

Towards Goal-oriented Process Mining

Mahdi Ghasemi

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

Abstract— The growth of our digital world makes it possible to record many types of events. In particular, the number of business processes whose events are being logged is significantly increasing. Process mining is an approach that exploits event logs to discover real processes executed in organizations, enabling them to (re)design and improve process models. Goal modeling, on the other hand, is a requirements engineering approach mainly used to analyze what-if situations and support decision making. Common problems with process mining include the complexity of discovered “spaghetti” processes and a lack of goal-process alignment. Current process mining practices are mainly oriented towards activities and do not profit from considering stakeholder goals and requirements to manage complexity and alignment. Involving goal-related factors can augment the precision and interpretability of mined models and lead to better opportunities to satisfy stakeholders. In this research, I propose two methods (for goal-oriented process and enhancement discovery, and for conformance checking) that show potentially synergetic effects achievable by combining process mining and goal-oriented modeling. The paper reports on the research method, the gap/problems being addressed, preliminary solutions, expected contributions, and main foreseen challenges.

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

I. INTRODUCTION AND MOTIVATION

Actors perform activities within business processes to reach their goals. The activities are observed as they are recorded in event logs. These logs, resulting from the execution of processes, might be characterized by various attributes such as identifiers, timestamps, activity names, transaction types (start, complete), resources, and associated costs. These logs are the main source of every study that aims to take *real-world processes* into consideration. *Process mining* is a maturing discipline, built upon process model-driven approaches and data mining, that is concerned with exploiting event data [1]. Various methods, tools and algorithms developed within the process mining community enable the analysis of event logs primarily through discovering underlying processes and creating process models. Process mining techniques are classified into three categories: (1) *Discovery*, where a model is being created using event logs; (2) *Conformance checking*, where the data generated from the model is compared with the actual data in event logs to discover differences between the model and reality; and (3) *Enhancement*, where the desired data is used to improve or/and extend an existing process model.

In the context of requirements engineering, *goal modeling* is used to support heuristic, qualitative, or formal reasoning about stakeholder and system goals, and ultimately what-if/trade-off analysis and decision making [2]. While process mining is an activity-oriented approach and therefore focuses on “*how*”, “*what*”, “*where*”, “*who*”, and especially “*when*” questions, goal-oriented modeling focuses mainly on answering “*why*” questions that are not considered by process mining [3].

Process mining techniques assume that cases within logs of a process pursue some similar goals and do not consider specific goals of individual cases or their satisfaction levels. This situation threatens the rationality behind the discovered models on one hand, and results in unstructured “spaghetti-like” processes on the other hand. Although they reflect reality, unstructured processes cover many exceptions, which hurts model understanding.

The process mining literature suggests some strategies for dealing with unstructured discovered processes [1]. The main idea is to take into account the frequency of activities and carefully filter out infrequent activities traces from the log before discovery. For example, keeping the activities that occur in at least 20% of cases is a way to simplify the model. In contrast to filtering strategies that change the log, abstraction techniques such as fuzzy mining [1] can be directly applied to the resulting process graph.

Goal-oriented approaches offer a way to document intentions and rationales, leading modelers to consider opportunities that stakeholders look for and vulnerabilities that they try to avoid. Such approaches also support answering “*what*” questions in a way that helps identify capabilities, services, and architectures required to satisfy stakeholder goals [2].

Therefore, a goal-oriented approach combined to process mining activities looks promising not only to identify capabilities, services, and architectures, but also to document and exploit goals and rationales and leverage them to improve process models and their realization. Process models that are aligned with goals are expected to produce high levels of goal satisfaction.

In this research, goals and intentions behind the activities within a business process will be considered. The objective in this work is to develop a goal-oriented process mining framework concerned not only with the sequencing of activities (as is conventionally the case in process mining), but also with process goals and satisfaction indicators.

