Supervised Intentional Process Models Discovery using Hidden Markov Models

Ghazaleh Khodabandelou, Charlotte Hug, Rebecca Deneckere, Camille Salinesi

Supervised Intentional Process Models Discovery using Hidden Markov Models

Ghazaleh Khodabandelou, Charlotte Hug, Rébecca Deneckère, Camille Salinesi
Centre de Recherche en Informatique
University of Paris 1 Panthéon-Sorbonne
Paris, France
Ghazaleh.Khodabandelou@malix.univ-paris1.fr, {Charlotte.Hug, Rebecca.Deneckere, Camille.Salinesi}@univ-paris1.fr

Abstract— **Since several decades, discovering process models is a subject of interest in the Information System (IS) community. Approaches have been proposed to recover process models, based on the recorded sequential tasks (traces) done by IS's actors. However, these approaches only focused on activities and the process models identified are, in consequence, activity-oriented. Intentional process models focus on the intentions underlying activities rather than activities, in order to offer a better guidance through the processes. Unfortunately, the existing process-mining approaches do not take into account the hidden aspect of the intentions behind the recorded user activities. We think that we can discover the intentional process models underlying user activities by using *Intention mining* techniques. The aim of this paper is to propose the use of probabilistic models to evaluate the most likely intentions behind traces of activities, namely Hidden Markov Models (HMMs). We focus on this paper on a supervised approach that allows discovering the intentions behind the user activities traces and to compare them to the prescribed intentional process model.**

Keywords—intention mining; process modeling; supervised learning; process discovery

I. INTRODUCTION

While many process models have been proposed to guide information systems engineering, still much remains to be done to understand their actual implementation. Analyzing the traces enacted by Information System Engineering (ISE) stakeholders allows detecting the gap between the prescribed process model and what is actually done by the stakeholders. Such gap occurs when the prescribed process model is under-matching (mediocre precision with the presence of noisy behaviors), over-matching (mediocre generalization which recovers only current behaviors) or ill-matching (missing or/and irrelevant behaviors). Any of these gaps could cause the failure of the project or at least be a catalyst to lower the quality of software products.

Several process mining approaches have been proposed to discover process models from event logs [6,7,10,17]. While these methods help to recover the sequences of activities and to extract information from those sequences, they overlook the hidden intentions that generate them. Intentions are important as they capture what the stakeholders intend to perform [1]. To fulfill their intentions, actors perform a set of activities; they also may change their intentions during the process enactment and sometimes have random behaviors. As a result, we believe that whereas sequences of activities represent a set of actors' behaviors executed one after another, the fundamental process behind them is mostly intentional. However, discovering intentional process model from event logs is part of a whole new field of researches, which we call *Intention Mining*.

The main objective of *Intention Mining* is to extract sequences of actors' activities from an event log to evaluate and predict the actors' intentions related to those activities [36][37],[38]. Knowing the intentions underlying actors' activities allows improving the actors' guidance through their tasks. As a matter of fact, if we know why an actor is performing some activities, we can have a better handling of what he may want to do on his next step and offer him better recommendations. Intention mining can be useful at different levels, for instance: at the design level, to identify the underlying intentions hidden in the logs in order to define the followed process model; at the analysis level, to identify the gap between a prescribed process model and the activities actors are actually performing; at the application level, to recommend steps to the actor at run-time, following logs history.

This work focuses on the discovery of the intentions hidden in the logs and on the intentional process model enactment. To rebuild the process model (the followed model that generates the activities' traces) and evaluate the intentions, we use Hidden Markov Models (HMMs) [8] and Map, an intentional process metamodel [1]. Map allows representing flexible process models, enacted in a dynamic way since the sections of a Map process model can be executed non-sequentially and as long as intentions are not completely fulfilled. We propose a two-step method. The first step consists in a training phase to estimate the parameters of HMM using sequences of activities whose intentions are known. The second step consists in recovering the intentions in a fully automated way by using the estimated parameters on a new dataset. In this paper, we describe a case study that consists in analyzing the traces produced by students designing entity-relationship diagrams.

This paper is organized as follow: in section 2, we present the related works. In section 3, we describe the technical contribution based on HMMs. In section 4, we present the results of the case study and we conclude this paper in section 5.

II. RELATED WORK

HMMs are applied in many fields where the objective is to rebuild the hidden states of observed processes, such as speech recognition [25] and bioinformatics [30].

A comparable kind of challenge arises in process mining, where the goal is to extract observed activities from event logs to recover the original workflow that produced the logs. These event logs contain the traces of process executions. In process mining approaches, the objective is generally to discover the sequence of tasks and activities underlying processes. To do so, these approaches extract the information from event logs using different algorithms and techniques such as classification and learning techniques [2].

One approach to discover the processes is Machine Learning (ML) [14, 18]. Depending on the type of inputs, there are various ML approaches: *supervised, unsupervised, semi-supervised and reinforcement*. S*upervised-learning* is based on two steps: (a) in a *training step,* a learning algorithm is applied on observations (independent variables) to train a classifier, (b) in a *prediction step,* the trained classifier can be tested using a testing set of observations, and most importantly, the classifier can be applied on any new dataset. Figure 1 illustrates this approach.

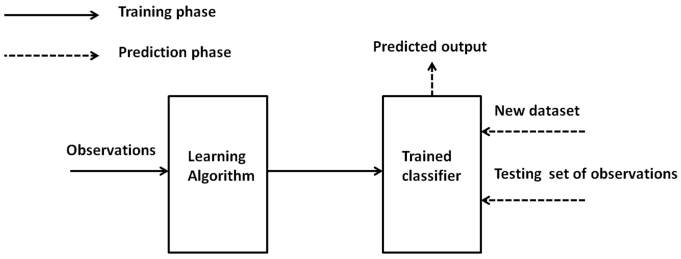

Fig. 1. Illustration of supervised learning.

The aim of this approach in process mining is to classify sequences of activities into classes by looking for similarities between them. Some approaches to classify traces exist, such as [17, 19].

In [17], a *trace clustering* approach is used to overcome the problems of non-well-structured processes by classifying event logs in terms of cases. This approach focuses only on activities' sequences and does not take into consideration the intentions behind these activities.

In [19], the *sequence clustering* approach has been proposed to tackle the problems of classical methods of classification by collecting the different methods into one. This approach aims at dividing the sequences into meaningful clusters or groups of similar sequences.

