Investigation of Performed User Activities in Overall Context with IT Analytical Framework

František Babič, Jozef Wagner, and Ján Paralič

Department of Cybernetics and Artificial Intelligence,
Faculty of Electrical Engineering and Informatics, Technical university of Košice,
Letná 9/B, Košice, 042 01 Košice, Slovakia
{frantisek.babic,jozef.wagner,jan.paralic}@tuke.sk

Abstract. Collaborative user activities represent an important source of knowledge and experiences that can be identified with suitable discovery methods. These activities are typically realized within collaborative environments which in several cases offer only a simple analytical module providing basic statistical information. However, these results are not sufficient for deeper understanding of performed activities, processes behind realized actions, hidden relations, etc. A motivation to discover and understand such knowledge from user activities led us to design a formal representation of historical events and method for their advanced analysis. This paper describes and evaluates a semi-automatic procedure of pattern definition and discovery that can assist and simplify manual evaluation of collected data by teachers, researchers or evaluators.

Keywords: user activities, event log patterns, analytical tool, knowledge discovery.

1 Introduction

Collaborative activities are driven by various elements and models of user behavior. This behavior represents an interaction between different users or between users and an environment. Its aim is to reach specified objectives in an effective and successful way. The decision process of determining which iteration or subset of realized activities is the most effective or most successful one is a very complex task and it needs to be supported by event logs. This complex evaluation can be performed based on a collected historical data representing investigated activities. We have identified several possible methods and approaches to analyze this kind of data (as described in Related work section), but our primary motivation was to automate a process of identification of patterns or knowledge creation processes in a large set of data, which is currently performed manually by end users. Typical example of this procedure can be found in collaborative learning environments. Such environment provides a virtual space for various learning courses having students which produce different outputs during their activities. When course finishes, it is necessary to evaluate used

W. Abramowicz et al. (Eds.): BIS 2012, LNBIP 117, pp. 284–295, 2012.

procedures and methods, taken decisions, created outputs, etc. A teacher or evaluator has three possibilities of how to get the expected assessment: he can use his notes created during the whole course; export and manually analyze data from learning environment within e.g. an Excel; or use some analytical tool (included in a learning environment, or from a third party). The last two options represent an interesting approach on how to process a huge set of data, but it often provides only limited possibilities of preprocessing and query customization. The generation of association rules can be understood as a typical representative of this approach in which most frequent actions or objects are aggregated into rules in form of if X and Y, then Z.

Our motivation was to provide a simple and intuitive visualization of performed user activities in which users have a possibility to more easily identify activity paths which led to an important point in a process, important decision or results. The proposed solution is oriented on collaborative processes within learning environment with no predefined model. This contrasts with business processes and analyses within field of process mining, which we also describe in the Related work section. In many cases the collaborative processes result in the creation of a new knowledge. This knowledge creation process is not known beforehand and it is often unique, so it is necessary to evaluate which procedure or sequence of performed steps led to most valuable knowledge.

This approach can be also adapted to some other systems as virtual learning environment. Interesting application case represents a software testing environment that provides all necessary features for effective realization of testing procedures. Typical software testing consists of relatively well structured processes, in which we can identify many similar procedures, decisions and steps' sequences. These findings can lead to continuous improvement of relevant testing practices in terms of cost reduction, shorter execution time and more effective management of human resources. This is objective of our cooperation with software company called RWE-IT (see subsection 4.4).

Identified process aspects represent one possible view on performed user activities which can be extended with other methods as e.g. text mining analyses on relevant text documents. For this purpose we designed a text mining library called JBOWL in order to develop own implementation of relevant algorithms based on available services. More about this approach is described in subsection 3.3 below.

The methods presented in this paper represent a joint work of research team containing members from two research areas at the Technical university of Košice: Artificial Intelligence and Business Information Systems. This research team offers a good collective space to share ideas and experiences within existing or new education programs, research and project activities.

The whole paper is organized into following main sections: introduction with motivation (this section 1); the presentation of related work (in section 2) with some findings for our research, the technical description of proposed framework (section 3); four application cases in section 4 that are divided into two categories (finished and planned) and a short summary which concludes the paper (section 5).

2 Related Work

The core of our framework lies in its analytical capabilities; therefore we focus this section on several relevant research directions which can be identified in current state of the art research.

