A Data-driven Framework to Facilitate Automated Requirements Engineering

Sachiko Lim[1]

Supervisors: Jelena Zdravkovic, Aron Henriksson
[1]Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden
sachiko@dsv.su.se

Abstract. Traditionally, requirements engineering has been stakeholder driven. With the advent of digital technologies, unprecedented amounts of data are continuously generated. Although such dynamic data are not created with the intention of eliciting requirements, they may include information about system requirements, including up-to-date user requirements, which would be difficult to obtain with traditional elicitation methods. Thus, in addition to domain knowledge, dynamic data from unintended digital sources can potentially serve as valuable requirements sources and support automated and continuous requirements engineering. However, most previous efforts to automate the requirements engineering process have focused on eliciting requirements from domain knowledge that are relatively static, or partially supporting automation of specific requirements engineering activities. There is, thus, a lack of a holistic framework to automate the requirements engineering process that is driven by dynamic data from unintended digital sources. To address this research gap, the PhD study aims at developing a novel and holistic framework for automating data-driven requirements engineering. A design science approach will be used to develop the envisaged framework. This paper reports on the research progress based on the first six months of the PhD study, which includes explicating research problems, formulating research questions, and presenting an initial overview of the envisaged framework as well as preliminary results of a systematic review on the state-of-the-art automated methods for eliciting requirements from dynamic data. The framework will support efficient and effective inclusion of important and relevant requirements for improving existing or developing new software systems.

Keywords: Requirements engineering, Big Data, Automation

1 Introduction

Successful development of information systems depends on the quality of requirements engineering. Poor-quality requirements engineering can result in scope creep, cost overrun, and project failure. According to a survey conducted by the Standish group, only 16.2% of the software projects completed on time and budget, while about 30% of the projects were withdrawn before completion and more than 50% of the projects cost nearly twice their original estimates [1].

Requirements engineering consists of three core activities: elicitation, documentation, and negotiation [2]. The main aim of the elicitation activity is to identify all the stakeholder requirements and translate them into functional and non-functional requirements. Requirements elicitation comprises three sub-activities: 1) identification of relevant sources of requirements, 2) elicitation of requirements from the identified sources, and, optionally, 3) elicitation of innovative requirements. The documentation activity aims to document the results of each requirements engineering activity, conforming to documentation rules and guidelines. Negotiation aims to achieve agreement among all stakeholders by resolving conflicts of needs and wishes.

In addition to the three core activities, two cross-sectional activities are performed in requirements engineering: validation and management. Validation aims to assess the quality of the outputs from the core activities in accordance with defined quality criteria. Management aims to deal with requests of requirements changes, to establish traceability among requirements, and to prioritize requirements [2].

Traditional requirements engineering has largely depended on domain knowledge that are obtained from stakeholders. With the capabilities of e- and mobile commerce as well as the advent of IoT, the digitalization of organizations and societies at large has been widespread, generating an unprecedented amount of high-velocity and heterogeneous data, which is often referred to as Big Data [3]. This digital transformation has spawned new opportunities to consider dynamic data from digital sources as potentially valuable sources of requirements, in addition to domain knowledge. Crowd-based requirements engineering (CrowdRE) is a good example that has taken advantages of such new opportunities. In CrowdRE, large amounts of implicit and explicit feedback from crowd users are embraced to elicit and mange requirements to facilitate user-centered and continuous software development and evolution [4]. The SUPERSEDE toolsuite supports combined analyses of end-user feedback and contextual data on software products that are collected using multi-modal feedback gathering techniques [5]. Harnessing both traditional and new data sources can complement each other to improve the quality of existing, or facilitate development of new, software systems [6].

In this study, *dynamic data is defined as raw data available in a digital form that changes frequently and has not already been analyzed.* Dynamic data certainly includes but is not limited to Big Data. However, it excludes static domain knowledge that is less frequently created or modified. Domain knowledge can be derived from internal (e.g., intellectual property, business documents, existing system's specifications, and goals) or external sources (e.g., standards, conferences, and knowledge from customers or external providers). *Unintended digital sources are defined as sources of data generated via digital technologies that are unintended with respect to requirements elicitation. Thus, dynamic data from unintended digital sources are the changeable digital data pulled from data sources that are created without the intention of eliciting requirements.* Of note is that the two terms "dynamic data" and "unintended digital source" together define the scope of this research. For example, although domain documents are often created without the intention of performing requirements engineering, they are not considered as dynamic data, thus are excluded.

