ProDiGy : Human-in-the-loop Process Discovery

P.M. Dixit
Eindhoven University of Technology
Philips Research, Eindhoven
Eindhoven, Netherlands
p.m.dixit@tue.nl

J.C.A.M. Buijs
Eindhoven University of Technology
Eindhoven, Netherlands
j.c.a.m.buijs@tue.nl

W.M.P. van der Aalst
RWTH, Aachen, Germany
Aachen, Germany
wvdaalst@pads.rwth-aachen.de

***Abstract*—Process mining is a discipline that combines the two worlds of business process management and data mining. The central component of process mining is a graphical *process model* that provides an intuitive way of capturing the logical flow of a process. Traditionally, these process models are either modeled by a user relying on domain expertise only; or discovered automatically by relying entirely on event data. In an attempt to address this apparent gap between user-driven and data-driven process discovery, we present *ProDiGy*, an alternative approach that enables interactive process discovery by allowing the user to actively steer process discovery. ProDiGy provides the user with automatic recommendations to edit a process model, and quantify and visualize the impact of each recommendation. We evaluated ProDiGy (i) objectively by comparing it with automated discovery approaches and (ii) subjectively by performing a user study with healthcare researchers. Our results show that ProDiGy enables inclusion of domain knowledge in process discovery, which leads to an improvement of the results over the traditional process discovery techniques. Furthermore, we found that ProDiGy also increases the comprehensibility of a process model by providing the user with more control over the discovery of the process model.**

***Index Terms*—Interactive Process Mining; User Driven Process Discovery**

I. Introduction

A process is a series of actions performed in order to achieve a particular task or goal. Processes are omnipresent, and can be ad-hoc, defined explicitly or incorporated implicitly in a system. For example, a manufacturing process which deals with production of some goods, may be implicitly incorporated in the production system. On the other hand, a loan application process in a bank may be explicitly defined in a process model used to configure the workflow management system of the bank. A process can also be viewed at various levels of abstraction. For example, a hospital visit of a patient can be represented by a very high level process common to all the patients in the hospital, by abstracting all the low level detailed activities specific to the particular patient. Alternatively, for the same patient, a different abstraction can be a process of all patients with a similar disease.

The human understanding of these processes is usually enabled by representing them as intuitive graphical models expressed in languages such as BPMN, EPC, Petri nets etc. Typically a domain expert who has a high level overview of the end-to-end execution of a process, models the *expected* process models. Since these modeling notations offer intuitive visualization of processes, they have proved to be valuable artifacts to enable communication of complex knowledge in a comprehensible way across people from dissimilar backgrounds. The process model as described by a domain expert is the best guess, of how a process acts, or should act. However, reality may not always conform to this expected behavior of the process.

The execution histories of processes, called event logs, can be easily extracted from the corresponding information systems. For example, the event logs of an administrative process can be extracted from an Enterprise Resource Planning (ERP) system, or the event logs of a careflow process in a hospital can be extracted from the Electronic Health Record (EHR) system. These event logs represent a rich source of information by giving a view on a process execution in reality. Thus, these event logs can be explored to investigate any problems in the process, and suggest improvements to the process. Statistical techniques, such as statistical process control charts, are very popular particularly in the context of manufacturing processes. These techniques typically focus on analyzing the numerical data values at the activity level and do not focus on the end-to-end process. Machine learning techniques, such as sequence mining, have also been used in order to explore common (or uncommon) sequences of process variants for analysis. However, sequence mining techniques can only be used to find sequential patterns and not discover end-to-end processes. These techniques can also not discover process behavior such as choices, concurrency and repetition of activities.

Most of the traditional statistical or machine learning techniques either dive deep into a particular aspect of processes, or miss out on the important *process oriented characteristics* such as choices or concurrency between activities. Process mining is a discipline which overcomes these pitfalls by taking into account the process context while analyzing the event data. Broadly, process mining techniques can be grouped in three categories: (i) process discovery techniques, which aim at discovering a process model using an event log; (ii) process conformance techniques, which analyze how well an event log conforms to a given process model; and (iii) process enhancement techniques, which extend an already existing process model with information from the event log [1]. Process conformance and enhancement techniques require a process

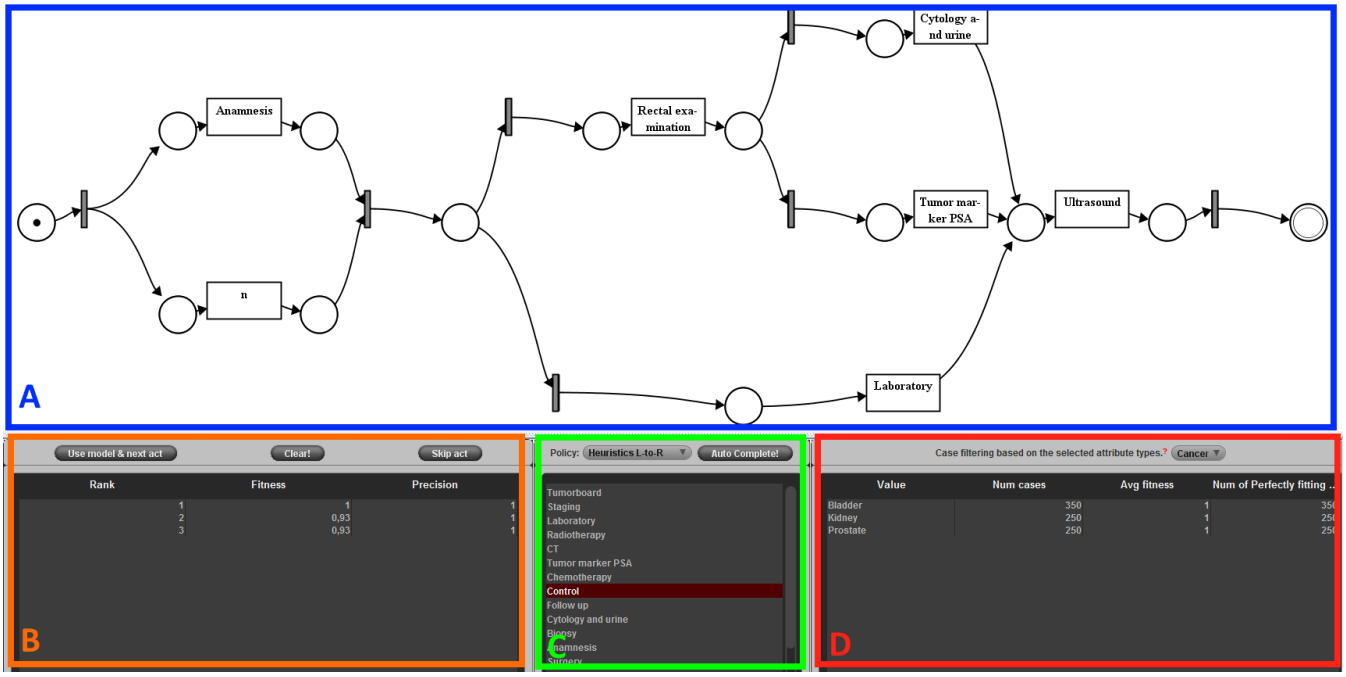


Fig. 1: The user interface of ProDiGy: the process model view (A) shows the process model interactively discovered by the user, the recommendations table (B) shows the ranked recommendations in a tabular form, the policy and activity selection view (C) shows the activities that can be added to the process model, and the process variants analysis table (D) shows the distribution of process variants based on case level attributes.

model to work with. A process model which is hand made or discovered, is always the starting point for in depth process analysis.

