Process Mining Versus Intention Mining

Ghazaleh Khodabandelou, Charlotte Hug, Rebecca Deneckere, Camille Salinesi

Process Mining *Versus* Intention Mining

Ghazaleh Khodabandelou, Charlotte Hug, Rebecca Deneckère, Camille Salinesi

Centre de Recherche en Informatique, Université Paris I Panthéon-Sorbonne
Ghazaleh.khodabandelou@malix.univ-paris1.fr,{Charlotte.hug,
rebecca.deneckere, camille.salinesi}@univ-paris1.fr

Abstract. Process mining aims to discover, enhance or check the conformance of activity-oriented process models from event logs. A new field of research, called intention mining, recently emerged. This field has the same objectives as process mining but specifically addresses intentional process models (processes focused on the reasoning behind the activities). This paper aims to highlight the differences between these two fields of research and illustrates the use of mining techniques on a dataset of event logs, to discover an activity process model as well as an intentional process model.

Keywords: process mining, intention mining, intentional process modeling

1 Introduction

Tracing and analyzing activities in Information Technologies (IT) is a field that appeared when the need to discover process models emerged [1]. Process mining aims to fill the gap between activity traces obtained from event logs and process models. So far, the mined process models are activity oriented models. Dowson [2] proposed a classification of process models into activity-oriented, product-oriented and decision-oriented models. Activity-oriented process models concentrate on the activities and tasks performed in producing artifacts and their ordering. Product-oriented process models are concerned about the successive product transformations. Decision-oriented process models introduce the concept of reasoning, choice and decision-making, the processes are then seen as teleological [3,4]. [37] introduced a new category called *intentional* process models [5,52,53,54]: they take into account the notions of intention and strategies of the process enactment.

Process mining techniques focus on activities, not addressing the intentional aspect of processes. We think that intentional models are accurate to represent the users' ways of thinking and working as they capture the human reasoning behind activities. We proposed in [5,52,53,54] a new field of research dedicated to intentional process mining called *intention mining*, closely related to process mining but addressing only intentional process models.

This paper aims to highlight the differences between process mining and intention mining. We will explain, for each field of research, their objectives, the representation and models they use and some of the mining tools already in place. Furthermore, we show, with the same set of event logs, how to discover an underlying process model,

firstly an activity-oriented process model – using process mining techniques – then an intentional process model – using intention mining techniques.

This paper is organized as follows. We browse a literature overview of process mining and intention mining in Section 2 and 3. Then, we compare the two approaches by applying discovery techniques to find a process model on a predefined dataset of traces and discuss the results in section 4. Section 5 concludes this paper.

2 Process Mining

This section browses a process mining literature overview.

2.1 Objectives

The process mining idea has initially emerged in the software engineering field with Cook and Wolf [6] and applying process mining on workflow log has been proposed for the first time in [7]. It is able to "close" the Business Process Management (BPM) life-cycle that presents how a process can be managed from its definition [1], through a requirements phase, to its execution and improvement. However, in each lifecycle the requirements phase is not well supported and there is no systematic or methodical manner to diagnose the requirements. Indeed, when redesigning a process, many non-serious problems, changes or crucial information of the actual process are not taken into account to improve the process model quality. According to [1], retrieving event logs containing information about the actual process allows having an insight into the followed process model. The event logs are recorded by the Information System (IS) and are generated by the actors' interactions with the IS. Each event in process mining is assumed to be an activity carried out by IS actors. An activity describes a well-structured step in a process. Van der Aalst proposes to classify process mining techniques into three categories [1]: Discovery, Conformance and Enhancement. In addition to these three classes, we add another emerging category: Recommendation.

- Discovery: Some techniques aim to discover process models by analyzing event logs. There is no a-priori information about them. For instance, event logs may be studied by α-algorithm [2] which automatically transforms them into a Petri net model presents the actors' behaviors recorded in the event logs.
- Conformance: Other techniques use an a priori model to check the degree of aligning between the actual followed process model (what actors are actually performing) and the pre-defined process model. These techniques can detect the deviations (about the who, what, when and where) and the model's intensity degree.
- Enhancement: It uses information recorded in event logs to improve and enrich the actual process model using methods of repair and extension. Repair has a mirror effect, (i.e. it tries to reshape the model to better illustrate reality). Extension allows widening the process model with new aspect by cross-correlating it with log.
- Recommendation: Some techniques aims to go a bit further about the studied process and use event logs to guess which activity may follow a current activity. [8] proposes recommendations based on URL traces. Schonenberg *et al.* propose and

experiment an approach based on recommendations which shows that the process performance is higher with an appropriate guided selection of activities [9].

2.2 Metamodels For Process Mining Results Representation

There are several metamodels for representing activity-oriented process models, such as EPCs [24], declarative models, Petri Nets BPMN, etc. However in this paper, we select only the latter two as they seem to be the most used in Process Mining.