We start with some specific areas of data mining research, relevant to our framework. As our primary role is education and many of the processes in the knowledge society deal with some form of learning, we have analyzed the area of educational data mining, which covers the exploitation of data from educational systems or environments [2]. It deals often with temporal, noisy or incomplete datasets [13]. Other important area is process mining, which usually concentrates on business processes and provides extraction of different useful information from event logs, such as actual version of process model, comparison with its planned version and identification of deviations, as well as provision of various performance statistics, time overview of activities, or social network analyses [15].

Complex event processing (CEP, also known as Event Stream Processing) represents an interesting new approach to analysis, providing techniques for processing large amounts of event logs, often in real time. These types of analysis are heavily tied to business processes, and are used e.g. in analysis and monitoring of financial transactions, stock markets and data from RFID chips. Most active research in this field is in SASE[1] *(Stream-based And Shared Event processing)*, ETALIS[2] *(Event-driven Transaction Logic Inference System)* and Cayuga[3]. Recently released open source library called Storm[4], developed by a Twitter, Inc., supports many of CEP techniques and can be used on diverse types of event logs.

The interactions investigation in virtual environment by means of Social Network Analyses is described in [8], [10] or [14]. All these approaches are based on collection of historical logs representing performed activities and offer possibilities to identify social structures, hidden interactions and relations, etc.

The PAnDit [6] "Pattern Analysis and Discovery Tool" works with user activities stored in custom log format and searches through them according to user defined patterns. These patterns are represented as rules and are created with the help of the tool's user interface. With this so called "interaction analysis" user can search for occurrences of various groups of events, or create a nontrivial filter to select interesting events. Searching is implemented in Prolog language, and results are presented on a time line as dots or poly lines.

The exploitation of suitable data mining methods to improve the testing management is described in [9] and [7]. Both examples are focused on reducing the testing time by reducing the number of test cases with identification of similar patterns or the most probably attributes of software behavior.

---

[1] http://avid.cs.umass.edu/sase/index.php?page=home
[2] http://code.google.com/p/etalis/
[3] http://www.cs.cornell.edu/bigreddata/cayuga/
[4] https://github.com/nathanmarz/storm

The aim of this section is to provide some theoretical overview about existing approaches or methods with similar motivation and focus on activity. All of these activities are based on collected data in different formats of logs. We have investigated several types of logs to identify advantages and disadvantages of each solution. Based on these findings we proposed a generic format that provides a rich data structure for analytical purposes. The traditional data mining methods with some extensions (education and process mining) in combination with requirements from teachers and researchers inspired us to define an analytical framework containing all necessary services described in following sections. The main aim is to provide a tool that enables deeper understanding of performed user activities by means of simple data presentation and management from the users' perspective.

3 Proposed Analytical Framework

The proposed technical solution for analytical framework is a combination of existing services (developed mainly within the integrated FP6 project KP-Lab) and newly designed and implemented services, based on identified requirements within selected application cases. Moreover, existing services are being adapted for the new conditions in order to fulfill the specific goals of particular cases.

The core of our solution contains services for event logging, logs storage, manipulation with logs, extraction and visualization based on different user queries [12]. Extracted and visualized information represent a complex view of user behavior during virtual activities or processes, e.g. timeline-based visualization, quantitative statistics, level of collaboration, tacit relations, patterns, etc. The successful realization of all these approaches depends on quality of collected historical data. For these purposes, a generic log format was designed containing all necessary information. The initial list of 12 parameters can be simply expanded if necessary. It contains information about time, object of interest, type of action, actor performing the action and arbitrary custom data. The detailed description is presented in [12].

3.1 Activity Paths and Patterns

Traditional approaches to the analysis of performed user activities, such as process and data mining, require that logged events contain information about process instances to which the event belongs. This information is then used for process discovery and modeling. If no model of a process and no explicit information about process instances are given, traditional analyses are of little use. However, in collaborative processes (research and innovative education processes) most of the time this is the case, as historical data about user activities often contain traces of unique, ill defined processes, which are of great value for the researchers.