Based on the nature of the mining algorithms, different process mining approaches have emerged [19], such as α-algorithm [10], directed acyclic graphs [15], hierarchical clustering [9], genetic algorithm [5], instance graphs [4] or inductive workflow acquisition [14].

All these algorithms require using event logs with some additional information. For instance, as many traces of execution of the same process are recorded, the *case id* (process instance identifier) of each process instance has to be known for each trace. Another necessary data for algorithms such as directed acyclic graphs and α-algorithm is a threshold (algorithm parameter). This threshold limits the reconstruction of the causal relations under a certain probability and prevents the noise. This kind of algorithms has a good performance when it is applied on well-structured processes e.g., workflow processes [[10], 13]. But the intentional aspect of a process is completely overlooked, while this aspect explains why the process is needed or has a particular content or structure rather than another. Therefore, in order to capture this kind of additional knowledge, another kind of algorithm is needed to capture the intentional insights into the processes.

In [14], a workflow is modeled with HMMs that reflect the process model on another level. This work does not address the intentional dimension of processes. As a result, it is for instance not possible to reason about the importance of intentions or to recognize whether an instance of workflow deviates from the goals of the project.

In this paper, we apply supervised HMMs in the line of several former works.

In [2], HMMs are considered as versatile and relevant for process mining but HMMs unsupervised approaches are complicated: (a) there are computational challenges due to time consuming iterative procedures, (b) the number of states (as algorithm inputs) should be known, (c) the result of HMMs is not very understandable for the end-user.

In [16, 27], three different inference algorithms illustrate process discovery – they infer process models from event logs: RNet, Ktail and Markov.

(a) RNet [31] is a statistical approach that characterizes a state depending on the past behaviors. RNet generates a Deterministic Finite State Machine (DFSM), i.e., each state has only one transition for each possible input. It is robust to noise but is very time consuming in the training phase, the size of the net increases with the number of token's types and it requires to evaluate many parameters.

(b) Ktail [26] is an algorithmic approach that evaluates the current state depending on future behavior. The input is a set of sample strings and the output is a Finite State Machine. One can control the complexity of the algorithm when the number of states increases by DFSM. The disadvantage of this method is that it is not very robust to noise.

(c) Markov is a hybrid, between the statistical and algorithmic approaches looking at the neighboring past and future behavior to define the future state. It is robust to noise with a complexity that is controllable and has a deterministic FSM. However, this algorithm tries to discover only the processes from event logs and does not consider the actors' intentions behind these processes.

Another work, [12], uses HMMs as a conformance checking technique by measuring similarities between Markov models using a distance metric. It enables the authors to evaluate the quality of mined processes. The workflows modeled in Petri nets are mapped to HMMs but the hidden states of processes are not taken into account.

Nevertheless, all these works consider the hidden states of HMMs as the instances of the process while we consider the hidden states as the intentions that generate the observed sequences of activities. We do not model a process only as a set of tasks as Petri nets do, but as the intentions behind the activities' sequences. Moreover, many classical techniques of classification like SVM (Support Vector Machine) [28] cannot deal with the noise (incomplete or irrelevant data) and do not support the variability of data sequences - they only accept a prescribed length of sequences whereas sequences of activities' lengths are variable depending on the actors' purposes.

III. TECHNICAL CONTRIBUTION

Our technical proposition consists in:

- Modeling an intentional process model using HMM. The HMM fits the intentional process model since it models hidden states, i.e., intentions, found in observable data, i.e., traces of activities.
- Estimating the parameters of the HMM based on traces obtained from practical uses of the intentional process model.
- Predicting the next intentions and consequently the next activities using the estimated parameters of HMM.
- Evaluating the intentions associated to a sequence of activities using the *Viterbi Algorithm* (VA), given the parameters of the HMM. We discuss the performances of this evaluation in section IV.

Our work relies on the assumption that the realization of traces of activities follows a stochastic process. A stochastic process represents an evolution, discrete or continuous, of a random variable. Thus, we need to choose a statistic model that allows (a) knowing the observed sequences; if they are significant or only accident outcomes, (b) analyzing the observed sequences over time, (c) modeling the latent states of these observed sequences, (d) extracting the characteristics of observed and latent sequences. Among the probabilistic models we select HMMs [21,25] for several reasons. As described in the previous section, the other methods do not fit the intentional model. HMMs are applied widely in bioinformatics' field [20, [30]]. Nevertheless, as indicated earlier, several applications of HMMs in process mining [12, 14] consider the hidden states as the process instance. The contribution of this paper consists in applying HMM to discover intentions from traces of activities, that we name *Intention Mining* [38].

HMMs are a special kind of stochastic process that captures the relations between an observable sequence (the activities) and their hidden states (associated intentions). More precisely, the framework of HMMs allows focusing on several problems such as (a) How to estimate the parameters of the HMMs? (b) What is the probability of a given activities sequence? (c) What are the most probable intentions associated to a given activities sequence? The last question is of great interest for our work since we aim at finding the hidden intentions associated to a sequence of traces of activities. We present in the following subsections an overview of HMMs.

A. Hidden Markov Models (HMMs)

1) Definition

A HMM is a statistical signal modeling formalism that allows modeling a sequence by a finite number of states that alternate. HMMs are very flexible due to latent data that allows modeling the structure of complex temporal dependencies. The systems modeled by HMMs are based on two complementary Markov processes, i.e., specific types of stochastic processes.

To understand what is a Markov process, let $(X_1, \dots, X_T)$ be a sequence of random variables with length T generated by a Markov chain of order m (or with memory m), where m is finite. The generating process is:

$$\mathbb{P}\left(X_t = x_t \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_1 = x_1\right) = \mathbb{P}\left(X_t = x_t \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-m} = x_{t-m}\right)$$

$$for\ t > m. \quad (1)$$

This means that for a Markov chain of order m, the transition to the next state depends only on the m previous states. The choice of m indicates how far in the past one has to look to know the probability of the next state.

In the context of HMMs, we need to define two Markov processes. A first Markov process models the hidden state of the system. A second Markov process models the observations of the system, knowing that this second process depends on the hidden states. In our framework, the hidden states are the intentions and the observations of the system are the traces of activities. The topological structure of HMMs allows defining, according to the context, dependences between hidden states and/or observed data from the past to the future ones.

2) Models of HMMs

Choosing the right order for Markov chains in an HMM is a challenge since the orders of the chains define the behavior of the HMM. As a first step, we will work with a *M1M0* model.

