Process Mining in the Education Domain

Awatef HICHEUR CAIRNS[1], Billel GUENI[1], Mehdi FHIMA[1], Andrew CAIRNS[1] and Stéphane DAVID[1]
Nasser KHELIFA[2]

[1] ALTRAN Research, [2] ALTRAN Institute
2 rue Paul Dautier
Vélizy-Villacoublay, 78140 FRANCE
[awatef.hichaurcairns, billel.gueni, mehdi.fhima, andrew.cairns, stephane.david, nesser.khelifa]@altran.com

Abstract— **Given the ever changing needs of the job markets, education and training centers are increasingly held accountable for student success. Therefore, education and training centers have to focus on ways to streamline their offers and educational processes in order to achieve the highest level of quality in curriculum contents and managerial decisions. Educational process mining is an emerging field in the educational data mining (EDM) discipline, concerned with developing methods to better understand students' learning habits and the factors influencing their performance. It aims, particularly, at discovering, analyzing, and providing a visual representation of complete educational processes. In this paper, in continuity of the work presented in [1], we investigate further the potential, challenges and feasibility of the educational process mining in the field of professional trainings. First, we focus on the mining and the analysis of social networks, from educational event logs, between courses units, resources or training providers. Second, we propose a clustering approach to decompose educational processes following key performance indicators. We have experimented this approach using the ProM Framework.**

Keywords-component; process mining; educational data mining; curriculum mining; key performance indicator; ProM.

I. INTRODUCTION

Recently, education and training centers have started introducing more agility into their teaching curriculum in order to meet the fast-changing needs of the job market and meet the time-to-skill requirements. Modern curriculums are no longer monolithic processes. Students can pick the courses from different specialties, may choose the order, the skills they want to develop, the level (from beginner to specialist), the way they want to learn (theoretical or practical aspects) and the time they want to spend. This need for personalized curriculum has increased with the emergence of collaborative tools and on-line training which often supplement and sometimes replace traditional face-to-face courses [2]. In fact, e-learning represents an increasing proportion of the in-company training, while addressing ever wider populations. The broad number of courses available and the flexibility allowed in curriculum paths could create, as a side effect, confusion and misguidance. Students may be overwhelmed by the offer and blurred on the time required to enter and remain in the job market. Moreover, teachers and educators may lose control of the education process, its end-results and feed-back [2]. In this modern education context, where students can access courses and curriculums on-line, from all around the world, the education systems enter a competitive market they are not used to, where they are increasingly held accountable for students' success. This situation creates additional pressure in higher educational institutions and training centers to achieve the highest level of quality in curriculum content and managerial decisions.

The use of information and communication technologies in the educational domain generates large amount of data, which may contain insightful information about students' profiles, the processes they went through and their examination grades. The deriving data can be explored and exploited by the stakeholders (teachers, instructors, etc.) to understand students' learning habits, the factors influencing their performance and the skills they acquired [3-4]. Rather than relying on periodic performance tests and satisfaction surveys, exploiting historical educational data with appropriate mining techniques enables in-depth analysis of students' behaviors and motivations. To answer these questions, there are increasing research interests in using data mining in education [3-4]. *Educational Data Mining* (EDM) is a discipline aimed at developing specific methods to explore educational datasets generated by any type of information system supporting learning or education (in schools, colleges, universities, or professional training institutions providing traditional and/or modern methods of teaching, as well as informal learning). EDM brings together researchers and practitioners from computer science, education, psychology, psychometrics, and statistics. EDM methods can be classified into two categories – (1) Statistics and visualization (e.g., Distillation of data for human judgment*)*, and (2) Web mining (e.g., Clustering, Classification, Outliers detection, Association rule mining, Sequential pattern mining and Text mining) [5]. However, most of the traditional data mining techniques focus on data or simple sequential structures rather than on full-fledged process models with concurrency patterns [6-7]. For instance, it is not clear how, given a study curriculum, EDM techniques could check automatically whether the students always follow it [6]. Precisely, the basic idea of *process mining* [8] is to discover, monitor and improve real processes (i.e., not assumed nor truncated processes) by extracting knowledge from event logs recorded automatically by Information Systems. Our research aims to develop generic methods which could be applied to general education issues and more specific ones concerning professional training or e-learning fields for [1], [9]:

- The extraction of process-related knowledge from large education event logs, such as: process models and social networks following key performance indicators or a set of curriculum pattern templates.

- The analysis of educational processes and their conformance with established curriculum constraints, educators' hypothesis and prerequisites.
- The enhancement of educational process models with performance indicators: execution time, bottlenecks, decision point, etc.
- The personalization of educational processes via the recommendation of the best course units or learning paths to students (depending on their profiles, their preferences or their target skills) and the on-line detection of prerequisites' violations.

In this paper, we focus mainly on (1) process model discovery, deriving from Key Performance Indicators; (2) social network discovery between training courses and training providers. For the first time, to our knowledge, the present approach handles a professional training dataset of a consulting company involved in the training of professionals. In this paper, we extend the work presented in [1]. The rest of this paper is organized as follows. Section II introduces process mining techniques and their application in the educational domain. Section III presents our approach for social networks mining and process models discovery. Section IV describes briefly the PHIDIAS platform. Section V discusses some related works. Finally, Section VI concludes the paper.

II. Educational Process Mining

A. Definition

Process mining is a relatively new technology which emerged from business community [8]. It focuses on the development of a set of intelligent tools and techniques aimed at extracting process-related knowledge from event logs. The complete overview of process mining application in the educational field (known as *educational process mining* [6-7]) is illustrated in Fig. 1.

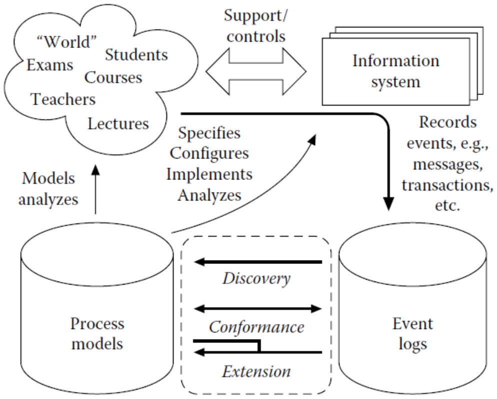


Figure 1. Process mining concepts

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). Typical examples of event logs in education may include students' registration procedures and attended courses, student's examination traces, use of pedagogical resources and activity logs in e-learning environments. The three major types of process mining techniques are:

Process model discovery: takes an event log and produces a complete process model able to reproduce the behaviour observed in this log. Examples of such techniques are control-flow mining algorithms, which allow the discovery of educational processes and learning paths based on the dependency relations that can be inferred from event logs, among student's actions or courses taken. This step is not limited to control-flow discovery. For instance, there are approaches to discover social networks, organizational structures and resource behavior from event logs. Typically, these approaches use the information about the performer (resource), e.g., the person or component initiating or completing some activity, to generate the relationships in social networks between these resources, following their involvement in the process execution [10].

Conformance checking: aims at monitoring deviations between observed behaviours in event logs and normative process models (generated either by traditional modelling or by process mining techniques). This evaluation can be made using metrics such as *fitness* (Is the observed behavior possible according to the model?) and a*ppropriateness* (Is the model *typical* for the observed behavior?). Major deviations from a normative model might also mean that the model itself does not reflect real world circumstances and requirements. *Compliance checking* is another kind of event log inspection, which aims at measuring the adherence of event logs with predefined business rules, constraints, temporal formulas or Quality of Service (QoS) definitions. An example of such a technique for auditing event logs is the LTL Checker which is used to verify properties (i.e., rules, constraints, etc.), expressed in terms of Linear Temporal Logic (LTL).

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. There are different ways to extend a given process model with these additional perspectives, e.g., decision mining, performance analysis, and user profiling.

Process mining techniques can also be used for operational decision support activities. For instance, based on historic information, it is possible to make predictions (e.g., the remaining flow time) for running cases or to recommend suitable actions (e.g., proposing the activity that will minimize the expected costs and time). Moreover, it is possible to check, on the fly, if running cases fit with normative process models or if desired properties (defined in Linear Temporal Logic) hold in these running cases.