For such purposes, we have developed and implemented an analytical methodology and tools for analysis of so called patterns, which identify parts of processes captured in logs. Analysis is interactive, iterative and consists of following steps:

1. Understanding of problem's domain, formulation of hypothesis
2. Acquiring logs of users' actions and basic understanding of them
3. Preprocessing and creation of a filter in order to select and prepare suitable data set for analysis
4. Creation of a pattern and its parts.
5. Performing search for a pattern occurrences in given data set
6. Interpretation of results, iteration *(back to step 3 or 4)*

Pattern is a collection (usually a sequence) of fragments, each describing a generalization of some activity. Searching finds occurrences of a given pattern in a data set. Fragments are matched with events, resulting in a search tree. Results are then represented as leaf nodes in a lowest level. Process of searching with a following pattern (uppercase symbols denote variables which bind to concrete values from event logs) is shown in Fig. 1:

```
(def f1 {:actor :X :type "opening" :entity "doc1"})
(def f2 {:actor :X :type "creation" :entity :Y})
(def f3 {:actor :X :type "link" :entity :Y :link-to "doc1"})
(def pattern [f1 f2 f3])
(search data pattern)
```

Above pattern finds situations where user (any user X) created a new document (Y) after reading particular existing document (doc1), and then linked the two (previously existing document doc1 with the newly created document Y) together. In our application case the search found two results (user a2 created documents doc4 and doc6 and linked both of them to doc1), depicted as green leaf nodes. Searching process operates with two variables, X and Y, defined in the patterns above, which are bound to values as the search progresses.

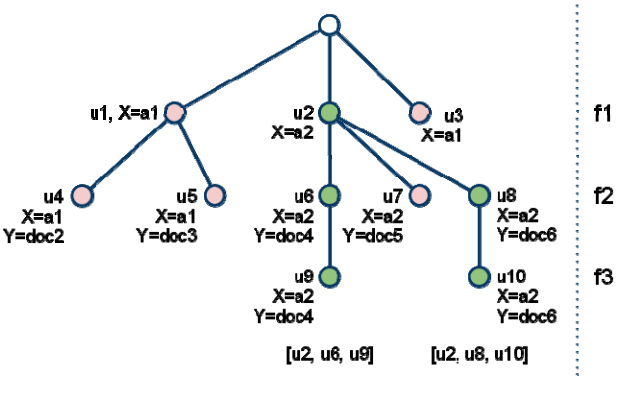

Fig. 1. Search tree for a pattern analysis

Depth-first search is used, with some optimizations which remembers environment (current fragment and variable bindings) of traversed nodes and does not expand new nodes if they happen to have same environments as the failed ones (note that node u3 does not expand).

Analysis was implemented within library called Piaget, in a functional language called Clojure, running on top of JVM (Java Virtual Machine). It provides an API for log import, preprocessing, pattern definition and searching. For the experiments, we have developed a simple user interface for both browsing and viewing of event logs and for pattern definition and search. Tools allow for creation of a pattern either from a scratch (step-by-step definition of all its parts), or by generalization of selected set of events from a log.

3.2 Quantitative Analyses

Quantitative analyses were designed to offer basic summarized information about performed user activities. Similar feature is available in most collaborative environments (like e.g. Moodle), but our aim was to provide unified middleware services on top of the logs which can be used by various graphical user interface (GUI) tools fulfilling different kinds of users' requirements. Such a separation of middleware analytic services and GUI services is not offered by any of current collaborative environments and presents our original contribution to the state-of-the-art providing higher flexibility for analytical as well as GUI services.

The data extraction services are implemented on the middleware level and a simple example of aggregation service is following: String *activityAggregation* (Query *query*, List<AggregationFunction> *aggregationFunctions*, Set<GroupBy> *groupBy*), where:

- *Query* parameter describes constraints used for filtering of the actions included in the aggregated view. Query object encapsulates the parameters from log format and two additional constraints for more specific description, as *filter* - set of key value pairs which will be compared with events custom properties, *excludeFilter* - true or false, whether include or exclude events which do not have properties from the filter present in them.
- *AggregationFunctions* specify the list of aggregation functions included in the view computed from the set of selected events as *NumOfActivities*; *NumOfActors*, *NumOfEntities*, *TimeSpan* - starting and ending date of investigated time period.
- *GroupBy* specify clause for the grouping of the result by actor, object or type of action.

The results depend on type of investigated environment, processes or activities, but it could be e.g. as number of participants involved and number of actions performed by each of them; number of shared objects used / changes made / versions produced; number of comments added; number of meetings, links, etc. in given time interval, within given group or with other constraints posed by the user in the analysis phase. The detailed description can be found in [11]. On the other hand we have starting a development process of our own GUI for quantitative analyses to provide simple accessible web interface with possibilities to define required queries and visualize the obtained results within friendly and understandable graphical format. This GUI will complement our analytical framework at the end of this year