A *M1M0* model means that among the different orders of Markov chains, we rely on a Markov chain of order 1 and a Markov chain of order 0. A Markov chain of order 1 is a process where the transition to the next state depends only on the current state (also called Markov property). A Markov chain of order 0 is a process where the transition to the next state does not depend on any state. We now define the order for each Markov chain.

Definition 1: Hidden process. For a given time *t*, the state of the system I_t is only dependent on the state of the system at the previous time step I_{t-1}. This definition applies to the intentions (model *M1*). A sequence of intentions is denoted by $I_{1:T} = (I_1, \dots, I_T) \in S^T$, with S being the set of intentions and T being the length of the sequence. A homogeneous Markov chain, which parameter is denoted by q, models the hidden process of intentions with:

$$q(u, v) = \mathbb{P}(I_{t+1} = v | I_t = u), \forall u, v \in S, \quad (2)$$

and

$$q(u) = \mathbb{P}(I_1 = u), \forall u \in S. \quad (3)$$

The parameter q in (3) contains the probabilities of intention at the initial state and in (2) the transition probabilities for the following intentions.

Definition 2: Observed process. For a given time t, the observation A_t does not depend on any previously observed sequence. This definition relates to the sequence of activities (model *M0*). We denote an observed sequence of actors' activities by $A_{1:T} = (A_1, \dots, A_T) \in R^T$, with R being the set of activities. The emission probability p of an observation $a \in R$ for a given intention $u \in S$, is given by:

$$p_u(a) = \mathbb{P}(A = a | I = u). \quad (4)$$

P and q are the parameters of the HMM, namely the *transition* and *emission* probabilities. The transition probabilities are the probabilities of a hidden state at time t to reach another hidden state at time $t + 1$ (or to stay in the same state) as shown in figure 2.

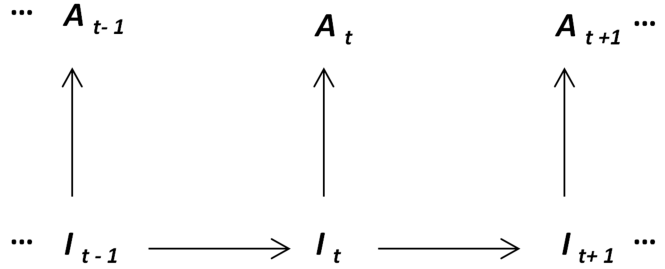

Fig. 2. Transitions of intentions and emissions of activities in a M1M0 HMM.

The emission probabilities are the conditional probability distribution of the observed variables in a given hidden state at time *t*. This model is a *M1M0* model since the hidden process is a first-order Markov chain and the observed process is a zero-order Markov chain. In other words, this means that the choice for executing an activity among all while enacting a process model is not isolated, but correlated to other activities achieved before.

As an example, we consider a case with three hidden intentions $\{I_1, I_2, I_3\}$ and four observed activities $\{a_1, a_2, a_3, a_4\}$. The associated *M1M0* model is described by the probability distribution of the initial intention and by two matrices: one 3×3 matrix for the transition probabilities of hidden intentions denoted by I (6) and a 3×4 matrix for the emission probabilities of observed activities for each intention, denoted by A (7).

$$I = \begin{pmatrix} q(I_1, I_1) & q(I_1, I_2) & q(I_1, I_3) \\ q(I_2, I_1) & q(I_2, I_2) & q(I_2, I_3) \\ q(I_3, I_1) & q(I_3, I_2) & q(I_3, I_3) \end{pmatrix} \quad (6)$$

$$A = \begin{pmatrix} p_1(a_1) & p_1(a_2) & p_1(a_3) & p_1(a_4) \\ p_2(a_1) & p_2(a_2) & p_2(a_3) & p_2(a_4) \\ p_3(a_1) & p_3(a_2) & p_3(a_3) & p_3(a_4) \end{pmatrix} \quad (7)$$

B. Estimation of parameters

Once we chose the model, we have to estimate the parameters of the HMM. There are two approaches to estimate the parameters of a HMM: *supervised* and *unsupervised*. The supervised approach is used when the segmentation of some activities' sequences into intentions is known while the unsupervised approach is used when the segmentation of any activities sequence is unknown.

We adopt the supervised approach in this paper. As indicated in section II, the supervised approach involves two phases. The first phase is a *learning phase* that consists in training the algorithm with sequences of activities to find the parameters of the HMM. For these sequences, the segmentation is known, i.e., the intentions related to the traces of activities are known. The second phase of the supervised approach, defined in subsection C, consists in the evaluation of the intentions related to a given activities' sequence.

The estimation of the parameters consists in finding the probability distribution of traces - the couple (p, q) defined in (2) and (4). If a sequence of activities $A_{1:T}$ is available and the corresponding intentions are known, the computation of the estimates for (p, q) consists in using the Maximum-Likelihood Estimation (MLE) [11]. This method estimates the parameters $q(u, v)$ and $p_u(a)$ such that they analytically maximize the likelihood of having simultaneously the intentions $I_{1:T}$ and the sequence of activities $A_{1:T}$. It amounts in counting the number of transitions from one intention to another and the number of apparitions of each activity during each intention as shown below:

$$\hat{q}(u, v) = \frac{Num(u,v)}{\sum_{x \in S} Num(u,x)}, \quad (8)$$

and

$$\hat{p}_u(a) = \frac{Num(a|u)}{Num(a)}, \quad (9)$$

where $Num(u, v)$ in (8) denotes the number of transitions from intention *u* to intention *v*, $Num(a)$ in (9) denotes the number of apparitions of activity a and $Num(a|u)$ denotes the number of apparitions of activity a while the intention is *u*.

These parameters allow learning the distribution of the activities and intentions in the process model. This *learning phase* is necessary for the next step to evaluate the most likely intentions associated to a given activities' sequence.

C. Evaluation of intentions' sequences

Once the parameters of the HMM are estimated, we want to be able to identify the most likely set of intentions associated to any sequence of activities. To do so, given a sequence of activity $A_{1:T}$ of length T, one could generate all the possible intentions of length T. Then, for each intention $I_{1:T}$, one could compute the probability $\mathbb{P}(A_{1:T}|I_{1:T})$. However, this is a brute-force search and it cannot be used to compare all the possible intentions for complexity reasons. For instance, if the number of intentions is C, the complexity will be C^T, which increases exponentially with *T*.

