Using Semantic Lifting for Improving Educational Process Models Discovery and Analysis

Awatef Hicheur Cairns[1], Joseph Assu Ondo[1], Billel Gueni[1], Mehdi Fhima[1], Marcel Schwarcfeld[1], Christian Joubert[1], Nasser Khelifa[2],

[1]ALTRAN Research, [2] ALTRAN Institute
Vélizy-Villacoublay, France
{awatef.hicheurcairns, joseph.assu, billel.gueni, mehdi.fhima, marcel.schwarcfeld, christian.joubert, nasser.khelifa}@altran.com

Abstract. Educational process mining is an emerging field in the educational data mining (EDM) discipline, concerned with discovering, analyzing, and improving educational processes based on information hidden in datasets and logs. These data are recorded by educational systems in different forms and at different levels of granularity. Often, process discovery and analysis techniques applied in the educational field have relied exclusively on the syntax of labels in databases. Such techniques are very sensitive to data heterogeneity, label-name variation and their frequent changes. Consequently, large educational process models are discovered without any hierarchy or structuring. In this paper we show how by linking labels in event logs to their underlying semantics, we can bring educational processes discovery to the conceptual level. In this way, more accurate and compact educational processes can be mined and analyzed at different levels of abstraction. We have tested this approach using the process mining Framework ProM 5.2.

Keywords: Semantic Process Mining, Educational Process Mining, Ontology, Semantic Matching, ProM.

1 Introduction

Nowadays, education and training centers promote personalized curriculums where students are free to choose the skills they want to develop (from beginner to specialist), the way they want to learn (theoretical or practical aspects) and the time they want to spend. This tendency is reinforced by the emergence of "e-learning" which represents an increasing proportion of the in-company trainings. Educational systems support a large volume of data, coming from multiple sources and stored in various formats and at different granularity levels [6], [16]. These data can be exploited by instructors to understand students' learning habits, the factors influencing their performance and their target skills. To answer these questions, there is an increasing research interest in using process mining in education [6],[10], [15], [16]. The idea of process mining [1] is to discover, monitor and improve real processes (i.e., not assumed processes) by extracting knowledge from event logs (recorded by an information system). However, the proposed approaches for process models extraction in the education field are somewhat limited because they rely on classical process mining techniques which are purely syntax oriented i.e. based on the labels in event logs [2]. For instance, we have encountered a massive professional training dataset of a worldwide consulting company where depending on the country and the region involved different names were used for the same training. So, the actual semantics behind the trainings' labels remain in the head of education management people (e.g. teachers, carrier advisors, etc.) who have to interpret them. To handle this question, semantic annotations on event logs could be used to prevent such interpretation efforts [2], [3]. To benefit from the actual semantics behind these labels, *semantic process mining* techniques were introduced in [2], [3], [4], leveraging mining and analysis techniques to the conceptual level. In this paper, we show how semantic process mining ideas may help to discover simplified educational process models and to extract more knowledge about their properties. For the first time, to our knowledge, a professional training dataset of a consulting company is taken as a case study to extract and analyze training paths annotated with semantic information. Also, we propose a (semi)automatic procedure used to associate semantics to training labels. The remainder of this paper is organized as follows. Section 2 summaries educational process mining techniques. Section 3 presents the semantic process mining core idea. Section 4 explains our approach to extract educational process models annotated with semantic information. Finally, section 5 concludes the paper.

2 Process Mining in the Educational Field

Process mining is a relatively new technology which emerged from the information technology and management science [1]. It focuses on the development of automated techniques to extract process-related knowledge from event logs. An *event* log corresponds to a set of process *instances* (i.e. *traces*) following a business process. Each recorded event refers to an *activity* and is related to a particular process instance. An event can have a *timestamp* and a *performer* (i.e. a person or a device executing or initiating an activity). *Educational Process Mining* (EPM) refers to the application of process mining techniques in the education domain [16]. Educational event logs may include students' registration procedures, student's examination traces or activity logs in e-learning environments. The three major types of process mining techniques are (cf. Fig. 1): *Process model discovery* takes an event log and produces a complete process model able to reproduce the behavior observed in this log. *Conformance checking* aims at monitoring deviations between observed behaviors in event logs and process models or predefined business rules and constraints. *Process model extension* aims to improve a given process model based on information (e.g., time, performance, case attributes, decision rules…etc.) extracted from an event log related to the same process. Regarding available process mining tools, the ProM Framework is the most complete and powerful one aimed at process analysis and discovery from all perspectives (process, organizational and case perspective) [8]. It is implemented as an open-source Java application with an extendable pluggable architecture.