This research is conducted in three phases. The first phase involves two systematic literature reviews related to the main concepts behind the proposed methods. In the first review [4], the status of process mining in general (and in healthcare in particular) was assessed and we concluded that process mining is a growing research topic that emerged in the last decade and that deserves to be better developed. In the second one [3], we focused on goal-oriented process mining. This review concluded that, although process mining and goal modeling are growing research topics with considerable amounts of publications, there are only a few rare studies conducted at their intersection, which we refer to as “*goal-oriented process mining*”. In order to fill this gap, in the second phase, the focus is the development of methods to reduce complexity and improve goal-process alignment based on all three categories of process mining. These methods aim to make a beneficial connection between event logs, as the main input of process mining activities, and goals (and their indicators and derived requirements), as the main concept behind goal modeling. In the last phase, I will evaluate the methods using existing logs, a controlled experiment, and a case study done in collaboration with healthcare organizations. As a Design Science Research (DSR) methodology is followed, I will iterate over the last two phases several times in order to learn from the studies and refine the methods accordingly.

The research questions of interest here are:

- **RQ1:** How can we enrich process mining activities such that they exploit both event logs (recorded in data logs) and measurable goals (derived from goal models) in order to generate a model easy to understand and aligned with goals?
- **RQ2:** How can the synergy between process mining and goal modeling assist requirements engineering activities related to the evaluation of and alignment between goals and prescribed processes?

In this research, two methods to address the aforementioned research questions are proposed.

The structure of this paper is as follows. After clarifying the context and introducing some relevant definitions in this section, the state of the art derived from our literature review will be highlighted in Section II. Then, Section III describes the research method for the thesis. A sketch of my solution to address the research questions is summarized in Section IV. In section V, the strategies for evaluating the proposed methods are discussed and then the challenges expected to face are briefly described in Section VI. The main expected contributions of the work are summarized in Section VII, and then the paper concludes and highlights future work.

II. STATE OF THE ART AND LITERATURE REVIEW

We have conducted two systematic literature reviews to consider the state of the art in the domains that the research builds upon. The first review [4] provides an overview of process mining in general, and then an example of application to the healthcare domain in particular. We concluded that process mining is a growing research topic (with 2371 publications at the time), particularly in the recent ten years. Also, we surveyed eleven published literature reviews on process mining selected

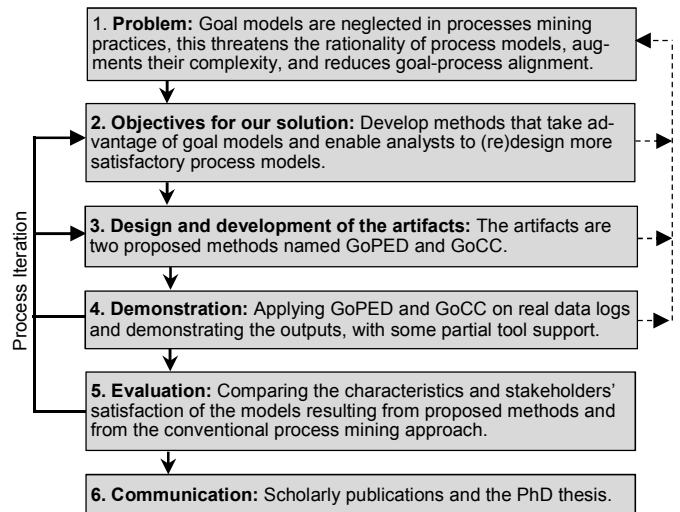


Fig. 1. My research method, instantiating DSR [6]

for further analysis along three main categories: algorithms, tools, and healthcare.

Narrowing down the research topic, we provided a second literature review [3], which analyzed 24 papers to assess the state of *goal-oriented process mining*. We concluded that there is no coherent line of research yet about the use of process mining in association with goals. Moreover, we observed that research about performance indicators that measure goals associated with process mining is also sparse. While the numbers of process mining publications and of goal modeling publications are trending up, goal-oriented process mining remains a gap between these two domains that this thesis aims to fill partially.

III. RESEARCH METHOD

Inspired by Hevner et al. [5], my selected research method is Design Science Research (DSR). In this method, understanding of the problem domain and of proposed solutions is achieved through the building and application of designed artifacts. New and innovative artifacts extend the boundaries of both human and organizational capabilities.

The version of DSR this work intends to follow is the one presented by Peffers et al. [6]. Their process consists of six activities in an iterative sequence shown in Fig. 1.