Petri Nets. Petri nets are mathematical modeling languages allowing to model concurrency and synchronization in distributed systems. They are used as a visual communication aid to model the system behavior [10] and represent the process mining results. A Petri net is a directed graph composed of three types of components: places, transitions and arcs. Each place represents a possible system state; when occurring events or activities, transitions allow going from a place to another. Arcs maintain the relations between transitions and places.

Business Process Model and Notation (BPMN). BPMN [11] is a graphical diagram to model business processes; it aims at providing an easy graphical way to model business procedures that is understandable by all business users. Furthermore, one can model complex business process easily and map it to other languages such as BPML (Business Process Modeling Language), BPEL4WS (Business Process Execution Language for Web Services) or UML. BPMN creates a standardized link to fill the gap between business process modeling and implementation procedures. It improves the possibilities of traditional notations by managing the complex nature of internal and business-to-business processes interactions.

2.3 Process Mining Algorithms

There are various process mining algorithms that aim at discovering underlying processes from event logs. These event logs are the results of actors' interactions during the executions of tasks in different processes and contain the information about actors' behaviors, such as activities, timestamps, actors' ID, instance of process, etc. We describe briefly in the following some algorithms used in process mining.

Inference methods. [12] compare three inference algorithms: RNet [13], Ktail [14] and Markov models [15] that infer process models with a tradeoff between accuracy and noise robustness: a) RNet is a statistical approach that characterizes current state depending on the past behaviors; b) Ktail is an algorithmic approach that evaluates the current state according to the possible future behaviors; c) Markov is a hybrid between statistical and algorithmic approaches looking at the neighboring past and future behaviors to define the future state. Later, [16, 17] proposed techniques for concurrency detection and a measure to quantify the variance between behaviors and process models.

α-algorithm [12]. This algorithm is proposed by Van der Aalst et al. to rebuild the causality in the Petri-net workflow from the existent relations in the event log. α-

algorithm takes the event logs as input, rebuilds process models by using simple XOR, AND splits and joins; thereby creates the workflow nets as output. α-algorithm cannot handle certain constructs of workflow nets such as loops and long-term dependencies. To overcome the difficulty of α-algorithm during complex situations, an extended algorithm has been proposed: α ++ algorithm [18] that proposes new relationships between event logs to handle long-term (implicit) dependencies.

Directed acyclic graphs [7]. In graph theory, a directed acyclic graph is a graph that has no cycle. In process mining, the events can be transformed into dependency graphs (workflow graphs) using directed acyclic graph, representing events and their causal relations without loop. However, using this kind of graphs to model the processes is delicate as loops exist in process models. To overcome this challenge, this approach tries to count the tasks frequencies and then fold the graph. Nevertheless, the results are partially satisfying and the model does not completely match the real process.

Inductive workflow acquisition [19, 20]. The aim of this approach is the acquisition of workflow models and their adaptation to changing requirements by finding the best Hidden Markov Models (HMMs) that reflects the process model. One can find the HMM by merging or splitting models. Each state of HMMs corresponds to an activity node. The event logs can be observed and generated into workflow nets by inductive learning.

Hierarchical clustering [21]. This algorithm separates a set of logs of a given process into clusters and finds the dependency graph for each log. This algorithm structures the clusters of event logs into a hierarchy tree. For each cluster, a workflow model is constructed and finally all the models are merged into a single one.

Genetic algorithm [22]. This algorithm provides process models (Petri nets) built on causal matrix (input and output dependencies for each activity). This approach tackles problems such as noise, incomplete data, non-free-choice constructs, hidden activities, concurrency, and duplicate activities. Nevertheless, it still remains a complex task as it requires the configuration of many parameters to deal with noise and irrelevant data.

Heuristic algorithm [23]. This approach is based on α-algorithm. It uses the likelihood by calculating the frequencies of relations between the tasks (e.g. causal dependency, loops, etc.) and construct dependency/frequency tables and dependency/frequency graphs. This approach can detect irrelevant logs. However, like the Genetic algorithm, Heuristic miner needs a complex configuration phase.

Colored Petri Nets (CPN) [24]. CPN is a graphical language for analyzing the properties of concurrent systems. CPN combines Petri nets with the CPN ML (based on functional programming language Standard ML). It can be used when concurrency and communication are crucial in the system modeling.

2.4 Process Mining Tools

Many tools emerged to support the process mining techniques. Among them, we can mention ProM [26], CPN tools [24], EMiT [27], Disco [28]. The ProM framework is a pluggable framework that supports various plugins for different techniques of process mining such as α-algorithm and its extensions. CPN tools allow modeling and analyzing CPN models [24] and simulating processes to analyze and check them. In [29,30], the authors present a combination of CPN tools and ProM framework plugin implemented to improve business processes modeling. Disco allows to automatically map the event logs with CSV and XLS extensions to the appropriate XES or MXML notations which are supported by ProM and to have an insight into the process from event logs very quickly. It optimizes performance, controls deviations and explores variations. However, it is a commercial tool and does not provide information about the used algorithms. EMiT is able to integrate timing information using an extended version of α-algorithm. This tool transforms the event log of commercial systems to XML format and mines to find the causal relations between logs and based on that, rebuilds a Petri net represented in a graphical model.

