multiple roles, e.g., a *Decision Maker* may be involved in service composition as a *Decision Support Engineer* or provide their own datasets or other decision services as a *Decision Service Provider*.

Finally, in the *Decision Making* phase, the *Decision Support Module* invokes the decision services as described by the previously assembled *Decision Support Composition*. For this purpose, additional run-time input by the *Decision Maker*

might be necessary. The resulting *Decision Recommendations* are forwarded to the *Decision Maker*. From a a *Decision Maker*'s point of view, the tailored DSS behaves similar to a traditional DSS during this phase.

4 Research Approach and Next Steps

My research approach can be aligned with Hevner's three cycle view of design science research [6] shown in Fig. 2. With respect to the *Relevance Cycle*, I already derived the need for tailored decision support systems from the application domain of energy distribution network planning (cf. Sect. 1). I also derived some high-level *Requirements* and furthermore evaluated existing *State-Of-The-Art* artifacts, methods and concepts (cf. Sect. 2), thereby partially addressing the *Rigor Cycle*. The *Design & Development* of the *Design Cycle* is partially addressed with the solution concept presented in the previous Sect. 3.

Next, I will focus on developing the details of the solution concept, i.e., the modules, documents, repositories and roles shown in Fig. 1. For this purpose, I plan to apply a design science research approach as shown in Fig. 2 to the individual phases of the solution concept. Again, I can utilize the application domain of energy distribution network planning and existing knowledge such as methods for requirements engineering, service composition and service orchestration. Working on the individual phases of the solution concept already provides a partial *Evaluation* of the overall solution concept.

Lastly, I want to prototypically implement the solution concept and conduct a *Field Test* by using the prototype in the application domain of energy distribution network planning through my research project. Based on the completed relevance, design and rigor cycle for both the overall concept and its individual phases, I expect to contribute design principles [10] for tailored decision support systems as an *Addition To Knowledge*.

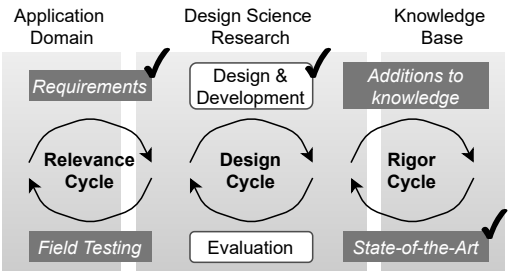


Fig. 2. Research approach (adapted from [6]) including progress

Acknowledgements I want to thank Prof. Dr. Gregor Engels and Dr. Christoph Weskamp of Paderborn University for their guidance in my dissertation project.

References

1. Bennett, N., Lemoine, G.J.: What a difference a word makes: Understanding threats to performance in a VUCA world. Business Horizons **57**(3), 311–317 (2014)
2. Fazal-Baqaie, M., Engels, G.: Software processes management by method engineering with MESP. In: Kuhrmann, M., Münch, J., Richardson, I., Rausch, A., Zhang, H. (eds.) Managing Software Process Evolution. Springer International Publishing (2016)
3. Fazlollahi, B., Parikh, M.A., Verma, S.: Adaptive decision support systems. Decision Support Systems **20**(4), 297–315 (1997)
4. Gottschalk, S., Kirchhoff, J., Engels, G.: Extending business model development tools with consolidated expert knowledge. In: Shishkov, B. (ed.) Business Modeling and Software Design (BMSD 2021). Lecture Notes in Business Information Processing, vol. 422, pp. 3–21. Springer International Publishing (2021)
5. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: State-of-the-art review. Journal of Universal Computer Science **16**(3), 424–478 (2010)
6. Hevner, A.R.: A three cycle view of design science research. Scandinavian Journal of Information Systems **19**(2), 87–92 (2007)
7. Kirchhoff, J., Burmeister, S.C., Weskamp, C., Engels, G.: Towards a decision support system for cross-sectoral energy distribution network planning. In: Breitner, M.H., Lehnhoff, S., Nieße, A., Staudt, P., Weinhardt, C., Werth, O. (eds.) Energy Informatics and Electro Mobility ICT (2021)
8. Mack, O., Khare, A.: Perspectives on a VUCA world. In: Mack, O., Khare, A., Krämer, A., Burgartz, T. (eds.) Managing in a VUCA World, pp. 3–19. Springer International Publishing, Cham (2016)
9. Maturana, S., Ferrer, J.C., Barañao, F.: Design and implementation of an optimization-based decision support system generator. European Journal of Operational Research **154**(1), 170–183 (2004)
10. Möller, F., Guggenberger, T.M., Otto, B.: Towards a method for design principle development in information systems. In: Hofmann, S., Müller, O., Rossi, M. (eds.) Designing for Digital Transformation. Co-Creating Services with Citizens and Industry. pp. 208–220. Springer International Publishing, Cham (2020)
11. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P., Jézéquel, J.M., Kühn, T., Mosser, S., Sahraoui, H., Syriani, E., Varró, D., Weyssow, M.: Opportunities in intelligent modeling assistance. Software and Systems Modeling **19**(5), 1045–1053 (2020)
12. Oliveira, M.I.S., Lóscio, B.F.: What is a data ecosystem? In: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age. dg.o '18, Association for Computing Machinery, New York, NY, USA (2018)
13. Stănescu, I.A., Ştefan, A., Filip, F.G.: Cloud-based decision support ecosystem for renewable energy providers. In: Camarinha-Matos, L.M., Baldissera, T.A., Di Orio, G., Marques, F. (eds.) Technological Innovation for Cloud-Based Engineering Systems. pp. 405–412. Springer International Publishing, Cham (2015)
14. Weskamp, C., Koberstein, A., Schwartz, F., Suhl, L., Voß, S.: A two-stage stochastic programming approach for identifying optimal postponement strategies in supply chains with uncertain demand. Omega **83**, 123–138 (2019)

Smart Contracts Classification and Vulnerabilities Detection

Giacomo Ibba

University of Cagliari, Department of Mathematics and Computer Science, Cagliari, Italy
giacomo.ibba14@studenti.unica.it

Abstract. Blockchain technology has become one of the most popular computer science trends, and in particular smart contracts programming has spread very quickly in the last years. One of the main problems with smart contracts is due to the identified vulnerabilities over the years. Honeypots in the smart contracts programming could cause the loss of a large amount of Ether, such as the DAO attack of the 18th June of 2016, in which attackers succeed in draining an amount of 3.6m Ether. Since then, the Solidity programming language has been widely improved, but nowadays, it is still possible to write vulnerable contracts. For this very reason, it is fundamental to detect and report an anomaly immediately by using machine learning techniques. Much work has been donein the smart contracts vulnerabilities detection field; indeed, different existing tools allow detecting smart contracts vulnerabilities. The problem with these tools is that they are limited to a subset of vulnerabilities and don't suggest how to re-write code to avoid the exposure itself. The research aims to build a tool integrating innovative machine learning techniques for detecting all known exposures and automatic code refactoring techniques to remove the possibility of the detected anomaly occurring.

Keywords: Blockchain · Smart Contract · Ethereum · LDA · Smart Contracts Vulnerabilities · Smart Contracts Trends.

1 MOTIVATION

In the last few years, developers and programmers started to take advantage of the blockchain's inner properties for several purposes. Traceability, Certification, Document Management are only a few of the use cases for which programmers want to exploit the advantages of the **Ethereum** blockchain and **smart contracts**. Smart contracts allow the execution of transactions without the help of third parties. Therefore, most of them are money transfer programs, token exchange [11], and games whose aim is to win Ether. Exploring the chain is possible to notice several sponsored contracts that hold Ether for a value of several millions of dollars. A security leak in such programs could be hazardous, leading to an irreversible loss of Ether [9], and for this reason, it is necessary to avoid writing **vulnerable** smart contracts. The most famous attack is the 'DAO Attack' [10] that caused an Ether loss for a value of 60M of dollars, and it is the perfect example of how a smart contract vulnerability could lead to dire consequences. Developers spotted several vulnerabilities that can lead smart contracts to unwanted behavior [2], but some are worse than the others in terms of potential Ether loss. For this

very reason, it is fundamental to detect possible dangerous patterns in the smart contract's code that may arise vulnerabilities. The research project aims to build a machine learning model able to recognize vulnerable smart contracts and detect the possible attack or problem related to the vulnerability itself. Since there are already existing tools allowing the recognition of vulnerable Solidity contracts, the ambition of the research is to build a machine learning model able to recognize all the already known vulnerabilities. After detecting the hypothetic vulnerability, the model re-writes the code such that the model removes the exposure. The research questions that focused my interest on this research are:

1. Is it possible to build a machine learning model able to classify smart contracts correctly?
2. Is it possible for the model to spot any of the already known vulnerabilities?
3. What are the principal vulnerabilities to which smart contracts are subject in 2021?
4. Is it possible to re-write the code such that the smart contract's vulnerability can be disabled maintaining all the original contract functionalities?

The most challenging part is the 4th task on the list since an automatic refactoring of the code is not trivial, and we must consider that smart contract code execution requires gas consumption. The new pattern will probably need a higher consumption of Ether to execute the code, so we need to consider techniques allowing to save gas.

2 STATE OF ART

The detection of vulnerable patterns in Ethereum smart contracts is a task of fundamental importance [16] since several contracts are handling an amount of Ether for a value of millions of dollars. A spotted vulnerability in such contracts could cause catastrophic consequences, as in the DAO attack. Therefore, automatic techniques for vulnerable patterns detection inside Solidity contracts are more than necessary. Researchers carried out several surveys about smart contracts' vulnerabilities [13], leading to unexpected loss of Ether or anomalous program behavior. Two of the most dangerous vulnerabilities are **reentrancy** and **denial of service** (Dos) since both lead to an Ether loss (in particular, an attacker used the reentrancy attack against DAO). Reentrancy consists of exploiting the peculiarity of Ethereum smart contracts of utilizing the code of other external contracts; an attacker can hijack these external calls whereby they force the program to execute malicious code. This attack can occur, for example, when a contract sends Ether to an unknown address, and the name of the vulnerability itself comes from the call back of a function on the vulnerable contract from the malicious program. Another critical vulnerability is the DoS [14] one, which consists of leaving a smart contract inoperable for a short period or, in some cases, permanently. Contracts inoperability can lead to a permanent loss of Ether since they can get trapped forever. The main problem is that there are several ways an attacker can make a contract inoperable, like looping through externally manipulated mapping or arrays or by allowing only the owner of the smart contract to execute some operations. Another common vulnerability regards arithmetic **overflows** and **underflows** since the Ethereum Virtual Machine specifies fixed-size data types for integers values. Therefore a variable of this type only

has a specific range of values it can represent. An arithmetic overflow or underflow occurs when an operation requires a fixed size variable to store a number (or piece of data) outside the range of the variable's data type. This kind of vulnerability allows attackers to create unexpected logic flows. Those listed above are only a few of the vulnerabilities spotted during years:

- Unexpected code execution due to the DELEGATECALL opcode.
- Unexpected loss of Ether due to improper use of the suicide/selfdestruct function
- Block timestamps manipulation.
- Improper iterations over array or mapping with a big amount of elements.