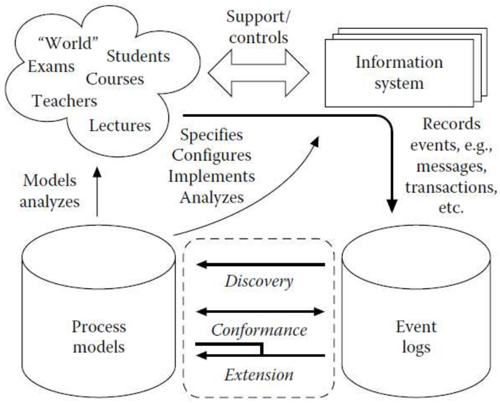

Fig 1. Process mining concepts

ProM supports a wide range of techniques for process discovery, conformance analysis and model extension, as well as many other tools like conversion, import and export plug-ins. The de facto standard for storing and exchanging events logs are the MXML (Mining eXtensible Markup Language) format or more recently the XES (eXtensible Event Stream) format. In practice, however, ProM presents certain issues of flexibility and scalability which limit its effectiveness in handling large logs from complex industrial applications [13]. We may get over these limitations by using the service oriented architecture of the ProM 6 framework. Theoretically, such architecture may allow the distribution of ProM's plugins over multiple computers (e.g., grid computing). We are recently testing such a construction in the development of an interactive and distributed platform tailored for educational process discovery and analysis. Let us note that, lately, educational process mining has emerged as a promising and active research field [6], [15], [16]. However, the application of process discovery techniques presents some challenges given the huge volume and the traces' heterogeneity often encountered in educational datasets. In fact, when analyzing event logs containing a lot of distinct traces, traditional process discovery techniques generate highly complex models (i.e. spaghetti models) [13]. In this case, the adoption of filtering, abstraction or clustering techniques may help reduce the complexity of the discovered process models [14], [17]. For instance, a clustering technique was proposed in [6] to improve both the performance and readability of the mined students' behavior models in the context of e-learning. In our previous work [10], we proposed a two-step clustering approach for partitioning training processes depending on an employability indicator. We think that *semantic process mining* techniques seem to be a promising area to explore in order to handle the issue of traces' heterogeneity and so to extract simplified process models.

3 Semantic Process Mining

The *semantic process mining* techniques, introduced in [2], [3] aim to analyze and extract process-related knowledge from event logs, at the conceptual (semantic) level

[4]. The challenges for mining and monitoring processes from a semantics perspective have been studied in the context of the European project SUPER [9]. The concept of semantic log purging was proposed in [12], taking a case study in the higher education domain. In [5], the authors proposed a combination of standard process mining techniques with semantic lifting procedures on the event logs in order to mine more precise process models. The core idea of semantic process mining is to explicitly annotate elements in event logs with the *concepts* that they represent. These concepts are formalized in generic or domain specific ontologies. Hence, semantic process mining techniques are built on the following three basic elements: *ontologies*, *ontology reasoners*, and *references* from elements in logs/models to concepts in ontologies [2]. First, ontologies define and formalize a set of concepts shared by (a group of) people to refer to things in the world and the relationships among these concepts. Second, the reasoner provides reasoning over the ontologies in order to derive new knowledge, e.g., subsumption, equivalence, etc. Finally, the references associate meanings to labels (i.e., strings) in event logs and/or models by pointing to concepts defined in ontologies. The discovery, conformance checking, and extension techniques rely on subsumption relations induced by these ontologies to raise the level of abstraction from the syntactical level to the semantical level. Thus, these techniques can be applied without requiring any modification of models or logs if the elements in different logs and models link to the same concepts (or super/sub concepts of these concepts). Let us note that all semantic plug-ins developed in ProM are based on the following concrete formats for the basic building blocks: Event logs are in the SA-MXML (i.e. Semantic Annotated Mining eXtensible Markup Language) file format. SA-MXML is a semantically annotated version of the MXML format which incorporates the model references (between elements in logs and concepts in ontologies). Ontologies are defined in WSML (Web Service Modeling Language) [7], [11]. The WSML 2 Reasoner Framework [18] is used to perform all the necessary reasoning over the ontologies.