Instead, the *VA* [23] is used to obtain, from a given observed sequence, the most likely hidden sequence of intentions that might generate it. The *VA* is commonly used in the context of HMM; this algorithm is able to calculate the probability that an observation or an intention has been changed into another, and radically simplifies the complexity of the search for the most likely hidden sequence. Thereby, the exponential complexity becomes linear.

This corresponds to the second phase of the supervised approach. As we mentioned earlier, to use the *VA,* it is necessary to know the estimated parameters obtained in the previous phase. We note that a given sequence of activities, with length T, may be generated by many related intentions with the same length; nevertheless, one sequence among all has the highest probability of emergence. In other words, this sequence is the most likely sequence of intentions, which generates the related sequence of activities. The mathematical description of this phase is presented hereafter.

Given a sequence of traces of activities $A_{1:T}$, the estimated parameters $\hat{p}_u(a)$ and $\hat{q}(u,v)$ and the initial probabilities for each intention, the *VA* tries to find the hidden associated intentions' sequence $\bar{I}_{1:T}$ which maximizes:

$$\mathbb{P}(I_{1:T}|A_{1:T}). \tag{10}$$

The problem can also be written as:

$$\bar{I}_{1:T} = \arg\max_{I_{1:T}} \mathbb{P}(I_{1:T}|A_{1:T}) \tag{11}$$

It is important to highlight that the VA's output is a sequence of intentions. It enables to account for the history of the users' activities. In the next section, we detail how we use HMM in a specific case study.

IV. APPLICATION OF HMMS ON REAL DATA

We conducted an experiment in order to obtain traces from students in computer science master's degree. The process model that was used to guide them was intentionally specified with Map. The dataset obtained from this experiment is discussed in detail in subsection C.

A. Context

In order to get traces, we recorded the activities achieved while creating an Entity-Relationship diagram according to the intentional process model (figure 3) prescribed through a website. The set of students is composed of 71 master students. Table I presents the profile of the students.

TABLE I. PROFILE OF THE STUDENTS

Total	Average age	Sex		Master degree	
		Male	*Female*	*1st year*	*2nd year*
71	24,4	54	17	53	18

The process model presented in Figure 3 is an instance of the Map metamodel [1] that provides a navigational structure that supports the dynamic selection of the next intention to be achieved and the appropriate strategy to achieve it. The Map metamodel allows to specify processes according to actors intentions, and it supports process variability by defining different strategies. Thereby confronted to a specific situation and a particular intention of the user, the process model reveals the alternative strategies to follow the intention, and the intentions to pursue. Hereafter, we give an overview of the *intentional process model* used in the case study.

B. Intentional process model

The *intentional process model* represents a process to guide users through the creation of Entity-Relationship diagrams based on [22]. As mentioned earlier, this process is specified with the Map formalism. According to this model, users can select eleven strategies to fulfill three intentions. These intentions are *Specify an entity, Specify an association* and *Stop.* Figure 3 shows this *intentional process model* as a directed graph. Each edge represents a strategy that a user can select to fulfill an intention (specified as a node) according to his/her situation. For instance, if the current situation is *Start* and the user's intention is to *Specify an entity,* there is only one strategy (by completeness of the model) to fulfill this intention. When the current situation is *Specify an entity,* there are four strategies (by completeness, by generalization, by specialization, by normalization) to fulfill the same intention. It is possible to continue progressing in the process by selecting the strategies that lead to the considered intentions but once the *Stop* intention is achieved, the enactment of the process is finished. The original map proposed in [22] was more elaborated but we choose to simplify it for this case study.

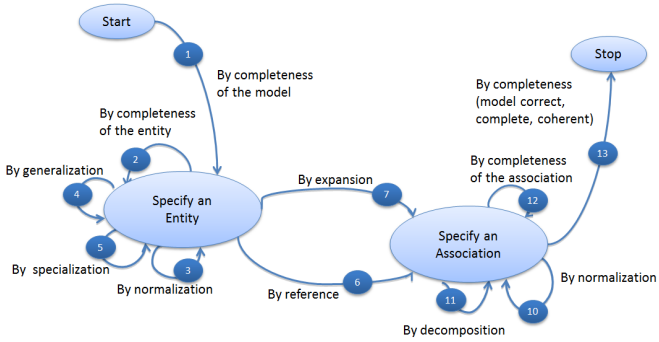

Fig. 3. Intentional process model. (from [22])

To fulfill an intention following a strategy, users have to carry out actions. The term of action differs from activity - the actions are performed by the users but the activities are the user's actions recorded by an IS tool. There are fifteen actions related to the Map. Table II gives the names of all the actions and Table III shows in detail the link between each section and its related actions.

As can be noted in table III, several actions (consequently several activities) jointly appear simultaneously in intentions *Specify an entity* and *Specify an association,* such as *delete an attribute.* Indeed, a given action can be executed to achieve several different intentions.

TABLE II. SECTIONS AND RELATED TRACE ACTIONS

Map Section		Related Trace Actions	Actions Codes
1	Specify an entity By completeness (model)	Create entity	A1
2	Specify an entity By completeness (entity)	Create attribute Link attribute to entity	A3, A4
3	Specify an entity By normalization	Delete attribute Delete Link attribute to entity	A10, A15
		Delete entity Delete attribute * (Delete association, Delete attribute *) *	A11, A10*, (A12, A10*)*
		Define primary key	A9
4	Specify an entity By generalization	Create entity Create generalization link	A1, A7
5	Specify an entity By specialization	Create entity Create specialization link	A1, A8
6	Specify an association By reference	Delete attribute Create entity Create association Link association to entity Link association to entity	A10, A1, A2, A6, A6
7	Specify an association By expansion	Create association	A2
10	Specify an association By normalization	Delete association (Delete attribute, Delete Link attribute to association)*	A12, (A10, A15)*
		Delete attribute	A10
11	Specify an association By decomposition	Create association Link association to entity Link association to entity	A2, A6, A6
12	Specify an association By completeness (association)	Create attribute Link attribute to association	A3, A5
13	Stop By completeness (final)	Check coherency Check completeness	A13, A14

** Iterative action*

TABLE III. RELATED ACTIONS AND CODES

Related Actions	Code
Create entity	A1
Create association	A2
Create attribute	A3
Link attribute to entity	A4
Link attribute to association	A5
Link association to entity	A6
Create generalization link	A7
Create specialization link	A8
Define primary key	A9
Delete attribute	A10
Delete entity	A11
Delete association	A12
Check coherency	A13
Check completeness	A14
Delete Link	A15

As a result, it is not trivial to know the intentions of all the actors when executing an action. Our purpose here is to provide a method to find the intentions hidden behind a sequence of activities. The next section applies the HMM previously defined to the dataset described in subsection A, i.e., the traces of users enacting the intentional process model to create Entity-Relationship diagrams.