3 Intention Mining

This section gives a literature overview of intention mining.

3.1 Objectives

The definition of "intention" according to [31] is: "a determination to act in a certain way; a concept considered as the product of attention directed to an object or knowledge […]". From a psychological point of view, intention is defined as follows: "Our common sense psychological scheme admits of intentions as states of mind; and it also allows us to characterize actions as done intentionally, or with a certain intention" [32]. Purpose in an IS context is also essential for any organization. IS are created to fulfill organization needs and their functionalities and properties are defined according to the objectives of the organization. According to [33], an intention is "an optative" statement, a state or a result that is expected to be reached or maintained in the future.

Some approaches called Intention Mining have been defined, however their analysis is not based on traces of IS activities; [34]'s approach uses classification techniques to classify home video content. The analysis is based on what is recorded with camcorders and provides categories to sort videos. They do not provide process models but some of their classification techniques could be reused in process models context.

In our point of view, Intention Mining aims at extracting sequences of actors' activities from sets of event logs to infer related actors' intentions. A set of activities corresponds to the achievement of an intention. Intention mining uses event logs as input and produces intentional process models. It is a field related to intentional process models. As for process mining techniques, Intention Mining tackles the four challenges of discovery, conformance, enhancement and recommendation:

- Discovery: Identifying the underlying actors' intentions and strategies from the event logs allows defining intentional process models.
- Conformance: Checking the conformity between a prescribed intentional model and its enactment allows measuring the gap between the prescriptions and what is actually done by users.
- Enhancement: The conformance checking allows identifying the distance between the model and the traces, which helps to define what is wrong with the model (which intention is never achieved, which strategy is never used, etc.).
- Recommendation: Using the event logs repository and the discovered intentional process models allows providing recommendations to IS actors at run-time, based on their supposed reasoning behind their activities.

3.2 Metamodels for Intention Mining Results Representation

Processes may be formalized in an intentional way. The common aim of goal-modeling approaches is to model the processes according to the purpose of the actors/projects/organizations. We quote among them i* [35], KAOS [36] and Map [37].

KAOS. This approach proposes to specify the system and its environment by a requirements model instance of a metamodel to support the goals, agents, and alternatives. It is based on a goals diagram where goals are related together through AND/OR decomposition links. KAOS uses goals to specify, analyze, negotiate, document and modify the systems requirements. To do so, the decompositions refine high-level goals identified by actors into thinner particle of goals. This refinement requires classifying goals according to their level of abstractions and linking the same goals at the same level of abstraction. This approach supports variability and have a well-structured semantic but is less involved in the intentional aspect of IS actors. Furthermore, KAOS has a rigid task-decomposition; modeling complex intentional processes is then difficult [28].

I*. The i* framework is a modeling language that aims at analyzing IS and the environments of organizations to model processes by focusing on the relationships between actors and theirs goals. It consists in two main models: the strategic dependency model (SD) and the strategic rational model (SR). The SD describes external relationships between actors (called strategic actors). The SR describes the internal relationships between actors. The i* framework is used to model the business strategy with organizational strategic goals. i* supports actor-oriented and goal-oriented concepts. The actor-oriented concept allows modeling requirements of IS by concentrating on the dependencies between actors' goals. Actors are autonomous entities with uncontrollable and non-cognizable behaviors. They are different and independent in their ways of reasoning and consequently have diverse goals. i* models aim at producing a conceptual framework to represent the processes involving agents, i.e., software actors and software systems. i* is able to assess the functional or non-functional requirements of systems so it can capture what, how and why a software component is developed. It recommends the use of the notion of non-functional requirements using "soft goals" identified as evaluation criteria. The alternatives then contribute to different degrees of satisfaction of these goals. However, this modeling language has an operational semantic for the tasks but not for the

goals and it is not used to model strategic goals. i* is not designed to be a variable framework therefore, it does not afford a high level of flexibility.

Map. This modeling language is an intentional process metamodel that allows formalizing flexible processes. Map supports variability for the goals and offers the possibility to follow different strategies by focusing on the intentional aspect when enacting methodological processes. During its enactment, a process is not limited to linear activities; actors, according to their context, have a variety of choices to execute a task. Map models (instances of Map metamodel) guide the actors by proposing dynamic choices according to their intentions. Map models can be executed non-sequentially and followed until the fulfillment of intentions. Thereby, map process models offer a better adaptability to the context of each actor. Map specifies processes in a flexible way by focusing on the intentions and the different ways to fulfill them. A map model is presented as a graph which nodes represent intentions and edges represent strategies. An edge is defined between two nodes where its related strategy can be used to achieve the intended target node. There may be several edges entering a node representing all the strategies to fulfill an intention. Fulfilling a specific intention with a particular strategy is related to a specific guideline defining the activities to perform. This model allows describing high level organizational intentions. The intentional Map metamodel has been introduced in the IS engineering domain [37] and was validated in several works: requirement engineering [38], method engineering [39], and enterprise knowledge development [40].