4 Case Study: Leveraging Educational Process Mining Techniques at the Semantic Level

Our motivating example is based on real-world training databases from a worldwide consulting company. This company has around 6 000 employees that are free, during their careers, to take different trainings aligned with their profiles. These trainings are provided by internal or external organizations. The data collected for analysis includes the employees' profiles (demographics data), their careers (i.e. the jobs/missions they did) and their training paths (the set of trainings taken during the past three years) (cf. Table 1). In what follows, we apply a process model discovery algorithm (e.g. the heuristic miner [8]) on a fragment of the training event log (cf. table 1), containing 1000 traces, 2419 events and 280 originators. We can see that the obtained result is an unreadable spaghetti like process model (cf. Fig. 2). This result can be explained by the heterogeneity in employees' training paths and the great number of different trainings' labels. Let us note that depending on the organization, the country and the region involved, different labels (i.e. string) were used for the same training. Moreover, some training courses can be seen as special cases of other trainings. For instance, the trainings "*Collective English*", "*Collective Face to Face*

English", "*English In Group*" are in fact the same training which is given different names following data sources. Moreover "*Collective Face to Face English"* is a variant of "*Face to Face English"*, which is a special type of the "*English"* training.

Table 1. Example of an educational event log

Matricul	Profil	Training_Id	Training_Label	Training_Orga_Id	StartDate	EndDate
7	CONSULTANT	Tr 850	EXCEL ELEARNING	Org 13	11/07/2011	31/12/2011
8	CONSULTANT	Tr 769	QF TEST	Org 135	26/04/2011	28/04/2011
9	CONSULTANT	Tr 252	INTERCULTURAL WORKING RELATONS : INDIA	Org 135	01/07/2011	01/07/2011
10	CONSULTANT	Tr 260	SELENIUM	Org 135	25/10/2011	26/10/2011
11	CONSULTANT	Tr 812	UML FUNCTIONAL ANALYSIS	Org 135	24/10/2011	27/10/2011
12	CONSULTANT	Tr 774	DESIGN PATTERNS AND APPLICATION C++	Org 135	08/12/2011	09/12/2011
13	CONSULTANT	Tr 1923	SQL BASIC	Org 135	03/04/2012	05/04/2012
14	CONSULTANT	Tr 813	C++ ADVANCED	Org 135	04/04/2012	06/04/2012
15	CONSULTANT	Tr 2014	XML BASIC AND XPATH	Org 135	10/04/2012	11/04/2012
14	CONSULTANT	Tr 1282	DESIGN PATTERNS AND APPLICATION IN C++	Org 135	13/09/2012	14/09/2012
...	....		...		......	

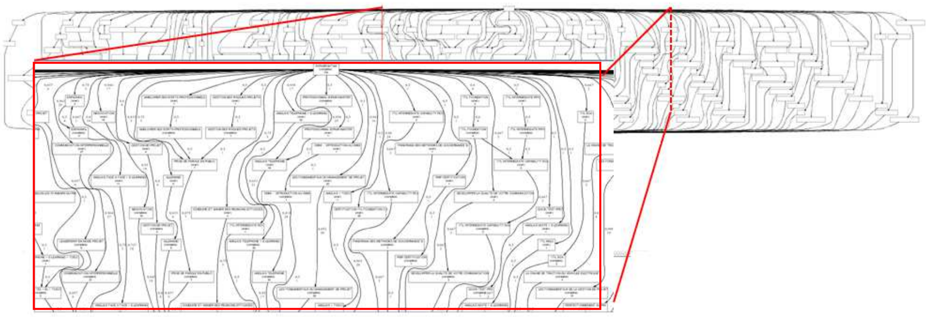

Fig 2. Fragment of a spaghetti process describing all trainings followed by the consulting company's employees during the last three years. The process model was extracted using the Heuristic Miner plug-in of ProM.

To handle this issue, we need to link different trainings which are variants or synonyms of the same training to a unique concept in a *training ontology*. Usually, there are two ways to achieve this. We can manually create all the necessary ontologies and annotate the necessary elements in educational event logs with ontologies' concepts. It is also possible to use tools to (semi)automatically discover ontologies based on the elements in these logs [4]. The discovered ontologies can be manually improved in a second step. Let us note that semantic process mining tools can also play a role in ontologies' extraction and enhancement from event logs. The ontology depicted in Fig. 3 is used to formalize the concepts for trainings in our example. It contains 42 concepts and 129 instances. We built this ontology manually taking as starting point the semantic description of trainings provided in training organizations' catalogues. We distinguished five super-concepts related to the training domain: *Communication*, *Staff Management*, *Project Management*, *Audit and Control*, *Information Technologies*.

