

What Requirements Engineering can Learn from Process Mining

Mahdi Ghasemi

School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
mghasemi@uottawa.ca

Abstract— Process Mining is an approach that uses event logs of systems or processes and turns them into valuable insights. The main characteristic of process mining techniques is that they focus on and exploit “real behavior” of a large number of stakeholders of a system or of a process. On the other hand, requirements engineering is concerned with requirements elicitation and analysis not only in terms of software specifications but also in terms of activities carried out within an organizational and social context. Furthermore, involving a large number of users/stakeholders has always been a challenge with traditional requirements engineering methods. Although both requirements engineering and process mining have gained increasing research attention, the synergy between these two domains is yet to be exploited. Such a synergy can help both domains benefit from their capabilities and mitigate their own challenges. The ability of process mining to exploit huge data logs can help requirements engineers cope with the above challenge. This paper aims to highlight how requirements engineering can benefit from process mining’s components such as execution logs, process discovery and conformance techniques for requirements elicitation, prioritization and validation.

Index Terms— Process Mining, Goal Modeling, Requirements Engineering, Event Logs, Data-Driven, Goal Mining

I. INTRODUCTION

Systems and *processes* are designed and performed in order to meet their stakeholders’ requirements and to achieve predefined *goals*. Activities that are performed among processes are often logged and stored in databases. These logs can enable process analysts and requirements engineers to optimize processes and to enhance compliance. Finding effective ways to improve processes considering the (often conflicting) stakeholders’ goals is an intricate task.

In the context of requirements engineering (RE), goals are recognized as essential components involved amongst RE processes [1]. Requirements are expected to explain why a system/process is needed based on current or predicted conditions and, also, to clarify how the system/process should be constructed to satisfy its goals [2].

Traditional RE approaches typically involve a limited number of stakeholders (or their representatives) in their activities, e.g. interviews or focus groups. Some advanced approaches, however, allow companies to take advantage of some feedback-gathering channels for direct interaction with key stakeholders.

Market-driven requirements engineering [3], which targets an open marketplace (as customers of a software product) rather than one specific customer, uses such advanced approaches. In current *Market-driven* approaches, however, developers cannot consider the diverse backgrounds of subgroups of users. This is because they do not continuously involve large groups of users who express their feedback over a variety of media [4].

Traditional RE methods face challenges when engaging large quantities of users in requirements engineering activities, especially when they should involve a crowd of software users who are beyond an organization’s reach [4, 5].

The elicitation of requirements from huge collections of data is becoming a research trend aligned with increasing research attention dedicated to big data [6]. Data-driven and user-centered approaches [7] aligned with *crowd-based requirements engineering* (CrowdRE) [4] are automated or semi-automated approaches that aim to collect and analyze data from a crowd to elicit validated user requirements [7, 8]. These approaches consider the identification, prioritization, and management of software requirements using online reviews and feedback about software products that are submitted by users on app stores or social media.

Process mining, situated at the intersection between traditional model-based process analysis and data analysis techniques, is an evidence-based approach that exploits event logs, produced by potentially large quantities of users, and infers valuable insights about processes.

Process mining techniques are classified into three categories: i) *process discovery*, where a model is being created using real event logs; ii) *conformance checking*, where generated data logs or discovered models are compared with the actual data in event logs or prescribed models to find deviations; and iii) *process enhancement*, where an existing process model is improved or/and extended by adding some new aspects [9].

Process discovery [9, 10], as the most prominent process mining techniques, aims to generate a process model, e.g., in BPMN [11], using event logs. It is noteworthy that in process discovery, there is no prior model involved. Event logs capture real behavior of stakeholders and, consequently, the discovered model is expected to represent the underlying process describing such behavior. There exist several discovery algorithms (e.g., α -algorithm [10, 12]) advanced in the process mining community and implemented in specialized tools (e.g., ProM [13]).

Conformance checking, the second category of process mining techniques, is a means to reveal where the real process has deviated from a prescribed or desired model. An existing process model is compared against observed behavior stored in event logs [14].

Process-oriented modeling primarily focuses on “how”, “what”, “where”, “who”, and particularly “when” questions, whereas, goal-oriented modeling focuses on addressing “who”, “what”, and especially “why” questions. Hence, the latter one offers a way to document *intentions* and *rationales*.