3.3 Text Mining Services

Important source of contextual information is not presented in the log files, but rather in textual form (reports, working papers, discussion contributions, etc). Therefore one type of analytical services in our framework is based on text mining techniques. For this purpose we utilized our JBOWL[5]. JBOWL is an open source Java library that was designed to support different phases of the text mining process and offers a wide range of relevant classification and clustering algorithms. Its architecture integrates several external components as JSR 173 – API for XML parsing or Jakarta Lucene for indexing and searching. The motivation behind the design of this library seven years ago [3] was existence of many partial implementations of different algorithms for processing, analyses and mining in text documents within our research team on one hand side and lack of equivalent open source tools at that time. The main aim was not to provide simple graphical user interface with possibility to launch selected procedures but to offer set of services necessary to create the own text mining stream customized to concrete conditions and specified objectives. At the time of beginning (2005) several similar alternative existed as Lucene, GATE and Weka, but each of these systems covered only limited range of features providing by JBOWL [3].

The initial set of JBOWL functions:

- management and manipulation of large sets of text documents,
- support for different formats as plain text, HTML or XML in both languages: Slovak and English,
- services for indexing, complex statistical text analyses and preprocessing tasks,
- interface for knowledge structures as ontologies, controlled vocabularies or database WordNet

has been continuously extended and improved based on new requirements or expectations expressed by researchers and students of our department. The last JBOWL update offers possibility to run the text mining tasks in a distributed environment within task-based execution engine. This engine provides middleware-like transparent layer (mostly for programmers wishing to re-use functionality of the JBOWL package) for running of different tasks in a multi-threaded environment [5]. The available newly added services will be used e.g. for aspect-based sentiment analysis or formal concept analysis [4].

4 Application Cases

The aim of the application cases described below was to apply our framework into real practices to evaluate its usability and contributions to other state-of-the-art tools. Following four cases can be divided into two groups: the first two present recently finished experiments and the second pair sketches out our future research plans.

[5] `http://sourceforge.net/projects/jbowl`

4.1 KP-Lab System

KP-Lab System represents an initiative in the domain of collaborative learning environments and provides necessary functionalities to support knowledge building and creation within groups of users utilizing tools for collaborative editing, commenting, tagging, chatting, etc. This system was designed and implemented within European IST FP6 project called KP-Lab[6]. Our team participated on the development of logging and analytical services that were described in previous sections.

All performed actions in virtual user environment were monitored and stored into separate log repository in order to obtain necessary historical data for analytical purposes. At the end of the project (May 2011) log repository contained more than 100 thousand user activities. Even after official end of the project, KP-Lab System is still in operation and many of project's user partners use it in their courses.

The collected data can be used for two purposes: on-line notification services based on user specified requirements and off-line analyses in two different ways: quantitative analyses or timeline-based analyses with pattern discovery (see Fig. 2).

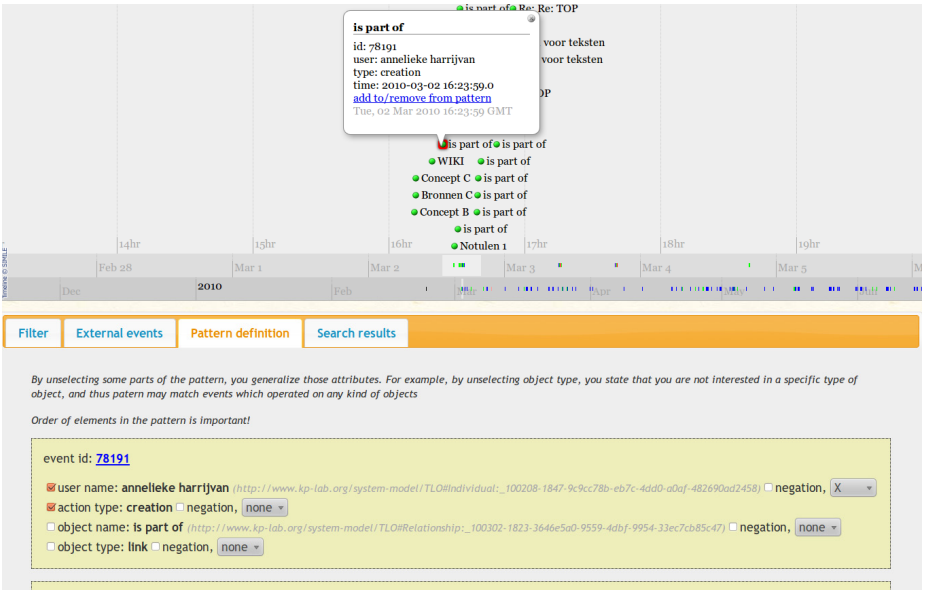

Fig. 2. Timeline-based visualization of one learning course within KP-Lab project with possibility to specify pattern definition