Dynamic data from unintended digital sources can be classified into three data types: human-sourced information data (e.g., social networks, blogs, internet searches on

search engines, contents from mobile phones), process-mediated data (e.g., electronic health records, commercial transactions, credit card payments) and machine-generated sources (e.g., sensor readings, mobile phone locations, Web logs) [7]. They, thus, comprise a broader range of data sources than explicit and implicit user feedback.

Unintended digital sources can include data relevant for new system requirements which otherwise could not be discovered from other sources. Including such requirements, which a current software system is not supporting, can bring business values in the form of improved customer satisfaction, cost and time reduction, and optimized operations [8]. Focusing on dynamic data also allows for capturing up-to-date user requirements, which in turn enables timely and effective operational decision-making. Moreover, dynamic data from unintended digital sources are machine-readable. Thus, they serve as a good basis for paving new ways for automated and continuous requirements engineering. A fitting requirements engineering approach can provide new opportunities and competitive advantages in a fast-growing market by extracting real-time business insights and knowledge from variety of digital sources.

2 Research Problem and Research Questions

Much effort has been made to facilitate automation of the requirements engineering process in which requirements are primarily derived from domain knowledge that are created by stakeholders. Arguably, either of the following aims have driven the majority of such aforementioned efforts:

1) to elicit requirements from existing domain knowledge which are derived from stakeholders (e.g., natural language (NL) documents [9] and models [10]),
2) to perform specific requirements engineering activities based on existing requirements (e.g., requirements prioritization [11], classification of NL requirements [12], and requirements validation [13]), and
3) to develop support tools to enhance stakeholders' ability or engagement to perform requirements engineering activities based on domain knowledge or existing requirements (e.g., tool-support for collaborative requirements prioritization [14] and requirements negotiation with rule-based reasoning [15]).

Nevertheless, there is a paucity of research on utilizing dynamic data from unintended digital sources to facilitate automation of the requirements engineering process. Moreover, although there are pioneering works enabling automated requirements engineering that are driven by dynamic data from unintended digital sources, by taking the focus on specific activities, such works have only partially supported the automation of the requirements engineering process. There is, thus, a lack of a holistic framework for automating requirements engineering driven by dynamic data from unintended digital sources.

Therefore, the aim of this PhD study is to develop a novel and holistic framework for automated and continuous requirements engineering of dynamic data from unintended digital sources, hereinafter referred to as dynamic data. The following main and sub-research questions were formulated to fill the knowledge gap:

How could the entire requirements engineering activities be efficiently and effectively automated when the requirements sources are dynamic data?

- How can requirements be elicited from dynamic data?
- With the given elicitation approach, how can documentation, negotiation, management and validation be supported through automation?
- How can machine learning techniques be applied to elicit innovative system requirements from dynamic data?

The envisaged framework will contribute to 1) utilizing the IoT and other digital technologies for eliciting system requirements, 2) facilitating efficient and effective inclusion of important and relevant requirements to develop new software systems, or continuously improving existing systems, and 3) alleviating the workload and human errors of requirements engineering by increasing the level of automation.

3 Overview of Research Framework

Fig.1 depicts an envisaged holistic framework for automating dynamic data driven requirements engineering activities that are mapped to activities of traditional requirements engineering. The framework is intended to be used by organizations that concern the evolution and development of software products, especially requirements engineers, to complement their requirements engineering process by considering new sources of requirements.

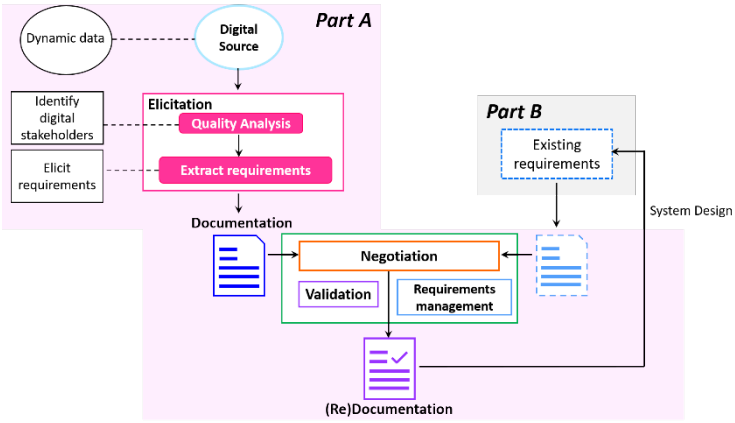

Fig. 1. Envisioned framework for automating dynamic data driven requirements engineering

A cycle of dynamic data driven requirements engineering (Part A) repeats as new data streams in. The process flow of the framework is described in more detail below, together with associated challenges to facilitate automation.