In order to produce a process model, there are typically two options. On one end of the spectrum, we have the process discovery techniques, which are automated and discover a process model directly from the event log with very limited user input. The user has very little influence over the actual process discovery, and usually it is not possible to incorporate domain knowledge during process discovery. Moreover, the resultant process models might be incomprehensible for the user. On the other end, we have the process editor based modeling tools which are completely user driven and use no historical evidence from the event logs for process modeling. However, ideally there should be something in between that bridges the gap. In this paper we introduce ProDiGy (**Pro**cess **Di**scovery **G**uided b**y** the user), which addresses this gap between the two approaches (Figure 1). ProDiGy is a tool that supports interactive process discovery by including human-in-the-loop, supported by effective visualizations and data feedback. Furthermore, ProDiGy supports interactive data analytics to explore process variants.

The main contributions of this paper are:

- Seamless integration of user driven and data driven approaches for process discovery, by providing recommendations to the user about possible locations of placing an activity within a process model and the impact of adding an activity at the specified location.
- Auto-complete/anything in between functionality: to switch efficiently between interactive discovery and automated discovery of a process model.
- Data support, feedback and analysis of the complete process as well as process variants during discovery of the process.

II. Related Work

ProDiGy is an interactive process discovery system that enables human-in-the-loop process discovery. Our approach sits at the intersection of the traditional user-driven process modeling techniques and the data-driven automated process discovery techniques, and hence is closely related to these two fields. Moreover, our approach is also related to *process model repair* techniques which repair a process model based on the information from the event log.

A. Modeling and discovery approaches

In this sub-section we discuss the techniques from the literature that deal either with user-driven process modeling or data-driven process discovery.

1) User-driven modeling: Business process management as a discipline has evolved with user-driven process modeling at its core [2]. Traditionally, a user involved in modeling of a process interviews participants across the different functions of the process, in order to get a holistic overview of the process [3]. The knowledge gained through interviews is then used to construct a human understandable graphical process

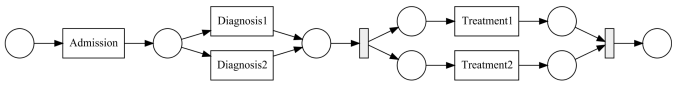


Fig. 2: A simple process model in a hospital setting. The first step in the process is Admission, followed by either one of Diagnosis 1 or Diagnosis 2. Diagnosis is followed by both Treatment 1 and Treatment 2, which can occur in any order.

model. Many process editing tools, both open-source and commercial, have been developed to support the user in efficient editing of a process model [4]. Most of these process editors offer users with limited or no support in decision making while modeling the process. With the turn of the century, there has been an added focus on recommendation based modeling support. However, most of these are text-based recommendations, which are generated based on query strings [5] or semantic similarity of activity descriptions based on ontologies [6], or based on historical evidence of a user's personalized preferences [7]. That is, these techniques are not data-driven and the historical evidence from the event log is not used to generate recommendations.

2) Data-driven process discovery: Automatically discovering process models from event logs started gaining attention in the early 2000's with the introduction of alpha algorithm [8]. Many discovery techniques followed [9]–[15] . Each of these discovery algorithms tackles the process discovery problem differently, and provides different assurances. For example, some discovery techniques generate process models that guarantee perfect fitness with the event log. However, in all these approaches, the input from user is rather limited, and the decisions that lead to the generation of process models are almost entirely data-driven.

B. Hybrid approaches

The area in between manual and automated process discovery is mostly unexplored. There have been some process model repair techniques, which take as input an event log and a process model and try to minimally change the process model to reflect the reality as depicted by the event log [16]–[19]. The initial model is typically hand made. However, the set up and goals of process model repair techniques is different compared to the human-in-the-loop process discovery. In such settings, the user has to first construct a process model without data support, which is then repaired using data. Alternatively, there have been approaches which let the user express some rules or constraints that the discovered process model must adhere to [20]–[23]. However, the user is restricted by the languages used to represent these rules, thereby limiting users expressiveness. Moreover, the user has no control over the actual discovery, and the resulting process model may adhere to all the rules, but may be extremely complicated. Our approach involves users explicitly in the process of process model discovery, and hence the user can discover a process model at the preferred level of complexity.

TABLE I: An example event log showing the flow of patients for process model from Figure 2.

Patient ID	*Activity*	*Timestamp*	*Age*	...
Patient 1	Admission	2017/02/02 14:45	68	...
Patient 1	Diagnosis 2	2017/02/02 15:15	68	...
Patient 2	Admission	2017/02/02 15:18	45	...
Patient 3	Admission	2017/02/02 15:40	89	...
Patient 1	Treatment 1	2017/02/02 16:41	68	...
Patient 2	Diagnosis 1	2017/02/02 16:45	45	...

III. Preliminaries

Before describing our approach in detail, we first introduce the preliminaries that are used throughout the paper.

A. Process models

Process models are used to model the flow of steps of a process. As discussed previously, most of the process modeling approaches allow modeling of more than just sequential flow of steps. That is, they allow for rich behavioral aspects, such as choice, synchronization, repeatable activities - either by duplication or by introducing loops, concurrency etc. In our approach, we use the Petri net representation of process models. Figure 2 shows a simple example of a process model represented by Petri nets.

B. Event logs

Event logs record the process notion of a system. Event logs record *what* happened i.e. an *activity* such as admission in the hospital, *when* did it happen i.e. the *time* when the patient was admitted, and for *which case*, i.e. who was the patient that was admitted. Next to this, event logs may also contain additional information pertaining to cases such as the age or gender of a patient, or pertaining to activities, such as which nurse admitted the patient. Table I shows an example event log with *Patient ID* as case identifier.

C. Quality dimensions

Having introduced process models and event logs, we now discuss two standard metrics which are used to evaluate the quality of a process model based on the event log. In order to calculate these metrics we use the alignment based conformance technique, described in [24].

1) Fitness: The *fitness* metric captures the goodness of fit of cases from an event log on a process model. For every case, the fitness value can span between 0 and 1. A perfect fitness score of 1 indicates that the model perfectly represents the behavior from the case. The values between 0 and 1 indicate the degree of fitness of a case and the process model. The averaged fitness values of all the cases in an event log indicate the overall fitness of an event log with a process model.

2) Precision: The *precision* metric captures how precise a given process model is with respect to an event log. That is, the precision metric penalizes a process model for adding *extra* behavior not present in the event log. Like fitness, the precision metric is scored between 0 and 1.