The above data-driven RE approaches mainly consider RE in the context of software requirements and the data that they use are the users’ feedback rather than their behavior. The approach introduced in this paper is different as in this approach, RE learns from process mining to use execution logs, process discovery and conformance techniques for requirements elicitation, prioritization and validation.

These two domains can learn from one another and propose synergic benefits that can only be achieved with two domains working together rather than in isolation. A recent systematic literature review [15] shows that such a synergy is yet sparse. This paper highlights the way in which requirements engineers can use process mining capabilities and exploit huge data logs in RE activities. To this end, after the above overview of the context, the state of the art resulting from our systematic literature review [15] will be highlighted in Section II. Then, Section III describes how RE activities can be improved using process mining capabilities. Finally, Section IV discusses future directions and who they could inspire and impact.

II. LITERATURE REVIEW AND STATE OF THE ART

We have conducted a literature review [15] to assess the state of the art at the intersection between process mining and requirements engineering. In that review, we aimed to find the share of goal-oriented and requirement engineering approaches among process mining studies, and to synthesize existing work on joint RE and process mining techniques. Using the main keywords of both domain in the queries we found and reviewed 24 relevant papers from four popular search engines in 2018. Our review exposed three main categories of studies: *goal modeling and requirements elicitation* (10 papers), *intention mining* (10 papers), and *performance indicators* (4 papers).

The first category focuses mainly on the use of process mining capabilities in associated with process goals. However, these studies have been conducted separately (not in a coherent research line) and not necessarily about requirement elicitation using process mining capabilities.

The second category exposes a meaningful line of linked studies on *intention mining* approaches [16]. These approaches shares objectives similar to process mining’s but here the main goal is to discover intentional process models (i.e., goal models), beyond activity process models.

The third category consists of papers that consider key *performance indicators*. This category linked the event logs to indicators to adjust models according to desired deviating behaviors and to assess impacts of process improvements.

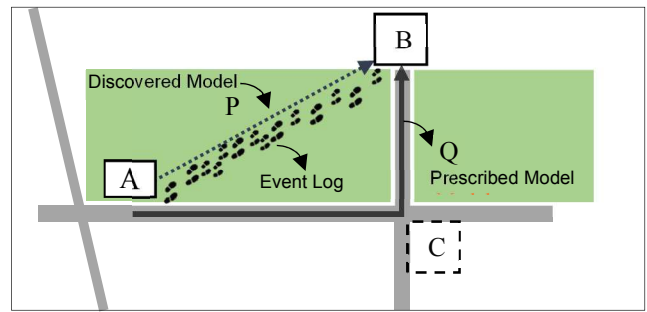


Figure 1 The footsteps of students against the sidewalks

These 24 studies showed considerable potential benefits in using process mining techniques associated with goal modeling and/or requirements elicitation. Yet, such benefits are neglected in current RE approaches.

III. REQUIREMENTS ENGINEERING USING PROCESS MINING

In the requirements engineering’s life cycle, modelers take into consideration the opportunities that stakeholders look for and the vulnerabilities they try to avoid. In general, a requirement is a feature of a system/process necessary to achieve a goal. There may be different plausible approaches to realizing that goal [17]. In a process mining-based approach, a requirements engineer can take advantage of *process discovery* algorithms to find the ways in which stakeholders have realized their goals. The discovered model represents the underlying process model manifesting the stakeholder’s preferences and priorities. Such a discovered model also enables analysts to compare real behavior with prescribed models, if any.

In particular, process discovery techniques use a form of crowdsourcing where event logs are generated from system users executing tasks. This can help requirements engineers generate *as-is* process models that reveal real behaviors recorded in logs. In this sense, process discovery can be employed as a data-driven substitute or complement to use case or scenario elicitation. In scenario-based requirements engineering [18], scenarios represent paths of possible behaviors through a use case, and they are considered to elaborate requirements. In fact, scenario-based RE and process mining both consider ways of generalizing instance-level data into models, except that in a process mining context such data is orders of magnitude larger. Scenario-based techniques (e.g. [19, 20, 21]), however, do not scale to process mining-size problems and the ability of process mining to deal with huge data can offer RE a solution for this challenge.