Several existing tools allow analyzing and detecting Solidity contracts vulnerabilities. *ReGuard* [8] parses a smart contract code to an intermediate representation and then performs a source-to-source transformation from the intermediate representation to C++. As output, the tool returns a bug report with all the possible patterns of reentrancy. Osiris [1] combines symbolic execution and taint checking to find integer bugs in Ethereum smart contracts accurately. *ContractFuzzer* [7] generates fuzzing inputs based on the ABI specifications of smart contracts, defines test oracles to detect security vulnerabilities, instruments the EVM to log smart contracts runtime behaviors, and analyzes these logs to report security vulnerabilities. Other existing tools help to detect Ethereum contracts vulnerabilities [3] [17] [4], but most of them are built to check a specific exposure or only a subset of them. Therefore, this research aims to carry out a machine learning-based tool to detect and find any known vulnerabilities. If a vulnerable pattern is spotted, the model should correct the code such that the code is not exposed anymore.
For what concerns the protection and correction of vulnerable smart contracts, only a few tools exist. *ÆGIS* [6] avoids exploitation of already deployed smart contracts reverting transactions in real-time based on pattern matching; these patterns encode the detection of malicious transactions that trigger exploits or scams, but it doesn't detect smart contracts vulnerabilities. *SmartShield* [18] is a bytecode rectification system that helps developers to carry out secure Solidity contracts, in particular, by checking three typical insecure patterns, and it also helps to write gas-optimized contracts. Although this last tool outclasses the existing ones, the focus is on a subset of vulnerabilities, so the problem persists.
Apart from this last point, we should consider that most contracts analyzed by these tools are dated back to 2019 or earlier, so, despite the Solidity programming language increased the security of smart contracts thanks to new pragmas. Nevertheless, several contract types are programmed using old pragmas versions of Solidity, especially Token, ICO, and Crowd-Funding programs; using these old versions of the pragma could easily lead to vulnerable patterns inside the Solidity contract. Accordingly, a scraping of 2020 and 2021 contracts is necessary, both for statistics and test purposes.

3 EXPECTED RESEARCH METHODOLOGY

We aim to collect a dataset of smart contracts that are large enough to contain many vulnerable programs. Many existing datasets could be helpful; the one that could be the most useful is *Smart Bugs* [5], a dataset of over 47.000 vulnerable contracts, and could

be the primary source of our dataset. Another interesting dataset is *Smart Corpus* [12], a dataset that contains about 10.000 contracts. The contracts included in this dataset were programmed using *pragma* versions between 0.4.x and 0.6.x that are known to be the versions allowing programmers to write vulnerable code effortlessly. By combining these two datasets, we have a significant number of contracts. Still, we must consider the possibility of duplicated programs between the two datasets, so we must drop any possible refuse before proceeding with the analysis. Another relevant problem consists of the date on which these contracts date back. Indeed, most of these contracts date back between 2017 and 2019, so the programs could be helpful to train the model in recognizing vulnerable patterns, but it doesn't give any information about the principal vulnerabilities to which current smart contracts are subject. Therefore, we need to add more contracts to our dataset, but they must be contracts from June 2021 to June 2020 to have updated statistics about how many vulnerable contracts we can find nowadays, what type of contract they are, and what is or what are the possible vulnerabilities to which the contract are exposed.

4 SMART CONTRACTS CLASSIFICATION

Once the smart contracts dataset is collected, the next step is to classify any program populating the dataset itself. The classification adds value to the work in two ways: the first one concerns statistics about smart contracts, especially for those written in the last year. The second aspect focuses on the creation of subsets of the dataset based on the classification. For example, any smart contract classified as 'Token' will be part of a 'Token Smart Contracts' dataset. Datasets of specific contracts are significantly essential to study different possibilities of building the same type of program, how the programming methodology of a particular contract kind changed over the years, and the types of contracts most present in the blockchain. An important aspect to focus on is the selection of features for the smart contracts dataset. The intention is to join numerical, continuous, and text attributes. The numerical features will include smart contract metrics [15] of interest for the classification, such as the number of events, the number of contracts defined inside a single program, the number of payable functions, etc. Continuous features include the contract's balance, the total number of transactions, and the highest and the lowest balance. In contrast, text features include:

- Source code.
- Application Binary Interface (ABI).
- Opcode.
- Information retrieved from transactions.

We can achieve the categorization of the dataset's samples by following two different paths. The first one involves using clustering techniques to group the contracts into clusters, such that we can categorize the different typologies of smart contracts. The second one consists of checking a sufficient sample of contracts manually, then label the selection, and finally, train a model to classify the remaining dataset's population.

5 VULNERABILITIES DETECTION AND CORRECTION

Once the contracts are classified, the next step is to look for vulnerable smart contracts, which is the core of the research. There are already existing tools that allow to label and analyze vulnerable Solidity programs. Still, they analyze only a subset of vulnerabilities, and primarily the only current tool for automatic correction fixes three security-related bugs that are state changes after external calls. It also analyzes missing checks for out-of-bound arithmetic operations and missing checks for failing external calls. These kinds of bugs can lead to reentrancy attacks, denial of services, and arithmetic overflows and underflows. Still, there are other vulnerabilities to detect and correct, so the ambition is to build a model to catch any known vulnerabilities and re-write the code to delete the exposure and prevent eventual attacks. The techniques for finding vulnerable patterns are various and include semantic extraction of patterns from abstract syntax trees, dependencies graphs, pattern recognition, and transactions monitoring. By combining these techniques, we should find vulnerable code successfully, but we must study how to optimize the detection and automatic correction process.
Two challenges arise from automatic code re-writing. The first one is the most obvious: the code re-writing aims to reformat the code to remove the vulnerable pattern, but the program must maintain its execution flow after the process. In other terms, the program must perform each of the original functionalities after the vulnerability removal. The second one is slightly subtle, and it concerns gas consumption. Re-writing the code could lead to a new pattern that requires a higher consumption of gas and, therefore, a higher Ether consumption. A user would like that is new not-vulnerable code doesn't need to consume too much respect to the previous version of the contract, but in some cases, it is not possible to maintain the same gas consumption. Consequently, it is critical to follow general guidelines for code writing so that the gas consumption (and Ether consumption) is optimized.

References

1. Osiris: Hunting for Integer Bugs in Ethereum Smart Contracts (12 2018). https://doi.org/10.1145/3274694.3274737
2. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on ethereum smart contracts (sok). In: Maffei, M., Ryan, M. (eds.) Principles of Security and Trust. pp. 164–186. Springer Berlin Heidelberg, Berlin, Heidelberg (2017)
3. Di Angelo, M., Salzer, G.: A survey of tools for analyzing ethereum smart contracts. In: 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON). pp. 69–78. IEEE (2019)
4. Feist, J., Grieco, G., Groce, A.: Slither: A static analysis framework for smart contracts. In: 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). pp. 8–15 (2019). https://doi.org/10.1109/WETSEB.2019.00008
5. Ferreira, J.F., Cruz, P., Durieux, T., Abreu, R.: Smartbugs: A framework to analyze solidity smart contracts. In: 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 1349–1352 (2020)
6. Ferreira Torres, C., Baden, M., Norvill, R., Jonker, H.: Ægis: Smart shielding of smart contracts. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2589–2591 (2019)

7. Jiang, B., Liu, Y., Chan, W.: Contractfuzzer: Fuzzing smart contracts for vulnerability detection. In: 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). pp. 259–269. IEEE (2018)
8. Liu, C., Liu, H., Cao, Z., Chen, Z., Chen, B., Roscoe, B.: Reguard: Finding reentrancy bugs in smart contracts. In: 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion). pp. 65–68 (2018)
9. Liu, J., Liu, Z.: A survey on security verification of blockchain smart contracts. IEEE Access **7**, 77894–77904 (2019)
10. Mehar, M.I., Shier, C., Giambattista, A., Gong, E., Fletcher, G., Sanayhie, R., Kim, H.M., Laskowski, M.: Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack. J. Cases Inf. Technol. **21**, 19–32 (2019)
11. Norvill, R., Fiz, B., State, R., Cullen, A.: Standardising smart contracts: Automatically inferring erc standards. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 192–195 (2019). https://doi.org/10.1109/BLOC.2019.8751350
12. Pierro, G.A., Tonelli, R., Marchesi, M.: Smart-corpus: an organized repository of ethereum smart contracts source code and metrics (2020)
13. Praitheeshan, P., Pan, L., Yu, J., Liu, J., Doss, R.: Security analysis methods on ethereum smart contract vulnerabilities: A survey (2020)
14. Samreen, N.F., Alalfi, M.H.: Smartscan: An approach to detect denial of service vulnerability in ethereum smart contracts (2021)
15. Tonelli, R., Destefanis, G., Marchesi, M., Ortu, M.: Smart contracts software metrics: a first study (2018)
16. Wang, S., Zhang, C., Su, Z.: Detecting nondeterministic payment bugs in ethereum smart contracts. Proceedings of the ACM on Programming Languages **3**(OOPSLA), 1–29 (2019)
17. Wang, W., Song, J., Xu, G., Li, Y., Wang, H., Su, C.: Contractward: Automated vulnerability detection models for ethereum smart contracts. IEEE Transactions on Network Science and Engineering (2020)
18. Zhang, Y., Ma, S., Li, J., Li, K., Nepal, S., Gu, D.: Smartshield: Automatic smart contract protection made easy. In: 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 23–34 (2020). https://doi.org/10.1109/SANER48275.2020.9054825

Group Recommender System: a Conversational Approach to Support Group Discussion

Hanif Emamgholizadeh[1][0000−0002−7159−3574]

Free University of Bolzano, Italy hemamgholizadeh@unibz.it

Abstract. With the widespread usage of internet, information overload has become a critical issue for internet users. In this scenario, recommender systems, which are tools that provide relevant suggestions about items that individual users could like to use, have been introduced. In some cases, the searched items are meant to be consumed by a group of users, e.g., a holiday package or a movie to watch. Hence, also group recommender systems (GRS), which are aimed at identifying items that can suit all the group members, have been developed. Some GRSs address this problem by constructing a group profile, i.e., a model that summarizes the preferences of the group by combining the preferences of the group members. By using this group, profile recommendations for the group are identified. But, a group decision, which should be supported by the GRS, is often made after a discussion, where the group members' preferences may evolve in a dynamic way. To model such a dynamic decision-making process, new GRSs have been introduced.
In my Ph.D. research, I am trying to improve the state-of-the-art approaches for modeling group decision-making dynamics by considering the social and personal aspects of the group members. This model can be utilized to anticipate the group choice, which can then be used to support the group members in improving their decision-making process. This support can be offered by providing more information about the item that is predicted to be the group's choice, providing new items similar to the group choice, etc.

Keywords: Recommender Systems · Group Recommender Systems · Group Dynamics · Food Recommender Systems

1 Introduction

Information overload has made recommender systems a necessary tool for daily life. Recommender systems are software or techniques that provide some suggestions about items that users like to use [11]. In addition to individual life, we are also a member of several groups and constantly engage in the different groups' decision-making process, a decision like where to go for a weekend in a family group, or where to eat with friends. Generally, a group of people starts a discussion about the potential options for an issue. In this discussion, the members support some items and try to persuade other members to accept their point

of view. These kinds of group activities embark on new needs which cannot be satisfied using individual recommender systems.

In individual recommender systems, the system's main goal is personalizing the recommendations for a specific user, that is, recommending items to a user concerning her personal preferences. Whereas in a GRS, the main objective is recommending items to a collection of users who have made a group for a specific purpose. Unlike individual recommender systems that deal with only one user, GRSs need to be able to aggregate the individual users' preferences and provide items to increase the group's satisfaction in their decision-making process compared to the situation that the group members do not use the system's support. By aggregation, the system tries to combine individual preferences using a mathematical function. In GRSs, the system deals with a group of people instead of an individual.