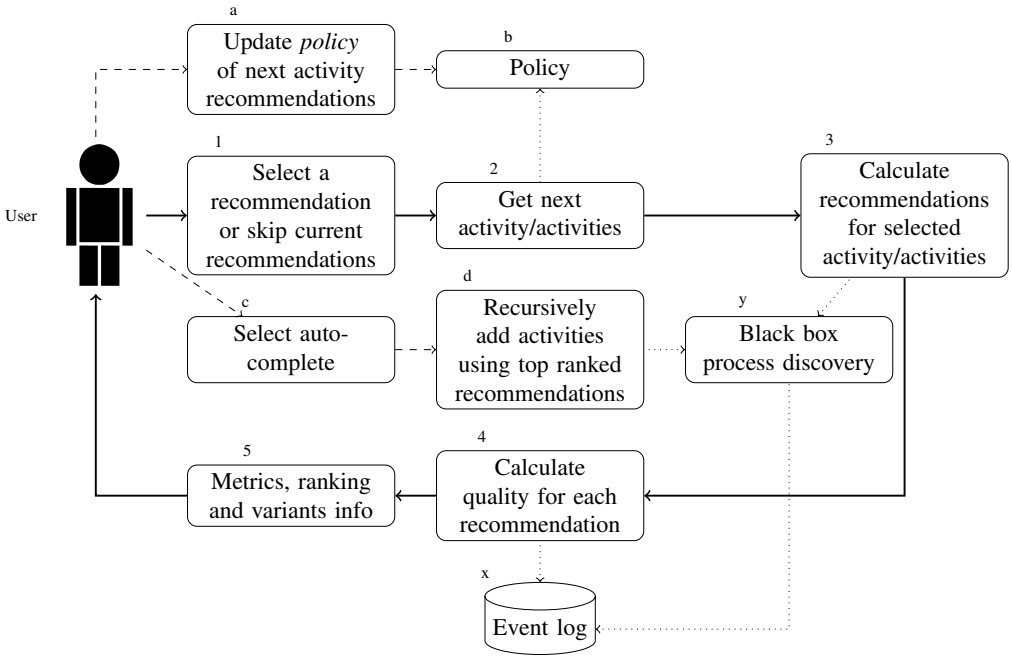


Fig. 3: Components of the ProDiGy system. The thick lines indicate the usual flow and interactions between the various components. Dashed lines indicate alternative flow and interactions between components. A dotted line between two components from *A* to *B* indicates that component *A* uses component *B*.

IV. Challenges of interactive process discovery

This section focuses on *the what*, i.e. what are the key challenges addressed by ProDiGy. Human-in-the-loop process discovery aims at merging the worlds of user driven process modeling without data support and automated process discovery which relies solely on data. In order to bridge the gap between these seemingly linked (yet fairly unexplored) disciplines, the following challenges are identified based on literature review:

- **C1: Recommendations for activity positioning in a process model.** Suggesting the position of an activity in a process model without overwhelming the user with too much information is one of the foremost challenges of human-in-the-loop process discovery techniques. The said activity can be placed at multiple positions within the process model. For example, suppose a new activity called *Treatment 3* needs to be added to the process model from Figure 2. *Treatment 3* can happen before *Treatment 1*, *Treatment 3* can happen after *Treatment 2*, *Treatment 3* can happen instead of *Treatment 1* and so on. Therefore, it is vital to recommend to the user the locations where a particular activity can be placed in the process model. Eventually the user decides the appropriateness of a particular recommendation based on the metrics, as well as the interpretability of the process model for a given recommendation. It should be noted that only those activities can be added to the process model, which are present in the event log.
- **C2: Evaluating the recommendations.** The recommendations for changing a process model need to be evaluated based on the event log by providing relevant information about each recommendation, using the quality dimensions. Ideally, the recommendations should be ranked automatically to assist the user in decision making. As an illustration, let's continue with the example from **C1** of adding *Treatment 3* to the model from Figure 2. If in the event log it was observed that *Treatment 1* and *Treatment 3* never occur together, then recommendations suggesting anything contradictory should be penalized and ranked low.
- **C3: Next possible activity recommendation.** Ideally, there should be a certain intuitive *flow* of choosing the activities during the process of process modeling. Since the process model is expanded incrementally, there should be a logical order in which the activities are added to the model. That is, there should be a way to decide the order or decide which activities belong together. For example, from a user's perspective, it would be easier to get recommendations of all the related activities together (or following one another), rather than getting ad-hoc recommendations of unrelated activities. Furthermore, at any given point during the process discovery, the user should be able to change the logic in which the next possible activity recommendation is generated.
- **C4: Mix and match - auto discovery and interactive discovery.** It is important to allow the user to switch effortlessly between automated discovery and interactive discovery. The user may decide to build a process model until a certain point, and then wish to *hand it over* to a

discovery algorithm to automatically complete the rest of the process model. That is, the user's focus may be to exhaustively discover a process model by using the available domain knowledge first, and then let an automated technique take over. Alternatively, the user may want to delegate some discovery tasks to an automated discovery algorithm, and then resume interactive process discovery when the so-called *quality* of the model is unacceptable.

- **C5: Process variant data analysis.** It is common to have multiple variants of processes in a large event log. For example, a hospital event log may contain cases about different patients suffering from different diseases. In reality, some highly frequent activities may actually be specific only to certain types of diseases. Hence, a high level process model that tries to encapsulate the most frequent behavior, may cater to some diseases better than others. Thereby, it is important to clearly provide such information to the user to enable informed decision making for analyzing the process variants independently, instead of just concentrating on the aggregated quality scores.

V. ProDiGy

This section focuses on *the how*, i.e. how the challenges discussed in Section IV have been addressed in ProDiGy. ProDiGy is an interactive system that automatically recommends and ranks positioning of next possible activities in a process model, while incorporating user's choices at every step.

A. Overview

Figure 3 shows the high level overview of the main components of ProDiGy. The usual flow and interactions between the components of Figure 3 is as follows: the user selects one (or skips all) recommendation(s) from the list of recommendations (1), the process model is updated based on the recommendation chosen by the user and the next candidate activities are selected (2) depending on the policy chosen (b). The information from the event log (x) is used by a black box process discovery algorithm (y) to come up with different recommendations for the newly selected activity/activities (3). The quality of each recommendation is calculated (4) by using the event log (x) which is then presented to the user along with ranking and process variants information (5). At any point of time, the user can change the policy used to predict the next activity/activities (a). Similarly, the user can decide to hand over the discovery task to the automated discovery technique at any point of time (c) and (d). The auto-complete mechanism recursively adds activities to the process model by using the black box process discovery algorithm (y) by selecting the top ranked recommendation until some termination condition is met.

B. Policy

Policy is a pluggable component used to decide the next activity/activities that should be added to the process model. Since process discovery in ProDiGy is incremental, the order in which the activities are added to the process model is important. There are a couple of reasons for this. Firstly, having a logical flow of adding activities incrementally may assist the user in better decision making. For example, adding all the semantically similar activities one after the other may help the user in structuring the overall process model better than adding unrelated activities in random order. Secondly, by following a strict order, it may become possible to discover a certain structure of process model which may otherwise become unreachable because of prior discovery decisions when adding activities randomly. As policy is a pluggable component, there could be many policies used to decide the next possible activities that are added to the process model. At any given point, the user can switch between policies by updating the policy using component (a) of Figure 3. Policies essentially cater challenge **C3** from Section IV. In theory, a policy can return multiple activities, and hence the recommendations presented to the user may contain process models containing some or all of those activities. However, here we present two policies which suggest a *single* next activity at a time. Hence, in the remainder we use the singular form of activity when referring to policies.