As a mock example, shown in Figure 1, the footsteps of students walking on campus can be considered as an event log of the process of “going from building A to building B”. The footsteps logged on the lawn (the more frequent the path, the deeper the footsteps) can represent event log. The path P is considered as the model that is discovered from the log. Such a model is not necessarily the same as the prescribed model that may have been previously designed (path Q). In this example, there exists a prescribed model, but there may also exist many cases where a priori models are not available. In such cases, process discovery can generate a model that can serve at least as a valuable start point for eliciting stakeholders’ approaches towards their goals.

When there exists a prescribed model, differences between that model and the discovered one or the logs (e.g., activities in the model not existing in the log or vice versa) can be further analyzed. These deviations can serve as a source for eliciting new requirements or validation and/or modification of the current ones. Accordingly, requirements engineers can use *conformance checking* capabilities to find new system/process requirements to improve the alignment between real practices and desired ones. Frequent deviations in an event log should trigger the requirements engineer to re-assess the suitability of the system's goals, requirements, and processes. In the example shown in Figure 1, the footsteps of students (event log) may not be aligned with the constructed sidewalks (prescribed model). These footsteps may show that the desired way is the shortest path between points *A* and *B*. The goal behind the students here is “to increase the speed” or “to decrease process time”, i.e., some goal (or requirement) that has not been included in the original goal model or documented requirements. As a result, the modeler can update the requirements or/and goal model and modify the prescribed process model to accommodate such a deviation. Conversely, a goal or a requirement that is entirely satisfied could be of more interest for the future.

Let us assume that the students will frequently use the sidewalk (prescribed model $A \rightarrow C \rightarrow B$) on snowing days. At this time there are very rare footsteps on the path $P (A \rightarrow B)$. Students have changed their behavior as their goals' or requirements' priorities are changed. In this case, the goal behind the students' behavior is “to increase safety (from sliding)” or “to decrease the risk of making shoes muddy”. Nevertheless, the requirements “to increase the speed” and “to decrease process time” are yet pursued but their priorities are lower than the safety-related requirements or goals raising in snowy conditions. Frequent deviations and/or conformances in different situations highlight some knowledge of importance about different requirements and goals.

Goal-oriented conformance checking (GoCC), recently introduced in [22], is a data-driven method that aims to find new requirements through event logs, instead of eliciting them from stakeholder goals or requirements documented in goal models.

IV. FUTURE WORK

So far, RE has neglected the potential benefits of process mining [15]. The learning capabilities of process mining can help attain evidence-based definitions of existing systems/processes, their goals and requirements. Such a combination enables RE to measure predefined goal satisfactions and requirements fulfillments [23]. It also enables requirements engineers to assess goal-requirements-process alignment and help them define relevant requirements for their modification or improvement.

Researchers interested in mining goals from event logs (i.e., goal mining) are invited to consider process mining capabilities combined with those of current goal modeling approaches. Also, requirements engineers who are experiencing scenario-based elicitation methods can exploit this combination to make models based on instance-level data logs, which are orders of magnitude larger than the requirements engineers often use. The main research questions in this line of research would be “how to use real data logs for the elicitation/modification of requirements to make

a better process/system?”. Another questions is “how to find the goals behind the stakeholders' activities to make/modify goal models and help improve stakeholder satisfaction?”. Also, “which challenges may raise in the course of using huge data logs to elicitate requirements?”. This approach will be expected to cope with the many challenges recognized in process mining activities [24]. The way in which those challenges can affect this approach is yet to be better considered and addressed. In particular, *data quality*, as the main challenge in process mining research and practice [25], deserves much attention in RE too.