A group of people engages in a discussion when they make a decision. Research has shown that the outcome of this discussion can be completely different from the predicted outcome considering the users' initial preferences [4]. The satisfaction of a group member also relies on the satisfaction of other group members. Although this change, known as emotional contagion, depends on personal characteristics and interpersonal relationships, its influence on group decision-making has been shown by some of the studies [8]. Additionally, people tend to align their opinion with other group members' perspectives even in straightforward tasks [8]. Hence, predicting group scores (scores that computed by a method to reflect how the group think about the item), recommending proper items to the group, and supporting group discussion and decision-making process are the objectives of GRSs.

The methods which consider recommending or score estimation problem for a group can be divided into three main classes [7]. The first class includes traditional methods that try to construct a group profile using linear methods, such as averaging group member preferences. This group profile can be considered as the preferences of the group's representative user. The second class of methods is heuristic methods that use some heuristics to aggregate preferences and recommend items [6, 2]. For example, Gorla et al. [6] developed a graph-based method that takes both negative and positive aspects of group members into account. In this model, two similarity-based graphs with positive and negative edges are constructed for the users and items. The main heuristics in this method are: (i) potential group choice will be close to the group's previous choices, and (ii) inappropriate items will be near the group's previously disliked items. Then, using a random walk, the recommendations for the groups are produced. Finally, new methods try to use machine learning approaches to train a model for estimating the choices of groups based on historical data [10]. Using different aspects of the groups and users, deep models try to find the proper embedding of groups and items. Then, these embedding can be used for choice prediction [1].

2 Research Questions and Future Works

Most of the studies on GRSs, which some of them have been introduced above, have been dedicated to predicting group scores for items in a static environment (single-shot). These scores are further used to determine which item will be liked by most group members, and the statement "liked by the most" shows how different approaches look at this problem. However, group decision-making is a dynamic process in which the group decision is being made during the group discussion [4, 5]. Discussion can happen in a chat-based environment or in face-to-face communications. In this discussion, the group members discuss the pros and cons of the current options and select one. During this discussion, members' opinions about the options can be changed based on other members' supports or rejection. To deal with this situation, recommender systems should continuously monitor the group behavior, interactions, and preferences and adjust the system and recommendations based on the changes in the members' preferences.

In my Ph.D. study, I will try to study this dynamic process and integrate it with the social and personal aspects of users. As discussed above, social relations and discussion topics also play a role in the preferences' changes during a discussion. For instance, in a hierarchical family, social relations play a more important role than personality. In this case, the social relations determine the group choice and probably preference changes. The topic is another determinative aspect. Group members' behavior can change based on the topic on which they are discussing. For example, if the group is discussing a sensitive political issue, the members' behavior can be completely different compared to the situation that they are discussing where should be the next dinner party for the political parties. These dimensions can have an important influence on the group dynamics and preference evolution.

As it is clear, creating users and group profiles concerning social relations and personal characteristics is not trivial. Developing algorithms that can monitor group dynamics (with respect to personal characteristics, social relations, and discussion topics), create users and group profiles, and adapt them with group preference change is the ultimate goal of this research. In other words, I will research how the users' preferences evolve during a group discussion, how much these evolving preferences are different for different groups, topics, and members' personalities, and how we can support a variety of groups with varying characteristics in their decision-making process. This support aims at making the decision process easy and pleasant for the members. In the following, I discuss the main problems I will tackle in this Ph.D. research.

- **Question1.** An essential question in GRS is *how we can effectively model the group discussion dynamics to recommend proper items and support the group better*. Based on personal and social relations, people behave differently in different group discussions. For instance, children behave differently in a family group compare to a friend group. Additionally, people behave differently, even in the same group, when they discuss different topics. For instance, the controversy level in making a political decision between the

parties is much higher than the controversy level when they make a decision about non-sensitive social issues. These contradicting behaviors in the different groups by the same people or the same group on various topics should be taken into account for modeling group dynamics. Nguyen et al. [9] only considered one aspect of the group, which was conflict resolution style; but as we explained above, there is some other information about the group, users, and topic under discussion which are able to provide more information about the group dynamics, as they provide us broader perspective toward the group and its dynamics. If we model a group discussion that takes into account this information, we can predict the group's preferences in different circumstances and adjust our recommender system with these new preferences. For instance, if during the discussion for selecting a destination for the next trip, a group decided to go to the beach, the system should put aside all of the recommendations related to mountains.

- **Question2.** A GRS is made of two entities: System and Group. The group consists of members. These members can be described based on social relations and personal attitudes. For modeling group discussion dynamics, we can consider each user's personal attributes, such as openness to new experiences, extrovertism, etc., and social features, like type of relations, structure of the friendship, etc. Hence, another research question is *whether we can efficiently combine users' social and personal aspects to the system.* If so, *how.*

  Most of the previous works have considered social and personal attitudes in the static environment [3]. Nguyen et al. [9] were one of the first who studied personal attitudes, specifically conflict resolution style, in a dynamic environment. In our future research, we will focus on textual information provided by group members. This textual information, expressed by the group members during a discussion, can indicate their current stand (short-term preferences) with respect to the discussion and the group's possible options. Other textual information can also be exploited. For instance, there are some studies on detecting users' personalities using users' textual data in the social networks [13]. Furthermore, based on social relations, group members act differently in the groups. For instance, one may be more amenable in a friend group decision-making process; whereas, the same person can be more resolute in a working group. We also try to integrate the members' social dimension into the dynamic GRS to find more accurate predictions, recommendations, and better support.
- **Question3.** Having defined the important social and personal attributes of users as well as the discussion model, our next question is *how to predict the group's final choice accurately and satisfactorily recommend items to the group for supporting the members.* With prediction, the GRS can estimate the final group choice. This final choice can be utilized to support the group by helping them to reach a consensus in a shorter time or even discover new options similar to this predicted final choice, which seems to be interesting to the group. Unlike the aforementioned studies, which try to predict the group scores to items in a single-shot situation, we will try to predict group choice

in a dynamic circumstance by monitoring the users' behavior and updating the group scores. The corresponding recommendations should help the group to keep their focus on the underlying question, consider new undiscovered areas, and have a more pleasant discussion.

I believe that the future recommender system should play a more active role in the decision-making process. These proactive GRSs should help the group members have a better experience in a discussion and aid them in finding a better solution to their problem, which can maximize the group members' satisfaction. Therefore, I want to develop a conversational GRS. This conversational GRS must play a proactive role in the group decision-making process. This method should be adapted with evolving group preferences and, in each step, support them to experience a more satisfactory discussion. This conversational method should directly deal with group members and help them end up with a choice that can maximize all group members' satisfaction.

An application domain in which I can implement and evaluate the discussed idea is the food domain, in which a group of people wants to select a restaurant to go to with respect to their preferred dishes. The problem I am going to address is developing a proactive GRS that adapts the group preferences evolution using the data provided by the users in only one session (implicit or explicit). This session is temporary interaction of the users with the system from the time they enter to until the moment that they leave the system. This is a challenging problem as we do not have historical data about both users and groups.

3 Research Results

I started my research by working on a project known as Predicting Group Choices from Group Profiles. As discussed above, GRSs aim to learn users' preferences and behavior from historical data to predict group choice. Although a significant number of studies have been dedicated to predicting group scores, a small portion of them has considered predicting group choice as a problem.

SDS theory [12], which investigated the problem of combining group members' individual preferences for predicting the discussion outcome, is the basis of our study. In his research, Stasser aimed at answering two questions: (i) how are the inclinations of individuals combined to reach a consensus in a group?, and (ii) what are the implication of the findings in a set of conditions to another situation? The method we are working on tries to train a machine learning model to predict the final choice of a group decision-making process. Indeed, the model intends to learn the group composition (distinguishable distributions), that is, how the group members think about the possible options (the first question of Stasser's work) and use this distribution to predict the final choice of a group (second question of Stasser's work). Our method automatically learns how to construct a group profile and how from a group profile, the group choice can be predicted, as Stasser also mentioned that this relationship could be learned.

In the future, we intended to improve our model to consider more group choices. In our group choice model, we only concentrated on group choices when

the alternative options are few. We want to examine to what extent our method is good in dealing with more options for the groups. We are also trying to use other group features, such as group size, discussion time, etc., to see whether they are informative enough to the model.

References

1. Cao, D., He, X., Miao, L., An, Y., Yang, C., Hong, R.: Attentive group recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 645–654 (2018)
2. Carvalho, L.A.M.C., Macedo, H.T.: Users' satisfaction in recommendation systems for groups: an approach based on noncooperative games. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 951–958 (2013)
3. Christensen, I.A., Schiaffino, S.: Social influence in group recommender systems. Online Information Review (2014)
4. Delic, A., Neidhardt, J., Nguyen, T.N., Ricci, F., Rook, L., Werthner, H., Zanker, M.: Observing group decision making processes. In: Proceedings of the 10th ACM conference on recommender systems. pp. 147–150 (2016)
5. Forsyth, D.R.: Group dynamics. Cengage Learning (2018)
6. Kim, H.N., Bloess, M., El Saddik, A.: Folkommender: a group recommender system based on a graph-based ranking algorithm. Multimedia systems **19**(6), 509–525 (2013)
7. Masthoff, J.: Group recommender systems: aggregation, satisfaction and group attributes. In: recommender systems handbook, pp. 743–776. Springer (2015)
8. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction **16**(3), 281–319 (2006)
9. Nguyen, T.N., Ricci, F., Delic, A., Bridge, D.: Conflict resolution in group decision making: insights from a simulation study. User Modeling and User-Adapted Interaction **29**(5), 895–941 (2019)
10. Ortega, F., Hernando, A., Bobadilla, J., Kang, J.H.: Recommending items to group of users using matrix factorization based collaborative filtering. Information Sciences **345**, 313–324 (2016)
11. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Recommender systems handbook, pp. 1–34. Springer (2015)
12. Stasser, G.: A primer of social decision scheme theory: Models of group influence, competitive model-testing, and prospective modeling. Organizational Behavior and Human Decision Processes **80**(1), 3–20 (1999)
13. Tay, L., Woo, S.E., Hickman, L., Saef, R.M.: Psychometric and validity issues in machine learning approaches to personality assessment: A focus on social media text mining. European Journal of Personality **34**(5), 826–844 (2020)

Resource Management in MEC Networks

Bin Xiang

Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, Italy
bin.xiang@polimi.it

Abstract. In the 5G and beyond mobile networks, Mobile Edge Computing (MEC) brings cloud-computing capabilities to the edge of the mobile networks, especially in close proximity to mobile users, making it possible to simultaneously address the stringent latency requirements of critical services and ensure efficient network operation and service delivery.
However, MEC services, on one hand, require significant investments from both network operators and service providers in terms of deploying, operating and managing edge clouds, and on the other hand, provide limited computational and storage resources by design. Besides, due to the large amount of tasks from users with high demands during peak hours, the latency requirements of different services can hardly be guaranteed. This issue can be tackled by massively deployed edge clouds that are attached to the base stations and connected to each other in a specific topology, as ultra-dense 5G-and-Beyond networks are built.
The goal of this research is to leverage cooperation among interconnected multiple MEC units and investigate joint resource optimization considering multiple aspects of network operations, with the target of enhancing the utilization efficiency of resources to further satisfy improved QoS and reduce network operation cost. Specifically, aggregated mobile traffic and user requests are considered based on their types (e.g., video, web, game, etc.) associated with different QoS requirements. This research jointly optimizes i) where to process the traffic and requests, ii) how to route network flows and iii) how to allocate and schedule the required resources w.r.t. communication, computation and storage. These problems are formulated into multiple mathematical models and both centralized and decentralized approaches are proposed to tackle them efficiently. The performance is evaluated in real-size network scenarios including both random geometric graphs and a realistic mobile network topology, showing the impact of the considered parameters (e.g., tolerable latency, network topology and bandwidth, computation, storage, etc.) on both the optimal and approximate solutions.

