Journal of Universal Computer Science, vol. 17, no. 10 (2011), 1474-1491
submitted: 30/10/10, accepted: 15/4/10, appeared: 1/6/11 © J.UCS

Mirroring of Knowledge Practices based on User-defined Patterns

Ján Paralič
(Technical University of Košice, Slovakia
Jan.Paralic@tuke.sk)

Christoph Richter
(Christian-Albrechts-Universität zu Kiel, Germany
richter@paedagogik.uni-kiel.de)

František Babič
(Technical University of Košice, Slovakia
Frantisek.Babic@tuke.sk)

Jozef Wagner
(Technical University of Košice, Slovakia
Jozef.Wagner@tuke.sk)

Michal Raček
(PÖYRY Forest Industry Oy Vantaa, Finland
Michal.Racek@poyry.com)

Abstract: Knowledge practices mirroring is one of the central elements of the Trialogical Learning Approach as it is supposed to be a driving force in processes of practice transformation and knowledge creation. The exploitation of historical logging data holds promise to provide a great deal of information about group activities without requiring additional efforts for recording of events by the users. Building on the Trialogical Learning Approach as well as related work in the fields of Data Mining and Knowledge Discovery, Computer-Supported Collaborative Learning, and Information Visualization, this paper suggests high-level requirements for mirroring tools in support of practice transformation and introduces a software tool called Timeline-Based Analyzer (TLBA) that was designed and developed in response to these requirements. One of the main TLBA features is the possibility to define patterns as sequences of relevant actions that resulted into critical moments in investigated practices. Such kind of patterns might also represent conceptually interesting practices that emerged within a particular context – either being positive (a sort of good practices), or negative (bad practices). The usability of the whole analytical solution has been tested through the first iteration of several practical experiments and case studies. One of them is described in this paper and illustrates how TLBA can be used to support collaborative analysis and mirroring. The results of these evaluations have been used for continued improvement of the TLBA in order to provide a stable and intuitive tool not only for researchers, but for daily use of teachers or other users.

Keywords: knowledge practices, collaborative system, timeline-based visualization, patterns
Categories: L.0.0, L.3.4, L.3.6

1 Introduction

Mirroring and reflecting on one's own and other people's working or learning practices is an essential meta-activity for any kind of project work. Analysis and reflection are thereby not understood just as a means of optimizing or improving a given way of working, but also as an active and productive process geared towards the advancement of collaborative knowledge practices. Collective reflection thereby exceeds the exchange of individual experiences and insights in that it aims at the development of a shared understanding and transformation of the collective knowledge practice.

To support collaborative analysis and reflection, participants need to have some kind of material evidence regarding the pursuit knowledge (working or learning) practices [Bolfíková et al. 2009]. Respective techniques include for example narrative methods of story-telling, the writing of diaries and reflection notes, but also log-file based tools that mirror collaborators' activities in terms of recorded interactions with some shared software application. The exploitation of historical logging data holds promise to provide a great deal of information about group activities without requiring additional efforts for recording of events by the users.

The goal of the proposed approach is to provide teams of students, teachers, and knowledge workers with tools and methods enabling them to reflect on their knowledge practices while being engaged in substantial project work over longer periods of time. Emphasizing the productive and open-ended nature of collective reflection, our approach is focused on the development of an analytic tool that allows users to depict and interpret the digital traces of their practices in an iterative and collaborative manner. We do not aim to identify useful indicators or patterns of successful collaboration a priori, but to empower students, teachers, and knowledge workers to identify meaningful indicators and patterns by themselves.

We consider it important that our data analysis tool and methods can be used in connection to any virtual collaborative system that provides respective traces of users' activities. The rest of the paper is organized as follows: we first define the position of our research with respect to relevant research areas in the rest of this section (1.1). In section 2 we outline high-level design goals for an analytic tool and introduce the Timeline-Based Analyzer (TLBA), which has been built to address these goals in section 3. Section 4 deals with an important analytical feature of the TLBA called patterns with detailed description of proposed representation and implementation procedure. Section 5 presents a case study, describing how the TLBA can be used to provide an important input for collaborative project retrospectives. The paper closes with a short summary in section 5.