3.3 Intention Mining Algorithms

In order to retrieve the intentions from the traces of actors' activities and due to the variability of traces in terms of length and nature, we propose some algorithms based on probabilistic and statistical techniques. The probabilistic models provide the information about the nature of data (i.e. if the observed data is a harvest of hazard or an intended result). Moreover, they model the data taking into account their temporal aspect. Since the observed side of data could hide the latent one, the probabilistic models can also formalize this concealed side and extract the characteristics of both observed and latent data. Hidden Markov Models (HMMs) [41] seem promising to mine intentions [52,54](concurrency can be handled in future extensions). Hereafter, we describe some algorithms used in HMMs which help mining intentions.

Viterbi Algorithm (VA) [42]. VA is commonly used to decode convolutional codes (to correct errors in noisy channels) and in the context of HMMs. It finds the most likely sequence of hidden states (Viterbi path) for a given observed sequence using trellis, which is a type of Finite States Machine (FSM) with states and transitions. In HMMs, the observations are generated by the underlying states; HMMs enable to find the hidden structure by estimating the parameters of observations sequences. Given an observed sequence, the VA uses various metrics to evaluate which path is the most likely one. In the context of Intention mining, the hidden states are the IS actors' intentions which generate the observations (actors activities' sequences). To find the correct path, we can use a brute-force method but the complexity of computation C^n can explode with the increase of the length of observations n for the number of states

of C. Nevertheless, the VA allows to compute the Viterbi path with a nC^2complexity, which is considerably lower than C^n. On the other hand, VA minimizes the error probability of states transitions by determining the most likely set of states. To infer the most likely set of intentions, the VA requires knowing the parameters of observations, i.e. the transition probabilities, initial probabilities of states and emission probabilities. These parameters can be estimated by Maximum-Likelihood Estimation (MLE). The transition probabilities can be calculated by counting the average frequency of states transition from one intention to another. We calculate the emission probabilities by counting the average apparition frequencies for a given activities sequence for a given intention. Thus, the VA is applicable when this information is available; this is possible in the case of supervised learning Moreover, the VA is particularly useful when activities can belong to several intentions.

Baum-Welch Algorithm (BWA) [43]. BWA is a special kind of Generalized Expectation-Maximization (GEM) algorithm [44]. In the case of unsupervised learning, the BWA allows computing the unknown parameters of HMMs. To do this, the BWA uses Forward-Backward algorithm [45]. The use of iterative algorithms such as the EM algorithm allows estimating the optimal solution fairly, accurately and quickly. As mentioned in the VA, if the parameters of observations sequences are known, the computation of these parameters is a simple task. Nevertheless, since in some cases we do not know this a priori information for the observed sequences, we cannot directly count the frequencies from data, therefore, the BWA uses Forward-Backward algorithm to estimate the expected frequencies.

3.4 Intention Mining Tools

There is no intention mining tool to our knowledge, as intention mining has not been applied yet to re-build intentional process models and discover goals behind activities. We aim to provide a tool that will implement the developed algorithms.

4 Case Study

In this section we illustrate our comparison with the use of process mining and intention mining techniques on the same dataset.

4.1 Dataset

The dataset, obtained from [46], contains the event log of claims handling in an insurance company. These claims are supported by two call centers in two different locations: Sydney and Brisbane. The log contains 46 138 events and 3 512 cases (claims). The incoming calls volume and average total call handling time are the same in the two centers. However, call center agents handle differently the incoming calls. The agents first handle the claims; the rest of the process is treated in the back-office of the insurance company. According to [46], although this dataset is synthetic with non-

noisy event log, the α-algorithm cannot mine correctly and extract the right model. In Equation (1), A represents the transition matrix of the activities: they are ordered from left to right and top to down: Incoming claim (A1) (first line, fist column), Brisbane checks if sufficient information is available (A2), Brisbane registers claim (A3), Sydney checks if sufficient information is available (A4), Sydney registers claim (A5), Determine likelihood of claim (A6), Assess claim (A7), Advise claimant on reimbursement (A8), Initiate payment (A9), Close claim (A10) and End (A11). This matrix shows the following activities for each activity, and in which proportion; for instance, A1 activity is followed at 48,97% by A2 activity and by A4 activity at 51,03% .

$$A = \begin{pmatrix} 0 & 0.4897 & 0 & 0.5103 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8866 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1134 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7952 & 0 & 0 & 0 & 0 & 0 & 0.2048 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.8342 & 0 & 0 & 0 & 0.1658 \\ 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0.4039 & 0.3990 & 0 & 0.1971 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5030 & 0.2389 & 0.2581 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2389 & 0 & 0.7611 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2581 & 0 & 0 & 0.7419 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1)$$

4.2 Process Mining Result

We choose to illustrate the process mining results with a Petri net representation of the process model found with α-algorithm (Figure 1). It was obtained using ProM tool by choosing the causal dependency parameter. The choice of this parameter is due to the fact that Petri nets allow the representation of causal dependencies between activities occurrences to model reactive systems [47].