Following DSR, I will hence address my research through two main steps: building and evaluation of an approach designed to meet the research objective, namely, a goal-oriented process mining framework that exploit goals to simplify processes and improve their alignment. So far, in the building step, I have defined the problem and its importance through literature reviews and the acquisition of context knowledge. Also, I have performed the early steps of the third DSR activity in Fig. 1, i.e., design and development of the methods.

The next section will highlight the proposed methods and then I will discuss the possibilities to perform the evaluation step and the challenges I need to cope with.

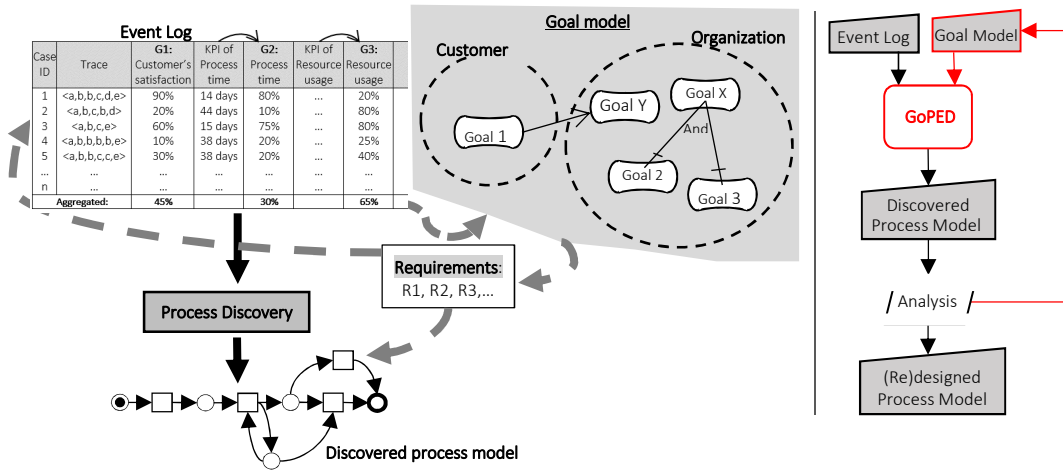


Fig. 2. Overview of goal-oriented process enhancement and discovery (GoPED), the first proposed method

IV. DESCRIPTION OF TWO PROPOSED METHODS

This section explores how process mining can exploit goal modeling to guide process discovery (RQ1) and identify opportunities for better aligning processes, goals and requirements (RQ2). There are two methods proposed as contributions to goal-oriented process mining. In the *Goal-oriented Process Enhancement and Discovery* (GoPED) method, I aim to manipulate the current process discovery scheme addressing goal-oriented concerns derived from goal models. The other method, *Goal-oriented Conformance Checking* (GoCC), will mainly focus on conformance checking and goal-process alignment.

A. Goal-oriented Process Enhancement and Discovery (GoPED)

Process discovery is the most prominent process mining technique. It takes an event log as input and produces a model without using any a priori information (see Fig. 2). There exist many techniques to extract a process model from event logs, e.g., the α -algorithm [1]. Many organizations find it surprising to see that process discovery techniques can discover real processes from merely some executions recorded in data logs. Process discovery is often the starting point for other types of analyses [1].

Another type of existing process mining is *process enhancement*, sometimes called *process extension*. The main idea is to extend an existing process model using some information of the actual process recorded in some event logs. For example, by using timestamps in the event log of a given process, one can extend the process to show bottlenecks.

In the existing literature, some perspectives like organizational, time, and resource perspectives have received some attention but a goal-oriented perspective is not visibly considered.

In the GoPED, the main idea is to exploit a wealth of information related to goals and their satisfaction levels that event logs may contain. In this method, in addition to usual attributes of events, some cases' attributes recorded in the event log and correlated to some goals and indicators are also needed. Such related goals and indicators are assumed to be also contained by

the corresponding goal model (e.g., specified using the Goal-oriented Requirement Language (GRL) standard [7]).

In usual process enhancement activities, the information of new perspectives are added to the process model already discovered in the process discovery step. Therefore, adding new information does not have any impact on the discovery phase. In GoPED however, the goal-related information influences the process discovery step and, hence, the resulting process model.