1.1 Related Work

Besides its foundation in the trialogical approach to learning [Paavola and Hakkarainen 2009], we draw on research in the areas of Process Mining and Educational Data Mining as well as computer-supported collaborative learning (CSCL). In the following paragraphs we briefly discuss related work in these fields.

Both process mining and educational data mining can be understood as extensions to classical data mining. While process mining is focused on the extraction of potentially useful information from event logs in general [Aalst et al. 2009], educational data mining is concerned with the exploitation of data from educational settings, especially from web-based learning environments [Baker and Yacef 2009].

The process mining tool ProM[1] [Aalst et al. 2007], developed in the Eindhoven University of Technology, represents state of the art in the process analysis. ProM tool offers a vast number of analyses of the event logs themselves, or analyses where also the process model is present. For the purposes of analyses, the ProM tool expects that it is possible to divide activities from the event log into separate sets, where each set represents one instance of a process, called “a case”. A given activity present in the event log must belong to exactly one process instance. This requirement is critical for the purposes of the subsequent analyses in the ProM tool.

Especially in the area of business processes several theories and approaches have been proposed, and applications have been devised to support the creation, execution and evaluation of a process model [Habala et al. 2009]. The main problem here is that the analysis of business processes is geared today mainly towards orchestration issues and/or performance improvements [Sabol et al. 2010], [Bartaloš and Bieliková 2009]. Our focus is different. The core issue here is the identification, comparison and analysis of knowledge practices projected into the learning/working process supported by an intelligent information system.

The use of process data from workplace or educational settings goes along with particular challenges. For example, [Perera et al. 2009] pointed out that the existing data mining algorithms are often insufficient given the temporal, noisy, correlated, incomplete, and sometimes small size of the data sets available. Furthermore, it has been argued for information visualization methods in support of human judgment, which are appropriate for the users supposed to make sense of the data collected, e.g. [Baker 2010]. Against this background, this paper explores requirements for process mining and interactive visualization techniques when applied to poorly understood and loosely structured data.

Hardless and Nulden proposed the Activity Visualization (AV) method as technical supporting functionality for understanding of learning processes in virtual environment. AV uses information from the environment to visualize aspects of the whole learning process in order to offer an opportunity to view activities, progress, and usage patterns from various perspectives [Hardless and Nulden 1999], e.g. analysis of message lengths to reveal usage patterns and relationships; message counts in relation to time; usage patterns showing complete overview of when, where, and how to give an idea of what has happened; possibility to give individual feedback as opposed to public messages; and mood indicators to improve understanding of context.

The idea to visualize logged events on a timeline appears in a couple of systems, from which we have selected here the most relevant ones. Semantic Spiral Timelines is an interactive visual tool aimed at the exploration and analysis of information stored in collaborative learning environments [Gomez-Aguilar et al. 2009]. It provides an interesting way of presenting the events in the form of spiral timeline that contains sequences of colour-coded events. These are ordered clockwise with the oldest data at the centre of the spiral and the outermost data depicting the most recent

[1] http://prom.win.tue.nl/research/wiki/

event. The actual form of visualization can be changed by various filters as orientation of individual person, selected type of actions, selected time interval, etc. The patterns are understood here as typical histograms of frequencies of various types of events in time.

A similar approach is described in [Zaïane and Luo 2001] that presents a constraint-based approach to pattern discovery. This approach comprises the definition of filters to reduce the search space during the preprocessing phase, the specification of constraints during the mining phase (used methods as association rule mining, sequential pattern analysis, clustering and classification) to accelerate and control pattern discovering and the definition of constraints in the evaluation phase to ease interpretation and comprehension. The results are visualized through intuitive graphic charts and tables in order to make the discovered patterns easy to interpret.

The OCAF framework [Avouris et al. 2002] shares similar motivation to our approach. It also places objects at the centre of the collaborative activity. Those objects are studied as entities with their own history and evolution procedures. Synergo [Avouris et al. 2007], which builds on the OCAF framework, already supports the automatic association of some kind of events, e.g. insertion, or modification event. The authors also list serious shortcomings of log analysis (especially quantitative indicators may be misleading if e.g. some events, although annotated, may be unimportant, but counted), so they recommend mainly qualitative analysis of collaborative activities.