Experiments were based on real world data collected from pilot courses in Netherlands, Finland, Austria and Sweden. Analyses were performed by researchers and teachers of given courses, and they reflect genuine and real-world problems arisen in these courses. This included problems such as finding whether students

[6] http://www.knowledgepractices.info/

contribute to each others work, whether they take note of others findings and additions within a virtual shared space, or whether they cooperate on categorization of documents collected during the course. For each of the teacher's hypothesis, a formalized pattern was created and used in a subsequent search. For some problems, teachers were searching for occurrences of a given pattern, but sometimes even the number of results itself and frequency of result occurrences led to the resolution of the hypothesis. More detailed descriptions of performed experiments and their evaluation is provided e.g. in [12].

4.2 Wikipedia

Experiments were performed also on the data taken from the English Wikipedia. The sheer amount of logged information and contributing users and a collaborative nature of the Wikipedia provided a suitable environment for the evaluation of our analytical tools. The goal of our experiments was to analyze the cooperation of Wikipedia contributors in fighting the vandalism. We wanted to analyze and identify the methods for maintaining the quality of created knowledge.

Data for our analysis consisted from log of articles changes containing more than four hundred million events. Log contained changes of the 3.6 million of articles by 14 million users.

By textual analysis of event's free text description, we have identified events which have represented a fix of the vandalized article by undoing edits, also called reverting. Further preprocessing filtered out all other types of events and sorted the data in chronological order.

Traditional analytical tools applicable to Wikipedia offer only a minimal support for analysis across multiple users and articles. We have thus oriented on such kind of situations, and formulated following problems which we have than analyzed:

- Find a diligent user who reverted multiple articles vandalized by the same person.
- Find a persistent vandal, by finding situations where multiple people had to revert the actions of one vandal on one article.
- Identify vandals who defaced multiple articles over short period of time, in a way that multiple persons had to fix his edits.

We have formulated a formal pattern for each of the problems and analyzed the results of the search. In each case, results unanimously identified persons we were searching for, which we further confirmed by looking at users' profile, history of users actions or articles modified by given person.

The size of data was a challenge for this analysis. Moreover, events lacked the information about the type of action (whether it is a revert or a fix), and we had to analyze the comments attached to the events. Scalability of our analysis could be improved by incorporating a distributed storage and computation facilities into our tools, such as Hadoop and Cascalog.

4.3 Moodle

Moodle is one of the most used virtual learning systems in the world. Departments and faculties at our university use several Moodle portals for their learning courses. Even though Moodle offers some basic statistics about object usage frequency, it is not sufficient and we are in the process of integration with our framework in order to provide more advanced analyses. In this case, we performed some initial experiments with our logging services to obtain necessary historical data from Moodle. These experiments with the description of used methods and obtained results are described in [1]. We're planning to provide an effective combination of learning system and analytical extension, applicable for our teachers, researchers and students.

4.4 Software Testing Environment

Testing phase represents important step in the software development process and is typically supported by suitable technological solutions. Traditional testing environment offers functionalities to specify testing goals, testing procedures, input data, involved participants, expected goals, etc. The standard testing process contains sequence of planned user actions, but sometimes it is necessary to extend these sequences with measures to solve the unplanned or unexpected situations in software behavior. All performed actions and activities within testing process are monitored and stored in order to identify the advantages or disadvantages of used methods, approaches and ways of dealing with situation described before. This broad collection of historical data represents an interesting source for application of suitable analytical methods such as data mining, timeline-based visualization, pattern identification and discovery, etc. The successful realization of this case requires collaboration with companies engaged in software testing, in our case the RWE IT[7]. The cooperation has started with identification of possible application directions such as modeling of testing procedures and finding bugs based on the analysis of testing logs generated by automatic testing procedures.

5 Conclusion

In this paper we have examined a potential of analytical services for less structured collaborative activities realized within different virtual environments. The main aim of our proposal is to provide features for semi-automatic identification of important or interesting patters in collected historical data representing these activities. Its usability (sufficient content of the logs, effective logging mechanism, adequate pattern representation and its successful discovery) was tested within two initial sets of experiments: a collaborative learning system and Wikipedia. In the case of KP-Lab System, teachers and researches evaluated our framework as a very good step towards replacing manual examination of huge datasets after each learning course. Experiments with logs from Wikipedia were based on specified hypothesis that were

[7] http://www.rweit-slovakia.com

verified with our pattern services. Our future work will deal with adaptation of described framework into new conditions of software testing environment and we will also continue with research in a domain of learning systems, i.e. planned integration with Moodle and a dissemination within our university. The text mining services will be integrated in our analysis in order to acquire the additional information from produced text documents during realized activities.