1) Alphabetical: The *alphabetical* policy, as the name suggests, orders activities purely based on their alphabetical ordering. At any point, if a user decides to skip an activity, then the next activity based on the alphabetical ordering is chosen. As evident, in the case of alphabetical ordering only one activity is selected at a time. Even though this is a very simple policy, it can still be useful as the user *knows* which activity would follow.

2) Log Heuristics: The *Log heuristics* based policy orders activities based on the information extracted from the event log. The idea is that the user starts building the process with the most common starting activities across all the cases from the event log, and ending it with the most common ending activities across all cases in the event log. The activities in between are also ordered according to their occurrences across the cases in the event log. This example policy is described in more detail below, but in order to do so, we first introduce some notations.

Let L denote an event log. We know that a case is a sequence of activities. It should be noted that the same activity can be repeated multiple times in a case, and hence can be present at multiple positions in the activity sequence of a case.

Let $\mathcal{A}$ denote the set of all the activities from the event log L. Then for any activity $a \in \mathcal{A}$, let $\#_i(a)$ denote the *total number of cases* for which the activity at the i^{th} position is a.

In order to compute the policy of log heuristics, we introduce a function $max(act)$. Function $max(act)$ returns "the" position where activity act occurs most frequently across all the cases from the event log. In case there are multiple positions, then the lowest position is chosen. Let $a, b \in \mathcal{A}$ be two activities, then we define a partial order $a \leq b$ as:

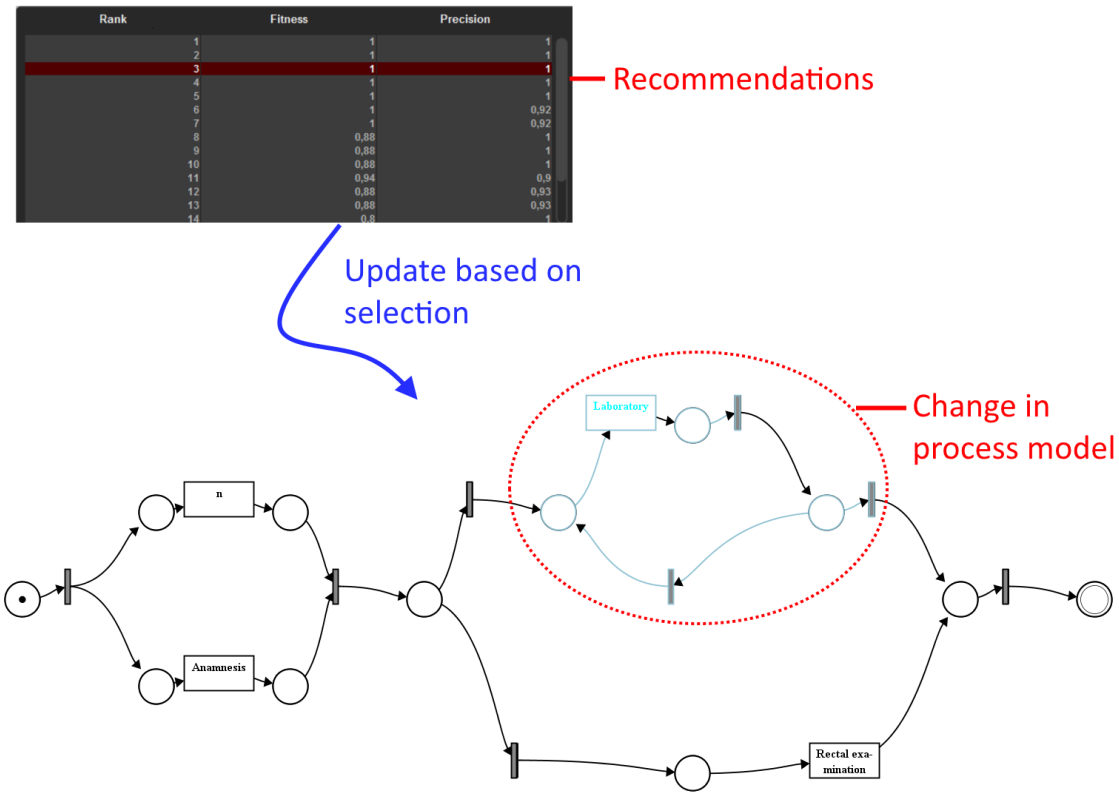


Fig. 4: Primary user interaction in ProDiGy. Based on the recommendation chosen from the recommendation table, the process model is updated temporarily to display the change in process model.

$$a \leq b \equiv max(a) < max(b)\ or$$
$$(max(a) = max(b) \wedge$$
$$\#_{max(a)}(a) \geq \#_{max(b)}(b))$$

$a \leq b$ defines a partial order, that can be made total to form a sequence. This sequence forms the policy of log heuristics, that suggests positioning of an activity based on the information from the event log, across all the cases and compared to all the other activities.

C. Recommendations

Recommendations component provides a list of possible edits that can be made to the current process model, to incorporate the next activity as suggested by the policy. As shown in Figure 3, we abstract from discussing the actual process discovery approach which comes up with these recommendations as it is outside the scope of this paper. There are multiple techniques possible that take as input a process model and an activity, and output multiple locations wherein the input activity can be added to the process model. Similar to the policy component, this is also a pluggable component, and any discovery algorithm which can incrementally add activities to a process model can be used. In this paper we use a concrete approach that is based on free-choice net synthesis rules [25]. However, as mentioned this is out of scope for this paper. In this approach, a single activity can be added at a time. For every possible position of the activity in the process model, we then compute the fitness and precision scores using the changed process model and the event log, using the alignment based conformance technique discussed in [24]. Each recommendation has its own fitness and precision score and hence indicates the goodness of positioning an activity at various locations in the process model. The fitness value indicates how well the event log fits a process model, whereas the precision value indicates how much extra behavior does a process model allow, compared to the event log. It should be noted that the fitness and precision values are calculated with respect to the activities present in the process model. Hence, the event log is filtered to contain only those activities present in the process model. This filtered event log is then used to compute the fitness and precision values. Also, it should be noted that the newly added activity can be added in multiple ways. For e.g., as a choice to a set of activities, or in parallel to some activities. Therefore, the precision or fitness scores would vary depending on the different ways in which the new activity is added. The recommendations are *ranked* based on a ranking criteria, consisting of a weighted average of fitness and precision values. By default the precision and fitness values are weighted equally, but these weights can also be determined by the user at any given point. Recommendations and ranking

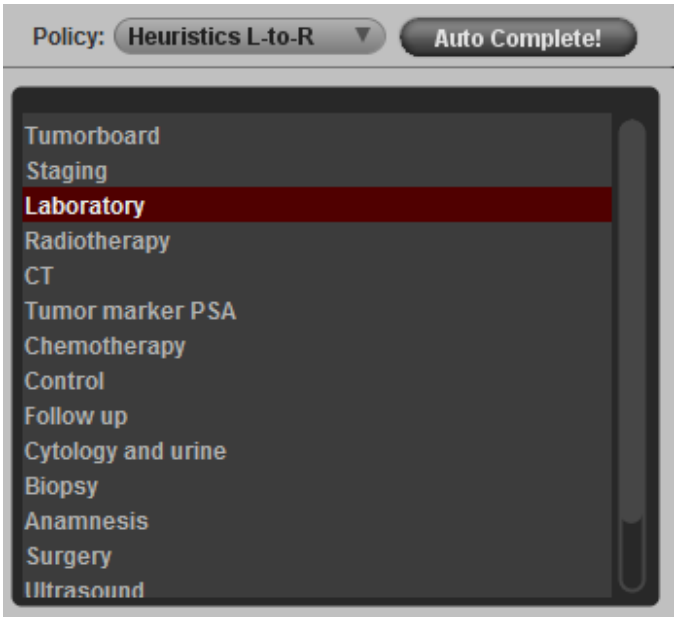


Fig. 5: Policy panel from ProDiGy, showing the dropdown for choosing a policy, Auto-complete button and activities list.