While discovering a process model, the main concern of process mining algorithms is that the model should be “representative” of the behavior realized in the event log [1]. Generally, when evaluating discovered models, there is a trade-off between four competing quality criteria. The model should allow generating the traces seen in the log (*Fitness*), should not allow generating the traces irrelevant to log (*Precision*), should not be too driven by examples in the log (*Generalization*) and, finally, should be the simplest model explaining the behavior observed in the log (*Simplicity*). The intended usage of the process model might be the main basis for balancing these criteria.

Besides the fitness-related criteria, there are two other phenomena related to event logs that should be considered during process discovery: *noise* and *incompleteness*. Noise refers to the problem of having “too much data” that are describing rare behavior, whereas incompleteness refers to the problem of having “too little data” to find the typical underlying process.

The steps of a process discovery algorithm, including the log preparation, and the algorithm’s performance, are usually only subject to the criteria just explained. In GoPED, there is a new input to the current process discovery approaches. This input might be a set of all goals including requirements, constraints, obligations, preferences, indicators, and stakeholder’s expectations derived from the goal model or documented requirements. The output of the approach will be yet a discovered process model. However, the discovered model, made in a goal-oriented approach, not only meets the aforementioned quality criteria of process discovery, but also complies with the *intentions* acquired from the goal model. In this approach, goals might be considered from different viewpoints, e.g., *customer* (case journey), *agent*, or *organization*. These viewpoints are enabled by the ability of

languages to support modeling goals of different actors, as in GRL.

As the new method enhances the usual data log by adding some goal-related attributes and then influences the usual steps of process discovery, it is called “goal-oriented process enhancement and discovery”.

Fig. 2 shows an overview of GoPED through an example. In this example, three different goals (G1, G2 and G3) are represented. Also, the logs have recorded the state of each goal-related attribute, e.g., *process time* is associated to the goal “to take a short process time”. Each of these goals may have associated Key Performance Indicators (KPIs), allowing to quantify its satisfaction level. Each indicator has a current value that should be evaluated against a set of parameters (*target*, *threshold* and *worst values*, sorted) [8]. The result is a normalized value ranging in [0%, 100%] showing the satisfaction level of a goal. The *threshold* value is mapped to 50% and the distance from the threshold determines the satisfaction level. For example, the goal “to take a short process time” is associated with the KPI “process time”, with *target*, *threshold* and *worst* values of 10 days, 20 days and 50 days respectively. Here, a 15-day processing time refers to +75% as satisfaction level. Using this scheme, the satisfaction level of the goals in the log of Fig. 2 are computed from the corresponding KPIs in the table.

In the GoPED method, the objective is to design a process model that fulfills one or many of the following criteria:

1. The model guarantees a threshold for satisfaction levels of one or multiple goals for all cases (e.g., in Fig. 2, $G1 \geq 60\%$, for all cases).
2. The model guarantees a threshold for the *overall* satisfaction level of each single case, where the satisfaction is computed using the goal model or simply by applying a function on some goals (row perspective). For example, the overall satisfaction of a customer could be defined as the average of her/his satisfaction levels about the *process time* and the *cost*.
3. The model guarantees a threshold for the *aggregated* satisfaction level of one goal rather than the level for individuals (column perspective). For example, in Fig. 2 the aggregated satisfaction level of G1 is 45%.
4. The model guarantees thresholds for *aggregated* satisfaction levels of multiple goals.
5. The model guarantees a threshold for the comprehensive satisfaction level of all goals for all cases (table perspective). This may be computed through a goal model (e.g., in GRL) or simply through a function. For example, in Fig. 2, assuming that the overall level of satisfaction equals the average of the three goals, the comprehensive satisfaction level of the table is the average of 45%, 30%, and 65%, which equals 46.6%.

To achieve such process models, GoPED selects a subset of traces that have fulfilled the given criteria. For example, if the objective is securing at least 50% as satisfaction level of G1 for all the customers, the cases 1 and 3 will be selected, both of which have a satisfaction level more than 50% for that goal. After such a selection, a process model is learned through the selected traces using process discovery algorithms. The discovered

model then can be compared to the model discovered from the whole log using the goal-oriented conformance checking method (described later). In GoPED, the overall satisfaction level of different goals (the last row in the log of Fig. 2) will be fed to the goal model in order to compute the overall satisfaction of each GRL *actor* (e.g., customer and organization in Fig. 2). This is also a means to quantify the satisfaction level of a case holder, e.g., a customer, using some indicators in the log (i.e., the second condition listed above).