Keywords: mobile edge computing · resource management · optimization

1 Introduction

In a network with the interconnected multiple MEC nodes, the efficient delivery of next generation services requires joint optimization of where to execute each

service request, how to route network flows and how to allocate and schedule the required resources w.r.t. communication, computation, storage, in order to meet the improved QoS.

The goal of this research is to study resource management for MEC networks with arbitrary topology. More specifically, several problems are addressed, mainly related to slicing[1], planning[2] and scheduling[3, 4] of edge resources for mobile traffic.

The first scenario is an edge network organized by multiple edge clouds, each of which is connected to the Radio Access Network at a certain location. All such edge clouds are connected through various topologies, typically organized in some kind of hierarchy. In this way, each edge cloud can serve end user traffic by relying not only on its own resources, but also offloading some traffic to its neighbors when needed. In this edge network, mobile user traffic coming from one access network is considered, which is aggregated according to the different types (e.g., voice, video, web, game, etc.) associated with different QoS requirements on user experienced latency. Each class of traffic can be segmented and processed on all edge clouds, and each edge cloud can slice its computation capacity to serve different classes of traffic. Specifically, for each class of incoming traffic, the corresponding mobile access network decides how to slice wireless network capacities while the co-located edge cloud decides i) whether to serve the traffic or to offload it to some other edge cloud nodes and ii) how to allocate computation capacities for the pieces of traffic. This decision depends on the QoS requirements associated with the specific class of traffic and on the current status of the edge cloud. The main objective in this context is to ensure that the infrastructure is able to serve all the possible types of traffic within the boundaries of their QoS requirements and of the available resources. A joint slicing of mobile network and edge computation resources is proposed. Therefore, the proposed optimization approach aims at minimizing the total traffic latency of transmitting, outsourcing and processing user traffic, under constraints of user tolerable latency for each class of traffic.

The first problem focuses exclusively on minimizing the latency of traffic in a hierarchical edge network, with the network and computation capabilities fixed. Further, the proposed second problem takes into account the overall budget that the operator uses in order to plan and allocate the computation capabilities in its edge network which also has an arbitrary topology. Multiple classes of mobile user traffic from multiple access networks (or multiple ingress networks) are considered. The main objective is to further operate cost-efficient edge networks through jointly planning the availability of computational resources at the edge, the slicing of the mobile network and edge computation resources, and the routing of heterogeneous traffic types to the various slices. Therefore, the proposed optimization approach in the second problem aims at minimizing the network operation costs and the total traffic latency of transmitting, outsourcing and processing user traffic, under constraints of user tolerable latency for each class of traffic.

Finally, the study focuses on serving different classes of user requests in MEC networks with arbitrary topology. Each type of request is regarded as an aggregated communication-computation demand, e.g. web, video, game and etc., which has to be accommodated in the network and requires some amount of bandwidth, storage and computation resources. The calendar (i.e., the starting time and duration) of the requests for the upcoming period is assumed to be known, which can be achieved under the condition that customers have announced their requirements in advance, or by using some history-based prediction tools. The main objective is to exploit the intrinsic flexibility of services demanded by different users, the starting time of which can be shifted without penalizing the utility perceived by the users while, at the same time, permitting a better resource utilization in the network. Therefore, The aim is to study an optimization framework that jointly considers several key aspects of the resource allocation problem in this context, specifically, through optimizing: i) admission decision (which requests are admitted and served by the network), ii) scheduling of admitted requests, also called calendaring, iii) routing of these flows, iv) the decision of which nodes will serve such requests as well as v) the amount of processing and storage capacity reserved on the nodes chosen, with the objective of maximizing the operator's profit.

The above proposed optimization models are first formulated as mixed-integer nonlinear programming (MINLP) problems, which are $\mathcal{NP}$-hard. To tackle them efficiently, Equivalent reformulations from MINLP to mixed-integer quadratically constraint programming (MIQCP) are performed, and based on that, further effective heuristics are proposed to facilitate the solutions of the problems, including the Sequential Fixing based approaches and a distributed algorithm based on Alternating Direction Method of Multipliers (ADMM) for the edge calendaring model. The performance of the proposed models and heuristics are evaluated in real-size network scenarios including both random geometric graphs and a realistic mobile network topology, showing the impact of all the considered parameters (i.e., different types of user traffic or requests, tolerable latency, network topology and bandwidth, computation, storage and link capacities) on both the optimal and approximate solutions. Results obtained demonstrate that near-optimal resource allocation solutions can be achieved by the proposed heuristics in short computing time.

2 Future Directions

Future directions of this research include investigating the following aspects: i) studying context-aware resource allocation problems in the MEC networks, exploiting the context information, e.g., the user traffic patterns for each service, user density for the location of each MEC unit, etc., analyzed by using the historical data. In this way, MEC network resources can be allocated and scheduled in advance so as to improve quality of services while reducing system costs; ii) taking into account the dependence among network service functions and focusing on the optimization of service function chain (SFC, a set of service functions

that must be executed in a specific order) deployment in the context of MEC networks.

Acknowledgments

Thanks for the supports of the EU projects H2020-MSCA-ITN-2016 SPOTLIGHT under grant No. 722788 and H2020-ICT-2020-1 PIACERE under grant No. 101000162.

References

1. Xiang, B., Elias, J., Martignon, F., Di Nitto, E.: Joint network slicing and mobile edge computing in 5G networks. In: IEEE International Conference on Communications (ICC) (2019)
2. Xiang, B., Elias, J., Martignon, F., Di Nitto, E.: Joint planning of network slicing and mobile edge computing: Models and algorithms. IEEE Transactions on Cloud Computing (2021)
3. Xiang, B., Elias, J., Martignon, F., Di Nitto, E.: Resource calendaring for mobile edge computing in 5G networks. In: IEEE International Conference on Communications (ICC) (2021)
4. Xiang, B., Elias, J., Martignon, F., Di Nitto, E.: Resource calendaring for mobile edge computing: Centralized and decentralized optimization approaches. Computer Networks (2021)

Decentralised Cloud Computing

Adrian Spătaru

Department of Computer Science, West University of Timisoara, RO
adrian.spataru@e-uvt.ro

Abstract. This paper identifies the current research directions in the field of Decentralised Cloud Computing.

Keywords: distributed systems, cloud continuum, smart contracts

1 Motivation

The Internet promised a global system where everyone makes its content available to others. Whereas this is true, having billions of people sharing their content and thoughts through social media platforms, this content is stored in a centralised manner. This has lead to numerous privacy breaches, in the sense that the stakeholders are collecting vast amounts of personal information which can be used for financial gains, but can sometimes be used maliciously. Regulations have been put in place to put the power in the hands of the data source (the content creators or consumers), requiring the platforms to inform the users about data collection policies and delete the data upon their request. Furthermore, the centralisation of Cloud resources may lead to network congestion because of the multitude of users accessing the same highway to these centres. In parallel, some file-sharing applications and volunteer computing services made use of peer-to-peer networks of computers to provide the storage and computational backbone. Recently, the Internet of Things and Edge Computing fields emerged, requiring that data is pre-processed at the edge of the network (near the data source), processed on Fog nodes, and later sent to the Cloud for archiving and in-depth analytics, forming a Cloud Continuum. The emergence of Blockchains has further pushed the research effort invested in peer-to-peer systems with a focus on decentralisation, security, and transparency. This research statement outlines several directions worthy of investigation in the framework of the Cloud Continuum, focusing on Blockchain technology as a notary for access control to Cloud, Fog and Edge resources, while providing security guarantees and accountability for both the Cloud Consumer and the Cloud Resource Provider.

2 Background

I am a Teaching Assistant and Post-Doctoral Researcher at West University of Timişoara and I have interest in distributed and parallel computing, having a

combination of theoretical analysis, modelling and technical skills. My current fields of interests are represented by Edge, Fog and Cloud computing as well as Blockhain technology, with the goal to combine the security and transparency advantages of the Blockchain with the versatility and ubiquity of the Cloud Continuum. Nevertheless, I am eager to provide support for processing data or accelerating algorithms using technologies like OpenMP, MPI, or CUDA in order to advance the research of the colleagues from my lab or researchers from other departments in our university.

2.1 Thesis

My PhD thesis investigated a decentralised Cloud platform, instantiated as a peer-to-peer network that offers computational resources like Virtual Machine Instances, Containers, and accelerators like Graphic Processing Units (GPU), Many Integrated Core (MIC) cards or Data Flow Engines (DFE). The resources can be provisioned through Smart Contracts deployed on a public Blockchain. The main contributions of the thesis are:

1. The design and implementation of the Gateway Service, a collection of components allowing for the definition, composition, optimisation, and deployment of Cloud Services which use HPC infrastructure like GPUs, MICs or DFEs. Together, the components enhance the flexibility of the Cloud Service Provider through the concept of Abstract Applications. Abstract Applications can be instantiated on different infrastructure, depending on the constraints of the user and the availability of the Cloud resources. A preliminary version of this work has been published [1], while the thesis presents the final version.
2. An architecture employing self-organisation of collaborative components which manage computational resources [2]. The Self-Organising Self-Management (SOSM) framework consists of a bottom layer representing the physical resources, and several layers of software components which manage the workload on the resources under control. The software components employ self organisation policies (exchange resources, split or merge) and self-management policies (task scheduling [5, 9], resource turn-off for electricity savings) in order to align themselves with a goal update received from a top layer component.
3. The design of protocols and mechanisms for the management of a decentralised Cloud platform composed of personal computers with a focus on the fault tolerance of the running Services and the Components that contribute to the management of the system. At first, several scheduling methods have been proposed and investigated to understand their behaviour on a real world use case [8]. A Component Administration Network is later introduced to monitor the state of the system components and storing checkpoints related to their state [10]. One such component is the Orchestrator, which ensures the fault tolerance of the Services. Moreover, the components are writing important events in the Application Instance Smart Contract, which are used

for evaluating a fair payment related the the amount of work done by each part of the system: resource, administration network, orchestrator.

The delivery model of Cloud Services has shifted focus from the provisioning of Virtual Machines to the management of on-demand Containers. The Orchestrator developed in the context of my PhD thesis is the first to allow the orchestration of hybrid applications, composed of mixed Virtual Machine and Container packed software. Furthermore, Fog and Edge use-cases require the design of Applications with a high level of heterogeneity, which is allowed by the specifications and tools described in [1].

The analysis provided in [8] has impact beyond a specific Blockchain system because it tackles a problem fundamental to information management. Ethereum 2.0 is adding a Proof of Stake consensus layer in order to increase the block production rate. Yet, gas limits (or other mechanisms for other Blockchains) will still be required to prevent denial of service attacks. The protocols described in [10] are designed to optimise the amount of work executed in the Smart Contract, increasing the throughput of the system with respect to the number of transactions (decisions) that can be mined in a block.

2.2 Other work

Machine learning techniques require a lot of time to train for predicting the consumption of individual customers due to the increasing number of customers connected to the smart grid. To reduce the number of individual predictions, one solution is to cluster customers with similar patterns, train the model for one time-series, and use this model to predict the behaviour of all customers in the same cluster [7]. The accuracy of our method is compared against standard ARIMA and the best found seasonal ARIMA model. Results on real-life data show an average deterioration of 30% in terms of Mean Average Percentage Error of the best found individual model.

Another study investigated the use of genetic algorithms for discovering parametric configurations that will improve the accuracy of predictions for the Weather Research and Forecast (WRF) numerical weather prediction system. The average prediction error was reduced in a small amount of iterations [3].

During my work on the ASPIDE Horizon 2020 Project, I contributed to the definition of the programming constructs of the DCEx programming model [6] designed for the implementation of data-centric large-scale parallel applications on Exascale computing platforms.