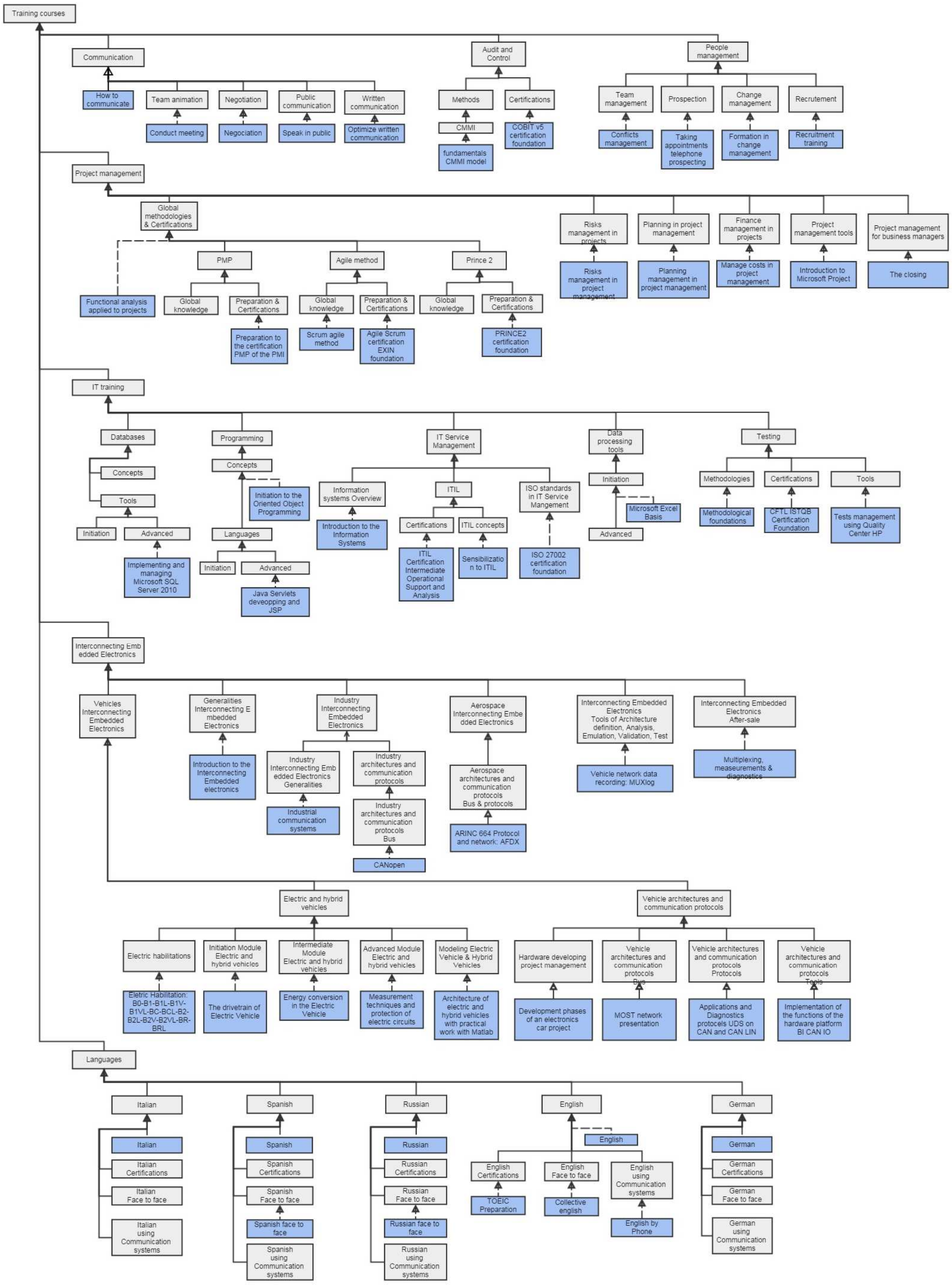

Fig 3. Fragment of the “Training ontology”: only some instances (i.e. training labels) are represented

These concepts are subdivided into sub-concepts which are in their turn subdivided into lower sub-concepts (cf. Fig 3). Trainings' labels are the instances of this ontology and each label is associated to one concept or sub-concept. To simplify the ontology depicted in Fig 3, we only represented one instance (training label) per concept. We used the tool WSMT (Web Service Modeling Toolkit) to implement the training ontology in the WSML format since it is supported by the ProM 5.2 framework. Moreover, the semantic process mining plug-ins existing in ProM 5.2 expect log elements to be connected with process ontologies (i.e., to be in the SA-MXML logging format). So to enrich the educational log of our example with semantic annotations from the *Training Ontology*, we implement a conversion plug-in in ProM 5.2. The latter takes as input the original *educational log* (in MXML format) and the *Training Ontology* (in WSML format) and produces the corresponding semantically annotated event log (in SA-MXML format).