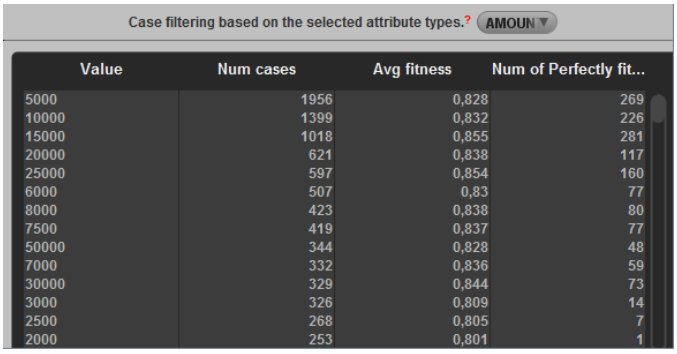
component addresses challenges **C1** and **C2** described in Section IV.

D. Auto-discovery

The *auto-discovery* component embeds the possibility of using traditional automated process discovery in ProDiGy. This component allows the user to pass the control to the black box discovery technique. The user provides some input to the algorithm similar to a traditional automated discovery technique, which requires some pre-configuration. The user sets the number of activities that should be added and a threshold for minimum fitness and precision values. The user can also update the weights for fitness and precision scores used for ranking the recommendations. Once these parameters are set, the auto-discovery feature iteratively adds activities to the process model, while the thresholds are met, by choosing the first ranked process model from the list of recommendations. The process models of the activities which did not satisfy the threshold criteria are skipped. After the desired number of activities are added, the user can resume the interactive process discovery. This component allows the user to switch between automated and interactive discovery and thus addresses challenge **C4** described in Section IV.

E. Process variants

The *process variants* component is used to analyze how different variants of the process behave for each recommendation. As described in the introduction of event logs, every case may have multiple features. For example a patient in a hospital, has additional attributes like age, sex, gender, type of disease etc. These features, when combined with the process model, can provide valuable insights. The user can compare the distribution of patients based on the process model. For example, it would be easy to investigate how a



Value	Num cases	Avg fitness	Num of Perfectly fit...
5000	1956	0,828	269
10000	1399	0,832	226
15000	1018	0,855	281
20000	621	0,838	117
25000	597	0,854	160
6000	507	0,83	77
8000	423	0,838	80
7500	419	0,837	77
50000	344	0,828	48
7000	332	0,836	59
30000	329	0,844	73
3000	326	0,809	14
2500	268	0,805	7
2000	253	0,801	1

Fig. 6: Process variants tabular view pouplated based on the selected case level attribute *AMOUNT*.

group of patients with a certain disease fit (or not fit) a given process model, compared to another group of patients with another type of disease. Since the values of fitness are pre-computed, these are re-used for providing the user with a certain feature specific distribution of cases. This component caters to challenge **C5**.

VI. Interface and Interaction

ProDiGy is composed of four display and interaction panels as shown in Figure 1. In this section we describe the user interface of each panel, and the type of interactions possible that enable human-in-the-loop process discovery.

A. Process model panel

The *process model panel* is panel **A** from Figure 1. This is a *view only* panel, that is, the user does not interact directly with the process model displayed in this panel. Whenever a user chooses a recommendation, the process model in this panel is updated by placing the new activity in the process model based on the recommendation. This is the largest panel of ProDiGy. Along with the process model view, there is an option to *undo* a previously selected recommendation, thereby allowing the user to revert the changes made.

B. Recommendations panel

The *recommendations panel* shows the recommendations of possible edits to the process model based on the chosen activity. This is a tabular panel, corresponding to panel **B** from Figure 1. Each row in the table corresponds to a unique way of adding an activity to the current process model. The table has three columns: *rank*, *fitness* and *precision*, which guide the user through the impact of each recommendation. Figure 4 shows the primary user interaction in ProDiGy. Based on the ranking, fitness and precision scores, the user selects one row from the recommendations table. The change is animated and projected on top of the current view of process model. The layout is changed minimally, and the new nodes added to the process model are colored differently, so that the user can easily spot the part of the process model that has changed. By navigating through different rows (that is recommendations)

the user can readily see the impact of each change. If a user is satisfied with a particular recommendation, then that recommendation can be made permanent by clicking *use model and select next activity* button. This button finalizes a recommendation, chooses the next activity based on the policy and generates new recommendations based on the activity chosen. Alternatively the user may click *skip current recommendations* button to skip all of the given recommendations, choose a next activity based on the the policy and generate new recommendations based on the newly chosen activity.

C. Policy panel

The *policy panel*, as shown in Figure 5, corresponds to panel **C** from Figure 3. This panel provides the user with options for choosing or updating a policy to select the next activity. The policy panel contains a policy selection box which contains all the available policies. The policy panel also contains a list box for activities which contains the activities from the event log. The user can scroll through the activities. The current activity as chosen by the policy is highlighted. Furthermore, the activities in the activity list are sorted according to the chosen policy. This provides an easy way for the user to find out the sequence of activities as suggested by the policy. The user can also interact and select any activity from the activity box. This action overrides the policy and gives user more control of the sequence in which activities should be added to the model. Whenever a new activity is chosen, either automatically by a policy, or manually by the user, the recommendations are recalculated for the newly selected activity. The policy panel also has the auto-complete button. As discussed previously, the auto-complete functionality lets the user switch between interactive discovery and automated discovery. Upon clicking the auto-complete button, the user is provided with a dialogue box to set the thresholds for minimum fitness and minimum precision values, as well as the number of activities that should be added automatically.

D. Process variants panel

The *process variants panel* corresponds to panel **D** from Figure 1. As shown in Figure 6, this panel is divided into two parts. The top panel contains a feature selection list. This lets the user choose the global case feature from the list. The bottom panel shows the distribution depending on the type of feature chosen. The distribution of cases is shown in a tabular format using the process model and event log. Whenever the user chooses a certain recommendation, the information in the feature table is updated to reflect the distribution based on the changed process model. Therefore, the user can navigate through different recommendations, and choose a process model better suited for a particular feature value, for e.g., patients suffering from a particular disease.