B. *The PROM Framewok*

Regarding available process mining tools, the ProM Framework [11] is the most complete and powerful one aimed at process analysis and discovery from all perspectives (process, organizational and case perspective). It is implemented as an open-source Java application with an extendable pluggable architecture, which enables users to write and import their own mining algorithms as plug-ins. These plug-ins can also be chained into `macro' plugins. 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 is the MXML (Mining eXtensible Markup Language) format or more recently the XES (eXtensible Event Stream) format, which is the successor of MXML. These two standards are adopted in the ProM Framework. In practice, however, ProM presents certain issues of flexibility and scalability, which limit its effectiveness in handling large logs from complex industrial applications [12]. 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.

C. *Process Mining Issues in the Education Domain*

Educational systems support a large volume of data, coming from multiple sources and stored in various formats and at different granularity levels [3-4]. The data come from face to face educational systems, such as traditional classrooms, or from distance education taken from interactive learning environments or computer-supported collaborative learning. For instance, face to face educational systems store only administrative and demographic information; i.e., students' profiles (e.g., grades and curriculum goals), who follows which program, takes which courses and exams. Computer and on-line education systems store more fine-grained data because they can record all the information about students' actions and interactions into log files and databases. This data includes resource usage logs (e.g., handouts, video recordings), assessment data, collaborative writing in wikis or versioning systems, and participation in forums [2]. Moreover, the cost-effectiveness quest of modern education systems leads them to record more information about (1) the learners' short-term satisfaction on programs, course units or resources, and (2) the long-term usefulness of the courses they have taken in entering and remaining in employment. Recorded information in educational systems are structured (logs, student registration information, student usage profiles, administrative information, etc.) or unstructured (interaction with teachers via chat, collaboration with other students via chat, etc.).

To discover a suitable process model, it is assumed that the event log contains representative sample of behavior. However, the application of process discovery techniques presents some challenges given the huge volume and the traces' heterogeneity often encountered in educational datasets.

Voluminous Data - Large Number of Cases or Events in event logs

Event logs in the education domain, particularly those coming from e-learning environments, may contain massive amounts of fine granular events and process related data. In fact, real-life experiences show that most of the contemporary process mining techniques/tools are unable to handle massive event logs [12], [14]. There is a need for research in both the algorithmic as well as deployment aspects of process mining. For example, we should move towards developing efficient, scalable, and distributed algorithms in process mining [12]. This issue was tackled in recent researches [14], [15], where clustering techniques were proposed for partitioning large logs into smaller parts that can be checked locally and more easily.

Heterogeneity and complexity: Large Number of Distinct Traces and Activities in event logs

Indeed, educational processes are unstructured and flexible by nature, with a lot of heterogeneous and distinct traces, reflecting the high diversity of behaviors in students' learning paths. Consequently, existing process mining techniques generate highly complex models (called spaghetti models) that are often very confusing and difficult to understand. Moreover, conformance and compliance checking may be complicated with heterogeneous and large scale event logs. One reason for such a result can be attributed to constructing process models from raw traces without due pre-processing. The adoption of filtering, abstraction or clustering techniques may help reducing the complexity of the discovered process models [12], [14], and hence their verification using conformance checking. The issue is to adopt a combination of simplification techniques which reduce the complexity of event logs without losing pertinent information allowing us to discover key concepts and process patterns from these logs.

Concept drift

Usually, when processes are mined and reconstructed from event logs, classic process discover techniques assumed that these processes were stable over the time of observation. But this might not be the case in educational processes. It is not uncommon for the subjacent curriculum to evolve over time and go through major changes from time to time. In fact, courses and study curriculums may be created, modified (e.g., identifier, name, content or structure) or deleted at any time during learning paths of students. Concept drift refers to

TABLE I. EXAMPLE OF AN EDUCATIONAL EVENT LOG STYLES

Matricule	Profil	Training_id	Training label	Training orga_id	Start_date	End_date
7	consultant	tr850	Excel e-learning	Org 13	11/07/11	31/12/11
13	consultant	Tr1923	C++ advanced	Org 135	03/04/12	05/04/12
14	consultant	tr813	Xml basics and XPath	Org 135	04/04/12	06/04/12
…	…	…	…	…	…	…

the situation in which the process changes while being analyzed [16]. An approach to deal with concept drift, in the context of process mining, was introduced in [16].

Interpretation of results by the end users

The models obtained by process mining discovery algorithms have to be comprehensible and useful for the end users' decision-making. For this purpose, visualization techniques and notation simplification are very useful for showing results in a way that is easier to interpret. For instance, instead of showing the whole obtained process model, as directly displayed by process mining algorithms, it is better to abstract its representation using suitable notations (e.g., usual academic notation) understandable by end users or in the form of a list of suggestions, recommendations and conclusions about the obtained results. Also, the interactive interfaces presented to the end-users have to be such as to facilitate the selection of the specific mining method to use with appropriate values for the key parameters to obtain good results/models. Moreover, the trustworthiness of the results should always be clearly indicated [14].

III. ANALYZING EDUCATIONAL PROCESSES USING PROCESS MINING TECHNIQUES

A. Motivating example

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 training courses aligned with their profiles. These trainings are provided by internal or external organizations. The data collected for analysis includes the employees' profiles (identifier, function, and number of years of service), their careers (i.e., the jobs/missions they did) and their training paths (the set of training courses taken during the past three years) (see Table I). Training mangers aim to gain more insight in employees' training paths and motivation so they can offer more personalized training courses, according to the job market needs.

In this section, we show how process mining techniques can be used to analyze the training processes underlying this dataset.

B. Preprocessing phase

Data pre-processing allows the transformation of original data into a suitable shape to be used by process mining algorithms. In our case study, the data being collected for analysis is stored in various databases. So, as a first step, we construct a consolidated log (stored as a CSV file) extracted from these databases using an ETL (i.e., *Extract*, *Transform and Load*) tool, gathering all employees' training courses and work experiences over the last three years. Given that we use the ProM 6.3 framework in process discovery and analyses, we transform this log into the MXML (Mining eXtensible Markup Language) format by using the ProM Import plug-in. Let us note that in our case, during this transformation, we stipule that an employee identifier corresponds to a process instance identifier (an employee training path is understood as a process instance). To obtain a less complex event log, we can use the variety of log filter plug-ins existing in the ProM framework [11]. For instance, the *Event log filter* plug-in enables the selection of only the desired activities in an event log. The *Log filter using simple heuristic* enables a user to select the most frequent activities appearing in an event log.

C. Dotted Chart Analysis

As a first step in our study of the training courses' dataset, we use the dotted chart plug-in of ProM to gain some insight in the underlying process and its performance. The dotted chart shows the spread of events over time by plotting a dot for each event in an event log which enables visually examining an event log and so highlighting some interesting patterns present in it [17]. The dotted chart has two orthogonal dimensions: time and component types. The time is measured along the horizontal axis of the chart. The component types (e.g., instance, originator, task, event type, etc.) are shown along the vertical axis. The dotted chart analysis plug-in of ProM is fully configurable. Based on the chosen component type, the events are rearranged. Fig. 2 illustrates the output of the dotted chart analysis of the training courses' dataset example using process instances as component type. In this chart, every row corresponds to a particular case of the training process, i.e., all the training courses followed by one employee during the last three years. Each training course is represented by two dots of the same color (one per starting date and ending date)

All the instances (one per trainee) are sorted by the first events of trainings, i.e., trainings are sorted by the first date of their occurrence. We can clearly see from Fig. 2 that each year, there are few trainings scheduling around the last three months (see inside the black circle). Also, almost no training course is scheduled during the summer (see inside the red circles).