To be able to record the actors' traces, we developed a web application with HTML, PHP, JavaScript and MySQL. This application records traces of executions carried out by the users during the creation of their diagrams in a database. Each record in event log usually contains information about the action that was executed, the process instance it belongs to, and the timestamp of the action execution. The model used to store the traces can be found in [3].

C. Estimation of the parameters of the HMM

Since we monitored the creation of *Entity-Relationship* diagrams by the users following the intentional process model, it was easy to know the intentions hidden behind the actions traces. The dataset is made of one sequence of actions' traces and the associated sequence of intentions. In total, we recorded 4141 actions traces produced by 71 students. Consequently, the length of the actions' traces sequence and the associated intentions is 4141. The knowledge of the intentions enables us to work with the framework of *supervised intention mining* to estimate the parameters of the HMM. As explained in section III, estimating the parameters of the HMM consists in estimating the coefficients of the matrices A and I. Since the intentional process model comprises 3 intentions and 15 actions, the size of I is 3×3 and the size of A is 3×15. We obtain the coefficients of the transition matrix I by counting the number of transitions from one intention to another and the coefficients of matrix A by counting the number of times each action trace appears for each intention.

$$\hat{A} = \begin{pmatrix} 0.1276 & 0 & 0.4020 & 0.4020 & 0 & 0 & 0.0008 & 0.0025 & 0.0068 & 0.0240 & 0.0102 & 0.0102 & 0 & 0 & 0.0138 \\ 0.0436 & 0.4782 & 0.1239 & 0 & 0.1239 & 0.0873 & 0 & 0 & 0 & 0.0768 & 0 & 0.0332 & 0 & 0 & 0.0332 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix} \quad (12)$$

Of course, the quality of the estimated coefficients depends on the length of the sequences used to calculate the estimates. If the length of the sequences is too short, the sequences will not capture all the user behaviors and the estimated coefficients will be of poor quality. On the contrary, if the length of the sequences is long enough, we can capture the typical behaviors of the students and the estimated coefficients will be satisfying. We can verify this phenomenon with our dataset. First, we estimate matrices A and I with the full length of the sequences, i.e., 4141. Then, for eleven different lengths of sequences (1, 5, 10, 20, 30, 40, 50, 100, 200, 300, and 400), we estimate matrices A and I again. We obtain eleven couples of matrices of different quality.

On figure 4, for each sequences' length, we represent the average difference between the coefficients of estimated matrices $\hat{A}$ and $\hat{I}$ and the coefficients of the matrices estimated with the full sequences of length 4141. The error of estimation of coefficients decreases with the length of the sequences. For instance, for an estimation over sequences of length 20, the error is 0.15 for the parameters of $\hat{A}$ and 0.08 for the parameters of $\hat{I}$. Nevertheless, for sequences of length 400, the errors for the transition and emission matrices decrease to 0.05 and 0.001 respectively.

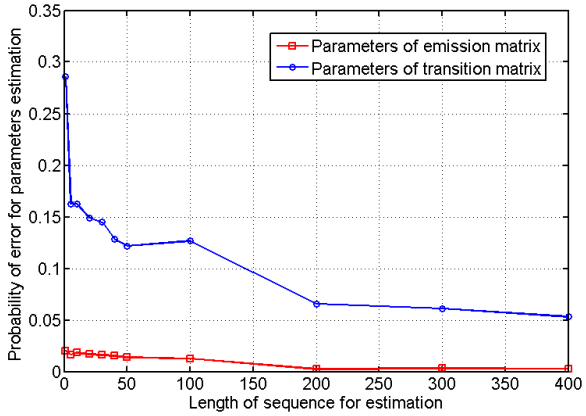

Fig. 4. Error of estimation for parameters depending on sequences' length.

Equations 12 and 13 give, respectively, the estimated matrix of emission of actions in a given intention and the estimated matrix of transition probability for the intentions. We obtain these results for sequences of length 4141. These results are the best estimation of the parameters.

$$\hat{I} = \begin{pmatrix} 0.9438 & 0.0522 & 0.0040 \\ 0.2775 & 0.6771 & 0.0454 \\ 0.4588 & 0 & 0.5412 \end{pmatrix} \quad (13)$$

In equation 13, it is important to highlight the coefficient $\hat{I}_{31}$ (0.4588). Indeed, this coefficient would mean that there is a possible transition from *Stop* to *Specify an entity*. But we want to make it clear that it is not the case: there is no possible transition from *Stop* to *Specify an entity*. The reason why this coefficient is not zero is that we treat the traces of different users sequentially. Consequently, when the trace of a student ends by the intention *Stop*, it is followed by the trace of another student starting with *Specify an entity*.

Regarding the intentions, Figure 5 illustrates the process model obtained by the transitions probabilities. We can deduce from these results that when actors enact *Specify an entity*, they have a high probability (0.9438) to continue performing this intention. They can switch to *Specify an association* with a probability of 0.0522, and they can go directly to *Stop* with a probability of 0.004. This last transition is very surprising since it is not allowed in the model given on Figure 3. It means that some of the students deviated from the prescribed process model. When they enact *Specify an association*, they mostly continue performing this intention with a probability of 0.6771 or switch to *Specify an entity* with a probability of 0.2775. Once again, this transition is not present in the prescribed process model. Consequently, this is a deviation from the prescribed process model. Finally, when they enact *Stop*, the only next intention possible is *Stop*. This is an expected result since two different actions are performed in this intention: *check coherency* and *check completeness*. As a result, the action *Check coherency* is always followed by *check completeness*. Once a student has performed the two actions of *Stop*, his/her trace is stopped. Consequently, the following intention comes from another student starting the process with the first intention (*Specify an Entity*).