Identify digital stakeholders: This step aims to discover potential data sources for eliciting system requirements, or digital stakeholders. It starts with extracting raw data from one or more data sources. To minimize risks of eliciting wrong requirements, the informativeness of the extracted data will be assessed. A major challenge is to develop source-specific methods to assess the informativeness. Human-sourced information

data are unstructured and include a significant proportion of non-informative data for requirements elicitation. Process-mediated data comprise quality issues such as incompleteness and errors as well as non-representativeness due to non-random sampling, which could produce misleading results and subsequent decisions [16]. Furthermore, it is challenging to combine data from different sources that use different data types, definitions, and periodicities [16]. Machine-generated data are subject to missing data, noise, and errors. It is also hard to choose the frequency of data collection that is sufficient to elicit "good-enough" requirements to control the cost of storing and analyzing data.

Elicit requirements: From the identified digital stakeholders, requirements will be extracted automatically, using different algorithms depending on the type of data sources. Eliciting requirements from data expressed in NL is a challenge due to the ambiguous and unstructured nature. There are pioneering studies which elicited requirements from human-sourced information sources such as twitter and app reviews [17] [18]. Another challenge is to investigate methods to elicit system requirements from process-mediated and machine-generated data that are not expressed in NL. Regardless of data source types, it is difficult to elicit requirements while achieving an optimal trade-off between completeness and correctness. Furthermore, it needs to be considered how to integrate requirements from different types of sources.

Requirements documentation: This step aims to automatically document the elicited requirements and assess their quality. A challenge of this step is to identify appropriate quality assessment criteria for each data source. For human-sourced information data, disambiguation of NL requirements is an issue to be solved. Previous studies applied natural language processing and machine learning techniques to tackle with the ambiguity issues in NL [19] [20]. Likewise, a challenge of using process-mediated and machine-generated data is to determine an optimal set of source-specific quality dimensions such as accuracy, consistency, completeness, and freshness [21–23]. At the end of this step, only the requirements that meet a given quality criteria should be saved, while the others are further analyzed or discarded. However, in what form those "quality" requirements are saved remains to be investigated.

Requirements elicitation and documentation from existing software systems: If domain knowledge is available and accessible, requirements are mined and specified automatically. (Part B in Fig.1). Note that having existing requirements is not a prerequisite for Part A.

Negotiation: The "documented" requirements within and across different sources should be matched to check for redundancy and/or conflicts. Identified redundant and/or conflicting requirements should be harmonized. A main challenge is to determine methods to automate the matching process and investigate how to match requirements from different data sources.

Management: This step aims to prioritize requirements, manage requirements changes, and establish requirements traceability. Main concerns for prioritizations include how to identify factors influencing the priority of requirements, how to optimally reflect different stakeholders' perspectives, and how to perform prioritization while taking into account dependencies among requirements [24, 25]. Main challenges for establishing traceability are: (i) to identify appropriate information retrieval techniques

to link requirements, (ii) to maintain traceability, while managing changes, and (iii) to define a suitable trace granularity [26–28]. Requirements change includes three core processes; change identification, change analysis, and change cost/effort estimation [29]. A major challenge is to develop methods to automate those processes.

Validation: The quality of requirements will be assessed in terms of consistency, completeness, and correctness (3Cs). Challenges include when to validate requirements, how to automate the validation process, and how to characterize properties of 3Cs for each data source.

Documentation of validated requirements: The validated "document" will be stored in a pool of existing requirements. In what form validated requirements should be saved remains to be considered.

4 Research Methodology

The PhD study will use design science approach, because it aims at solving a practical problem by creating a framework for dynamic data driven requirements engineering as an artefact. To address our research questions, we will follow the five activities of the design science research process as described in detail below [30]. Each activity will be performed iteratively to refine the design of the framework.

Explicate problem: Since the requirements engineering process starts with requirements elicitation and the activity is a key driver for successful development of information systems [2], a systematic literature review is ongoing, focusing on automated requirements elicitation. The specific aims of the review are two-fold: 1) to understand the state-of-the-art automated methods to elicit requirements from dynamic data, and 2) to identify gaps in existing methods and requirements for the elicitation process in our envisaged framework.

A comprehensive query-based search was performed in six electronic databases: Scopus, Web of Science, ACM Digital Library, IEEE Xplore, EBSCOhost and ProQuest. Those databases were selected because they together cover the top ten information system journals and conferences [31]. In total, 1390 non-duplicate articles were identified, among which 40 met our inclusion criteria to proceed to full-text screening. We will also conduct literature reviews on existing methods to automate other requirements engineering activities.