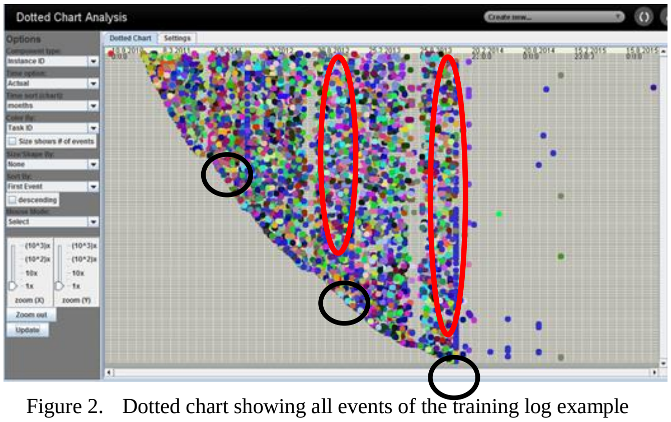

Component type: Instance ID
Time option: Actual
Time sort (chart): months
Color By: Task ID
☐ Size shows # of events
Size/Shape By: None
Sort By: First Event
☐ descending
Mouse Mode: Select

Figure 2. Dotted chart showing all events of the training log example

As a second step, we apply a process model discovery algorithm on only a fragment of our dataset example containing employees' training courses over one year, to get a big picture about the nature of professional training processes. The process model was constructed (using the Heuristic Miner plug-in of ProM [11]) based on an event log containing 8884 events, 2272 training courses performed by 404 different training providers. We can see that the obtained result is an unreadable spaghetti like process model (see Fig. 3).

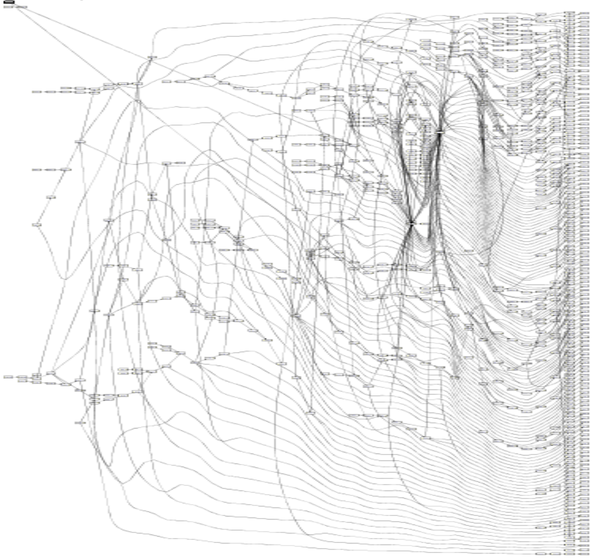


Figure 3. Fragment of a spaghetti process describing all training courses followed by 2980 employees during one year.

D. Social Network Mining

In this section we show the use of the social miner algorithms [10] implemented in the ProM 6.3 framework [11], to examine and assess interactions between training providers and between training courses following their involvement in students' training paths.

According to [18], a social network is defined as a network of interactions or relationships (represented as edges) between entities (represented as nodes). Social Network Analysis (SNA) refers to the collection of methods, techniques and tools in sociometry aiming at the analysis of the structure and composition of ties in social networks. The results of SNA might be used to [18]:

- Identify individuals that are communicating more often with each other (community)
- Identify the individuals with (a) the most outgoing connections (influence), (b) the most incoming connections or in degree (prominence), (c) the least connections (outlier)
- Identify the individuals or groups who play central roles.
- Distinguish bottlenecks (central nodes that provide the only connection between different parts of a network), as well as isolated individuals and groups.

In the EDM context, SNA is usually used to evaluate interactions between students in their collaborative learning tasks, communication actions and online discussions. It can help to understand the group dynamics (structure and content) of educational communities or to quantify the performance of students in teamwork [19-20].

In our case study, we aim to mine and analysis key interaction patterns between training providers and training courses using social mining techniques. To analyze these notions, we rely on two important SNA measures [18]:

Degree Centrality of a node (i.e., the number of nodes that are connected to it): This measure represents the popularity of a node (in our case, training courses or training providers) in a community (in our case, training paths or curriculums).

Betweenness Centrality of a node: In social network context, a node (i.e., training provider or training course) with high betweenness centrality value means that it performs a crucial role in the network, because this node enables the connection between two different groups (i.e., two different training paths or curriculums). If this node is the only bridge linking these two groups and for some reason this node is no longer available, the change of information and knowledge between these two groups would be impossible.

In the process mining field, social mining techniques aim to extract social networks from event logs based on the observed interactions between activities' performers (i.e., resources), depending on how process instances are routed between these performers. These interactions can be generated following one of these five kinds of metrics: (1) transfer of work, (2) delegation or subcontracting of tasks, (3) frequent collaboration (working together) in cases, (4) similarity in executed tasks and (5) reassignment of tasks.

According to our case study, we apply these various metrics to mine social networks between training providers and training courses. Our goal is to find the most pertinent metric allowing us to deduce key interaction patterns between training providers or training courses involved in employees' learning paths. Let us note that, in order to generate social networks between training courses, we replace originator IDs by training IDs of the same events during the event log conversion step in *ProM import*. In what follows, we use the social network plug-ins of ProM 6.3 (based on the four metrics mentioned above) to generate social networks. In the resulting graphs, each node represents a training provider (resp. a training course) where the names have been anonymized for privacy reasons. The oval shape of the nodes in a graph visually expresses the relation between the in and out degree of the connections (arrows) between these nodes. A higher proportion of ingoing arcs lead to more vertical oval shapes while higher proportions of outgoing arcs produce more horizontal oval shapes. We use different views (a ranking view, a stretch by degree ratio, etc.) and two SNA measures (i.e., degree centrality and betweenness centrality) when generating these graphs depending on the key concepts and patterns we want to extract.

Handover of work metric:

Within a case (i.e., process instance) there is a handover of work from individual *i* to individual *j* if there are two subsequent activities where the first is completed by *i* and the second by *j*. In our case, this metric allow us to discover the flow of trainees (specified by the direction of the arrows) between training providers and courses. For instance, in Fig. 4, two providers are connected if one performs a training course causally followed by a course performed by the other provider. In Fig. 4, we distingue two groups of providers strongly related to each other (clustered in cliques) following their causal involvement in training paths. Training providers without arc are those which offer very stand-alone training courses without causal dependency with others. In Fig. 5, we distinguish the most important training courses (trainings with Id 4 et 1) which play central roles in training paths. Training courses or providers with high betweeness centrality represent the ones which connect two different learning paths. In Fig. 6, the size of training courses (with high betweenness) indicates their crucial role as a bridge (i.e., intermediate trainings) between different types of training courses.

Using SNA's measures (betweenness, degree), we can deduce that:

- Training providers or courses with *high degree* are the most popular and prestigious ones, playing a central role in training paths.
- Training providers or courses with no connection with others represent outliers, providing very specific skills, not involved in training paths.
- Nodes with no incoming arcs are training providers (or training courses) who only initiate learning processes (i.e., give the basics for training paths), while nodes with no outgoing arcs are training providers (or courses) who perform only final trainings (i.e., complete training paths with the most required skills).
- Training courses strongly connected to each other hint popular or typical curriculums (or learning paths). The direction of the edges gives the order of training courses followed by students in such curriculums.
- Training courses or providers (with high betweenness) indicates their crucial role as a bridge (i.e., offering intermediate trainings) between different types of training paths.

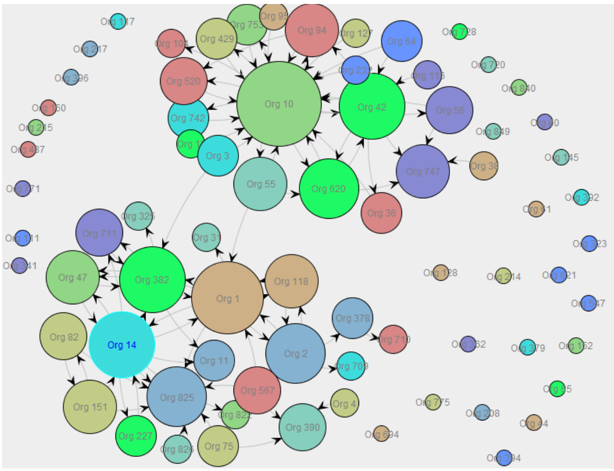


Figure 4. Social network showing handovers between providers of the top 80% of followed training courses using a size by ranking view i.e., the size of a provider's node (which depends on its degree) indicates the importance of its involvement in training paths.