I also had a support role in a study related to assessing the freshwater quality remotely using land cover data [4], implementing an automation tool for the aggregation of information stored in several files within sub-directories. The tool proved of great use for the researchers from the Geography and Biology Department, reducing the time to publication and human errors.

3 Research Objectives

3.1 Formal validation of decentralised Clouds

Proof of Work Blockchains are a Byzantine Fault Tolerant mechanism which ensures a cryptographically secure and transparent ordering of valid transactions, while discouraging Sybil attacks (multiple identities attack). Assets that are stored on the Blockchain benefit from its advantages, yet not all assets can be linked with the real world object they describe. If this link is weakly enforced, the system is vulnerable to different types of attacks. In our context, the assets are the computational resources and the components involved in managing the Cloud System. The fault tolerance enforced over the platform proposed in my PhD thesis discourages a byzantine adversary from attempting some types of attacks, for example having multiple identities of the same computational unit. Nevertheless, these aspects require formalisation in the domain of Distributed Systems Security.

3.2 Data locality and privacy

Peer-to-peer systems are advantageous compared to central servers from a data transfer point of view, yet several optimisations can be made with regard to data movement. This aspect is worthy of investigation in the context of the Cloud Continuum, incorporating aspects of data locality to improve the efficiency of the system. This will imply the collection of information describing the topology of the network and the employment of policies to ensure replication and fast access. Moreover, data should be encrypted in order to allow for any rules related to the privacy and access control.

3.3 Standards and Specifications

A survey is needed to investigate the architecture and philosophy differences between the different efforts in the direction of decentralised management of computational resources. Some important aspects to be investigated are:

- is there an ontology describing the entities that are part of such systems?
- are there standards or specifications put in place to allow communication between components developed by different parties? are there contradictions? is there a complete standard?
- which are the mechanisms to assess the quality of service in a decentralised, trustless environment ?
- are there any open-source libraries that can be linked to ease the development of such a system?
- is simulation software available to investigate different organisation, management and security policies?

To summarise, my current objectives are related to the different components involved in a decentralised platform that manages the access of data and computational resources. The purpose is to interconnect personal computers with business infrastructure in order to facilitate the processing of data and availability of services at three different layers: Edge, Fog and Cloud.

References

1. Dragan, I., Fortiș, T.F., Neagul, M., Petcu, D., Selea, T., **Spataru**, t.: Application blueprints and service description. In: Heterogeneity, High Performance Computing, Self-Organization and the Cloud, pp. 89–117. Springer (2018)
2. Filelis-Papadopoulos, C., Xiong, H., **Spataru**, t., Castañé, G.G., Dong, D., Gravvanis, G.A., Morrison, J.P.: A generic framework supporting self-organisation and self-management in hierarchical systems. In: Parallel and Distributed Computing (ISPDC), 2017 16th International Symposium on. pp. 149–156. IEEE (2017)
3. Oana, L., **Spataru**, t.: Use of genetic algorithms in numerical weather prediction. In: Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2016 18th International Symposium on. pp. 456–461. IEEE (2016)
4. Şandric, I., Satmari, A., Zaharia, C., Petrovici, M., Cîmpean, M., Battes, K.P., David, D.C., Pacioglu, O., Weiperth, A., Gál, B., et al.: Integrating catchment land cover data to remotely assess freshwater quality: a step forward in heterogeneity analysis of river networks. Aquatic Sciences **81**(2), 26 (2019)
5. Selea, T., **Spataru**, t., Frincu, M.: Reusing resource coalitions for efficient scheduling on the intercloud. In: Cluster, Cloud and Grid Computing (CCGrid), 2016 16th IEEE/ACM International Symposium on. pp. 621–626. IEEE (2016)
6. Talia, D., Trunfio, P., Marozzo, F., Belcastro, L., Garcia-Blas, J., del Rio, D., Couvee, P., Goret, G., Vincent, L., Fernandez-Pena, A., de Blas, D.M., Nardi, M., Pizzuti, T., **Adrian Spataru**, Justyna, M.: A novel data-centric programming model forlarge-scale parallel systems. In: EuroPar Workshop LSDVE (2019)
7. **Spataru**, t., Frincu, M.: Using cluster information to predict individual customer consumption. In: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017 IEEE PES. pp. 1–6. IEEE (2017)
8. **Spataru**, t., Ricci, L., Petcu, D., Guidi, B.: Decentralized cloud scheduling via smart contracts. operational constraints and costs. In: The International Symposium on Blockchain Computing and Applications (BCCA2019) (2019)
9. **Spataru**, t., Selea, T., Frincu, M.: Online resource coalition reorganization for efficient scheduling on the intercloud. In: International Conference on Algorithms and Architectures for Parallel Processing. pp. 143–161. Springer (2016)
10. **Adrian Spătaru**: Decentralized and fault tolerant cloud service orchestration. Scalable Computing: Practice and Experience **21**(4), 709–725 (2020)

Using Machine Learning to Speed Up Optimization for Graphical Models

Aleksandra Petrova[1 [0000−0001−7485−5309]]

Universitat Politècnica de Catalunya - UPC, Barcelona, Spain
aleksandra.petrova@upc.edu

Abstract. The weighted CSP (WCSP) framework is a soft constraint optimization framework with a wide range of applications. This range is achieved by having scientists from different fields use the toulbar2 solver for their respective problems. Often these problems are looked into the respective fields, however no information is collected or shared from these solutions to improve optimization of other problems. The goal of this PhD research is to utilize the information of the wide range of problems in optimization and integrate Machine Learning to better optimize the algorithms. For that, a combination of statistical analysis on the different problems and Machine Learning modeling will be used. The statistical analysis will entail discovering different patterns that can be found in the problems, which point to utilizing this information to future unknown problems. Based off of that, the models will be built with the aim of discovering those patterns which will speed up the optimization of future unseen problems.

Keywords: Weighted CSPs · Tree Decomposition · Discrete Optimization · Machine Learning · Constraints Programming.

1 Introduction

Constraints Programming is a form of problem solving in which the problem is modeled as a set of decision variables, each having a set of possible values, and a set of constraints restricting the allowed combinations of values to variables. Constraints Programming (CP) is a promising field for the future of Artificial Intelligence (AI), as a lot of what it tackles influences the real world. Not only that, but CP is key to reaching the goal of logical reasoning in AI, as well as improving optimization which is of huge importance for advancing Computer Science. I am specifically addressing a weighted version of these problems called *weighted constraint satisfaction problems* (WCSPs). This framework has real world applications in areas such as: *Bioinformatics*, *Computer vision*, *Satellite observations*, *Probabilistic models*, *Airplane landing*, *Warehouse location problems* and many others.

One of the main focus of CP so far has been to develop better and more efficient algorithms. In the context of WCSPs the goal is to reach the optimum in the quickest way possible. This includes algorithms that preprocess the instances

so that the difficulty of the constraints can be minimized, that prune the search space and that search the space efficiently. While there has been a great deal of advances in all 3 areas, making the solvers very efficient, the problems are addressed as separate entities, meaning each problem is tested with different algorithms for pruning and search exploiting to obtain the best result. Researchers that work in the field of WCSPs have developed an intuition for knowing which problem will be better tackled with which algorithms. This is not the case for scientists that come from other fields and use WCSP solvers.

Despite the solution of one particular instance being important in the context of that instance, the information that is used for solving it is not stored or used to aid other similar instances. Instances of one problem can also often end up having similarities with each other. Even different problems may end up also having similarities. However, when solving them, people are unaware of such similarities existing in another area of problems, which then results in simply testing many solving strategies to obtain the best optimization for that problem. One approach to this issue is to use the power of Machine Learning to make more informed decisions when solving the problems. My main research goal is to create tools and algorithms for efficient solving Weighted Constraint Satisfaction Problems by extracting information from existing problems and using Machine Learning to make informed decisions when solving other problems.

2 State of the art

We can get more knowledge about the optimization problem we are working on by looking at its graph. Often problems that have acyclic graphs can be solved in linear time. The same doesn't apply to cyclic ones. To be able to know this information we can look into the tree decomposition of a problem. Using this algorithm we turn a graph into a tree [4]. This decomposition is very often used for solving problems, such as guiding a dynamic programming algorithm which solves the problem bottom-up.

To be able to benefit from the usage of the dynamic programming algorithm, the graph needs to have small tree width [8, 3]. Width is the measure of the size of the largest cyclic part, where the time and space complexity are exponential. Solving using tree decomposition is encountered in a different type of problems such as Bayesian Networks, Markov networks [7], Weighted CSPs [5, 9], and others.

In the case that the width is not small the solving relies on using search methods such as Depth-first search or Best-first search. These algorithms are more feasible in comparison to heuristic search algorithms as they can get exponential in problem size. Luckily the algorithms have been better optimizing, by solving large instances in a good time frame through pruning [6].

Another approach to tree decomposition regardless of the width is through adapting heuristic search to the problem decomposition. This is done by using AND/OR search [12]. By using this search the memoization is moved from tabular to an implementation of dynamic programming. The challenge here posed

is to use all the pruning techniques which are developed for heuristic search [14, 11, 2].

For the algorithms to be solved there is no attention paid to the choice of the root, but the algorithms do pick one. In the case of dynamic programming this is not of importance since the worst case complexity is a tight bound of average complexity. In the case of other algorithms it has been reported that the root can have an effect on the performance [10]. One such algorithm which has been focusing on using backtracking tree decomposition is the BTD algorithm inside toulbar2 [14], and the Russian doll search exploiting a tree decomposition [13].

So far there has been little focus in the CP community for researching the impact that the choice of root has to the overall optimization. One of the situation where the selection of root has been prioritized is the paper of Abseher et al. [1]. In their paper they create a Machine Learning algorithm which focuses on automating the selection of the root based on features of the decomposition. Their focus is on the dynamic programming approach to solving tree decomposition. The results are achieved by using 5 different problems which result in an overall 27.33% average improvement in comparison to not using their algorithm.

Besides looking into the decomposition itself and the information it can provide for the solution of the problem, another area in which research has been expanding is using into the search space. One such study is Learning Branching Heuristics for Propositional Model Counting [15]. Here the authors used evolution strategies to learn branching heuristics for SAT solvers. With their solution they are able to reduce the step count and generalize to larger instances from the same family.

3 Current work

From the start of my PhD in December 2020, until now I have focused on firstly familiarizing myself with the current research in CP and WCSP solving. During my literature overview I learned about the evolution of the search algorithms, the impact that they have made and different ways of preprocessing the problems. More precisely, I have become familiar with **toulbar2** [1] which is arguably the best general purpose WCSP solver. I have learned the many options that the solver provides and the huge impact that chosing ones o anothers may have in performance. I have seen that a given instance can be trivially solved with some solving options and run forever with some others and, most importantly, there are no precise guidelines telling the user which options work best.

One key aspect of WCSPs and other Graphical Model frameworks is the exploitation of problem structure and the best way to see such structure is through the notion of *tree decomposition*. Most algorithms that take advantage of tree decomposition must start by selecting a tree root.

The focus of my current work is on researching on how to make a more informed decision by having information about the Tree Decomposition of a wcsp

[1] https://miat.inrae.fr/toulbar2/

instance. Through solving all the instances with the default settings, and looking into their Tree Decomposition, it is visible that there are nodes that, if chosen as root, allow for a more efficient traversal of the search space. This hypothesis was confirmed through analyzing the Central Processing Unit (cpu) time when solving the instance with different nodes as the root. To be able to identify the pattern of which node might be the best root, different measures were created and used. These measured included: Cluster Size (Number of variables that the node contains), Domain Aware Cluster Size (Taking the domain sizes into account), Cluster Decomposition Size (Largest sub-problem after assignment), and Cluster Height (Longest path from the root). By correlating the cpu time and the measures, no direct relationship was revealed. This experiment was conducted on 30 hand picked instances.