In addition to the goal satisfaction, a process model that guarantees some specific *requirements* derived from goal models may be targeted as well. For example, a constraint may forbid performing activity *e* immediately after activity *b*, i.e., the trace $\langle \dots, b, e, \dots \rangle$ is not allowed. This kind of requirements might be represented in Linear Temporal Logic (LTL). LTL is a logic that, in addition to classical logical operators, also contains some temporal operators [1, 9]. van der Aalst et al. [10] took advantage of LTL to introduce an LTL-based language in the context of process mining. GoPED can enhance the process model through two potential paths (the two arcs leaving *Requirements* in Fig. 2). The first path checks each trace, and the acceptable traces will constitute the input for process discovery. The second path, however, will prune the raw process model inferred from the raw log to satisfy one of the aforementioned criteria.

The process model resulting from GoPED is learned through some cases that are carefully selected based on their goals and satisfaction levels. Therefore, as irrelevant cases together with unsatisfied ones have been filtered out, the discovered model is more likely well-structured and, also, oriented to satisfy predefined goal-related criteria.

B. Goal-oriented Conformance Checking (GoCC)

Conformance checking, the second type of process mining activity, is mainly concerned with comparing what *has* happened in reality with what *should have* happened. Here, an existing or *prescribed* process model is compared to an event log or a discovered model. Such a comparison highlights where the real process deviates from the prescribed process. Conformance checking allows to quantify the level of conformance and to diagnose differences.

Basically, one of the main usages of current conformance checking schemes is to measure the four fitness criteria that were introduced before, especially *fitness* and *precision*. These measurements are computed by replaying the event log on top of the process model. Here, the goal models or the requirements are not taken into account.

The second method, shown in Fig. 3, does not deal with process discovery algorithms. Instead, it waits for the process model discovered through mining as a representation of real-world behaviors. Then, it takes this real process as an input together with the prescribed process model. The new input here is the goal model. The real and prescribed processes are compared to find and analyze any misalignment. Such an analysis enables us to update the goal models on one hand, and to revise the prescribed (designed) process model on the other hand. Different metrics and algorithms are developed to check conformance of models

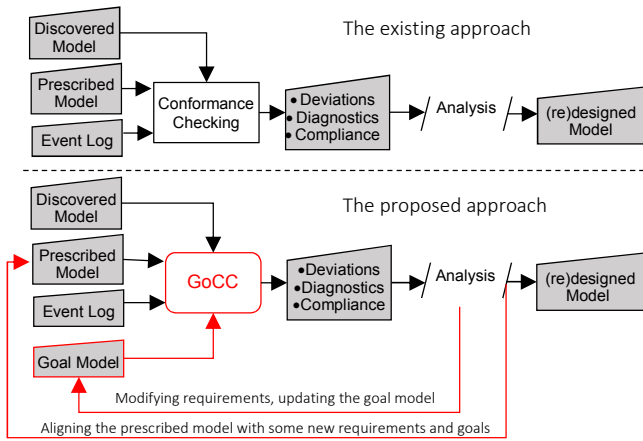


Fig. 3. The second proposed method: goal-oriented conformance checking (GoCC)

to models, logs to models, and logs to logs and find misalignments and degrees of deviation. There exist some concepts introduced in process mining dealing with stakeholders' requirements. *Auditing* is one potential usage of conformance checking to check whether business processes are performed within the boundaries set by stakeholders [11]. It is also possible to check the logs in terms of specific properties (e.g., constraints), sometimes called "business rules". For example, in an insurance company process, the activity "check the claim" should always be followed by an activity "decide" and also, eventually, either "reject the request" or "pay compensation" should happen, but not both. In the GoCC method, the usage of LTL checkers is highlighted in order to evaluate how the real processes comply with the requirements derived from goal models. Here, the corresponding indicators are defined to represent the level of satisfaction of different requirements and then feeding the indicators to the goal models and analyzing the results. To this end, I take advantage of GRL's capabilities in addressing *what-if* questions. Such questions are answered by GRL through forward propagation. Here, a given strategy initializes some intentional elements in the GRL model (goals/indicators, often leaves in the graph) and then these values are propagated in a bottom-up way to higher-level intentional elements of the model, and even to actors. This enables monitoring the impact of given action on a high-level goal or actor.