4.1 Semantic Matching Between Training Labels and Concepts

In order to help end users in the comprehension of the underlying semantics of training courses, we develop a (semi)automatic procedure, which can be used to associate a concept (of the training ontology) to a training label. The association used is based on the importance of the words in a label or in a concept. We assume that each word of a label L plays the same semantic role and hence has the same importance as well as the other words constituting L. We also suppose that at least one of the words characterizing a concept, or one of their synonyms, appears in all the labels associated to it. Therefore, there is an intersection between the set of the words of a label and the set of the words characterizing its associated concept. To build our technique we develop the following modelling: consider $W = \{w_1,\ldots,w_n\}$ a set of words, we consider a training label TL_i as succession of w_j, noted $TL_i= w_1$ƀ+… ƀ+ w_m, where $w_j \in W$ and the symbol ƀ represents blanks and all articles, pronouns, etc. For instance the label "*Introduction to Information Systems*" contains the set of words W = {*Introduction*, *Information*, *Systems*, *Management*} separated with three blanks and the preposition 'to'. We consider L_i the set of the words that contains TL_i, so $L_i = \{w_1,\ldots,w_m\}$ and in our case we assume that $card(L_i)$ represents the length of the label TL_i (we note $Len(TL_i)$), for example *Len*(*Introduction to Information Systems*) = 3. We also consider $C_j = \{w'_1,\ldots,w'_k\}$ as the set of the words characterizing a concept C_j.

Word importance: is a metric, or a weight, reflecting the importance of a word in a label according to our hypothesis given below. As each word plays the same role in a label we compute its importance *wp* as follow: $wp(w) = 1/ Len(TL)$ where $w \in L$. for the label TL = "*Management in Information Systems*", $Len(TL)=3$ and $wp(Management)=1/3$. This *wp* reflects clearly the relation between the length of a label and the importance of its word. A small label, like ones using only one or two words, gives a great semantic importance to its word that are considered like keys, whereas long labels use lot of words for their description giving its words a small semantic role.

Word concept weight: the weight of a word w in a concept C, noted $cw(w)$, corresponds to the sum of all word importance of w, or one of its synonyms, in all the labels associated to the concept C: $cw(w)= \sum wp_{TLi}(x)$, *where* $i \in \{1,\ldots,h\}$ *and* TL_i is associated to C. For instance, consider the concept characterized by the following words ("*management*", "*project*"). If "management" appears three times in the labels with the following *wp*: ½, ½ and ⅓ therefore *cw("management")= ½+½+⅓ =1.3*. This metric establishes a monotone relation between the frequency of the word in the labels and its importance, and it is clear that more a word is used, more it is important and more it will be used to characterize a concept.

Concept matching: to generate automatically the concept C associated to a label TL we create first a word weight table as follow:

1. We compute the set of all the words of all the labels contained in the training catalogue. We note this set as LW.
2. We create a matrix $M = (a_{i,j\ 1\leq i\leq n,\ 1\leq i\leq m})$ $a_{i,j}$ is the $wp(i)$ in the label j, $n = card(LW)$ and m is the number of all training Labels.
3. For each word w in LW, we sum its $wp(w)$ computed in the previous step and we store the result in the returned table.

After constructing this table, for a label TL we compute the semantic intersection between L and C as follow: $L \cap C = \{w_j, w_j \in L \wedge w_j \in! C\}$. $w_j \in! C$ means that w_j or a synonym of it is included in C. Then we compute the score of matching between L and C, noted $SC(L,C)$ as the sum of the *concept weight* of each element of $L \cap C$. We repeat this operation for all the concepts we have and then we associate L with the concept having the high score. If we have the concepts $C_1,\ldots,Cn$ then L will be associated to C if $SC(L,C) = Max(SC(L, C_n))$. The semantic importance we use in our matching is simplified compared with approaches doing deep semantic analysis using sophisticated techniques because we do a significant human effort to define the Ontology with different level, and we stress on the concepts of the level 2 to enrich them with words that are generally and mostly used to define the labels associated to each concept of this level. We remark that if we have two or more concepts having the same $Max(SC(L, C_n))$ we infer a conflict and in this case we need a user's intervention to choose what concept to associate to the label. We have tested this matching technique on Altran catalogue containing 128 labels and 35 concepts. Fig 4 depicts the obtained results. Let us note that in these tests we have identified some cases where we have not identified matching between labels and concepts.