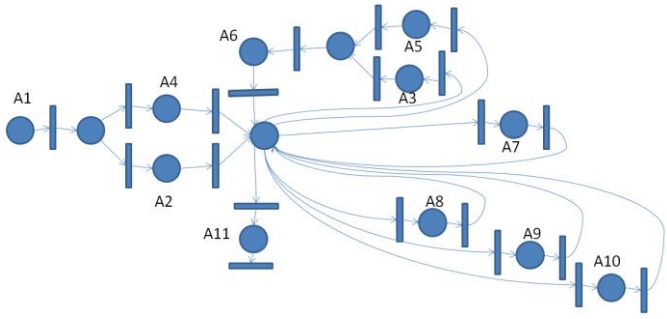

Fig. 1. The process model represented as a Petri net.

When comparing the links between the activities in (1) and the Petri net obtained from ProM, we perceive that there are transitions allowed in Petri net but not in matrix A (1). For instance, Petri net allows the sequence A5, A6, A7, A5, A6 whereas this sequence is not possible in A since there is no transition between A7 and A5.This means that both Petri net and the transition matrix fit the dataset, but Petri net is more

general and allows cases that are not present in the dataset, contrary to the transition matrix. Thus, Petri net, for this dataset, suffers from an under-fitting problem.

4.3 Intention Mining Result

In this paper, we choose the Map representation to illustrate the intentional process model found by intention mining. i* and KAOS will be studied further in our next works on intention mining. They focus on the operational level rather than the organization level and do not raise the issue of alignment [48], whereas Map allows working on strategic goals. As mentioned earlier in section 3.3, both VA and BWA are able to associate a sequence of intentions to a sequence of activities. However, these two algorithms need to be fed with the set of intentions, which is not the case here as this set is not yet defined. We then implement a new algorithm to find (a) the groups of strategies used by the agents of the insurance company during the claims treatments, and (b) the groups of intentions linked to those strategies.

Strategies Miner Algorithm. To construct the map process model, we have to find the strategies that the agents of the insurance company use to fulfill their intentions. To do this, we define a strategy by the triplet $\{s, \mathcal{A}, e\}$, where s is the starting activity, e is the ending activity and $\mathcal{A}$ is a set of activities in between. The realization of this strategy is a sequence of activities starting with s, followed by a sequence of activities comprising all the elements in $\mathcal{A}$ and finishing with e. We have developed two simple rules to organize the activities into strategies. The first rule is based on what we call a bottleneck activity. A bottleneck activity a is an activity for which there are at least two possible preceding activities, and for every possible preceding activity the transition to activity a occurs with a probability of 1. Of course, the initial and final activities are excluded from this definition. Then, a bottleneck activity can only be at the beginning of a strategy. The second rule is that for any strategy, there is a sequence of activities such that the probability of having this sequence is higher than a threshold β. The first rule accounts for the fact that if there are compulsory activities, these activities do not represent different strategies. The second rule accounts for the fact that the activities composing a strategy have to be frequently used sequentially. Note that for some strategies, the order of activities is not significant since the activities can be performed simultaneously or in a random order. However, the first and the last activities of the sequence must always be the same. Thereby, we can group the activities into strategies. We say that these rules define a set of strategies with parameter β. The strategies miner algorithm is given as follows in pseudo-code.

```
Inputs: cases
Outputs: strategies
Begin
  Make a strategy out of each case
  Merge equal strategies
  Divide strategies at the bottleneck(s)
  Merge equal strategies
  For each strategy {s, 𝓐, e}
    if max ℙ(𝓐, e|s) < β then
     divide strategy
```

```
   endif
 EndFor
End
```

Table 1 presents the strategies obtained with threshold $\beta = 0.01$ and $\beta = 0.3$. The column *probability* indicates the probability of having the corresponding strategy among the cases of the dataset.

	β=0.01				β=0.3			
Strategy index	**Starting activity**	**Ending activity**	**Set in between**	**Proba-bility**	**Starting activity**	**Ending activity**	**Set in between**	**Proba-bility**
1	A1	A11	A3	0.0555	A1	A3	A2	0.4342
2	A1	A11	A4	0.1045	A1	A5	A4	0.4058
3	A1	A2	A3	0.4342	A1	A4	none	0.5103
4	A1	A5	A4	0.4058	A11	None	none	1.0000
5	A6	A11	none	0.1392	A1	A2	none	0.4897
6	A6	A11	A7	0.1381	A6	none	none	0.8400
7	A6	A11	A7,A8,A9,A10	0.5626	A6	A7	none	0.7007
8	none	none	none	none	A8	A11	A9, A10	0.5626

Table 1. Discovered strategies with threshold $\beta = 0.01$ and $\beta = 0.3$

The algorithm discovers 7 strategies with $\beta = 0.01$ and 8 strategies using $\beta = 0.3$. With $\beta = 0.3$, we find that the first strategy begins with A1 activity (incoming claim), followed by A2 activity (Brisbane checks if sufficient information is available) and ends with A3 activity (Brisbane registers claim) with a probability of 0.4342.