E. Design evolution

The overall design of ProDiGy went through a number of iterations. Enabling an incremental way of interactively incorporating domain knowledge was the main guiding principle behind the design of the system. After a number of revisions influenced by various methods from literature, ProDiGy ended up with four sub-components as discussed earlier. The idea behind designing the tool decomposed into four sub-components is to clearly separate the steps of the workflow in the usage of the system. The process model is the core component of process discovery and hence is the largest component and is placed centrally. The recommendations panel contains possible recommendations for changes in a process model. This is the *first* step of the workflow, wherein the user goes through the recommendations and chooses one. Since it is natural to navigate (or read through) a workflow from left-to-right, this panel is placed on the left hand side of the tool. The policy panel shows all the activities and highlights the next activity that is chosen. This next activity is chosen and highlighted after a recommendation is made permanent by the user. Hence this is typically the second step of the workflow, and hence this panel is placed on the right side of the recommendations panel. The process variants panel shows the statistical information based on the impact of the current process model on the case features. The information in this panel is dependent on the chosen activity and recommendation, and hence this panel is placed on the right side of the policy panel.

Having discussed the idea behind the overall design, and positioning of various panels, we now briefly discuss the design decisions made within some of the panels. The policy panel (panel **C** from Figure 1) shows the activities as a list. The idea behind this is to let the user freely navigate through different activities, and check the natural sequence of following or preceding activities based on the chosen policy. The design of recommendations panel underwent multiple changes. Initially, the recommendations panel was designed as *tabs*, wherein each tab contained a new process model based on the recommendation. However, this made the impact of change to the process model difficult to comprehend for the end user. Hence, after exploring multiple design ideas, it was decided to show the changes in process model projected on the current process model. Also, by using different colors the user can easily view the impact and the change in the process model. Furthermore, it was decided to provide the recommendations in a tabular format. The tabular design was chosen to provide a holistic view showing all the recommendations that is easy to sort and navigate.

VII. Implementation

ProDiGy is implemented as a plug-in in the process mining toolkit ProM [26]. We use an alignment based approach for computing the fitness and precision values, which is also already present as a plug-in in ProM [24]. ProDiGy can be accessed through the nightly build version of ProM (Interactive Process Mining package), and can be downloaded from the process mining website. The user interface of ProDiGy is as shown in Figure 1.

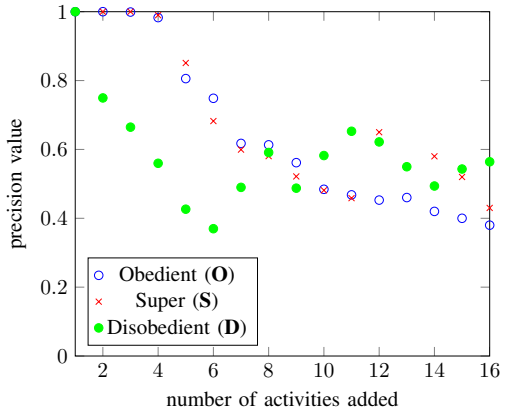


Fig. 7: Change in the precision values of the process model, after choosing a recommendation, i.e. adding an activity to the model.

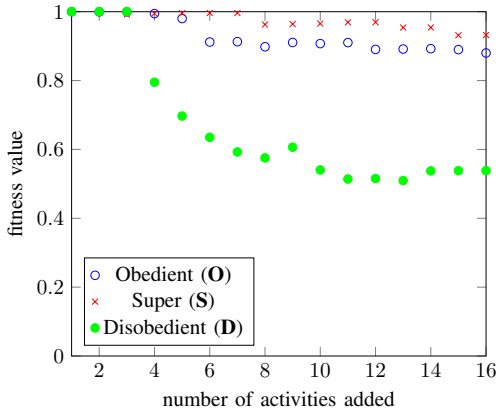


Fig. 8: Change in the fitness values of the process model, after choosing a recommendation, i.e. adding an activity to the model.

VIII. Evaluation

We conducted two types of evaluations in order to understand the usage, behavior and implications of ProDiGy. The first type of evaluation is objective and deals with analyzing a real life data set to compare the performance of various settings of ProDiGy, along with the comparison with some of the state-of-the-art automated process discovery techniques. The second type of evaluation is subjective wherein the domain experts explore ProDiGy based on some tasks using a synthetic data set.

A. Real life event log

We use a real life event log to demonstrate the functionalities of ProDiGy and evaluate the outcomes of interactive process discovery. The event log used is publicly available and contains information about patients suffering from sepsis disease from a hospital [27]. Each sepsis patient goes through a pathway of activities performed within the hospital. The information is extracted from the ERP system of the hospital and contains 15,214 activity occurrences for 16 activity classes for a total of 1050 sepsis patients.

We use ProDiGy to discover three process models. For all the three models, the weights for both fitness and precision values used during recommendations are kept at a default value of 1. The log heuristics based policy is chosen to predict the next activity. Using these settings, the following types of process models are discovered:

1) **Obedient user (O)**: For each activity, the first ranked recommendation is chosen. Each activity is added only once. This corresponds to the auto-complete feature of ProDiGy.
2) **Disobedient user (D)**: For each activity, a random recommendation is chosen from the recommendations list. Each activity is added only once. This is used to evaluate the non-conformant user who ignores the ranking criteria.
3) **Super user (S)**: Process model is discovered by using insights about the sepsis process from the literature [28]. This user can be considered as a domain expert who knows the clinical protocols and has a high level overview of the process, and can therefore, use the insights to guide the process discovery. This user may thus choose to ignore the ranking information during the process of process discovery if it contradicts with the clinical protocols.

The precision and fitness values of each type of user are compared after addition of every activity to the respective process models in Figure 7 and Figure 8. The fitness scores for user **O** and user **S** are fairly similar and are always above 0.9 after each iteration. The precision values for user **O** gradually decrease over time with each iteration. However, there is no clear pattern for the precision values of user **S**. This is because user **S** explicitly makes use of the domain knowledge, and does not rely entirely on the precision and/or fitness values. It can also be noted that both the fitness and precision scores of user **S** are higher than user **O**. User **D** chooses random recommendations. This is reflected by the fitness and precision scores, which follow no particular pattern and have a low value on an average. Even though the precision score of the final model (after adding activity 16) is higher for user **D**, the fitness score of user **D** is considerably lower than the other users.

Furthermore, as ProDiGy can also be viewed as a process discovery algorithm, we compared the results with some of the state-of-the-art process discovery techniques. We discovered process models by using the default settings of the automated discovery techniques. Subjective measures, such as comprehensibility and generalization of the process model are inherently taken into account by the users in ProDiGy. However, automated discovery techniques completely rely on the information from the event log. Hence, the automated process discovery techniques usually cannot take into account subjective measures such as simplicity of the process model. In order to keep the comparison between the approaches fair, in Figure 9, we compare the process models entirely based on

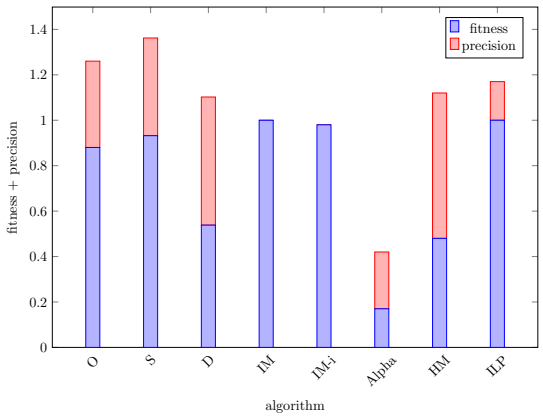


Fig. 9: Fitness and precision scores of the process models discovered by obedient (O), disobedient (D), super (S) users along with the automated discovery algorithms: Inductive miner (IM), Inductive miner infrequent (IM-i), Alpha miner (Alpha), ILP Miner (ILP), Heuristics miner (HM).