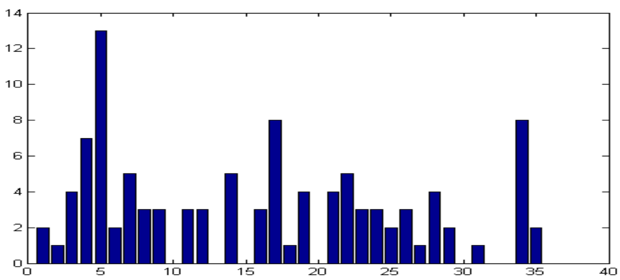

Fig 4. The number of labels (ordinate) associated to each one of the 35 concepts (absciss) of our case study

This is due to the use of some abbreviations that are hard to decrypt. In these tests, concepts contain only words that we find in labels and we do not need in this case to search synonyms. We plan in the future to use a dictionary in order to enhance the identification of synonyms.

4.2 Educational Process Models Mining at the Conceptual Level

After constructing a semantically annotated educational log, we specify the level of abstraction (i.e. concepts in the training ontology) used as a base for the mining and the analysis of training processes. To achieve this, we use the *filter* plug-in "*Ontology Abstract Filter*" implemented in Prom 5.2, which allows us to choose the required level of abstraction [8]. The *Ontology Abstract Filter* plug-in takes as input a semantically annotated event log (in SA-MXML format) and produces as output another event log where the names of tasks (i.e. trainings) are replaced by the names of the chosen concepts. The produced log can also be exported as an SA-MXML log. After this step, we may apply a control-flow mining algorithm (e.g. the *Heuristic Miner* plug-in) to extract the educational process model relaying on the concepts chosen in the previous step. We may choose concepts at different level of abstractions. When we use only the concepts at level 2 of the *Training Ontology* tree (i.e., the concepts "Communication", "Language", "Testing", "Audit_And_Control", "IT_Service_Management"…etc."), a process model like the one in Fig. 5 could be discovered.

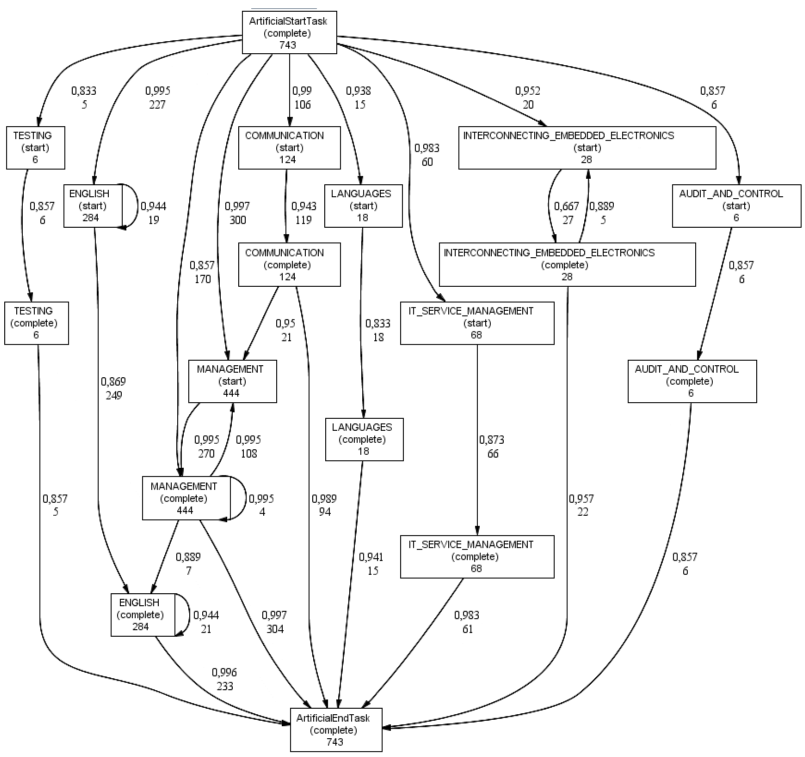

Fig 5. Training process model mined using the heuristic miner plug-in where only the concepts at the level 2 of the tree for the ontology "TrainingOntology" (cf. Fig.3) are considered.