We evaluated the discovered strategies with different values of β (from 0.01 to 0.9). The adjustment of the threshold to 0.3 is justified by the two following constraints: (1) the lower the value of β, the higher the number of activities per strategy, which is more interesting at the intentional level. (2) β is chosen such that no strategy contains more activities than half of the total activities number (here 5,5). For instance, in table 1 when $\beta = 0.01$, strategy 7 represents a set of six activities which is more than half of initial activities. Therefore, $\beta = 0.3$ satisfies these two constraints as the strategies found by this threshold are all inferior to 5.5 activities.

The next step is to infer the intentions from these strategies. However, even if this algorithm is able to find the groups of strategies, for now it is not able to find the intentions. In this paper, the intentions and the links between them (step b) will be determined by human inference and reasoning. In the future, we will automate this step using ontologies of activities and natural language analysis techniques [51]. The results are given in next section.

Process modeling using Map. Figure 2 presents the intentional Map process model manually built from the strategies found in Table 1 with $\beta = 0.3$. By inference, we determine intentions that can be fulfilled by strategies that have the same nature and we name the strategies according to the activities.

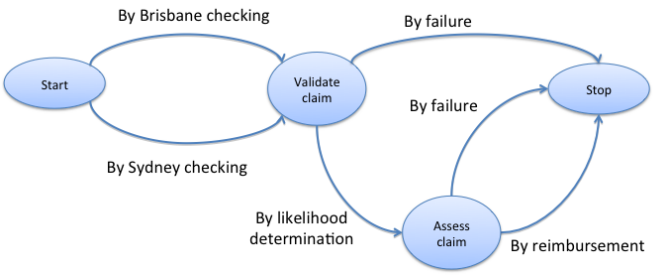

Fig. 2. The process model represented as intentional map.

Strategies 1, 2, 3 and 5, for instance, will be four ways to reach the same intention, which is *Validate the claim*. Strategies 1 and 2 show that there are two ways to validate the claim: either by *Brisbane checking* or *Sydney checking*. Strategies 3 and 5 correspond to the same strategies 1 and 2, as they correspond to a control activity that fails to validate the claim (the intention *Validate the claim* is not achieved). Strategy 7 corresponds to the set of activities that will allow to *Assess the claim* by *likelihood determination*. As for strategies 3 and 4, strategy 6 corresponds to a set of activities that does not allow reaching an intention, as there is an activity that fails (the claim cannot be assessed as the likelihood determination failed). Strategy 8 allows reaching the *stop the process* intention after the *reimbursement* of the client (note that activities A9 and A10 can be enacted in any order). Strategy 4 corresponds to the activities that end the process, so the intention reached in this case is *stop the process* when there has been a *failure* in the normal claim process (either in the claim checking or in the likelihood determination). This step will be automated in future works.

4.4 Discussion

This case study shows that either process mining or intention mining allows discovering a process model from a set of event logs. Process mining offers really good techniques to mine activity process models, not only to discover the models, but also to define the gap between the models and traces, and even to make recommendation about the possible following activities at run-time. However, these techniques focus only on the operational aspect of the process (activity oriented).

We strongly believe that intentional process models are an adequate formalism to guide users through the enactment of their activities. They allow modeling processes in a flexible way (the notion of sequence does not exist), variability can be introduced (alternatives path can be followed, different strategies can be used to achieve the same intention), thus, we think this kind of model allows more creativity than activity oriented process models. Intention mining for intentional process modeling is a new field of research and techniques are only emerging, however, we think that it will offer promising concepts and tools to mine intentional process models. Using intention mining techniques will help to promote this type of process models, as discovering them will be easier (as shown in the case study). Moreover, the techniques and algorithms that we are currently defining will also help to recommend tasks, based on the

supposed reasoning of the users (theirs intentions), as close as possible to human ways of thinking and working.

We think that intention mining will not be hampered by the same problems identified in process mining [50], as intentional process models are flexible. For instance, the problems of hidden tasks (no-recorded tasks) or duplicate tasks (a process model with same task twice) should be overcome with an intentional modeling, where activities are of less importance with a representation on a higher level. The concept of loop is also usual in intentional modeling - for instance in map process models, a section can be enacted several times, until the intention is achieved - whereas it is often a difficult problem to handle in process mining. However, intention mining is not the answer to these process mining problems, it is only another field of mining research, aiming to work on another kind of process models. It will have problems on its own: for instance, to define concurrent and exclusive strategies to achieve the same intention will be quite a problem to solve automatically, without human-expert inference.