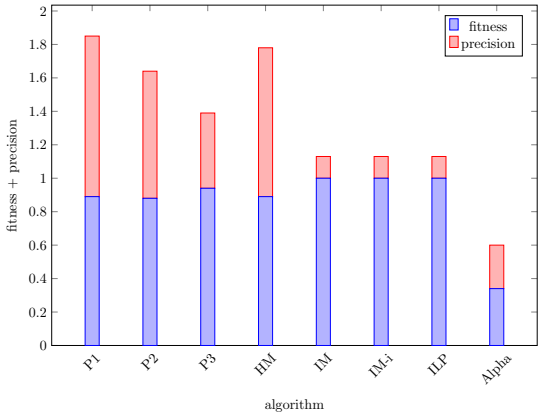


Fig. 10: Fitness and precision scores of the process models discovered from the noisy event log by three participants (P1, P2, P3) along with the automated discovery algorithms: Inductive miner (IM), Inductive miner infrequent (IMi), Alpha miner (Alpha), ILP Miner (ILP), Heuristics miner (HM). The process models are evaluated on the original noise free event log to compute the quality measures.

the fitness and precision values. As evident from Figure 9, the process models discovered using ProDiGy outperform other approaches. The outcome of obedient user (**O**) is comparable to the automated process discovery techniques but not as good as the super user (**S**). It should be noted that the two variants of Inductive miner (IM and IMi) find process models with almost perfect fitness. However, their precision values could not be computed within a practical time frame (more than 5 hours). This is probably because these process models are too imprecise and contain most of the activities in choice with a loop back, thereby allowing *any* execution sequence with unlimited repetitions. This comparison clearly demonstrates the advantages of human-in-the-loop process discovery enabled by ProDiGy, compared to the traditional automated process discovery algorithms.

B. User Study

Following the evaluation based on real life data, we also conducted a user study. The overall goal of the study is to understand the usability of ProDiGy and to gain insights through user feedback. The final outputs generated by the users using ProDiGy are also evaluated and compared with the traditional automated process discovery techniques.

The user study was conducted in the context of oncology patient flow in a hospital based on simulated data. We use Figure 2 in [29] as our reference process model, which contains the expected workflow of three cancer types: prostate, bladder and kidney. This process model was used to simulate data of 850 patients: 250 each of prostate and kidney, and 350 of bladder. Furthermore, in order to replicate the reality, noise was introduced in the data by removing 3% of random activity occurrences from random locations and adding 3% random activity occurrences at random locations in the event log. This 'noisy' event log was used throughout the user study. Three industry based health-care researchers participated in this study. As a first step, all the participants were provided with a brief introduction of the tool. Next, the participants were asked to perform couple of tasks: (i) discover an end-to-end process model by using a filtered (noisy) event log containing only prostate cancer patients, and (ii) discover a high level process model for all the patients using the entire (noisy) event log. The above tasks were used to guide the domain experts in exploration of ProDiGy, and their feedback was actively registered during the usage of the tool via an unstructured interview. Furthermore, after their usage of the tool the users were presented with a questionnaire to analyze the things that could be improved, were interesting and missing.

C. Results

In this section, we discuss the key findings of the user evaluation. For convenience, the participants are labeled as **P1**, **P2** and **P3**.

1) Usage patterns of ProDiGy: Even though every expert was given a standard introduction to the tool, there were differences in the way the tool was used. **P1** relied almost entirely on the ranking information noting that he *"trusts the data more"*, and therefore he always chose either the first or second ranked recommendation from the recommendations list. Contrary to this, **P2** used his domain knowledge most of the time, and relied on the recommendations when he was *"unsure what to do, or where to add an activity"*. **P3** added the activities he was familiar with first, and then relied on the recommendations to decide the fate of unfamiliar activities. The users were satisfied with the features and the workflow of the tool, and they deemed the system easy to learn. There were suggestions made by the participants based on their specific way of discovering a process model. For example, **P2** who relied more on his domain knowledge, navigated back throug-

hout the interactive process discovery session using the *undo* button, and said that he *"misses a redo button"* to navigate forward. Similarly, **P1** who relied more on the information from the event log noted that *"the recommendations helped in guiding the process discovery, however sometimes the ranking difference between recommendations was hard to spot due to a very small difference in fitness and precision values"*. In most of the intermediary steps, the participants chose one of the top-3 recommendations. Also, none of the participants selected a recommendation ranked lower than 5 at any given point. ProDiGy supported both types of users sufficiently: the one's relying on using domain knowledge as well as the one's relying on using the information from the event logs for process discovery, thereby supporting multiple ways of discovering process models.

2) ProDiGy improves the results of process discovery: All the three participants were satisfied with the process models discovered by them using our tool. The comparison of fitness and precision scores for the process models discovered by the participants and some automatically discovered process models is shown in Figure 10. In most of the cases, the process models discovered by the experts performed better than the automated process discovery techniques. In general, the process models discovered by the experts were deemed simpler to understand and/or more appropriate by the respective experts, compared to the process models discovered by the automated techniques. As **P1** responded about one of the process models discovered automatically, *"its extremely complicated and doesn't make much sense, the interactively discovered process model is easier to interpret"*. All the participants agreed that ProDiGy enabled them to have more control over process discovery and suggested that the interactive process discovery enabled discovery of substantially better process models, especially in terms of intelligibility, compared to the traditional automated process discovery techniques. An interesting observation here was that even though the three participants started off with an empty process model and with the same event log, there were also a few structural differences in the final process model. Partly this could be attributed to the usage patterns. However, another reason for these differences could be the fact that each participant had specific preferences to certain structural constructs in the process model over the others. For example, **P1** preferred sequential constructs over concurrency, whereas **P2** and **P3** had no preference between sequence and concurrency. This is also reflected in the Figure 10, wherein the process models discovered by almost all the participants have a similar fitness score, however, there is a notable difference in the precision score of each process model.

3) Experts gained insights during process discovery: An unprecedented finding was that during interactively discovering the high level process model, the experts were able to gain insights in to the individual pathways of prostate, bladder and kidney cancer patients. **P3** noted *"the information from process variants panel assists in finding out the right overall process model, but at the same time indicates which activities are applicable for which type of patients"*. Process variants information was also used by the **P2** to guide the process discovery in order to *"discover the process model that best fits a specific cancer type"*, rather than relying only on the aggregated fitness and precision values. The strength of the traditional process discovery techniques lies in understanding the data to make certain decisions. But the decisions made during process discovery are not visible to the end user. Since our tool enables active user involvement in process discovery, it also leads to exploring intermediary patterns, which may otherwise remain unexplored or hidden. For example, information about the link between the activities which might not be connected directly in the final process model, however, were connected during the intermediate steps of interactive process discovery. Hence, by encouraging the users to interactively discover a process model, ProDiGy can help in gathering insights during the process of process discovery, as noted by **P3**: *"ProDiGy is surprisingly good for exploratory purposes"*.