Conformance checking is, also, a means to reveal where the real process has deviated from a prescribed process model. In case of exposing undesirable deviations, the need for a better control of the process is implied. However, GoCC also leverages desirable deviations to reconsider the requirements and update the goal models by taking into account the deviations that stakeholders accept. For instance, sometimes the footsteps of students walking on campus are not necessarily aligned with the constructed sidewalks. These footsteps can show the desired line with the shortest path between two points. The students' goal here is "to increase the speed", a goal (or requirement) that has not been included in the original goal model. Therefore, the modeler can update the goal model and the prescribed process model through such a deviation. Also, a goal that is entirely satisfied could be made more ambitious for the future. Hence, when

a deviation in an event log is considerably frequent, GoCC, as a form of requirements engineering technique that exploits crowdsourcing, can find a new requirement (or new goal). When the purpose is to (re)design a process model, GoCC considers event data specifically to find deviations and to address how actual participants in the process-to-be performed it themselves. This data-driven method will find some requirements through the logs, instead of eliciting them from stakeholder goals or situations documented in the goal models.

V. EVALUATION STRATEGY

Design Science Research suggests five types of evaluations of designed artifacts: observational (case study, field study), analytical, experimental (controlled experiment, simulation), testing (functional, structural), and descriptive [5]. In this research, in addition to an ongoing example used throughout the thesis chapters to illustrate the proposed methods, my evaluation plan consists in four steps:

1. Prototype tool (Excel-based or programmed) to select traces in a complex log according to specified goal-oriented quality criteria.
2. Use of existing logs (e.g., datasets of a Dutch financial institute and a Dutch academic hospital from a process mining repository website [12]) augmented with goal-oriented data and goal models to test the methods.
3. A controlled experiment where an infrastructure to collect logs and goal-oriented data will be set up, processes and their tasks will be executed by students, process models will be mined from the logs, and the quality and usefulness of the models will be assessed through a questionnaire based on the perception of the participants.
4. A healthcare-oriented case study with real data and real users. There are two potential research groups interested so far, one at the *Institut du savoir Montfort* at Montfort Hospital in Ottawa (Canada) and the other at the Western Norway University of Applied Sciences in Bergen (Norway), where they already have an ongoing research project on process mining in healthcare.

VI. TECHNICAL CHALLENGES

The challenges expected in this research are rooted in the recognized challenges of both process mining and goal modeling. Moreover, a new challenge is the absence of real-world logs that include some goal-related attributes beside the usual event characteristics (e.g., timestamps). Although there are some potential KPIs that can be extracted from existing logs, the goals of each trace and their satisfaction levels are seldom available.

The "incompleteness" of an event log when discovering a model is another technical challenge that should be taken into consideration. As discussed in Section IV, an event log used in process discovery may not be a good representative for the process under consideration because of *noise* and *incompleteness*. GoPED is likely robust when dealing with noises because they could be filtered out while considering the traces' goals. However, incompleteness can be a challenge when the selection of traces results in too few events for use in discovery. As an extreme example, assume that in GoPED we select very few cases

based on a high level of satisfaction. Although the model discovered from a small log offers a high level of satisfaction, it becomes over-fitted and unusable for many of the cases. For that reason, GoPED is supposed to select as many traces as possible that satisfy the goal-oriented quality conditions. In these cases, one may consider a third research task for eliciting, through other means, activities that are poorly represented in event logs.

Another technical challenge related to GoPED is about complexity in the selection of traces based on the 5th quality condition. Finding the largest subset of cases whose comprehensive satisfaction level from all goals for all cases is higher than a threshold is quite sophisticated. The reason is that the selection of one case can decrease the aggregated level of one goal and, simultaneously, increase the aggregated level of another goal. Such selection of subset is not mathematically trivial.