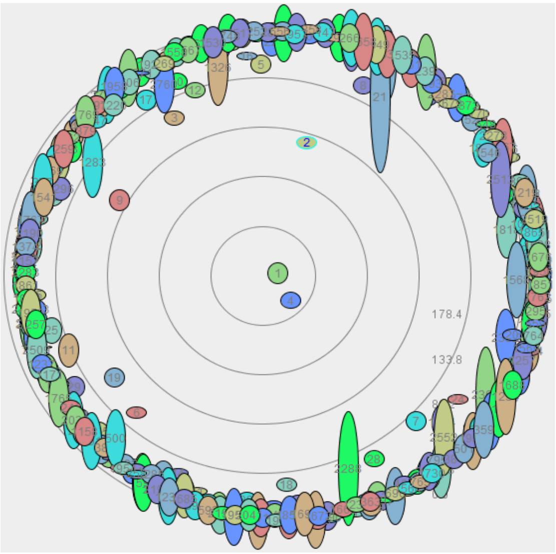


Figure 5. Social network showing handovers between training courses using (1) a ranking view on degree, i.e., courses most involved in training paths are more central in the graph, and (2) a stretch by degree ratio, i.e., the oval shape of the courses' nodes indicates their position in training paths flow.

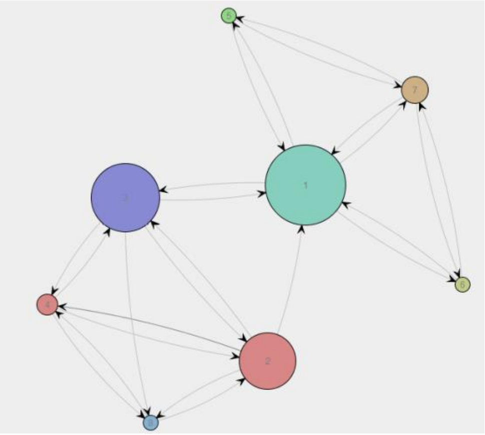


Figure 6. Social network showing handovers between the top 60% of followed training courses, using a ranking on betweenness centrality and a size by ranking view.

Subcontracting metric: A resource *i* subcontracts a resource *j*, when in-between two activities executed by *i* there is an activity executed by *j*. In this case, the start node of an arc represents a contractor and the end node means a subcontractor (see Figs. 7 and 8). In our case study, this metric allow us to extract complementary patterns between training courses and providers.

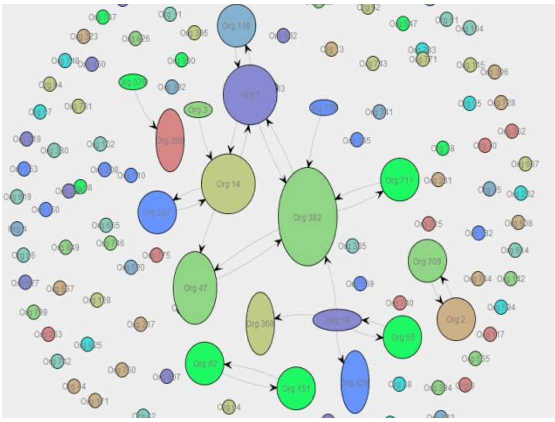


Figure 7. Social network showing subcontracting between training providers of the top 90% of followed training courses.

Using SNA measures, we deduce that:

- Nodes (i.e., training providers or courses) with a high out-degree of centrality (indicated by a horizontal oval shapes) usually play the role of contractors (the main providers or trainings, which give basic skills in these training paths).
- Nodes (i.e., training providers or courses) with a high in-degree of centrality (indicated by a vertical oval shapes) usually act as subcontractors (providers or trainings, which give complementary notions or skills allowing to enhance the notions given by contractors in these training paths).

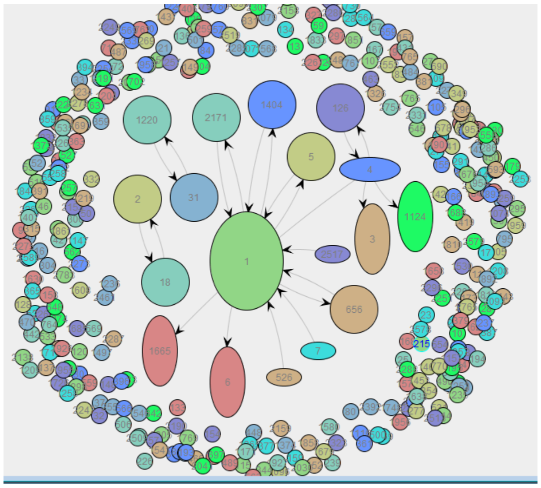


Figure 8. Social network showing subcontracting between the top 80% of followed training courses.

Working together metric: This metric ignores causal dependencies but simply counts how frequently two recourses are performing activities for the same case (see Figs. 9 and 10). In this case, *high density* means that a lot of training providers or courses are involved together in training paths. We can deduce from this social network the most popular curriculums (training providers or courses that work together, i.e., are involved together in training paths). The only difference with the handover metric is that this latter gives us the order followed by students in such curriculums.

Similar task metric: This metric determines who performs the same type of activities in different cases. In our study case, this metric makes sense only to generate relationship between training providers (see Fig. 11). In this case, it allows us to detect training providers who perform the same kind of trainings in curriculums.

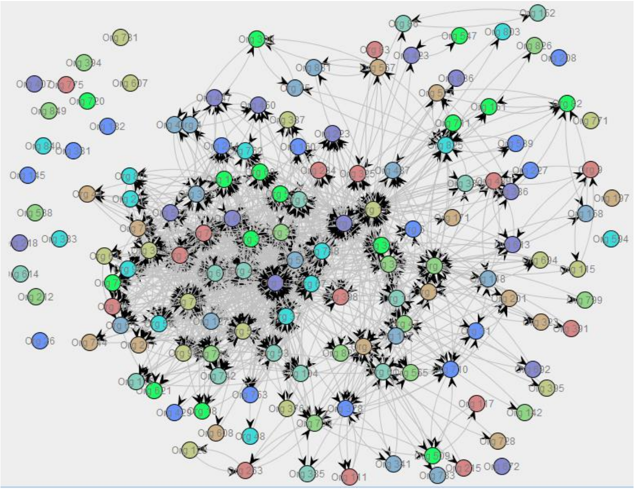


Figure 9. Social network based on working together between training providers. Providers that are often involved together in training paths are related and clustered in cliques. Training providers without arc are those which offer very stand alone rainings.

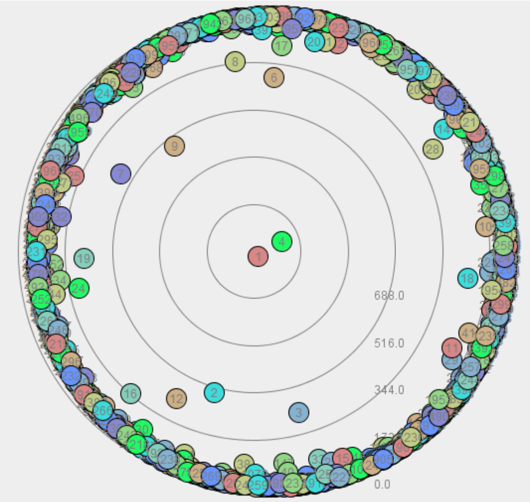


Figure 10. Social network based on working together between training courses using a rancking view on degree, i.e., courses most involved together in training paths are more central in the graph.