5 Conclusion

In this paper, we presented the objectives, algorithms, models and tools for process mining and intention mining. The developed case study showed that, on this specific dataset, either process mining or intention mining techniques allow discovering a process model. However, intention mining (for intentional process models) is a recent emerging research field and a lot of work is still needed to develop full algorithms.

Our next step will be to automate the merging of strategies to define intentions and name them using ontologies and natural language analysis techniques. We will also provide recommendations using map process models and traces.

References

1. Van der Aalst W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, 1st Ed. , Berlin (2011)
2. Dowson M.: Iteration in the Software Process, Proc 9th Int. Conf. on Soft. Eng. (1998)
3. Veblen Th.: Why is Economics not an Evolutionary Science?, The Quarterly Journal of Economics, vol. 12, N°4, pp. 373–397 (1898)
4. Ralph P., Wand Y.: A Teleological Process Theory of Software Development, JAIS Theory Development Workshop, vol. 8, N°23 (2008)
5. Hug C., Deneckère R., Salinesi C.: Map-TBS: Map process enactment traces and analysis, Procs. of RCIS'12, Valencia, Spain, pp. 204–209 (2012)
6. Cook J.E., Wolf A.L.: Discovering models of software processes from event-based data, In: ACM Trans. on Soft. Engineering and Methodology, vol.7, N°3, pp.215–249 (1998)
7. Agrawal R., Gunopulos D., Leymann F.: Mining Process Models from Workflow Logs, Proceed. of the 6th Int. Conf. on Extending Database Technology: Advances in Database Technology, LNCS 1377, Springer, pp.469–483 (1998)
8. Mobasher B, Cooley R., Srivastava J.: Automatic Personalization Based On Web Usage Mining, Communication of ACM, vol. 43, N°8, 2000, pp. 142–151 (2000)

9. Schonenberg, H., Weber, B., & Dongen, B. V. : Supporting Flexible Processes through Recommendations Based on History. BPM, vol. 5240, pp. 51–66. Springer (2008)
10. Van der Aalst W., Stahl C.: Modeling Business Processes: A Petri Net Oriented Approach, MIT Press, Cambridge, MA (2011)
11. http://www.bpmn.org, consulted 1 February 2013.
12. Cook J., Wolf A.: Automating process discovery through event-data analysis, Proceed. of the 17th Int. Conf. on Software Engineering, ACM Press, pp.73–82, (1995)
13. Das S., Mozer M.C.: A unified Gradient Descent/Clustering Architecture for Finite State Machine Induction, In: proc. of the 1993 Conf., N°6 in Advances in Neural Information Processing system, Morgan Kaufmann, pp.19-26 (1994)
14. Biermann A.W., Feldman J.A.: On the Synthesis of Finite States Machines from Samples of Their Behavior, In: IEEE transactions on Computers. vol. 21, N°6, pp. 592–597 (1972)
15. Baum Leonard L.E., Petrie T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains, The Annals of Math. Stat., vol. 37, N°6, pp. 1554–1563 (1966)
16. Cook J.E., Wolf A.L.: Event-based detection of concurrency. In: Proceed. of the 6th Int. Symposium on the Foundations of Software Engineering (FSE-6), pp. 35–45 (1998)
17. Cook J.E., Wolf A.L.: Software process validation: quantitatively measuring the correspondence of a process to a model, In: ACM Transactions on Software Engineering and Methodology, vol. 8, N°2, pp. 147–176 (1999)
18. Wen L., Wang J., Sun J.G.: Detecting Implicit Dependencies between Tasks from Event Logs, In: Asia-Pacific Web Conference on Frontiers of WWW Research and Development (APWeb 2006), Lecture Notes in Computer Science, Springer, pp. 591–603 (2006)
19. Herbst J., Karagiannis D.: Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models, In: Proceed. of the 9th Int. Workshop on Database and Expert Systems Applications, pp.745–752 (1998)
20. Herbst J.: Dealing with concurrency in workflow induction, In: European Concurrent Engineering Conference, SCS Europe (2000)
21. Greco G., Guzzo A., Pontieri L.: Mining Hierarchies of Models: From Abstract Views to Concrete Specifications, In: Proceed. of BPM 2005, pp. 32–47 (2005)
22. Van der Aalst W., Medeiros A., Weijters A.:Genetic Process Mining, Applications and Theory of Petri Nets, LNCS 3536, Springer, pp. 48–69 (2005)
23. Weijters A. and van der Aalst W.: Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integ. Computer-Aided Engineering, 10(2), pp. 151–162 (2003)
24. Van der Aalst, W. M.: Formalization and verification of event-driven process chains, In: Information and Software technology, vol.41, N°10, pp. 639-650 (1999)
25. Jensen K., Kristensen L.M.: Coloured Petri Nets, Springer, Berlin (2009)
26. Van Dongen B., de Medeiros A., Verbeek H., Weijters A., van der Aalst W.: The prom framework: A new era in process mining tool support, In: Applications and Theory of Petri Nets 2005, vol.3536, Springer-Verlag, pp. 444–454 (2005)
27. Van der Aalst W., Van Dongen B.: Discovering workflow performance models from timed logs, Engineering and Deployment of Cooperative IS, pp.107-110 (2002)
28. Thevenet L. H., Salinesi C.: Aligning IS to Organization's Strategy: The I n S t A l Method, In: Advanced Information Systems Engineering, Springer Berlin, Heidelberg, pp. 203-217 (2007)
29. Medeiros A., Günther C. W.: Process mining: Using CPN tools to create test logs for mining algorithms, Procs. of the 6th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pp. 177–190 (2005)
30. Rozinat A., Mans R.S., Song M., van der Aalst W.: Discovering colored Petri nets from event logs. In: International Journal STTT, vol.10, N°1, pp. 57–74 (2008)
31. Merriam-Webster Dictionary. Merriam-Webster (2002)