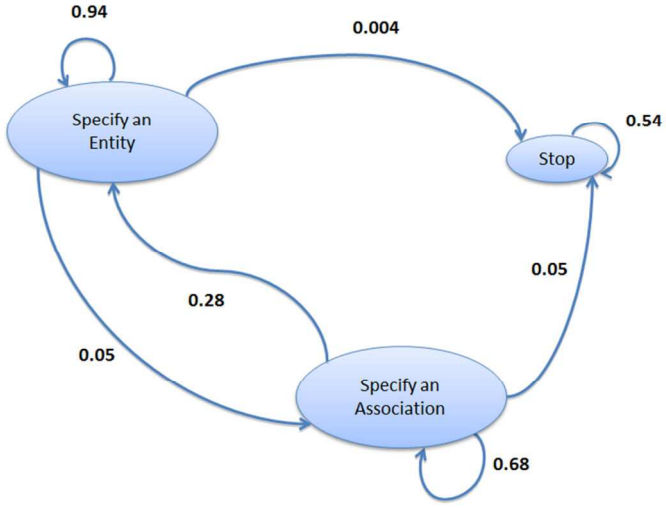

Fig. 5. Discovered process model obtained by transitions probabilities.

Regarding the emission matrix, we can verify that the actions $\{A_2, A_4, A_5, A_6, A_7, A_8, A_9, A_{11}, A_{13}, A_{14}\}$ only appear for a unique intention whereas the other actions $\{A_1, A_3, A_{10}, A_{12}, A_{15}\}$ can appear for several intentions. It is consistent with the definition of the traces of activities given in subsection B and it means that finding the intentions associated

to these latter actions is a difficult task. With a trivial method, one cannot distinguish which action belongs to which intention.

In terms of actors behaviors, we learned that the most executed actions to *Specify an entity* are A_3 and A_4: *Create attribute* and *Link attribute to entity*. The most executed action to *Specify an association* is *Create an association*. This knowledge is useful to understand the behaviors of the students.

D. Intentions recovery for random actions traces

Once we estimated the parameters of the HMM, we can use them to find the hidden intentions behind any sequence of actions traces, by using the *VA*. Our interest here is to determine the minimum length of sequences required to estimate the parameters of the HMMs in order to recover the right intentions with the *VA*. Consequently, we compare the performances of the *VA* with the eleven couples of estimated matrices obtained with the estimation lengths of 1, 5, 10, 20, 30, 40, 50, 100, 200, 300, and 400.

The comparison protocol is the following: for 1000 test sequences of actions of length 1500, we apply the *VA*, with each couple of matrices, on each sequence of actions. For each test sequence of actions, we get eleven predicted intentions. By comparing predicted intentions to the actual ones, and averaging the results over the 1000 realizations, we obtain the error percentage associated to each couple of matrices. We present the results in figure 6. It is interesting to note that the number of errors with the *VA* decreases when the estimated matrices are good. More precisely, with matrices estimated with sequences of length 200, we have an error percentage lower than 5%, which is satisfying.

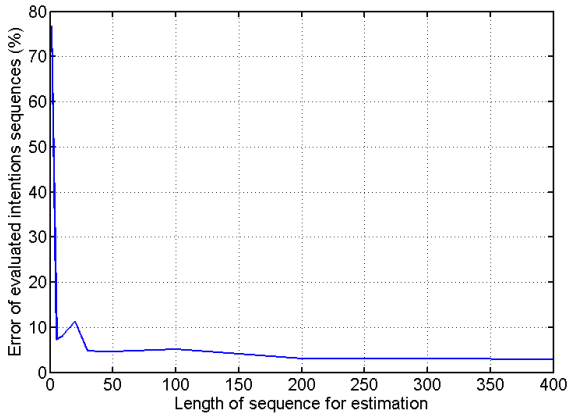

Fig. 6. Error of evaluation of intentions depending on sequences' length for estimation.

E. Validation

1) Measures of method validation

In order to evaluate the results that are provided by application of the *VA* on real data, we choose the recall, precision and F-factor (a combination of recall and precision) measures [29]. Before explaining these measures, we give a brief overview of some terms according to the context of this paper.

The True Positives (TP) represents the number of intentions correctly assigned by the *VA* as belonging to the right class of intention.

The False Negative (FN) represents the number of intentions, which were not assigned to the right class of intention.

The False Positive (FP) represents the number of intentions incorrectly assigned to the class of intention.

We evaluate the accuracy of the *VA* prediction by checking if the prediction matches to the actual intentions. Table IV shows these definitions:

TABLE IV. PREDICTION AND INTENTION

		Intention	
		True	**False**
Prediction	**True**	True positive (TP) (correct result)	False positive (FP) (unexpected result)
	False	False negative (FN) (missing result)	True negative (TN) (correct absence of result)

Recall is the ratio between the number of intentions correctly identified by the *VA* and the number of intentions in the dataset. Note that it does not take into account the number of intentions falsely identified by the algorithm. Recall is defined by the following expression:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{14}$$

Precision denotes the ratio between the number of intentions correctly identified by the *VA* and the number of intentions identified by the algorithm. Precision is defined by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{15}$$

In general, it is possible to increase recall to reduce precision and vice versa. F-score is a combination of precision and recall:

$$\text{Fscore} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision+recall}} \tag{16}$$

This is also known as F-1 measure, because precision and recall are weighted equally. F-1 measures the effectiveness of intention's recovery, considering the same importance for recall and precision.

2) Results of method validation

We calculate the recall, precision and F-score for the three intentions. Figure 7 shows these measures averaged over the 1000 test sequences for the intention 1: *Specify an entity*. The first ascertainment shows that the three curves are stabilized at an estimation sequence length of 200. It means that from a length of 200, the *VA* provides stable results. The result of recall expresses that the algorithm finds 99,50% of actions related to *Specify an entity*. This means that almost all the activities associated to intention 1 are identified with the *VA*.

Now the question is: does the algorithm associate several actions to intention 1 while, in fact, they belong to other

intentions? The value of the precision ratio is the answer to this question. The precision result stabilizes at 97%, which means 3% of the actions are ill associated to *Specify an entity* while they belong to other intentions.

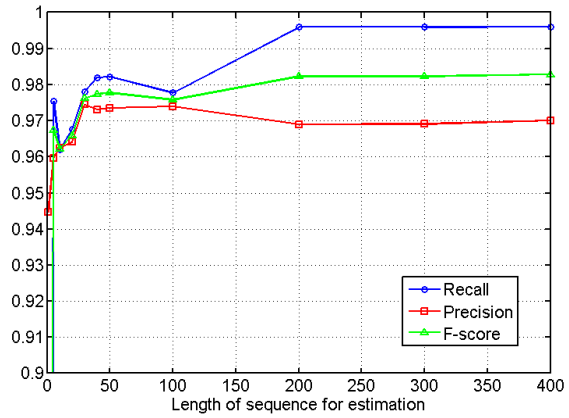

Fig. 7. The ratios of recall, precision and F-score for intention 1.