It contains 18 events (nodes) and 30 arcs which is more compact than the model extracted before the semantic abstraction (cf. Fig 2). Let us note that during the abstraction phase we deliberately replace the labels of the different kind of English trainings by their concept at level 1 (i.e. *English*). We can see that the mined model in this case is more compact (i.e., has a higher abstraction level) than the one in Fig 2. In this model we can see that trainings associated to the concept "*Management*" are taken 444 times. Also, there are seven trainees who took an "*English*" training after a "*Management*" training. The frequency associated to this relation in the educational log is 0,889.

4.3 Educational Process Analysis at the Conceptual Level

In our case study, process mining advantages are not limited to the discovery of employees' training processes. In fact, training advisors and directors of training organizations often need to check (off-line or on-line) whether trainees' paths conform to established career paths, trainings' prerequisites or business rules. The semantic LTL checker plug-in of ProM 5.2 is the perfect tool for auditing educational processes at the conceptual level [2]. This tool can be used to verify the same formula (e.g. generic formula such as prerequisite) on a set of different event logs as long as the arguments of this formula and the elements in these logs link the same concepts (or super/sub concepts of these concepts). There is a set of predefined formulas in the semantic LTL model checker plug-in. It is also possible to tailor the semantic LTL checker plug-in to express specific types of constraints encountered in the educational domain [16]. All these properties can be easily coded using the LTL language and imported into the user interface of the plug-in. In what follows we want to check if the rule "A *Project Management* training must be taken before a *Project Management Professional Certification (PMP)* can be taken" was always respected (prerequisite check). We define this property in a LTL file as follows:

```
formula c2_is_a_prerequisite_of_c1 ( c1 : ate.WorkflowModelElement, c2 :
ate.WorkflowModelElement) :=
{   <h2> Is the training C2 a prerequisite for the training C1? </h2>}
   ( <> (activity == c2) /\ (activity != c2 _U activity == c1) ) ;
```

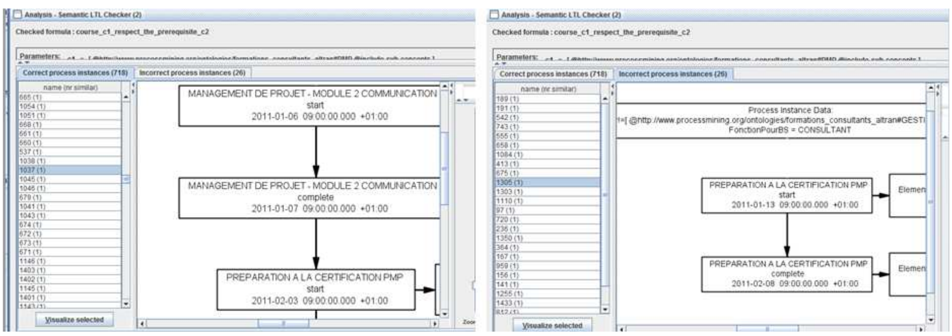

Fig 6. The results returned by the semantic LTL Checker plug-in while verifying the PMP prerequisite

Fig 6 shows the result displayed when this property is checked. We can see that there are 26 trainees who took the *PMP* training while they didn't take the *Project Management* training before (i.e., incorrect case instances). There are also 718 trainees that satisfy this property (i.e. they took the "*PMP*" training after a "*Project Management*" training).

5 Conclusion

In this paper we showed how by associating semantic annotations to educational event logs, more accurate and compact educational processes can be extracted and analyzed at different levels of abstraction. Also we developed a semantic matching procedure allowing to link training labels to the right concepts of a training ontology, in a (semi)automatic way. In future works, we will investigate how concepts from ontologies can be associated to training providers. We can then benefit from these semantic annotations in mining social networks and organizational models between training providers [1], [10], at the conceptual level. We plan also to conduct a case study in an on-line education setting that would illustrate the benefit of process mining approaches, at the syntactic and semantic levels, to mine and understand students' behaviors. Another important step in our works is to develop new clustering and classification techniques which take into account semantic annotations on event logs [14], [17]. For instance, trace clustering techniques [14] can be extended to partition event logs depending on trace similarities at the conceptual level. To implement our approach, we are currently developing an interactive and distributed platform tailored for educational process discovery and analysis. This platform will allow different education centers and institutions to load their data and access advanced data mining and process mining services [10]. Moreover, in order to optimize and enhance platform response time, our platform will allow distributing heavy analysis computations on many processing nodes.