Outline artefact and define requirements: The artefact type of the PhD study is a holistic framework to automate dynamic data driven requirements engineering that are outlined in Fig. 1. Functional and quality requirements on the outlined framework will be elicited.

Design and develop artefact: We will first get a number of ideas to be part of the outlined artefact based on critical analysis of existing automated requirements engineering methods. We will then decide which methods should be part of the design and development of the envisaged framework by conducting small-scale experiments that compare the performance of several candidate methods. This will aid evidence-based selection of the methods that best meets our requirements. We will then design and develop a set of automated methods for each requirements engineering activity.

Demonstrate artefact: We will perform an illustrative or real-life case study to show to what extent our designed framework can address the explicated problems as intended. The framework will be further refined and finalized based on the results of the case study.

Evaluate artefact: The proposed framework will be evaluated quantitatively for verification, using a combination of formative and summative evaluations. They focus on the effectiveness and efficiency of the framework. Effectiveness will be assessed in terms of completeness and correctness of algorithms to automate each requirements engineering activity, using standard metrics such as recall and precision, respectively [31]. Efficiency can be assessed by measuring the time that is required for the framework to perform the entire requirements engineering process. The performance of proposed framework will be compared to that of the state-of-the-art artefacts, whenever possible.

5 Conclusion

In addition to conventional domain knowledge, widespread digitalization of organizations and societies at large has created new opportunities to automate requirements engineering activities that are driven by dynamic data. There is, thus, a growing need of a framework to harness such fast-growing and large amounts of data for requirements engineering and gain near real-time insights and knowledge out of them. Nevertheless, previous studies have disproportionately focused on eliciting requirements from static domain knowledge, or on supporting automation of specific activities of requirements engineering. There is a paucity of research on automating the entire requirements engineering process that are driven by dynamic data. Therefore, the ultimate aim of the PhD study is to develop a novel and holistic framework to address the research gap, using design science methodology. As the progress of the first six months of the study, this paper explicated research problems, formulated research questions, and presented an initial overview of the envisaged framework with associated challenges as well as preliminary results of the systematic review on automated requirements elicitation from dynamic data. The framework will help requirements engineers leverage dynamic data to elicit innovative system requirements, facilitate efficient and effective inclusion of important and relevant requirements to develop new software systems, or continuously improve existing systems, and alleviate the workload and human errors of requirements engineering by increasing the level of automation.

References

1. The Standish Group: The Standish Group Report CHAOS, https://www.projectsmart.co.uk/white-papers/chaos-report.pdf.
2. Pohl, K.: Requirements engineering: fundamentals, principles, and techniques. Springer, Heidelberg ; New York (2010).
3. Chen, H., Chiang, R.H.L., Storey, V.C.: Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly. 36, 1165–1188 (2012).

4. Maalej, W., Nayebi, M., Johann, T., Ruhe, G.: Toward data-driven requirements engineering. IEEE Software. 33, 48–54 (2016). https://doi.org/10.1109/MS.2015.153.
5. Perini, A.: Data-Driven Requirements Engineering. The SUPERSEDE Way. In: Lossio-Ventura, J.A., Muñante, D., and Alatrista-Salas, H. (eds.) Information Management and Big Data. pp. 13–18. Springer International Publishing (2019).
6. Groen, E.C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., Hosseini, M., Marco, J., Oriol, M., Perini, A., Stade, M.: The Crowd in Requirements Engineering The Landscape and Challenges. Ieee Software. 34, 44–52 (2017). https://doi.org/10.1109/MS.2017.33.
7. Firmani, D., Mecella, M., Scannapieco, M., Batini, C.: On the Meaningfulness of "Big Data Quality" (Invited Paper). Data Science and Engineering. 1, 6–20 (2016). https://doi.org/10.1007/s41019-015-0004-7.
8. Ferguson, M.: Big Data-Why Transaction Data is Mission Critical To Success. Intelligence Business Strategies Limited. https://public. dhe. ibm. com/common/ssi/ecm/im/en/iml14442usen/IML14442USEN. PDF. (2014).
9. Slankas, J., Williams, L.: Automated extraction of non-functional requirements in available documentation. In: 2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE). pp. 9–16 (2013). https://doi.org/10.1109/NAturaLiSE.2013.6611715.
10. Nogueira, F.A., De Oliveira, H.C.: Application of heuristics in business process models to support software requirements specification. Presented at the ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems (2017).
11. Shao, F., Peng, R., Lai, H., Wang, B.: DRank: A semi-automated requirements prioritization method based on preferences and dependencies. Journal of Systems and Software. 126, 141–156 (2017). https://doi.org/10.1016/j.jss.2016.09.043.
12. Abad, Z.S.H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., Schneider, K.: What Works Better? A Study of Classifying Requirements. Presented at the Proceedings - 2017 IEEE 25th International Requirements Engineering Conference, RE 2017 (2017). https://doi.org/10.1109/RE.2017.36.
13. Kamalrudin, M., Hosking, J., Grundy, J.: MaramaAIC: tool support for consistency management and validation of requirements. Automated Software Engineering. 24, (2017). https://doi.org/10.1007/s10515-016-0192-z.
14. F. Kifetew, D. Munante, A. Perini, A. Susi, A. Siena, P. Busetta: DMGame: A Gamified Collaborative Requirements Prioritisation Tool. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 468–469 (2017). https://doi.org/10.1109/RE.2017.46.
15. Ahmad, S., Jalil, I.E.A., Ahmad, S.S.S.: An Enhancement of Software Requirements Negotiation with Rule-based Reasoning: A Conceptual Model. Journal of Telecommunication, Electronic and Computer Engineering (JTEC). 8, 193–198 (2016).
16. Hand, D.J.: Statistical challenges of administrative and transaction data. Journal of the Royal Statistical Society: Series A (Statistics in Society). 181, 555–605 (2018). https://doi.org/10.1111/rssa.12315.
17. Guzman, E., Ibrahim, M., Glinz, M.: A Little Bird Told Me: Mining Tweets for Requirements and Software Evolution. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 11–20 (2017). https://doi.org/10.1109/RE.2017.88.