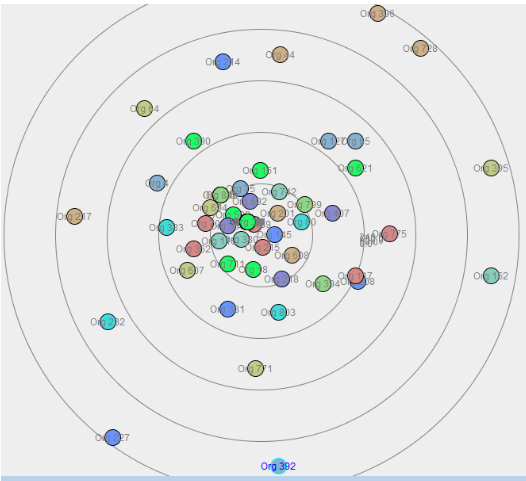


Figure 11. Social network (based on similarity of tasks) between training providers of the top 70 % followed training courses using a ranking view on degree, i.e., providers who perform the most similar collection of trainings are grouped together in the center of the graph.

This experience shows that social network analysis based on event logs is a powerful tool for analyzing coordination patterns between training courses and training providers [9]. Such an approach can also be used to mine interesting patterns about students' behaviors in on-line environments based on resources' usage logs and various interaction logs (e.g., with an intelligent tutoring system).

E. *Process model discovery using a Two-step Clustering Technique*

In order to handle the complexity and heterogeneity of the training paths encountered in the education domain, we propose a two-step clustering approach as a preprocessing step. Our goal is to identify the best training paths by dividing a training event log into homogenous subsets of cases following both their structural similarity and an employability indicator indicating the effectiveness of a training path. In our two-step clustering approach, training paths are firstly partitioned following performance indicators (employability factor and period of unemployment) then training path in each obtained cluster are partitioned further following their structural similarity (see Fig. 12).

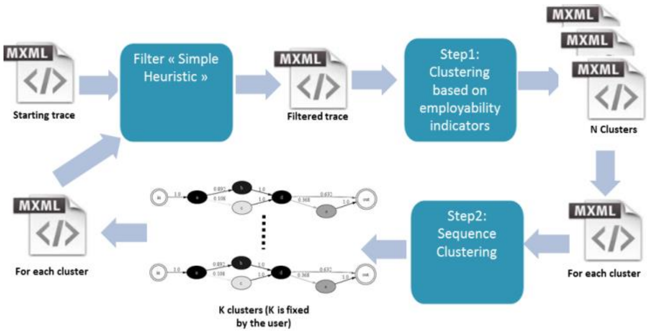

MXML
Starting trace
Filter « Simple Heuristic »
MXML
Filtered trace
Step1: Clustering based on employability indicators
MXML
MXML
MXML
N Clusters
MXML
For each cluster
Step2: Sequence Clustering
K clusters (K is fixed by the user)
MXML
For each cluster

Figure 12. The two-step clustering procedure

A. *First step*

This step consists of creating clusters of similar trainees' profiles based on a training path performance indicator expressed via two criteria. The first one, called employability, concerns the matching between the obtained skills after a training course and those required by a mission. The second criterion represents the time period between a training course followed by an employee and a new mission on which the employee is staffed after it.

a) *Matching criterion:* The criteria that models the matching between skills acquired during a training course and the ones required for a given job/placement, is considered as a real number included between 0 and 1. Hence, this criteria do the matching between a training course followed by an employee, with an identifier « $i \in \{1, \dots, 3340\}$ », and a job/placement will be noted « A_i » with $A_i \in (0,1)$. The set of skills obtained by an employee, identified by i, during his/her trainings is expressed as follows :

$$\mathcal{F}^i = \{F_1^i, F_2^i, \dots, F_{n_i}^i\}$$

Where n_i is an integer greater than or equal to 1. Generally, n_i is at least equal to 3 and less than 10. We note also that for all $\in \{1, \dots, n_i\}$, « F_j^i » indicates that the training course « j » is followed by the employee « i ». For example, $F_1^{10} = Anglais$ means that the employee «10» has followed the English training course. In the same way, the set of skills required by a given job/placement on which the employee « i » has been staffed is noted as follows:

$$\mathcal{M}^i = \{M_1^i, M_2^i, \dots, M_{m_i}^i\}$$

Where m_i is an integer greater than or equal to 1. Generally it is equal to 4 or 5. Also, for all $k \in \{1, \ldots, m_i\}$, « M_k^i » indicates that the skill number « k » is required for the job under consideration. For instance, $M_1^{10} = Anglais$ means that the found job/placement for the employee number « 10 » requires English language skill. In addition, the required skills by a given job/placement are weighted according to their importance for the success of this job. This weighting is modeled as follow:

$$\mathcal{P}^i = \{P_1^i, P_2^i, \ldots, P_{m_i}^i\}$$

Where for all $j \in \{1, \ldots, m_i\}$, $0 < P_j^i < 1$ is the weight associated to the competence « M_j^i » and $\sum_{k=1}^{m_i} P_j^k = 1$. Therefore, the matching criteria between skills obtained by training courses and skills required for a given job/placement is calculated by the following formula:

$$A_i = \sum_{k=1}^{m_i} P_j^i \times \mathbb{I}_{\{M_k^i \in \mathcal{F}^i\}}$$

With $\mathbb{I}_{\{M_k^i \in \mathcal{F}^i\}}$ is an indicator computed by the following rule:

$$\mathbb{I}_{\{M_k^i \in \mathcal{F}^i\}} = \begin{cases} 1 \; si \; M_k^i \in \mathcal{F}^i \\ 0 \; si \; M_k^i \notin \mathcal{F}^i \end{cases}$$

Hence, the distribution characterizing this matching criterion, using our training catalogue and employee information recorded in our example training courses' dataset, is given Fig. 13.

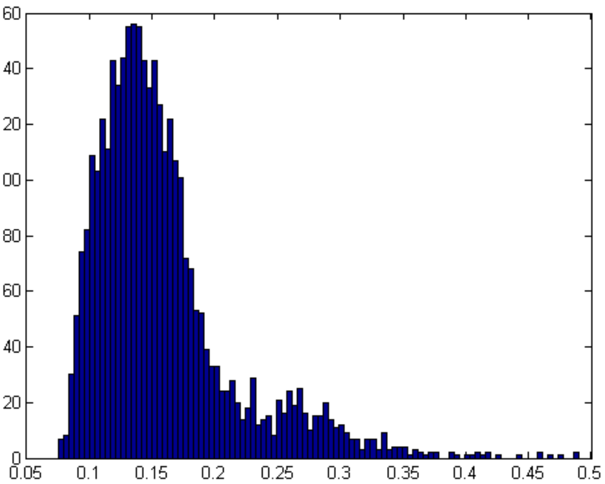


Figure 13. Matching ditribution between training courses and jobs for the employees of our training dataset example.

b) *Time period between training courses and jobs/placements:* This criterion represents, for an employee *i*, the time period between the end of a given training course and the start of his/her next job/placement. This criterion follows the *log normale* probability law. This law is widely used in the modeling of survivor duration. In fact, using the durations, expressed in working days, we obtain the estimated parameters for the used log normale law as follow:

$$\hat{\mu} = 3.16445 \quad [3.11872, 3.21018]$$
$$\hat{\sigma} = \; 1.12863 \quad [1.09721, 1.16191]$$

The graphic representation of the fit of this law is given in Fig. 14. Let us note that we normalize the durations according to the max one in order to have a criterion value comprised between 0 and 1. The goal of this normalization is to homogenize the duration criterion with the matching one.

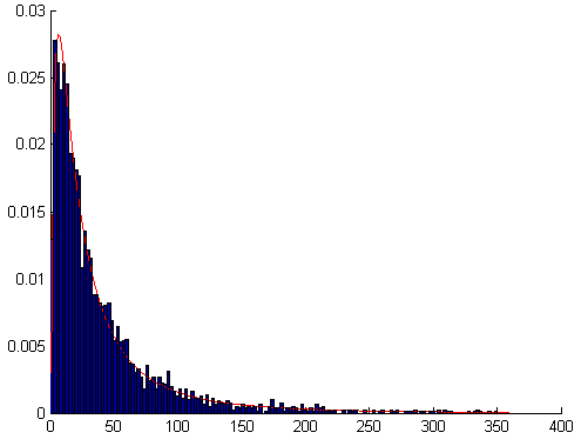


Figure 14. Density probability of the log normal law describing the time between training courses' end and the beginning of new jobs for the employees of the training courses' dataset example

c) *Clustering according to duration and matching criterion:* In these experiments, we do clustering based on the matching and duration criteria defined below. This clustering will help us identify class of training paths for employees that allow them to be staffed on jobs shortly after a training course.