The approach described in [Avouris et al. 2007] has been implemented in the Collaboration Analysis Tool (ColAT) for building interpretative models of activities in form of a multilevel structure. ColAT provides very good support for fusion of data from multiple sources (not only logs, but also e.g. audio or video recordings). The crucial point in this kind of analysis is the multi-level annotation of events. The first, so called operations level corresponds to the original events in the log file in XML format of the events, defined in [Kahrimanis et al. 2006]. The second, so called actions level, enables to group events from the first level. These manually selected groups of events can be categorized according to a typology defined by the researcher. Similarly, the third - activity level entries are created by grouping actions from level 2, using again a different annotation schema than on the previous level.

Drawing on this work, the main aim of the approach proposed in this paper is to enable users to mirror and reflect on their collaborative work or learning practices by themselves. Rather than 'outsourcing' the analysis to some external expert or algorithm, we put focus on the knowledge worker, student and teacher as analysts of their own activities. Furthermore, due to the loosely structured and emergent nature of knowledge practices we emphasize qualitative approaches taking into account the contextual knowledge of the actors involved. That means, the proposed application called TLBA provides a historical perspective, not planning functionality for future organization of user activities, while it can lead to improvement of future practices as a result of reflection. For example, in a selected collaborative learning environment (e.g. KP-Lab System or Moodle), a teacher or supervisor creates a workspace with some tasks, expected outputs, list of participants with responsibilities, etc. The course or project will be carried out by several teams, each of them performing particular collaborative practices and creating and working with various kinds of artefacts. The whole working process around these shared objects can be visualized and analyzed through the TLBA.

2 Mirroring of knowledge practices

Collaborative analysis and reflection on knowledge practices can take quite different forms and might involve different groups of stakeholders. This fact motivated several usage scenarios that can be covered by the proposed analytic approach:

- Monitoring can be understood as the tracking and analysis of activities as they unfold in time. Monitoring activities might be carried out by a supervisor or by one or more of the persons actually enrolled in the activity at stake. The primary aim in this case is to raise awareness of and to support the analysis of critical events in the course of collaborative action.
- Project retrospectives have been suggested as an important element of project-based work. In comparison to the monitoring activities, collaborative retrospectives provide a more formal and systematic approach aimed to reflect on past activities and to derive lessons learned for future work.
- The analysis and comparison of practices across teams or projects provide the third main usage scenario. Again, this kind of analysis can be carried out by the collaborators themselves or a supervisor.

The analysis of these usage scenarios led to four main design goals for an analytic tool in support of collaborative reflection:

1. *Supporting the explorative analysis* of computer-supported knowledge practices: rather than confronting users with predefined queries and indicators, an application should enable users to make sense of the data by themselves and in collaboration with others. Towards the end, users should be able to filter, aggregate, search, and annotate the data. Furthermore, they should be able to follow traces of material evidence of their activities on different levels of abstraction, providing overview and detailed information when needed.
2. *Openness for external events*: as collaboration seldom takes place in virtual environments alone, but often comprises a complex mixture of computer-supported teamwork, face-to-face meetings and work on non-digital artefacts, users should be able to complement automatically recorded data by other sources of data, including their memories.
3. *Supporting multiple perspectives and intergroup comparison*: a tool should allow users to share and articulate different perspectives, but also allow for comparisons across groups, in order to foster cross-fertilization and exchange. Therefore it should provide mechanisms for the exchange of queries, patterns, and views as well as the possibility to annotate and comment on events.
4. *Providing meaningful and comprehensible visual metaphors which can be easily customized to the information needs of various types of users*: to

account for different backgrounds and levels of expertise a tool should offer visual formats that are meaningful and comprehensible for all stakeholder groups. In addition, user interface mechanisms for data manipulation and analysis need to be powerful but yet intuitive to be accessible even for occasional users with limited background on data analysis.

3 Timeline-Based Analyzer

The Timeline-Based Analyzer (TLBA) represents a tool that visualizes selected events that were recorded by a virtual collaborative environment in chronological order based on users' interaction with the objects created. Figure 1 describes the integration of TLBA into a virtual collaborative environment.