In both methods, we assume that the goal models and requirements will not change in unforeseen ways. For example, one can assume that the desired processes and the stakeholders' preferences that are extracted from logs do not change over time. In particular, in the GoCC method, finding some new requirements and/or modifications of some current ones is based on frequent deviations. Here, it is assumed that the past behavior of the users is valid as a basis for designing a model for the future. Accordingly, *concept drift* [13], as an important phenomenon in the task of learning a model from data, should be considered.

Note that GoPED and GoCC may also be used in combination. The requirements and the goal models may be modified through GoCC and, consequently, such a modification may impact some entities involved in GoPED. In the current research phase, these two methods are presented separately to improve the understandability and avoid confusion.

VII. EXPECTED CONTRIBUTIONS

Two proposed methods (GoPED and GoCC) addressing our two research questions are the main expected contributions of this research. These methods will take goal-related attributes into account in the three main activities of process mining: process discovery, conformance checking and process enhancement. Such goal-oriented methods promise to contribute towards improving the rationality of the process discovery activities while simplifying the resulting models. Also, GoCC will be useful to improve the goal and process models based on insights inferred from extended event logs. Finding new requirements by exploiting the real behaviors of crowd-recorded event logs can contribute to requirements engineering engagements and improve the soundness of goal models and their requirements.

VIII. CONCLUSION

Our literature reviews [3, 4] show that process mining has neglected the potential benefits of goal models so far. Accordingly, in this paper, the goal-oriented method GoPED is proposed. This method extends the log with some new attributes related to goals and exploits these attributes to refine the log. Compared to the whole raw log, the refined log promises the generation of a model easier to understand and better satisfying the goal-related criteria (RQ1). Also, process compliance activities can be influenced by the goals and intentions of stakeholders,

leading to the GoCC method. GoCC promises to improve goal and process models using the insights obtained from comparing prescribed models with discovered ones. It is possible that goals of traces be achieved while some deviations between a prescribed model and real behaviors are exposed. In such cases, GoCC considers event data specifically to learn how actual participants executed the processes. This will allow finding some new requirements, or modifying some old ones elicited from stakeholders. These proposed methods are to be implemented, demonstrated and evaluated in the next steps of this research.

ACKNOWLEDGMENT

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

REFERENCES

- [1] W. van der Aalst, *Process Mining Data Science in Action*, 2 ed., Springer-Verlag Berlin Heidelberg, 2016.
- [2] D. Amyot and G. Mussbacher, "User Requirements Notation: The First Ten Years, The Next Ten Years," *Journal of Software*, vol. 6, no. 5, pp. 747-768, 2011.
- [3] M. Ghasemi and D. Amyot, "From Event Logs to Goals: A Literature Review of Goal-oriented Process Mining," *Requirements Engineering*, 2018. Submitted.
- [4] 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.
- [5] R. H. Von Alan, S. T. March, J. Park and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, 2004.
- [6] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007.
- [7] D. Amyot, J. Horkoff, D. Gross and G. Mussbacher, "A lightweight GRL profile for i* modeling," in *International Conference on Conceptual Modeling*, 2009.
- [8] J. Horkoff, D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida and J. Mylopoulos, "Strategic business modeling: representation and reasoning," *Software & Systems Modeling*, vol. 13, no. 3, pp. 1015-1041, 2014.
- [9] D. Giannakopoulou and K. Havelund, "Automata-based verification of temporal properties on running programs," in *Proceeding of 16th International Conference on Automated Software Engineering*, 2001.
- [10] W. van der Aalst, H. De Beer and B. F. van Dongen, "Process Mining and Verification of Properties: An Approach Based on Temporal Logic," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, 2005, pp. 130-147.
- [11] W. van der Aalst, K. van Hee and J. v. W. Computer, "Auditing 2.0: Using process mining to support tomorrow's auditor," *Computer*, vol. 43, no. 3, 2010.
- [12] "IEEE TF on Process Mining - Event Logs," Data, 4TU. Centre for Research, 2018. [Online]. Available: <http://data.4tu.nl/>.
- [13] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69-101, 1996.