Definition of the cluster number: To get these classes we use the « K-means » technique [21-22], where the optimal number of clusters is determined using a method based on the average silhouette of many clustering where the number of the clusters is varied (the number of clusters *K* is varied between 2 and 5). For more details on this silhouette method, interested readers may refer to [23]. The obtained results are presented in the Fig. 15. When analyzing this figure, we identify a breaking down of the progression of the average silhouette when *K=3*, this means that the clustering is optimal when we do a clustering with 3 partitions (i.e., clusters).

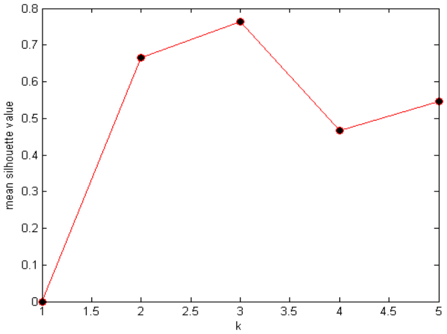


Figure 15. Silhouette Graphical analysis is used to determines the optimal number of clusters. X-axis represents number of clusters and Y-axis indicats asscoiated Silhouette scores.

Clustering for K=3: According to the results obtained in the previous analysis, we apply the "*K-Means*" method, based on the matching and duration criteria, with *K=3*, on our training courses' dataset example. The obtained results are given in Fig. 16.

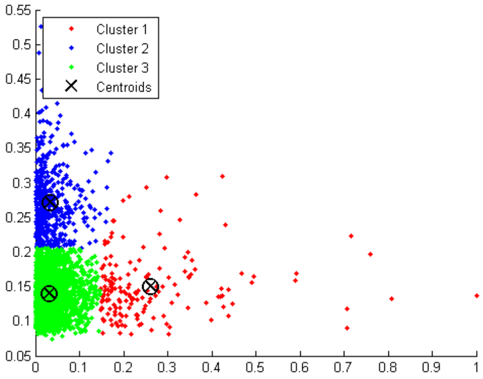


Figure 16. Results of the K-means clustering method applied on our training courses' dataset example using the matching and duration criteria. X-axis represents time period (normalized) between training and the next job. Y-axis corresponds to employability score in (0,1).

K-means method combined with Silhouette Graphical Analysis show that we can identify three trainees groups. Cluster number 2 (Blue group in Fig. 16) represents efficient trainees with high employability score and small employability duration. Instead of, Cluster 1 (red points in Fig. 16) corresponds to inefficient trainees who have small employability score and need more time to find a new job. Finally, green cloud points in Fig. 16 exhibits Cluster number 3 which regroups medium trainees who find quickly a new while they have a small employability score.

We use the fuzzy miner plug-in of ProM (given its robustness to noises) to discover the process model from the training traces of the trainees grouped in the first cluster. We obtain clearly identifiable training paths, as illustrated in Fig. 17. Let us note that these training paths correspond to the least performing ones regarding employability factor and period of unemployment.

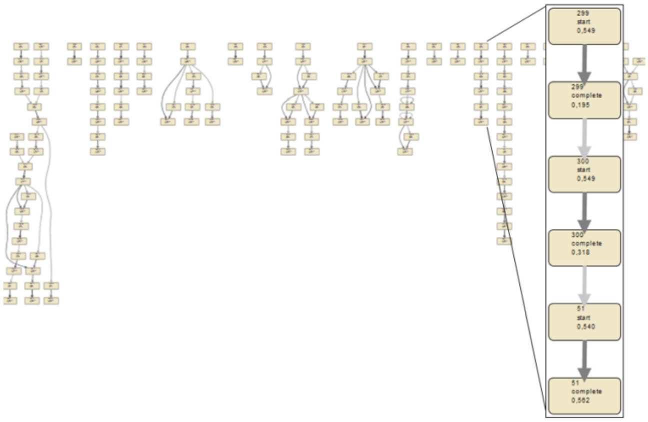


Figure 17. A fragment of the process model showing all the training patterns of cluster 1

Fig. 18 illustrates the process model discovered from the training traces grouped in the second cluster (using the fuzzy miner). Clearly, it is a spaghetti process. The process model discovered from Cluster 3 is even more complex. We can see then that training paths underlying the clusters 2 and 3 are less regular and are more complex that the ones discovered from the first cluster. Let us note that these training paths correspond to the highest performing ones regarding employability factor and period of unemployment.

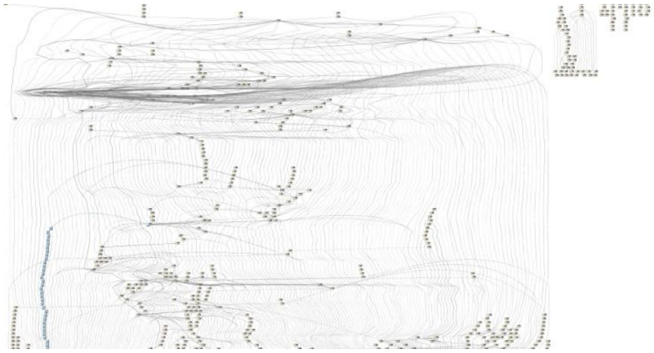
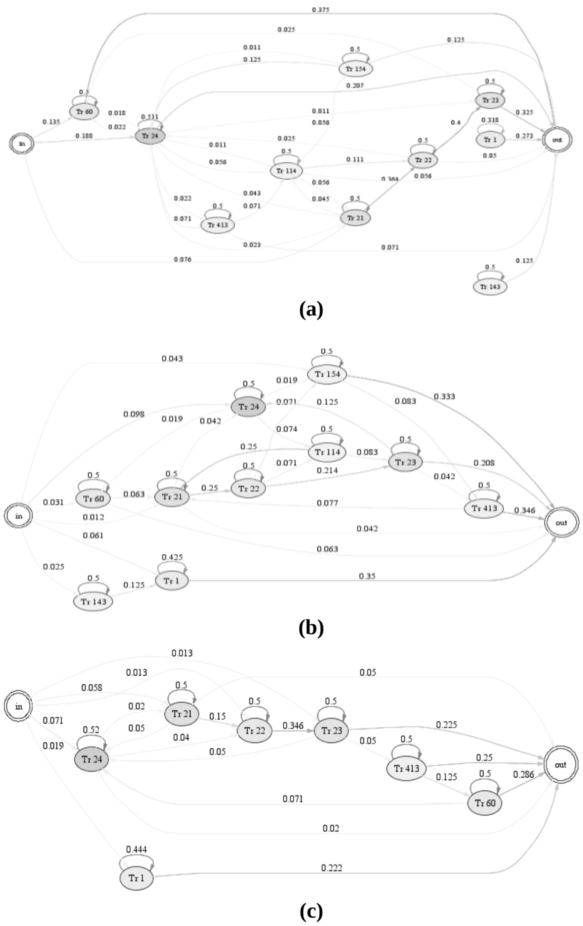


Figure 18. Fragement of the process model (spaghetti-like) underlying cluster 2

In order to obtain simpler training process models, the clusters two and three will be analyzed separately in the second step of our approach. The second step of our approach, simplify further the discovered process models, in the first step, by grouping training paths from each clusters following their structural similarity. Let us note that the first step facilitates the detection of the process patterns and enhances analysis performance because it reduces the searching scope from the whole trainees' information to limit ones for each inferred clusters.

B. Second step

We group training paths (traces in log events) from each of the last two complex clusters discovered in the first step, following their structural similarity using the Sequence clustering technique proposed in [24]. Instead of extracting features from traces, sequence clustering focuses on the sequential behavior of traces. Also, each cluster is based on a probabilistic model, namely a first-order Markov chain. The sequence clustering technique is known to generate simpler models than trace clustering techniques developed in [25]. In our example, when we apply the sequence clustering technique on the second group of trainees with an average employability (i.e., the second cluster of the first step), we obtain three more clusters (cluster 2.1, cluster 2.2 and cluster 2.3). Fig. 19 shows the training process models obtained from the three clusters obtained above, where only transitions occurring above the threshold of 0.05 are represented.