Acknowledgments. The work presented in this paper was partially supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/1147/12 (40%); the Slovak Research and Development Agency under the contract No. APVV-0208-10 (30%). This work is also the result of the project implementation Development of the Center of Information and Communication Technologies for Knowledge Systems (project number: 26220120030) supported by the Research & Development Operational Program funded by the ERDF (30%).

References

1. Babič, F., Wagner, J., Jadlovská, S., Leško, P.: A logging mechanism for acquisition of real data from different collaborative systems for analytical purposes. In: SAMI 2010: 8th International Symposium on Applied Machine Intelligence and Informatics, Herľany, Slovakia, pp. 109–112. IEEE (2010)
2. Baker, R.S.J.D., Yacef, K.: The State of Educational Data Mining in 2009: A Review and Future Visions. Journal of Educational Data Mining 1(1), 3–17 (2009)
3. Bednár, P., Butka, P., Paralič, J.: Java Library for Support of Text Mining and Retrieval. In: Proceedings of the 4th Annual Conference Znalosti 2005, pp. 162–169 (2005)
4. Butka, P., Pócsová, J., Pócs, J.: A Proposal of the Information Retrieval System Based on the Generalized One-Sided Concept Lattices. In: Precup, R.-E., Kovács, S., Preitl, S., Petriu, E.M. (eds.) Applied Computational Intelligence in Engineering. TIEI, vol. 1, pp. 59–70. Springer, Heidelberg (2012)
5. Butka, P., Sarnovský, M., Bednár, P.: One Approach to Combination of FCA-based Local Conceptual Models for Text Analysis - Grid-based Approach. In: Proceedings of the 6th International Symposium on Applied Machine Intelligence, SAMI 2008, Herľany, Slovakia, pp. 131–135 (2008)
6. Harrer, A., Lingnau, A., Bientzle, M.: Interaction Analysis with dedicated logfile analysis tools – a comparative case using the PAnDit tool versus manual inspection. In: Ninth IEEE International Conference on Advanced Learning Technologies, pp. 405–407. IEEE, Washington (2009)
7. Ilkhani, A., Abaee, G.: Extraction test cases by using data mining; reducing the cost of testing. In: Proc: Computer Information Systems and Industrial Management Applications CISIM 2010, Poland, pp. 620–625. IEEE (2010)
8. Martinez, A., Dimitriadis, Y., Rubia, B., Gomez, E., de la Fuente, P.: Combining qualitative evaluation and social network analysis for the study of classroom social interactions. Computers and Education 41(4), 353–368 (2003)
9. Muthyala, K., Naidu, R.: A novel approach to test suite reduction using data mining. Indian Journal of Computer Science and Engineering 2(3), 500–505 (2011)

10. Nurmela, K.A., Lehtinen, E., Palonen, T.: Evaluating CSCL log files by Social Network Analysis. In: Proc. CSCL 1999 Conference, pp. 434–444. Stanford University, Palo Alto (1999)
11. Paralič, J., Babič, F., Wagner, J., Simonenko, E., Spyratos, N., Sugibuchi, T.: Analyses of Knowledge Creation Processes Based on Different Types of Monitored Data. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) ISMIS 2009. LNCS, vol. 5722, pp. 321–330. Springer, Heidelberg (2009)
12. Paralič, J., Richter, C., Babič, F., Wagner, J., Raček, M.: Mirroring of knowledge practices based on user-defined patterns. The Journal of Universal Computer Science 17(10), 1474–1491 (2011)
13. Perera, D., Kay, J., Yacef, K., Koprinska, I., Zaiane, O.: Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. IEEE Transactions on Knowledge and Data Engineering 21(6), 759–772 (2009)
14. Rabbany, R., Takaffoli, M., Zaïane, O.R.: Analyzing Participation of Students in Online Courses Using Social Network Analysis Techniques. In: Proc. EDM, pp. 21–30 (2011)
15. Van der Aalst, W.M.P., et al.: ProM. The Process Mining Toolkit. In: Proceedings of the BPM 2009 Demonstration Track, Ulm, Germany. CEUR-WS.org, vol. 489 (2009)