F-score is a compromise between recall and precision, as mentioned earlier; we consider an equally weight for both of them. We find an accuracy of retrieval for the intention to *Specify an entity* around 0.98. In other words, when the user's intention is *Specify an entity,* the *VA* is able to find this intention with an excellent accuracy.

We applied the same measures for the *Specify an association* intention and present them in figure 8. For these measures, the three curves are stabilized around the estimation sequences' length of 200, too. From a sequence length of 200, the recall value is 0.80. This means the *VA* finds 80% of the actions related to the *Specify an association* intention. The recall is lower than for the previous intention, because the *VA* identifies the actions associated to intention 2 as actions associated to intention 1. That is why the recall of intention 2 is 80%. However, the precision of intention 2 is better and stabilizes at 96%.

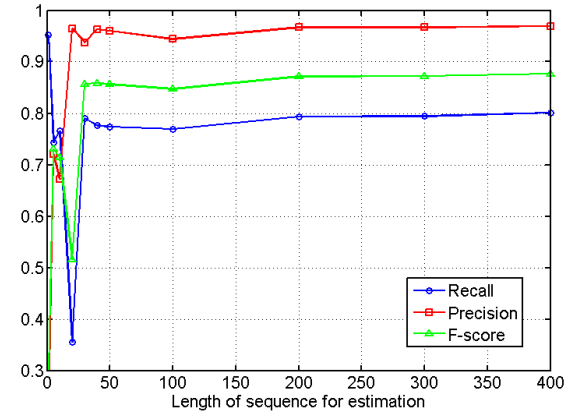

Fig. 8. The ratios of recall, precision and F-score for intention 2.

In other words, 4% of the actions are ill-associated to *Specify an association* while they probably belong to *Specify an entity*. The reason of this confusion is the existence of common actions between these two intentions.

The accuracy of retrieval for *Specify an association intention*, F-1, is measured at 0.87. In other words, when the user intention is *Specify an association*, the *VA* is able to find this intention with a very good accuracy.

Regarding the third intention, *Stop*, all the three measures stabilize at 1, as shown in figure 9, because the actions associated to *Stop* are not common to other intentions. It is then trivial to identify this intention.

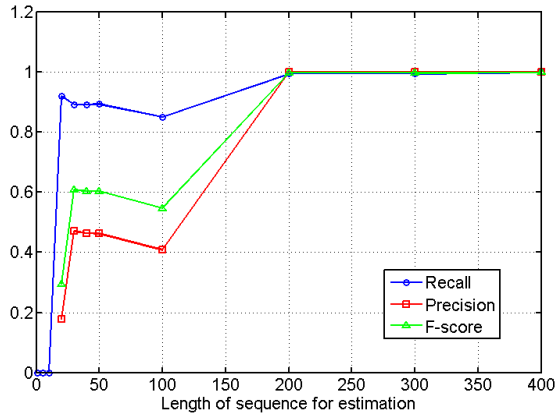

Fig. 9. The ratios of recall, precision and F-score for intention 3.

The following table shows the results obtained for the recall, precision and F-score measures for each predefined intentions.

TABLE V. RECALL, PRECISION AND F-SCORE FOR INTENTIONS

	Recall	**Precision**	**F-SCORE**
Intention 1	99,50%	97%	0.98
Intention 2	80%	96%	0.87
Intention 3	100%	100%	1

These results show that we were able to find the intentions behind the activities with satisfying accuracy. The results of F-score prove the efficiency and performance issued of from the application of a HMM applied on our dataset.

V. PERSPECTIVES AND CONCLUSION

This paper shows that HMM is an effective model to retrieve intentions from traces of activities. We used a supervised learning approach and the results shown in our first experimentation are promising. We were able to find the intentions behind the activities with satisfying accuracy, efficiency and performance (table V). Moreover, we also obtain the possibilities of transition from one intention to another (matrix $\hat{I}$) and the probabilities for apparition of activities in each intention (matrix $\hat{A}$), which is a first step to discover the intentional process model.

Another contribution of this work is the definition of a new field of research called *Intention Mining,* in line with previous works of our team on intentional process models (Nature [32],

Map [1]), process models alignment (InStAll [33]) and process guidance (Mentor [34], Crews-L'Ecritoire [35]). The main goal of *Intention mining* is to understand actors' behaviors from event logs in order to offer them a better guidance through the enactment of processes [36],[37],[38].

In further works, we will tackle the same problematic of intention discovery using unsupervised learning approach. In this other kind of approach, we do not know the segmentation of activities sequence, i.e., the links between intentions and activities sequences are unknown. We may intent to use the Baum-Welch algorithm [23] procedure, which is a variation of the more general Expectation-Maximization algorithm [12,24].

Another aspect that we plan to study is to find the apparition probability of a given activities' sequence assuming the parameters of the HMM are known. This perspective is interesting because many different intentions can lead to the same sequence of activities. Consequently, the probabilities for all the possible intentions must be added to get the probability of a given activities sequence.

Since the number of possible intentions increases exponentially with the length of the sequence, the computation of this probability can be quite complex. A solution to overcome this complexity issue is to use Forward-Backward algorithms [10] an inference algorithm for HMMs that compute, for a given observed sequence, the corresponding hidden states.

ACKNOWLEDGMENT

We would like to thank Sébastien Sion for the development of the first version of the web site and the Master Miage students for participating in the experiment.

REFERENCES

[1] C. Rolland, N. Prakash, A. Benjamen, "A Multi-Model View of Process Modelling," Requirements Engineering, Vol.4, N. 4, Springer-Verlag London Ltd , 1999, pp. 169-187.

[2] W.M.P. Van der Aalst," Process Mining: Discovery, Conformance and Enhancement of Business Processes," 1st Ed., 2011, pp. 184-352.

[3] C. Hug, R. Deneckère, C. Salinesi, "Map-TBS: Map process enactment traces and analysis," Proceedings of RCIS'12, International Conference on Research Challenges in Information Science, Valencia, Spain, 2012, pp. 204-209.

[4] B. van Dongen, W. van der Aalst, "Multi-Phase Process Mining: Building Instance Graphs", International Conference on Conceptual Modeling, LNCS 3288, pp. 362-376, Springer, 2004.

[5] W. van der Aalst, A. Medeiros, A. Weijters, "Genetic Process Mining", Applications and Theory of Petri Nets 2005, LNCS 3536, pp.48-69, Springer, 2005.