32. Bratman M. E.: Intention, Plans, and Practical Reason, Harvard University Press (1987)
33. Jackson M.: Software Requirements & Specifications, a Lexicon of Practice, Principles and Prejudices, ACM Press. Addison-Wesley (1995)
34. Mei T., Hua X.S., Zhou H.Q., Li S.: Modeling and Mining of Users' Capture Intention for Home Videos, vol.9, N°1, IEEE Transactions on Multimedia, pp. 66–77 (2007)
35. Yu E.: Modelling Strategic Relationships for Process Reengineering, PhD thesis, University of Toronto, Dpt of Computer Science (1995)
36. Dardenne A., van Lamsweerde A., Fickasu S.: Goal-directed requirements acquisition, Sci. Comput. Program, vol. 20, N°1–2 (1993)
37. Rolland C., Prakash N, Benjamen A.: A Multi-Model View of Process Modeling, Requirements Engineering, vol. 4, N° 4. Springer-Verlag London Ltd (1999)
38. Prakash N, Rolland C.: Systems Design for requirements expressed as a map. Proceed. of the conference IRMA 06, Washington DC (2006)
39. Kornyshova E., Deneckère R., Salinesi C.: Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach. ME 07, Switzerland (2007)
40. Barrios J, Nurcan S.: Model Driven Architectures for Enterprise Information Systems, 16th conf. CAISE'04, Springer Verlag Lettonie (2004)
41. Juang B.H., Rabiner L.R.: A Probabilistic Distance Measure for Hidden Markov Models, AT&T Technical Journal, vol.64, N°2, pp. 391–408 (1985)
42. Forney G.D.: The Viterbi Algorithm. Proceeding IEEE, vol. 61, N°3, pp. 268-278 (1973)
43. Baum L.E., Petrie T., Soules G., Weiss N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains, Ann. Math. Statist., vol. 41, N°1, pp. 164–171 (1970)
44. Moon T.K.: The expectation-maximization algorithm, Signal processing magazine, IEEE, vol. 13, N°6, pp. 47–60 (1996)
45. Yu S. Z., Kobayashi H.: An efficient forward-backward algorithm for an explicit-duration hidden Markov model, Signal Processing Letters, IEEE, vol.10, N°1, pp. 11-14 (2003)
46. http://www.processmining.org, consulted 1st February 2013.
47. Pvan Glabbeek R., Goltz U., Schicke J.: On causal semantics of Petri nets. CONCUR 2011–Concurrency Theory, pp. 43-59 (2011)
48. Bleistein S. J., Cox K., Verner J., Phalp K.: B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process, In: formation and Software Technology, vol.48, N°9, pp. 846–868 (2006)
49. Van der Aalst W., Weijters A.: Process mining: a research agenda. Computers and Industry, vol. 53, N°3, pp. 231–244 (2004)
50. Van der Aalst W., Weijters A.J.M.M., Maruster L.: Workflow Mining: Discovering Process Models from Event Logs. In: IEEE Transactions on Knowledge and Data Engineering, vol. 16, N°9, pp. 1128–1142 (2004)
51. Lallé S., Luengo V., Guin N. : Méthode semi-automatique pour le développement d'un ensemble de techniques de diagnostic des connaissances. Conférence EIAH(2013)
52. Khodabandelou G., Hug C., Deneckère R., Salinesi C.: Supervised Intentional Process Models Discovery using Hidden Markov Models, In: Procs. RCIS'13, Paris, France (2013)
53. Khodabandelou G., Hug C., Deneckère R., Salinesi C., Bajec M., Kornyshova E., Janković M.: COTS Products to Trace Method Enactment: Review and Selection, In Procs. of ECIS'13, Utrecht, Netherlands (2013)
54. Khodabandelou G., Contextual Recommendations Using Intention Mining on Process Traces, Procs. of RCIS'13 Doctoral Consortium, Paris, France (2013)