Based on the results, amount of instances, and the fact that the hypothesis was confirmed, the experiment is being repeated currently. All of the available data is being used, selected and preprocessed to ensure correct results. The aim is to later use Machine Learning algorithms for recognizing the pattern, as simple measures based on syntactical features of the Tree Decomposition do not seem to predict well a good root.

4 Objectives and Methodology

Based on the state of art and the achievements in the field so fat, the idea of this PhD is to take information from the problems themselves before attempting to solve them and make a more informed decision on which algorithms work best. Similarly information from other problems would be added to find similarities which may give additional insight into which model would perform the best. To be able to find the similarities between instances of the same problem or instances from others, we will harness the power of Machine Learning. This is done by first examining the data statistically in order to find areas where the problems may show similarities, preprocess the data and then use an algorithm which will identify those patterns well.

The aim of this PhD thesis is to enhance the optimization abilities of the existing algorithms using Machine Learning. Specific objectives are:

- To better understand the different types of problems that are commonly solved, and how information from them can be used to solve other problems alike.
- To understand the information that the search space provides and how it can be used to make informed decisions when searching.
- Create a Machine Learning algorithm that uses the information to optimize the search.

This PhD uses problems from different fields that the researchers a the INRAE MIAT in Toulouse and UPC, IIIA-CSIC in Barcelona have collected for the purpose of studying CP. The solver toulbar2 will be used for solving the problems, preprocessing the problems and extracting information regarding the

search space of them. For the purpose of analyzing the data and modeling the Machine Learning models languages such as python, R and C++ will be used. To be able to parallelize a cluster or different machines will be used for solving the problems, using toulbar2 for other problem related information acquisition or training and testing the Machine Learning models.

5 Future work

At the current stage of the analysis of solving the problems using a different root based on the tree decomposition, my main focus is first to understand the data better. Run the experiments with different roots and obtain statistical information and measures that could show why certain roots are a better option. Once this is ready a correlation will be used to see if there is any indication of which measure might be useful to predict a good root. Since in the previous experiment the measures already were not providing good information on what may be a good root, the next step would be to use Machine Learning to better identify the potential pattern. For this I would like to use the class of deep learning methods known as Graph Neural Networks. This type of method would be suitable for the structure that the problems get once we obtain their Tree Decomposition. To be able to build the Neural network I will first collect features from the data that will help the network make a better distinction from what is a good root and what is not. Afterwards the model will be built and trained with a good sample of the data that represents different types of problems that can be encountered. Last the model will be tested on unseen data, so the generalization of it can be evaluated. In the case of the Graph Neural Networks not performing well in the task of identifying the best root, alternative Machine Learning models such as building a more standard classification type of Neural Network that hierarchically classifies the cpu times, or creating a non-neural network method where the cpu times are encoded into classes and then the best root is selected.

One of the most challenging problems in CSP is to optimize the search space. So far there have been numerous improvements done using techniques for traversing the search space. While that has certainly made the optimization quicker, it has also made the algorithms more complex by adding more layers to them. I want to approach this problem from the perspective of Machine Learning, so that the space can be traversed by using the information that we gather from it as we go through it. This entails understanding the different types of search spaces that can appear commonly and how information can be propagated through them. The challenge will then be to incorporate Reinforcement Learning into it, as the agent will take the information that is being shared and make more efficient decisions along the way. To be able to build the Reinforcement Model myself, I will study the different types of models that could be applied here. Next I will also work on representing the space in the best way possible, so that the rewards can be maximized. Finally, a strategy for the agent will be built. The challenges of combining CSP and Reinforcement Learning are numerous, especially making sure that with Reinforcement Learning the search becomes even more optimal,

but the posibilities for reaching better optimization though informed decisions makes it an exciting area of research.

References

1. Abseher, M., Musliu, N., Woltran, S.: Improving the efficiency of dynamic programming on tree decompositions via machine learning. Journal of Artificial Intelligence Research **58**, 829–858 (2017)
2. Allouche, D., de Givry, S., Katsirelos, G., Schiex, T., Zytnicki, M.: Anytime hybrid best-first search with tree decomposition for weighted CSP. In: Pesant, G. (ed.) Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9255, pp. 12–29. Springer (2015)
3. Bertelè, U., Brioschi, F.: On non-serial dynamic programming. J. Comb. Theory, Ser. A **14**(2), 137–148 (1973)
4. Bodlaender, H.L., Grigoriev, A., Koster, A.M.C.A.: Treewidth lower bounds with brambles. Algorithmica **51**(1), 81–98 (2008)
5. Cooper, M.C., de Givry, S., Sánchez-Fibla, M., Schiex, T., Zytnicki, M., Werner, T.: Soft arc consistency revisited. Artif. Intell. **174**(7-8), 449–478 (2010)
6. Cooper, M.C., de Givry, S., Schiex, T.: Graphical models: Queries, complexity, algorithms (tutorial). In: Paul, C., Bläser, M. (eds.) 37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France. LIPIcs, vol. 154, pp. 4:1–4:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
7. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
8. Dechter, R.: Bucket elimination: A unifying framework for reasoning. Artif. Intell. **113**(1-2), 41–85 (1999)
9. Dechter, R.: Reasoning with Probabilistic and Deterministic Graphical Models: Exact Algorithms, Second Edition. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2019)
10. Jégou, P., Terrioux, C.: Combining restarts, nogoods and bag-connected decompositions for solving csps. Constraints An Int. J. **22**(2), 191–229 (2017)
11. Marinescu, R., Dechter, R.: AND/OR branch-and-bound for graphical models. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005. pp. 224–229. Professional Book Center (2005)
12. Nils, N.: Artificial Intelligence: A New Synthesis. Morgan Kaufmann (1998)
13. Sanchez, M., Allouche, D., De Givry, S., Schiex, T.: Russian doll search with tree decomposition. In: 21st International Joint Conference on Artificial Intelligence. p. 6. Morgan Kaufmann (2009)
14. Terrioux, C., Jégou, P.: Bounded backtracking for the valued constraint satisfaction problems. In: Rossi, F. (ed.) Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2833, pp. 709–723. Springer (2003)
15. Vaezipoor, P., Lederman, G., Wu, Y., Maddison, C.J., Grosse, R., Lee, E., Seshia, S.A., Bacchus, F.: Learning branching heuristics for propositional model counting. arXiv preprint arXiv:2007.03204 (2020)

Guaranteeing Privacy and Fairness in Personalized Systems

Giacomo Medda[1 [0000−0002−1300−1876]]

University of Cagliari, Cagliari 09124, Italy
giacomo.medda@unica.it

Abstract. The main problem to tackle in my research plan is the unfairness of the outcomes of recommender and biometric systems. Nowadays, social problems such as fairness among demographic groups, fight for human rights and discrimination are relevant topics for every person, and these problems are related to the biases present in various contexts, ranging from healthcare to jurisdiction systems. Artificial intelligence (AI) solutions are trained with data collected from real world scenarios and the biases inside the data can directly or indirectly affect the outcomes of AI systems. It is important to focus on the unfairness of the results of such systems to find the causes of these problems and on techniques that could help to mitigate this lack of fairness. AI is used in many scenarios of our everyday life, hence, the decisions taken with the support of these solutions can lead to consequences that mark these biases, resulting in group of individuals being systematically discriminated. Solutions to overcome these problems are necessary to make AI systems reliable for everyone and be sure that people are treated equally. This document aims to face the unfairness problem without using demographic information in order to strengthen the privacy of the systems: an equal quality, accuracy, classification of AI solutions would ensure fairness among the users without knowledge of the sensitive information, and here the advantages of such an approach are presented and examined.

Keywords: Recommender systems · Speaker recognition systems · Machine learning · Fairness · Consumer fairness

1 Background and Current work

Artificial intelligence and machine learning (ML) approaches have been widely used in the literature to solve a vast diversity of problems, amongst which are automatic speaker recognition (ASR) systems and recommender systems (RecSys). The former aims to confirm or refute the user's identity based on an enrolled speech model [5], the latter is a subclass of information filtering that seeks to suggest relevant items to users, e.g. movies, books, jobs. Current solutions to achieve these goals are often optimized to provide the highest accuracy in recognising a user or recommending items, while other concerns do not receive the same consideration. Recent literature uncovered algorithmic discrimination in sensitive contexts, raising attention to other beyond-accuracy objectives, amongst which

is fairness, which has started to be more relevant in different domains of ML, such as recommender systems [22, 9, 18, 4, 21, 7, 2, 14, 8], speaker recognition [12, 10], facial recognition [13, 19, 20] and others [15, 6, 23, 16, 24, 17, 1, 3].

The first problem the literature deals with is the correct definition of fairness. In [15, 23], the principal notions of fairness used in ML literature are defined and explained, but each one of them marks a particular type of unfairness and none of them can individually represent the right definition of fairness. In the context of ranking algorithms, [21] states that "There is no single definition of what constitutes a fair ranking, but that fairness depends on context and application": in learning to rank there are different aspects than need to be satisfied and offering equal ranking quality to all users could not guarantee a fair ranking. Current work makes use of these fairness notions or define new ones [7, 21, 13], but, in general, the ones listed in [15] are the most widely used.

The second problem is the mitigation of unfairness in ML. The different techniques proposed in the literature can be divided in 3 groups: pre-processing, in-processing and post-processing. Pre-processing techniques include strategies that balance the training data on the basis of demographic information [9, 12, 10], strategies that sample the users according to a particular constraint [1], strategies that create indexes to assign a minority/majority value to users and items in RecSys and use these indexes to guide the learning process [4]. In-processing techniques are based on added constraints during the training phase, in particular, modifications of the loss function [2, 4, 8, 14, 24] by considering other variables, such as expected exposure [8], computation on indexes [4], pairwise comparisons [2]. Post-processing techniques include the application of randomization algorithms [8], ad hoc algorithms to reduce the bias disparity [22, 17], fair reranking approaches [18].

2 Goals and Results

Even though the literature shows interesting results with respect to the mitigation of unfairness in ML systems, the most part of these studies is based on techniques that depend on demographic information. This raises problems of privacy and generalization when approaches of group fairness are taken into consideration: the former problem is relevant for every entity that interacts with the AI system, since personal information should be provided to properly apply the mitigation technique; the latter draws the attention on subgroups, for which an equal treatment is not guaranteed as well as the demographic groups considered by the mitigation algorithm. On the other hand, individual fairness solutions are based on the idea that similar users should be treated equally, but it is not straightforward to define a correct similarity notion.

My goal is based on the idea that unfairness could be solved by guaranteeing the same accuracy, quality for each user: this definition does not need notions of similarity and does not need any demographic information. Aiming at giving outcomes of same quality for all the individuals would guide the learning process to not advantage specific users and not depend on sensitive information. Never-

theless, demographic data could be relevant in some situations, e.g. in RecSys when the "items" to recommend are people, such as candidates recommended to a recruiter, where the diversity of gender, age or other demographic attributes is important to guarantee fair outcomes; hence, even if the users were treated equally, some indirect discrimination could be present because of the order of the candidates.