(a)

(b)

(c)

Figure 19. Training process models obtained, from the second cluster of the first step, using the sequence clustering techniques (second step of our approach)

When we apply the sequence clustering technique on the third group of trainees with the less important employability factor (i.e., the third cluster of the first step), we obtain five more clusters (cluster 3.1, cluster 3.2, cluster 3.3, cluster 3.4, cluster 3.5). Fig. 20 shows the training process model underlying the cluster 3.4, where only transitions occurring above the threshold of 0.05 are represented.

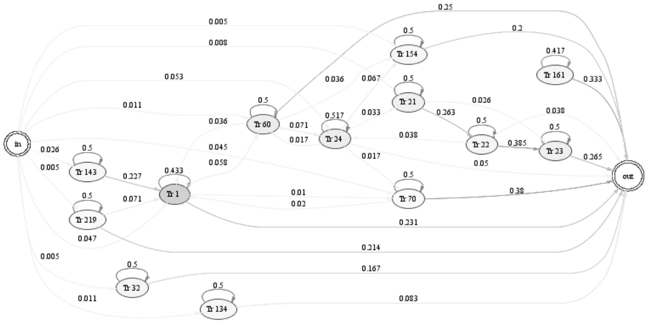


Figure 20. One of the training process model obtained, from the third cluster of the first step, using the sequence clustering techniques (second step of our approach)

IV. PHIDIAS: A PLATFORM FOR DISTRIBUTED EDUCATIONAL PROCESS MINING

To implement our approach, we aim to develop an interactive platform tailored for educational process reconstruction and analysis. This platform will allow different education centers and institutions to load their data and access to advanced data mining and process mining services. Such a platform has to address several issues related to:

- The heterogeneity of the applications and the data sources;
- The connection to some web portals and desktop applications to allow users dealing with the data and exploiting analysis results;
- The ability to add easily new data sources and analysis services;
- The possibility to distribute heavy analysis computations on many processing nodes in order to optimize and enhance platform response time.

To reach these targets we adopt an SOA architecture using an Enterprise Services Bus (ESB) depicted in Fig. 21. This architecture is composed of the following elements: data sources, Enterprise Service Bus, business applications and tools, web services, web portals and connectors. The core of this architecture is the application bus, which guarantees the interoperability and integration of the data sources and applications. For reasons of succinctness we limit the description of the platform to its main components as follow:

Enterprise service bus (ESB)

We have chosen to use ESB architecture in order to have a flexible architecture allowing easily plugging of new applications, data sources and web portals. This integration is done using connectors defining how the data source or the application will be connected to the bus. We recall that process mining is an application that needs a large number of computations in addition to the required capacities to handle data integration and run business and web server applications. This is why we add a resources optimization layer to the ESB. At this level we consider many features in addition to memory and CPU like transmission times.

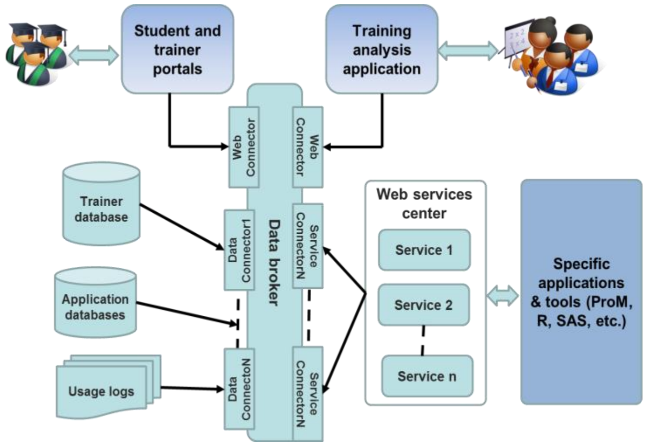


Figure 21. PHIDIAS Architecture

Connectors

The connectors are software components which define how each part of the system will communicate with the ESB. It is one of the main parts of the architecture allowing the system to be flexible and extensible. Its role is the intermediation between a data source, software components, and the web services and applications; and the ESB. This intermediation has different forms; this is why we define three kinds of connectors as follow:

- ***Data connectors:*** these connectors define how transforming the data from their original format to the one for the ESB and how to execute queries on the original sources and then return the results to the ESB.
- ***Web connectors:*** to connect a web portal to the ESB we need to manage the transformation of the user actions recuperated from the HTML blocs to some actions executed on the ESB. This transformation consists of converting for example a user click on a web page to a call of some functions executed on the ESB, then the results are returned back and the connector has to format them in HTML tags.
- ***Service connectors:*** for each service we associate a connector. This connector will store the service interface and transform all the ESB calls to the correct call with the correct parameters. Then it recuperates the results and transforms them to the standard format of the ESB used.

V. Related works

Clustering techniques can be used as a preprocessing step to handle large and heterogeneous event logs by dividing an event log into homogenous subsets of cases following their similarity [25-26], [14-27]. One can then discover simpler process models for each cluster. For this purpose, several clustering techniques have been developed and implemented in ProM, such as the Disjunctive Workflow Schema (DWS) plug-in [26] and the Trace Clustering plug-in [28]. Moreover, in [27], the authors propose a combination of trace clustering and text mining to enhance process discovery techniques such that: (1) Trace clustering is applied with the purpose of dividing the event log traces into sub-event logs; (2) A combination of text mining and data mining is proposed with the purpose of finding interesting patterns for the atypical cases. In [29], the authors propose an approach that uses the starting time of each process instance as an additional feature to those considered in traditional Clustering in Process Mining approaches. By combining control-flow features with the starting time, the clusters formed share both a structural similarity and a temporal proximity. Sequence clustering technique was proposed in [24]. This technique differs from Trace Clustering in several ways. Instead of extracting features from traces, sequence clustering focuses on the sequential behavior of traces. Also, each cluster is based on a probabilistic model, namely a first-order Markov chain. Despite of its success, clustering of event logs still remains a subjective technique. A desired goal would be to introduce some objectivity in partitioning the log into homogenous cases. We found out that the sequence clustering technique seems to be the most adequate to partition efficiently training event logs, in the second step of our approach. However, there are some questions we have to investigate when partitioning the process mining problem into smaller problems such as how to combine the results of the individual sub-problems into solutions for the original problems [14]. An important point to discuss when using decomposed process discovery, is how to assess the quality of a decomposition before starting the time-consuming actual discovery algorithm. In [14], the authors defined three quality notions (cohesion, coupling and balance) that can be used to assess a decomposition, before using it to discover a model or check conformance with.

Recently, Educational Process mining or Curriculum mining has emerged as a promising and active research field in Educational Data Mining, dedicated to extracting process related-knowledge from educational datasets. Beyond limitations of EDM, EPM enables greater insights into underlying educational processes. For instance, in Pecheniskiy et al [30], process mining tools, such as process discovery and analysis techniques, were used to investigate the students' behavior during online multiple choice examinations. In [31], the authors use process mining techniques to analyze a collaborative writing process and how the process correlates to the quality and semantic features of the produced document. Analysis techniques were also applied to check the conformance of a set of predefined constraints (e.g., prerequisites) with the event logs. In [6], the authors proposed a technique relying on a set of predefined pattern templates to extract pattern-driven education models from students' examination traces (i.e., by searching for local patterns and their further assembling into a global model). Under the project "CurriM" [6-7], the authors developed the first software prototype for academic curriculum mining, built on the ProM framework. This tool monitors the flow of curriculums in real-time and return warnings to students (before taking new courses) if prerequisites are not satisfied. Recently, two clustering approaches were proposed in [20], grouping students relying on their obtained marks and their interaction with the Moodle's course. Their aim was to improve both the performance and readability of the mined students' behavior models in the context of e-learning. Finally, in our previous work [9], to handle traces heterogeneity issue, we showed how by associating semantic annotations to educational event logs, 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.