4) Study limitations: The overall feedback by all the three participants was positive and led to a lot of suggestions and possibilities for improvements. However, we are aware that the user study has some limitations. Firstly, even though the participants had more than 50 years of health-care experience between them, the number of participants is limited and hence the results may be biased. Owing to the sample size of the study, we did not conduct structural interviews and statistical analysis of the results. The participants from the current study also had some basic exposure to the field of process mining, whereas tutorials may be needed for other populations. This paper serves as a starting point of building and evaluating interactive process discovery. In the future, after incorporating all the suggestions made by the participants, we would like to evaluate and validate the tool with a broader population.

IX. Conclusion and future work

In this paper, we presented a novel approach for human-in-the-loop process discovery. We first identified the key challenges of interactive process discovery and proposed solutions in order to address these challenges. The solutions proposed were designed and implemented in ProDiGy, which was tested on a real life event log and evaluated by users in the health-care domain. Our evaluation demonstrates that interactive process discovery can outperform the traditional automated process discovery techniques. Furthermore, the user study also demonstrated that the tool is easy to learn and the visual analytics techniques incorporated in the tool are intuitive. To our knowledge, this is the first application that involves the user during process discovery, thereby giving users more control over the comprehensibility of the final process model. In the future, we plan to extend the tool to include the suggestions made by the participants of the user study. The foremost feature to be included is a collection of new metrics to guide the user in further distinguishing the recommendations suggested, especially when the differences between the fitness and/or precision scores are not significant. One such metric could be based on the degree of coverability

of the newly added activity (activities) in the event log. We believe such new metrics, can be effectively combined with the pre-existing metrics to assist the users in better decision making. Also, we plan to extend human-in-the-loop techniques to other areas of process mining, such as interactive process analytics to understand the "health" of a process. ProDiGy can serve as the key enabler for enabling such interactive process analytics, wherein the user can investigate the control flow aspect of the process using ProDiGy, and project other (data) attribute information on the process model while discovering the process model. Another interesting direction could be constructing user profiles in order to build a recommendation system based on a user's background and preferences. The information from user profiles could be used directly in the recommendation algorithm in order to populate process model constructs preferred by the users.

REFERENCES

[1] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016. [Online]. Available: https://doi.org/10.1007/978-3-662-49851-4

[2] M. Hammer, *What is Business Process Management?* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3–16. [Online]. Available: https://doi.org/10.1007/978-3-642-00416-2_1

[3] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and Parallel Databases*, vol. 3, no. 2, pp. 119–153, Apr 1995. [Online]. Available: https://doi.org/10.1007/BF01277643

[4] R. S. Aguilar-Savn, "Business process modelling: Review and framework," *International Journal of Production Economics*, vol. 90, no. 2, pp. 129 – 149, 2004, production Planning and Control. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925527303001026

[5] A. Koschmider, T. Hornung, and A. Oberweis, "Recommendation-based editor for business process modeling," *Data Knowl. Eng.*, vol. 70, no. 6, pp. 483–503, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2011.02.002

[6] M. Born, C. Brelage, I. Markovic, D. Pfeiffer, and I. Weber, *Auto-completion for Executable Business Process Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 510–515. [Online]. Available: https://doi.org/10.1007/978-3-642-00328-8_51

[7] N. N. Chan, W. Gaaloul, and S. Tata, "A recommender system based on historical usage data for web service discovery," *Serv. Oriented Comput. Appl.*, vol. 6, no. 1, pp. 51–63, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1007/s11761-011-0099-2

[8] W. M. P. van der Aalst, A. J. M. M. 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, Sept 2004.

[9] J. M. E. M. Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, *Process Discovery Using Integer Linear Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 368–387.

[10] A. K. A. de Medeiros, T. A. J. M. M. Weijters, and W. M. P. van der Aalst, "Genetic process mining: an experimental evaluation," *Data Mining and Knowledge Discovery*, vol. 14, no. 2, pp. 245–304, 2007.

[11] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs containing infrequent behaviour," in *Business Process Management Workshops*. Springer, 2014, pp. 66–78.

[12] J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev, "Deriving Petri nets from finite transition systems," *IEEE transactions on computers*, vol. 47, no. 8, pp. 859–882, 1998.

[13] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser, "Process mining based on regions of languages," in *International Conference on Business Process Management*. Springer, 2007, pp. 375–383.

[14] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, pp. 1–34, 2006.

[15] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "A genetic algorithm for discovering process trees," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[16] D. Fahland and W. M. P. van der Aalst, "Repairing process models to reflect reality," in *Business process management*. Springer, 2012, pp. 229–245.

[17] A. Polyvyanyy, W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. T. Wynn, "Impact-driven process model repair," *ACM Transactions on Software Engineering and Methodology*, vol. 25, no. 4, pp. 28:1–28:60, Oct. 2016. [Online]. Available: http://doi.acm.org/10.1145/2980764

[18] A. Armas-Cervantes, M. L. Rosa, M. M. Dumas, L. García-Bañuelos, and N. R. van Beest, "Interactive and incremental business process model repair," *QUT eprints*, 2017.

[19] M. L. Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, and L. Garca-Bauelos, "Apromore: An advanced process model repository," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7029 – 7040, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417410013758

[20] A. J. Rembert, A. Omokpo, P. Mazzoleni, and R. T. Goodwin, "Process discovery using prior knowledge," in *Service-Oriented Computing*. Springer, 2013, pp. 328–342.

[21] P. M. Dixit, J. C. A. M. Buijs, W. M. P. van der Aalst, B. F. A. Hompes, and J. Buurman, *Using Domain Knowledge to Enhance Process Mining Results*. Cham: Springer International Publishing, 2017, pp. 76–104. [Online]. Available: https://doi.org/10.1007/978-3-319-53435-0_4

[22] G. Greco, A. Guzzo, F. Lupa, and P. Luigi, "Process discovery under precedence constraints," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 4, pp. 32:1–32:39, Jun. 2015.

[23] F. M. Maggi, A. J. Mooij, and W. M. P. van der Aalst, "User-guided discovery of declarative process models," in *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*. IEEE, 2011, pp. 192–199.

[24] A. Adriansyah, B. F. van Dongen, and W. M. P. van der Aalst, "Towards robust conformance checking," in *Business Process Management Workshops*, ser. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2011, vol. 66, pp. 122–133.

[25] J. Desel and J. Esparza, *Free choice Petri nets*. Cambridge university press, 2005, vol. 40.

[26] B. F. van Dongen, A. K. de Medeiros, H. M. W. Verbeek, T. A. J. M. M. Weijters, and W. M. P. van der Aalst, "The prom framework: A new era in process mining tool support." in *ICATPN*, vol. 3536, 2005, pp. 444–454.

[27] F. Mannhardt, "Sepsis cases - event log," ser. Dataset. Eindhoven University of Technology, 2016. [Online]. Available: https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460

[28] F. Mannhardt and D. Blinde, "Analyzing the trajectories of patients with sepsis using process mining," in *RADAR+EMISA 2017*. CEUR-WS.org, 2017, pp. 72–80.

[29] S. Wagner, M. W. Beckmann, B. Wullich, C. Seggewies, M. Ries, T. Bürkle, and H.-U. Prokosch, "Analysis and classification of oncology activities on the way to workflow based single source documentation in clinical information systems," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, p. 107, Dec 2015. [Online]. Available: https://doi.org/10.1186/s12911-015-0231-x