18. Chen, N., Lin, J., Hoi, S.C.H., Xiao, X., Zhang, B.: AR-miner: Mining Informative Reviews for Developers from Mobile App Marketplace. In: Proceedings of the 36th International Conference on Software Engineering. pp. 767–778. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2568225.2568263.
19. Gleich, B., Creighton, O., Kof, L.: Ambiguity detection: Towards a tool explaining ambiguity sources. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 6182 LNCS, 218–232 (2010). https://doi.org/10.1007/978-3-642-14192-8_20.
20. Yang, H., Willis, A., De Roeck, A., Nuseibeh, B.: Automatic detection of nocuous coordination ambiguities in natural language requirements. In: Proceedings of the IEEE/ACM international conference on Automated software engineering. pp. 53–62. ACM (2010).
21. Li, X., Dong, X.L., Lyons, K., Meng, W., Srivastava, D.: Truth finding on the deep web: is the problem solved? Proceedings of the VLDB Endowment. 6, 97–108 (2012). https://doi.org/10.14778/2535568.2448943.
22. Manzoor, A., Truong, H.-L., Dustdar, S.: On the evaluation of quality of context. In: European Conference on Smart Sensing and Context. pp. 140–153. Springer (2008).
23. Sha, K., Shi, W.: Consistency-driven data quality management of networked sensor systems. Journal of Parallel and Distributed Computing. 68, 1207–1221 (2008). https://doi.org/10.1016/j.jpdc.2008.06.004.
24. Lehtola, L., Kauppinen, M., Kujala, S.: Requirements prioritization challenges in practice. In: International Conference on Product Focused Software Process Improvement. pp. 497–508. Springer (2004).
25. Gupta, A., Gupta, C.: CDBR: A semi-automated collaborative execute-before-after dependency-based requirement prioritization approach, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054588910&doi=10.1016%2fj.jksuci.2018.10.004&partnerID=40&md5=50e9d383e341cb178e68c1ab401a1efe, (2018). https://doi.org/10.1016/j.jksuci.2018.10.004.
26. Cleland-Huang, J., Settimi, R., Zou, X., Solc, P.: Automated classification of non-functional requirements. Requirements Eng. 12, 103–120 (2007). https://doi.org/10.1007/s00766-007-0045-1.
27. Egyed, A., Grunbacher, P.: Automating requirements traceability: Beyond the record & replay paradigm. In: Proceedings 17th IEEE International Conference on Automated Software Engineering,. pp. 163–171. IEEE (2002).
28. Kannenberg, A., Saiedian, H.: Why software requirements traceability remains a challenge. CrossTalk The Journal of Defense Software Engineering. 22, 14–19 (2009).
29. Jayatilleke, S., Lai, R.: A systematic review of requirements change management. Information and Software Technology. 93, 163–185 (2018). https://doi.org/10.1016/j.infsof.2017.09.004.
30. Johannesson, P., Perjons, E.: An introduction to design science. Springer (2014).
31. Meth, H., Brhel, M., Maedche, A.: The state of the art in automated requirements elicitation. Information and Software Technology. 55, 1695–1709 (2013). https://doi.org/10.1016/j.infsof.2013.03.008.