My research into RecSys laid the foundations to support this goal idea by analysing some important aspects regarding the causes behind the unfairness of the models. In particular, the data and training features have been studied to evaluate their influence on the outcomes, but also the learning process has been examined to find rules to understand how RecSys outcomes evolve over the training epochs. On the other hand, I have already accomplished some results in the other field of my research focus: speaker recognition systems. The study of such systems is reinforced by a framework that includes state-of-the-art ASR systems, automatic data pipelines and functionalities to work on beyond-accuracy objectives. This framework has been used to produce two publications:

1. *Improving Fairness in Speaker Recognition* [12]: in this work, I and my colleagues applied a balancing strategy of the training set based on two demographic attributes and evaluated the fairness improvement in terms of disparity of equal error rate on two state-of-the-art neural networks. It has been presented at the Symposium on Pattern Recognition and Applications (SPRA) 2020 and has been selected as the best presentation of the conference.
2. *FairVoice Biometrics: Impact of Demographic Dataset Imbalance on Group Fairness in Deep Speaker Recognition* [11]: this paper extends the first one by considering a third neural network and evaluating the fairness improvement of the balancing strategy in terms of some state-of-the-art fairness metrics, extracted from [15]. It has been presented at the Interspeech 2021 conference.

3 Research Plan

The purpose of this document is to highlight the problems related to several unfairness mitigation approaches and to propose a different point of view of the fairness concern. In particular, recent literature tackles the unfairness by relying on sensitive information, which can harm the privacy of the users and make the systems dependant on demographic attributes. This research plan aims to examine these studies and approach the problem without taking into account private data and without using tools to find similarities among users.

Currently, my main focus is fairness in recommendation systems and automatic speaker recognition systems: the first step has been the analysis of different solutions of the literature of fairness in these two fields and in machine learning in general, from which I adjusted and perfected my idea. The plan is based on the goal described in Section 2 and it is structured on the following research questions:

– RQ1: Can the addition of n sensitive attributes $S = \{s_i | i \in [1::n]\}$ reveal a systematic unfairness against the new protected/unprotected groups when a mitigation approach is applied on demographic information not in S?
– RQ2: Can the prediction of missing or corrupted labels of sensitive attributes lead to decisions that damage the image of the users?
– RQ3: To what extent the continual learning can help on mitigating the unfairness by guaranteeing the same service quality to all the users?

The first question aims to understand if the state-of-the-art innovations in group fairness can guarantee a thorough equity among all the demographic groups present in a dataset. Consumers of a recommender system or speakers are humans and given that we can distinguish a multitude of groups and subgroups: satisfying the fairness for all of these groups and subgroups would be practically impossible, made harder by the fact that new demographic groups establish in society day after day. The second question aims to highlight the risk of predicting missing or corrupted labels and the damage that these actions could cause to people. This strategy is applied in several works of the literature [1, 17, 24] and even if it is affirmed that these predictions cannot be perfectly accurate, studies on bias are carried on using mispredicted labels that could damage the image of the users. The third question aims to understand the strength of continual learning for obtaining equal quality among all users, which, as stated before, would guarantee fairness. Continual learning makes possible to not forget previous knowledge and fine tune the model: these are features that can be combined with an equalization rule that guide the learning process on assigning equal quality.

My next activities will be guided by the aforementioned research questions:

1. Task 1: many studies leverage sensitive information to mitigate the unfairness of machine learning systems. All of these solutions will be gathered, reproduced and studied using datasets with several sensitive attributes. This approach permits to evaluate the unfairness impact on demographic groups that have not been taken into account by the mitigation algorithm, since the unfairness against subgroups, e.g. black Asian females, white African males, needs to receive attention as well.
2. Task 2: in every field of the machine learning literature there are works that make use of labels prediction. Missing or corrupted data can be solved by predicting the labels, but this could be dangerous and harmful for the users. All the predictions approaches and the unfairness mitigation techniques used with predicted labels will be gathered and reproduced to evaluate the consequences of mispredicted labels.
3. Task 3: machine learning models can be trained on a part of a dataset and the other part could be sliced in pieces in order to train the model with a new piece at a time without forgetting the previous training phases. This approach could be strengthened by a regularisation rule that aims to equalise the quality offered to all the users of the dataset. This activity will study the benefits of such an approach and what fairness goals could be reached by applying this technique on several datasets.

These listed points make the basis for the writing of scientific articles, aiming at publications on international high level journals and presentations at the most important conferences. The conferences targets are the World Wide Web (WWW) Conference, the ECIR European Conference on Information Retrieval, the International ACM SIGIR Conference on Research and Development in Information Retrieval, the Conference on Information and Knowledge Management (CIKM), the Conference on User Modeling, Adaptation, and Personalization (UMAP), the ACM International WSDM Conference, the International Conference on Web and Social Media (ICWSM), the International Conference on Machine Learning (ICML), INTERSPEECH Conference, the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). The journal targets are the Journal on Information Processing & Management, the Journal on User Modeling and User-Adapted Interaction, the Journal on Intelligent Information Systems Springer, the Journal on Intelligent Systems IEEE, the Journal on Knowledge-Based Systems Elsevier, the IEEE Signal Processing Magazine.

I will give the possibility to other researchers and non-experts to examine the positive impact of my studies by also creating web-based demos or ready-to-use programs, which will be presented in conferences to show the benefits that my projects could bring to the fairness of ML solutions. I expect my works to be integrated into real applications in order to guarantee fairer services to users and prevent issues of discrimination in ML models. Satisfying these goals means creating fairer models, which will socially improve the daily experience with such systems, treating everyone equally.

References

1. Awasthi, P., Beutel, A., Kleindessner, M., Morgenstern, J., Wang, X.: Evaluating fairness of machine learning models under uncertain and incomplete information. pp. 206–214. Association for Computing Machinery (2021). https://doi.org/10.1145/3442188.3445884
2. Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E.H., Goodrow, C.: Fairness in recommendation ranking through pairwise comparisons. pp. 2212–2220. Association for Computing Machinery (2019). https://doi.org/10.1145/3292500.3330745
3. Black, E., Fredrikson, M.: Leave-one-out unfairness. pp. 285–295. Association for Computing Machinery (2021). https://doi.org/10.1145/3442188.3445894, https://doi.org/10.1145/3442188.3445894
4. Bobadilla, J., Lara-Cabrera, R., Ángel González-Prieto, Ortega, F.: Deepfair: Deep learning for improving fairness in recommender systems. CoRR **abs/2006.05255** (2020)
5. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: Deep speaker recognition. In: INTERSPEECH (2018)
6. Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A.: Algorithmic decision making and the cost of fairness. pp. 797–806 (2017)
7. Deldjoo, Y., Anelli, V.W., Zamani, H., Bellogín, A., Noia, T.D.: A flexible framework for evaluating user and item fairness in recommender systems. User Modeling and User-Adapted Interaction (2021). https://doi.org/10.1007/s11257-020-09285-1

8. Diaz, F., Mitra, B., Ekstrand, M.D., Biega, A.J., Carterette, B.: Evaluating stochastic rankings with expected exposure. CoRR **abs/2004.13157** (2020)
9. Ekstrand, M.D., Tian, M., Azpiazu, I.M., Ekstrand, J.D., Anuyah, O., McNeill, D., Pera, M.S.: All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. vol. 81, pp. 172–186. PMLR (6 2018)
10. Fenu, G., Lafhouli, H., Marras, M.: Exploring algorithmic fairness in deep speaker verification. Computational Science and Its Applications – ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part IV **12252**, 77–93 (8 2020). https://doi.org/10.1007/978-3-030-58811-3_6
11. Fenu, G., Marras, M., Medda, G., Meloni, G.: Fair Voice Biometrics: Impact of Demographic Imbalance on Group Fairness in Speaker Recognition. In: Proc. Interspeech 2021. pp. 1892–1896 (2021). https://doi.org/10.21437/Interspeech.2021-1857
12. Fenu, G., Medda, G., Marras, M., Meloni, G.: Improving fairness in speaker recognition. pp. 129–136. Association for Computing Machinery (2020). https://doi.org/10.1145/3393822.3432325
13. de Freitas Pereira, T., Marcel, S.: Fairness in biometrics: a figure of merit to assess biometric verification systems. CoRR **abs/2011.02395** (2020), https://arxiv.org/abs/2011.02395
14. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Recommendation independence. vol. 81, pp. 187–201. PMLR (6 2018)
15. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. arXiv e-prints p. arXiv:1908.09635 (8 2019)
16. Mehrotra, A., Celis, L.E.: Mitigating bias in set selection with noisy protected attributes. pp. 237–248. Association for Computing Machinery (2021). https://doi.org/10.1145/3442188.3445887
17. Pujol, D., McKenna, R., Kuppam, S., Hay, M., Machanavajjhala, A., Miklau, G.: Fair decision making using privacy-protected data. pp. 189–199. Association for Computing Machinery (2020). https://doi.org/10.1145/3351095.3372872
18. Sacharidis, D., Mouratidis, K., Kleftogiannis, D.: A common approach for consumer and provider fairness in recommendations. vol. 2431, pp. 1–5. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2431/paper1.pdf
19. Serna, I., Morales, A., Fierrez, J., Cebrian, M., Obradovich, N., Rahwan, I.: Sensitiveloss: Improving accuracy and fairness of face representations with discrimination-aware deep learning. arXiv preprint arXiv:2004.11246 (2020)
20. Serna, I., Peña, A., Morales, A., Fierrez, J.: Insidebias: Measuring bias in deep networks and application to face gender biometrics. pp. 3720–3727 (2021)
21. Singh, A., Joachims, T.: Fairness of exposure in rankings. pp. 2219–2228. Association for Computing Machinery (2018). https://doi.org/10.1145/3219819.3220088
22. Tsintzou, V., Pitoura, E., Tsaparas, P.: Bias disparity in recommendation systems. arXiv e-prints p. arXiv:1811.01461 (11 2018)
23. Verma, S., Rubin, J.: Fairness definitions explained. pp. 1–7. Association for Computing Machinery (2018). https://doi.org/10.1145/3194770.3194776
24. Wang, J., Liu, Y., Levy, C.: Fair classification with group-dependent label noise. pp. 526–536. Association for Computing Machinery (2021). https://doi.org/10.1145/3442188.3445915

Log-based behavioural differencing applied to an industrial case

Céline Deknop

ICTEAM institute, UCLouvain, Belgium
celine.deknop@uclouvain.be

Abstract. When working with a complex process, it is difficult to get a clear idea on how exactly changes to the input can impact the output. Visualising how the steps of such process evolves with input can help understanding and/or boost confidence in the produced result.
We took a suitable existing industrial process, and created a visualisation for it using an adapted log-based behavioural differencing algorithm that highlights the parts of the process that really changed. We show that our visualisation scales well, propose that our solution could be applied to other use cases, and provide a replication package to allow others to try for themselves.

Keywords: Differencing, Visualisation, Refactoring, Legacy, Migration, Industrial

1 Introduction

This paper, presented as a poster at Informatics Europe's first *Early Career Researchers Workshop* during its annual ECSS2021 conference, reports on the first 2 years of my PhD research. This research was conducted in the context of an Innoviris AppliedPhD project, in collaboration between UCLouvain university and the Raincode Labs company in Belgium. The goal of such projects is to stimulate research while keeping in touch with the reality of the industry. In this particular project, we explore the discipline of code differencing in an industrial context.

We start by introducing the chosen discipline of differencing and present its limitations (Section 2). Next, we introduce Raincode Labs and the use case we will focus on Section 3. Section 4 then presents the technique that we used along with the limitations that we encountered when applying it to the industrial data obtained from Raincode Labs. In Section 5, we present some improvements over the first implementation and reactions from our validation participants. Finally, Section 6 gives a brief conclusion, an overview of possible future work, and provides a link to a replication package that can be used by anyone wishing to give our tool a try.

2 A brief history of differencing

The algorithm used to perform differencing that we are most accustomed to (e.g., to discover differences between two commits on GitHub) has been created in 1976 by Hunt and McIlroy [7] and has stayed mostly unchanged since. While this algorithm remains quite limited, more advanced techniques might benefit a software company like Raincode.