VI. CONCLUSION

In this paper, we studied the potential of process mining techniques in the educational domain. Particularly, we show how social mining techniques (implemented in ProM 6.3) can be used to examine and assess interactions between originators (training providers), training courses or pedagogical resources, involved in students' training paths. We also proposed a two-step clustering approach to extract the best training paths depending on an employability indicator. Our future work will continue in several directions. We intend to combine the approach proposed in this paper with other process mining techniques, which allow discovering interaction patterns from email datasets [32] in order to discover interactions patterns between students in their collaborative learning tasks, communication actions and online discussions. Moreover, the proposed architecture will be implemented and deployed and tested on a distributed environment connected to several data sources and applications. This will allow us calibrating and ameliorating our optimization technique either for storage or computation capacities. To enhance the usability of our platform, we are also working on designing an intuitive graphical interface for non-experts that automatically sets parameters and suggests suitable types of analysis. We also plan to conduct a case study that would illustrate the feasibility of process mining approaches in an on-line education setting. Another important step in our works is to investigate further clustering techniques in event logs decomposition to extract typical or atypical training paths depending on domain specific performance indicators and/or on a set of predefined patterns (describing training path templates). We intend also to develop new clustering and classification techniques taking into account semantic annotations on event logs. For instance, trace clustering techniques can be extended to partition event logs depending on trace similarities at the conceptual level. We also intend develop classification techniques to split semantically annotated event logs based on traces' distance from a set of process models or templates, defined at the conceptual level.

ACKNOWLEDGMENT

This work is done by Altran Research and Altran Institut in the context of the project PERICLES (http://e-pericles.org/).

REFERENCES

[1] A. Hicheur Cairns, B. Gueni, M. Fhima, A. Cairns, S. David and N. Khelifa, "Custom-designed professional training contents and curriculums through educational process mining," The Fourth International Conference on Advances in Information Mining and Management (IMMM 2014), Jul. 2014, pp. 53-58.

[2] C. Romero, S. Ventura and E. Garcia, "Data mining in course management systems: moodle case study and tutorial," Computers & Education, 51(1), 2008, pp. 368-384.

[3] C. Romero, S. Ventura, M. Pechenizkiy and R. Baker, "Handbook of Educational Data Mining," Boca Raton, FL: CRC Press, Taylor&Francis, 2010.

[4] T. Calders and M. Pechenizkiy, "Introduction to the special section on educational data mining," ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), 2011, pp. 3-6.

[5] C. Romero and S. Ventura, "Data mining in education," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol 3, 2013, pp. 12–27.

[6] N. Trčka and M. Pechenizkiy, "From local patterns to global models: towards domain driven educational process mining," Ninth International Conference on Intelligent Systems Design and Applications (ISDA 2009), Nov. 2009, Pisa, Italy, pp. 1114-1119.

[7] N. Trčka, M. Pechenizkiy and W.P.M. van der Aalst "Process mining from educational data (Chapter 9)," In Handbook of Educational Data Mining, CRC Press, pp. 123-142.

[8] W. M. P. Van der Aalst et al., "Process mining manifesto," International Conference on Business Process Management Workshops (BPM 2011), 2011, pp. 169–194.

[9] A. Hicheur-Cairns et al., "Using semantic lifting for improving educational process models discovery and analysis," 4th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2014), Italy, Nov. 2014, pp. 150-161.

[10] W.M.P. van der Aalst and M. Song, "Mining social networks: Uncovering interaction patterns in business processes," International Conference on Business Process Management (BPM 2004), Lecture Notes in Computer Science, vol. 3080, Springer, Berlin, 2004, pp. 244-260.

[11] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters and W. M. P. van der Aalst, "The prom framework: a new era in process mining tool support," International Conference on Applications and Theory of Petri Nets (ICATPN'05), Berlin, Heidelberg, 2005, pp. 444-454.

[12] M. Reichert, "Visualizing large business process models: challenges, techniques, applications," 1st International Workshop on Theory and Applications of Process Visualization Presented at the BPM 2012 (TAProViz'12), Tallin, Sep. 2012, pp. 725-736.

[13] J. C. Bose, R. S. Mans and W. M. P. van der Aalst, "Wanna improve process mining results?," IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, Apr 2013, pp. 127-134.

[14] B.F.A Hompes, H.M.W. Verbeek and W.M.P. van der Aalst, "Finding suitable activity clusters for decomposed process discovery," the 4th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2014), CEUR

Workshop Proceedings 1293, Milan, Italy, November 19-21, 2014, pp. 16-30.

[15] J. Munoz-Gama, J. Carmona and W.M.P. van der Aalst, “Conformance checking in the large: partitioning and topology,” International Conference on Business Process Management (BPM 2013), 2013, pp. 130-145.

[16] R.P.J.C. Bose, W.M.P. van der Aalst, I. Zliobaite and M. Pechenizkiy, “Handling concept drift in process mining,” 23rd International Conference on Information Systems Engineering (CAiSE'2011), Springer, 2011, pp. 391-405.

[17] M. Song and W.M.P. van der Aalst, “Supporting process mining by showing events at a glance,” Seventeenth Annual Workshop on Information Technologies and Systems (WITS'07), Montreal, Canada, December 8-9, 2007, pp.139–145.

[18] C. Aggarwal, “Introduction to social network data analytics,” In Social Network Data Analytics, Springer, 2011, pp. 1-15.

[19] P. Crespo and C. Antunes, “Social networks analysis for quantifying students' performance in Teamwork,” Educational Data Mining (EDM 2012), 2012, pp. 234-235.

[20] G. Obadi, P. Drázdilová, J. Martinovic, K. Slaninová and V. Snásel, “using spectral clustering for finding students' patterns of behavior in social networks,” International Workshop on Databases, Texts, Specifications, and Objects (DATESO), vol 567, 2010, pp. 118-130.

[21] G. A. F. Seber, “Multivariate observations,” Hoboken, NJ: John Wiley & Sons, Inc., 1984.

[22] H. Späth, “Cluster dissection and analysis: theory, FORTRAN programs, examples,” Translated by J. Goldschmidt, New York: Halsted Press, 1985.

[23] L. Kaufman and P. J. Rousseeuw, “Finding groups in data: an introduction to cluster analysis,” Hoboken, NJ: John Wiley & Sons, Inc., 1990.

[24] G.M. Veiga and D.R. Ferreira, “Understanding spaghetti models with sequence clustering for ProM,” Business Process Management Workshops, vol. 43, 2010, pp. 92–103.

[25] M.Song, C.W. Günther and W.M.P. van der Aalst, “Trace clustering in process mining,” Business Project Management (BPM 2008), vol. 17, Springer, Heidelberg, 2009, pp. 109–120.

[26] A.K.A. De Medeiros, A. Guzzo, G. Greco, W.M.P. van der Aalst, A.J.M.M. Weijters, B.F. van Dongen and D. Saccà, “Process mining based on clustering: A quest for precision,” Business Process Management International Workshops (BPM 2007), Brisbane, Australia, Berlin: Springer-Verlag, pp. 17-29.

[27] J. Weerdt, J. Vanthienen and B. Baesens, “Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes,” IEEE Congress on Evolutionary Computation (CEC), 2012, pp. 1-8.

[28] R.P. Jagadeesh Chandra Bose and W.M.P. van der Aalst, “Context aware trace clustering: towards improving process mining results,” In Proceedings of the SIAM International Conference on Data Mining, SDM, May, 2009, pp. 401–412.

[29] D. Luengo and M. Sepúlveda, “Applying clustering in process mining to find different versions of a business process that changes over time,” Business Process Management Workshops, 2011, pp. 153-158.

[30] M. Pechenizkiy, N. Trčka, E. Vasilyeva, W.P.M. van der Aalst and P. De Bra, “Process mining online assessment data,” Educationnal Data Mining (EDM'09), 2009, pp. 279-288.

[31] V. Southavilay, K. Yacef and R. A. Calvo, “Process mining to support students’ collaborative writing,” Educational Data Mining conference proceedings, 2010, pp. 257-266.

[32] W.M.P. van der Aalst and A. Nikolov, “EMailAnalyzer: an e-mail mining plug-in for the ProM framework,” International Conference on Business Process Management, 2007.