Acknowledgments. This ongoing work is being carried out by Altran Research and Altran Institute within the context of the PHIDIAS project.

References

1. Aalst, W. M. P., and al.: Process Mining Manifesto, In BPM 2011 Workshops Part I, LNBIP 99, pp. 169-194 (2012).
2. Alves de Medeiros, A.K., van der Aalst, W.M.P., Pedrinaci, C: Semantic Process Mining Tools: Core Building Blocks. In 16th European Conference on Information Systems, pp. 1953-1964. Galway, Ireland (2008).
3. Alves de Medeiros, A. K., . van der Aalst, W. M. P: Process Mining towards Semantics. Advances in Web Semantics I, LNCS 4891, pp 35-80 (2009).
4. Alves de Medeiros, A. K. and al: An Outlook on Semantic Business Process Mining and Monitoring. In R. Meersman, Z. Tari & P. Herrero (Eds.), The Confederated

International Conferences On the Move to Meaningful Internet Systems, LNCS, vol. 4806, pp. 1244-1255, Springer-Verlag (2007).

5. Azzini, A., Braghin, C., Damiani, E., Zavatarelli, F: Using Semantic Lifting for improving Process Mining: a Data Loss Prevention System case study. In the 3rd International Symposium on Data-driven Process Discovery and Analysis, CEUR-WS.org, pp. 62-73 (2013).
6. Bogarín, A., Romero, C., Cerezo, R., Sánchez-Santillán, M: Clustering for improving educational process mining. In Proceedings of the Fourth International Conference on Learning Analytics And Knowledge. ACM, New York, NY, USA, pp. 11-15 (2014).
7. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language WSML: An Overview. In Y. Sure and J. Domingue, editors, ESWC, LNCS, vol. 4011, pp. 590-604. Springer (2006).
8. van Dongen, B. F., de Medeiros, A. K. A., Verbeek, H. M. W., Weijters, A. van der Aalst, W. M. P: The Prom Framework: a New Era in Process Mining Tool Support. In ICATPN'05, Gianfranco Ciardo and Philippe Darondeau (Eds.). LNCS, vol. 3536, pp. 444-454, Springer-Verlag, Heidelberg (2005).
9. European Project SUPER - Semantics Utilised for Process Management withing and between Enterprises. http://www.ip-super.org/
10. Hicheur Cairns A. and al: Custom-Designed Professional Training Contents and Curriculums through Educational Process Mining. In The Fourth International Conference on Advances in Information Mining and Management, pp 53-58 (2014).
11. Lausen, H., de Bruijn, J., Polleres, A., Fensel, D: The WSML Rule Languages for the Semantic Web. W3C Workshop on Rule Languages for Interoperability, W3C (2005).
12. Ly, L.T, Indiono, C., Mangler, J., Rinderle-Ma, S.: Data transformation and semantic log purging for process mining. In Proceedings of the 24th international conference on Advanced Information Systems Engineering, J. Ralyté and al. (Eds.).LNCS, vol. 7328, Springer-Verlag (2012)
13. Reichert, M.: Visualizing Large Business Process Models: Challenges, Techniques, Applications. In 1st Int'l Workshop on Theory and Applications of Process Visualization Presented at the BPM 2012, LNBIP, vol. 132, pp. 725-736, Springer-Verlag (2013).
14. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, Springer, Heidelberg, pp. 109–120 (2009).
15. Trcka, N., Pechenizkiy, M: From Local Patterns to Global Models: Towards Domain Driven Educational Process Mining. In Proc. 9th International Conference on In telligent Systems Design and Applications, pp. 1114-1119. IEEE Computer Society (2009).
16. Trčka, N., Pechenizkiy, M., van der Aalst, W.: Process Mining from Educational Data (Chapter 9). In Handbook of Educational Data Mining. pp. 123-142. CRC Press (2010).
17. Veiga, G.M., Ferreira, D.R.: Understanding Spaghetti Models with Sequence Clustering for ProM. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 92–103, Springer, Heidelberg (2010).
18. WSML 2 Reasoner Framework (WSML2Reasoner). http://tools.deri.org/