ACKNOWLEDGMENT

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] A. van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in *Proceedings of the IEEE International Conference on Requirements Engineering*, IEEE, 2001, pp. 249-261.
- [2] D. Ross and K. S. Jr, "Structured analysis for requirements definition," *IEEE transactions on Software Engineering*, pp. 6-15, 1977.
- [3] B. Regnell and S. Brinkkemper, "Market-driven requirements engineering for software products," in *Engineering and managing software requirements*, Springer, 2005, pp. 278-308.
- [4] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini and others, "The crowd in requirements engineering: The landscape and challenges," *IEEE software*, vol. 34, no. 2, pp. 44-52, 2017.
- [5] V. Dheepa, D. J. Aravindhar and C. Vijayalakshmi, "A novel method for large scale requirement elicitation," *International Journal of Engineering and Innovative Technology*, vol. 2, no. 7, pp. 375-379, 2013.
- [6] P. Spoletini and A. Ferrari, "Requirements Elicitation: A Look at the Future through the Lenses of the Past," in *25th IEEE International Requirements Engineering Conference*, IEEE CS, 2017, pp. 476-477.
- [7] W. Maalej, M. Nayebi, T. Johann and G. Ruhe, "Toward data-driven requirements engineering," *IEEE Software*, vol. 33, no. 1, pp. 48-54, 2016.
- [8] E. C. Groen, J. Doerr and S. Adam, "Towards crowd-based requirements engineering a research preview," in *International Working Conference on Requirements Engineering: Foundation for Software Quality*, Springer, 2015, pp. 247-253.
- [9] W. van der Aalst, *Process Mining Data Science in Action*, 2 ed., Springer-Verlag Berlin Heidelberg, 2016.
- [10] W. van der Aalst, T. Weijters and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128-1142, 2004.
- [11] OMG, "Business Process Model and Notation (BPMN) Version 2.0," 2011.

- [12] J. Li, D. Liu and B. Yang, "Process Mining: Extending alpha-Algorithm to Mine Duplicate Tasks in Process Logs," *Advances in Web and Network Technologies, and Information Management*, pp. 396-407, 2007.
- [13] "ProM Tools," Process Mining Group, Eindhoven Technical University, [Online]. Available: <http://www.promtools.org/>. [Accessed 2018].
- [14] A. Rozinat and W. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64-95, 2008.
- [15] M. Ghasemi and D. Amyot, "From Event Logs to Goals: A Literature Review of Goal-oriented Process Mining," *Requirements Engineering*, 2018. Submitted.
- [16] G. Khodabandelou, C. Hug, R. Deneckere and C. Salinesi, "Process mining versus intention mining," *Lecture Notes in Business Information Processing*, vol. 147, pp. 466-480, 2013.
- [17] P. Ralph, "The illusion of requirements in software development," *Requirements Engineering*, vol. 18, no. 3, pp. 293-296, 2013.
- [18] A. Sutcliffe, "Scenario-based requirements engineering," in *11th IEEE International Requirements Engineering Conference (RE03)*, IEEE CS, 2003, pp. 320-329.
- [19] H. Saiedian, P. Kumarakulasingam and M. Anan, "Scenario-based requirements analysis techniques for real-time software systems: a comparative evaluation," *Requirements Engineering*, vol. 10, no. 1, pp. 22-33, 2005.
- [20] A. G. Sutcliffe, N. A. Maiden, S. Minocha and D. Manuel, "Supporting scenario-based requirements engineering," *IEEE Transactions on software engineering*, vol. 24, no. 12, pp. 1072-1088, 1998.
- [21] E. Ramezani Taghiabadi, D. Fahland, B. F. van Dongen and W. M. P. van der Aalst, "Diagnostic Information for Compliance Checking of Temporal Compliance Requirements," in *International Conference on Advanced Information Systems Engineering*, Springer, 2013, pp. 304-320.
- [22] M. Ghasemi, "Towards Goal-oriented Process Mining," in *Requirements Engineering Conference (RE), 2018 IEEE 26th International*, IEEE CS, 2018.
- [23] S. F. Gohmann, J. Guan, R. M. Barker and D. J. Faulds, "Requirements Fulfillment: A Missing Link between Requirements Determination and User Acceptance," *Information Systems Management*, vol. 30, no. 1, pp. 63-74, 2013.
- [24] W. van der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen and J. Buijs, "Process mining manifesto," in *International Conference on Business Process Management*, Springer, 2011, pp. 169-194.
- [25] M. Ghasemi and D. Amyot, "Process mining in healthcare: a systematised Literature Review," *International Journal of Electronic Healthcare*, vol. 9, no. 1, pp. 60-88, 2016.