[6] J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens, "A robust F-measure for evaluating discovered process models," in Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on, 2011, pp. 148-155.

[7] G. Greco, A. Guzzo, L. Ponieri, and D. Sacca, "Discovering expressive process models by clustering log traces," IEEE Transactions on Knowledge and Data Engineering , vol. 18, no. 8, pp. 1010-1027, 2006.

[8] L. E. Baum Leonard and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," The Annals of Mathematical Statistics, vol. 37, no.6, pp. 1554–1563, 1966.

[9] G. Greco, A. Guzzo, L. Pontieri, "Mining Hierarchies of Models: From Abstract Views to Concrete Specifications", Proceedings of the 3rd International Conference on Business Process Management, BPM 2005, pp. 32-47, 2005.

[10] W.M.P van der Aalst, A.J.M.M. Weijters and L. Maruster, "Workflow Mining: Discovering Process Models from Event Logs," IEEE Trans. on Knowledge and Data Eng, Vol. 16, no. 9, pp. 1128-1142, 2004.

[11] W. Enders, "Applied econometric time series," Hoboken, N.J.: Wiley, cop. 2004.

[12] A. Rozinat, M. Veloso, and W.M.P. van der Aalst, "Evaluating the quality of discovered process models," in 2nd Intl. Workshop on the Induction of Process Models, Antwerp, Belgium, 2008, pp. 45-52.

[13] A.J.M.M. Weijters and W.M.P. van der Aalst, "Rediscovering Workflow Models from Event Based Data using Little Thumb," Integrated Computer-Aided Engineering, 2003, Vol.10 (2), pp. 151-162.

[14] J. Herbst, D. Karagiannis, "Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models", Proceedings of the 9th International Workshop on Database and Expert Systems Applications, pp.745-752, 1998.

[15] R. Agrawal, D. Gunopulos, F. Leymann, "Mining Process Models from Workflow Logs", Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology, LNCS 1377, pp.469-483, Springer, 1998.

[16] J. Cook and A. Wolf, "Discovering Models of Software Processes from Event-Based Data," ACM Transactivitys on Software Engineering and Methodology, vol.7, nom. 3, 1998, pp. 215–249.

[17] S. Minseok, C.W. Günther and W.M.P. van der Aalst, "Trace Clustering in Process Mining Business Process Management Workshops," Lecture Notes in Business Inf. Processing, 2009, Vol.17, pp. 109-120.

[18] F. Eibe and I. H.Witten, "Data Mining: Practical Machine Learning Tools and Techniques," 3rd Ed., Morgan Kaufmann Publishers Inc. San Francisco, USA, 2005.

[19] D. Ferreira, M. Zacarias, M. Malheiros and P. Ferreira, "Approaching Process Mining with Sequence Clustering: Experiments and findings, "in Alonso, G., Dadam, P., Rosemann, M. (Eds) BPM 2007. LNCS, vol. 4714, pp. 360–374. Springer, Heidelberg, 2007.

[20] R. Durbin, S. Eddy, A. Krogh and G. Mitchison,"Biological sequence analysis Probabilistic models of proteins and nucleicacids", Cambridge University Press, New York, 1998.

[21] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceed. IEEE, Vol.77, no.2, 1989, pp. 257-286.

[22] S., Assar, C., Ben Achour, S., Si-Said, "Un Modèle pour la spécification des processus d'analyse des Systèmes d'Information," In proceedings of 18ème Congrès INFORSID, pp.287-301, 2000.

[23] G.D. Forney, "The Viterbi Algorithm," Proceeding IEEE, vol.61, no.3, pp. 268-278, 1973.

[24] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," Ann. Math. Statist., Vol.41, no.1, pp. 164-171, 1970.

[25] B.H. Juang and L.R. Rabiner, "Hidden Markov Models for Speech Recognition," published in Technimetrics by American Statistical Association and American Society for Quality, vol.33, no.3, pp. 251-272, 1991.

[26] A.W. Biermann and J.A. Feldman, "On the Synthesis of Finite States Machines from Samples of Their Behavior," IEEE transactions on Computers, vol.21, no.6, pp. 592-597, 1972.

[27] J. Cook, A. Wolf, "Automating process discovery through event-data analysis", Proceedings of the 17th International Conference on Software Engineering, pp.73-82, ACM Press, 1995.

[28] T. Joachims, "Text categorization with support vector machines," Technical report, LS VIII Number 23, University of Dortmund, 1997.

[29] C. Goutte and E. Gaussier, E., "A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation," In Proceedings of the 27th European Conference on Information Retrieval, pp. 345-359.

[30] A. Enright, S. van Dongen, C. Ouzounis, "An efficient algorithm for large-scale detection of protein families", Nucleic Acids Research, Vol.30, no.7, pp.1575-1584, 2002.

[31] S. Das and M.C. Mozer, "A unified Gradient Descent/Clustering Architecture for Finite State Machine Induction," In proceeding of the

1993 Conference, no.6 in Advances in Neural Information Processing system, pp.19-26. Morgan Kaufmann, 1994.

[32] M. Jarke, C. Rolland, A. Sutcliffe, R. Domges, "The NATURE of Requirements Engineering", Shaker Verlag, Aachen, 1999.

[33] L-H., Thevenet, C. Salinesi, "Aligning IS to Organization's Strategy: The InStAll Method", Proceedings of CAiSE 2007, pp 203-217, Trondheim, Norway, 2007.

[34] G. Grosz, "MENTOR: a Step Forward in Guidance for Information System Development", Proceedings of the fifth Workshop on the Next Generation of CASE Tools, Utrecht, The Netherlands, June 1994.

[35] M., Tawbi, C., Souveyet, C., Rolland, "L'ECRITOIRE a tool to support a goal-scenario based approach to requirements engineering", Information and Software Technology journal, Martin Shepperd, Ed., Elsevier Science B.V, 1998.

[36] G. Khodabandelou, C. Hug, R. Deneckère, C. Salinesi, M. Bajec, E. kornyshova and M. Jankovic, "COTS Products to trace method enactment: review and selection", 21st European Conference on Information Systems, Utrecht, Netherland, 2013.

[37] G. Khodabandelou, "Contextual Recommendations using Intention Mining on Process Traces," 7th IEEE International Conference on Research Challenges in Information Science, Paris, France, 2013.

[38] G. Khodabandelou, C. Hug, R. Deneckère, C. Salinesi, "Process Mining versus. Intention Mining," International Conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD), Juin 2013, Valencia, Espagne.