We therefore explored what techniques are out there, and how they could be applied to the reality of Raincode's day-to-day business. Some techniques, like Kim and Notkin's *LSdiff* [8], while interesting, seemed too object-oriented for our specific use case. We explored a bit the modelling community (UMLDiff [11], ADDiff [9]), as well as the domain of code clones and mining (ROSE [13], CloneDiff [12]), before deciding on the technique we will detail in this paper: log-based behavioural differencing, as proposed by Goldstein et al. [6].

3 Raincode Labs and their migration projects

Raincode Labs is an independent compiler company that provides services for migration and modernisation of legacy systems. Among thoses services, we will focus on one: PACBASE migration [10].

PACBASE is an aging fourth generation language that allows engineers to use concise macros to generate COBOL code, instead of writing COBOL code directly. PACBASE support having ended in 2015, reliance on it has turned into a liability. Ideally the language would be retired, and companies could maintain the generated COBOL code. Yet, the COBOL code generated by PACBASE is not the most readable for humans, and rewriting all COBOL code from scratch is not feasible.

Raincode's PACBASE migration refactors PACBASE-generated COBOL code into human-readable COBOL using a set of refactoring rules that are applied iteratively on the codebase, producing new COBOL files that are maintainable by the company. Since the code being migrated is often a critical part of the core business of the client company, it is important for Raincode Labs to ensure that the client will feel "at home" in the new code. For that, the exact set of refactoring rules can be tailored to the client's needs.

Picking the exact set of refactoring rules is the first step of a migration project. First, all of them are presented to the client, so they have an idea of what the rules achieve. A first set of rules is then picked following advice from Raincode engineers. Next, a small subset of programs are migrated by Raincode engineers using the chosen set of rules. This allows the client to look at the result of a migration to get a clearer idea of what it does. After that and again with the help and advice of Raincode engineers, the set of rules can be refined. The partial migration is run again to see if the output is better, and some iterations of this process can happen until a client is fully satisfied.

This part of the process is precisely where our application of advanced differencing could be applied: there are around 140 refactoring rules available, and

while Raincode engineers are experts in the domain and have great intuition, it is sometimes difficult for them to give a precise justification of why a specific rule should be turned on or off. It is also difficult for them to explain exactly how certain changes to the chosen rule set would impact the output, because all rules can interact with one another. To help in this matter, Raincode engineers would like to have more objective arguments as to why a rule should be picked or not, and our tool could help to do just that.

4 Log-based behavioural differencing

The initial idea we based our tool upon is explained in detail in Goldstein et al.'s paper [6]. It can be summarised as follows:

1. Take logs from two executions of a system you want to analyse, and sanitise them (remove any information you don't want in the final graphs).
2. Create a Finite State Automaton (FSA) from those logs. The original authors advise to use the KTails algorithm [2], which is what we did. Many tools exist for this, we chose Synoptic [1].
3. Compare the resulting FSAs using the algorithm proposed by Goldstein et al.
4. Visualise the resulting differences.

This approach is easily applicable to Raincode's migration projects since they already generate logs describing the exact order in which rules got triggered, along with some more details (timestamp, exact lines of code on which the rule was applied). In our case, we decided to use only the names of the rules being triggered, and exclude any errors or warnings.

As for the visualisation, we decided to keep close to the idea of Goldstein et al., keeping a graph structure with START and END nodes, and transitions between nodes representing the probabilities to change from one state to another in the automata. However, we decided to take it a little further, with the addition of colours (red/orange/green for a removal/modification/addition).

We therefore set out by reproducing the original algorithm and applying it to our industrial data (for more details about the original algorithm and our results, see our VISSOFT paper [5]). The results of this first experiment were a bit disappointing: with our log files having a mean of around 280 lines, even after their transformation to an automaton, our resulting graphs contained on average 120 nodes, which was too large to analyse at a glance, as can be seen on Figure 1.

5 Shrinking the graphs and getting feedback

After closely observing our first outputted graphs, we noticed a trend that could be exploited to make them more readable: due to the very iterative nature of Raincode's migration process, our graphs had very long lines of nodes without

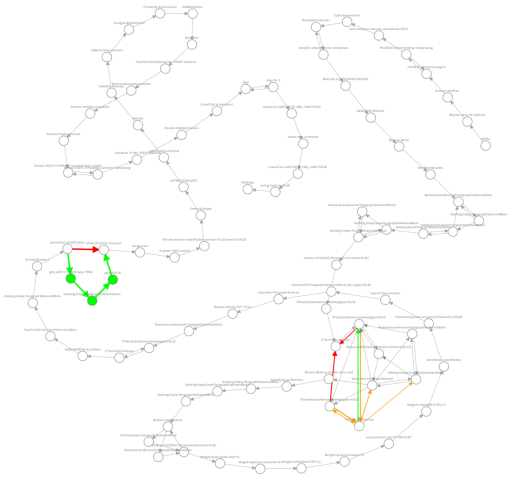


Fig. 1. Result of applying the algorithm from [6] as-is.

changes, that felt uninteresting and took up unnecessary space. We also noticed that most changes happened in only a few places of the graphs, in clusters.

Those observations lead us to create a merging algorithm to hide away uninteresting nodes, enabling us to highlight the more interesting information. Two nodes can be merged if:

- None of them are added or removed;
- No link towards them has seen a probability change;
- No link from them has seen a probability change.

When applying this merge algorithm to our graphs, we managed to reduce them from a mean of 123 nodes, to one of 33 nodes, a 75% reduction. We also decided to not just drop the nodes that were merged: they merely get hidden and can be inspected in a separate window, when the merged node is clicked. The resulting graphs felt readable enough to present them to Raincode engineers and get their feedback. For example, the merged version of the graph of Figure 1 is shown in Figure 2.

These graphs were shown to Raincode engineers during semi-structured interviews in which we focused on getting their opinion both on the usability of our tool in the context of their work, and in a more general way outside of their work. We also included some open question to allow them to express their likes and dislikes (details about our method, the exact questions and their results can be found at [4]).

The first result that jumps out is that both of them agreed that the merged version of the graphs was a strict improvement over the non-merged one. After seeing both versions, neither of them wished to use the unmerged variant, even if they would occasionally want to click on a merged node to see what is inside.

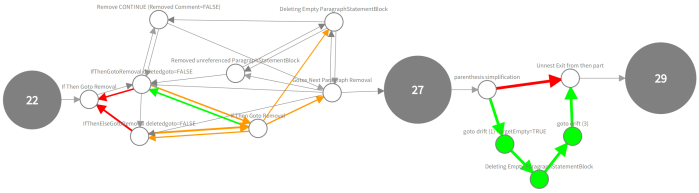


Fig. 2. Our merged graph (same input as 1)

The second result is that our tool seems pretty intuitive: with the help of a short two-page manual, both engineers were able to get situated and stop referring to the manual after 5 to 6 minutes. We did identify two points that required some verbal explanation on our part. First, the meaning of the orange color, its explanation in the user manual being too scientific to be user-friendly. Second, the feature of seeing inside a node when clicking on it was not intuitive enough for them to try and click, and should have been put forward more clearly.

Finally, when asked how exactly they would use our tool in their work, both interviewed participants came up with different ideas: one would use it as described earlier, to give justifications to a client, but the other (less client-oriented in those projects), would use it as a debugging aid, to make sure that changes to the migration process itself didn't have unwanted consequences.

6 Conclusion and replication

To conclude, we believe we produced a tool that can be used "as is" by our collaborating company, and that it could be adapted to other use cases fairly easily. Both engineers who assessed our tool had different ideas on how it might be helpful to their company, which suggests it is fairly versatile. Since finishing our validation, we also had the occasion to discuss about the tool and its purpose with several people and came up with various ideas, from the straightforward idea of comparing logs from boot executions of two Linux distributions to determine whether the second would be a good substitute for the first, to the more challenging idea of using our representation to analyse how a development team sticks to a development method (like scrum).

A good path to future work would be to use our tool in a completely different context to see how useful it can be there, and perform a second round of validation. For this, if anyone wishes to try, we have created a full replication package, with a VM and our tool installed, along with an obfuscated version of our data (Raincode's client data being confidential). We also provide documentation on how to obtain the input from another source of log, and how to run the tool on other data. All instructions to find and run the replication package can be found on GitHub [3], or else feel free to contact us.

Acknowledgements

Thanks go out to my advisors Kim Mens (UCLouvain) and Vadim Zaytsev (UTwente), the people at Raincode Labs: Johan Fabry, Yannick Barthol and Boris Pereira, and finally to Alexandre Bergel. We also thank the Innoviris funding agency for funding our CodeDiffNG Applied PhD research project.

References

1. I. Beschastnikh *et al.*, "*Synoptic GitHub page*," Online: https://github.com/ModelInference/synoptic, 2021.
2. A. W. Biermann and J. A. Feldman, "On the Synthesis of Finite-State Machines from Samples of Their Behavior," *IEEE Transactions on Computers*, vol. C-21, no. 6, pp. 592–597, 1972, DOI: 10.1109/TC.1972.5009015.
3. Céline Deknop, "Replication package and instructions," https://github.com/CelineDknp/VISSOFTArtifact, 2021.
4. ——, "Validation data," https://github.com/CelineDknp/PACBASEValidationData, 2021.
5. C. Deknop, K. Mens, A. Bergel, J. Fabry, and V. Zaytsev, "A scalable log differencing visualisation applied to cobol refactoring," in *2021 Working Conference on Software Visualization (VISSOFT)*, 2021, pp. 1–11.
6. M. Goldstein, D. Raz, and I. Segall, "Experience Report: Log-Based Behavioral Differencing," in *Proceedings of the 28th International Symposium on Software Reliability Engineering (ISSRE)*, 2017, pp. 282–293, dOI: `10.1109/ISSRE.2017.14`.
7. J. W. Hunt and M. D. McIlroy, "An algorithm for differential file comparison," no. #41, 1976.
8. M. Kim and D. Notkin, "Discovering and Representing Systematic Code Changes," in *Proceedings of the 31st International Conference on Software Engineering (ICSE)*. IEEE, 2009, p. 309–319. [Online]. Available: https://doi.org/10.1109/ICSE.2009.5070531
9. S. Maoz, J. O. Ringert, and B. Rumpe, "ADDiff: Semantic Differencing for Activity Diagrams," in *Proceedings of the 19th Symposium on the Foundations of Software Engineering and the 13rd European Software Engineering Conference (FSE)*, T. Gyimóthy and A. Zeller, Eds. ACM, 2011, pp. 179–189.
10. Raincode Labs, "PACBASE Migration: Flexible Process," https://www.raincodelabs.com/pacbase/, 2021.
11. Z. Xing and E. Stroulia, "UMLDiff: An Algorithm for Object-Oriented Design Differencing," in *Proceedings of the 20th International Conference on Automated Software Engineering (ASE)*. ACM, 2005, pp. 54–65.
12. Y. Xue, Z. Xing, and S. Jarzabek, "Clonediff: semantic differencing of clones," in *Proceeding of the Fifth ICSE International Workshop on Software Clones (IWSC)*, J. R. Cordy, K. Inoue, S. Jarzabek, and R. Koschke, Eds. ACM, 2011, pp. 83–84. [Online]. Available: https://doi.org/10.1145/1985404.1985428
13. T. Zimmermann, P. Weibgerber, S. Diehl, and A. Zeller, "Mining version histories to guide software changes," in *Proceedings. 26th International Conference on Software Engineering*, 2004, pp. 563–572.

www.informatics-europe.org