Central elements of the TLBA are middleware services for logging purposes on the one and an execution engine for specified queries on the other hand. Specified queries are retrieved from a user interface based on user requirements or expectations. Collected events are stored in a separate database called awareness repository, in order to provide necessary historical data for time projections of performed user actions or whole practices.

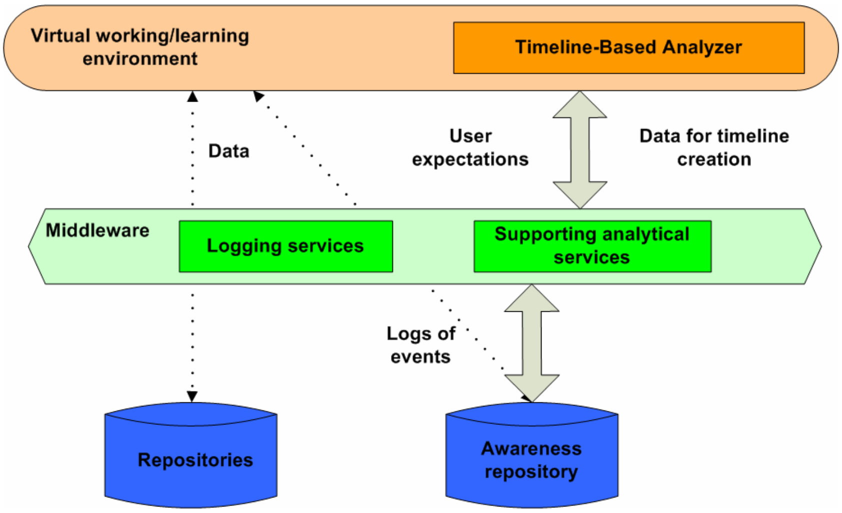


Figure 1: Architecture of TLBA integration with related supporting services.

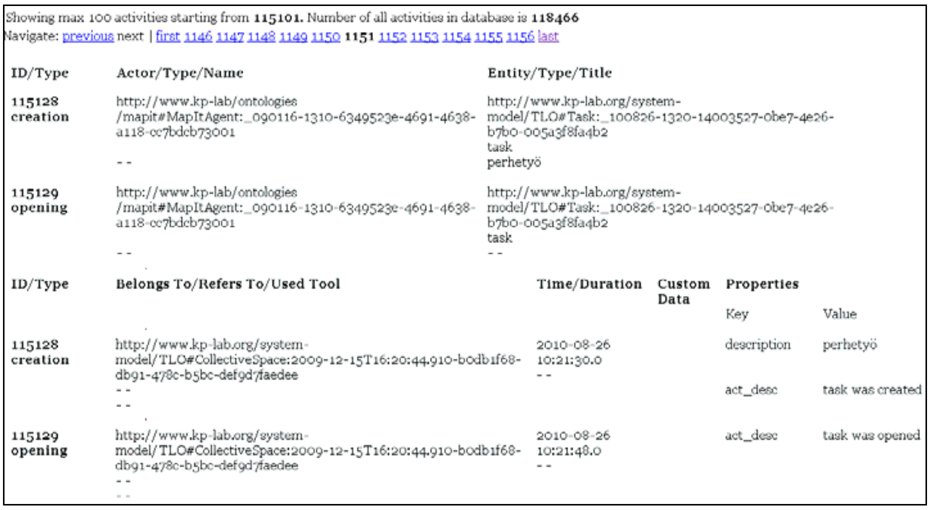
The TLBA allows describing and storing of external events, which cannot be recorded by the collaborative environment and enables defining and searching for 'patterns' in the logs recorded. All events are stored in the awareness repository in a predefined format of log. This generic format was designed to provide complex information for analytical purposes and can be adapted to new, specific requirements by adding new parameters or removing some of the existing ones. While the TLBA, as described in this paper, is currently integrated and tested within the KP-Lab System [Markkanen et al. 2008] [Lakkala et al. 2009], the format of log was also tested through several experiments with other collaborative systems such as Moodle or Claroline [Babič et al. 2010].

3.1 Log of events

The proposed log format contains 12 parameters that describe each event in details:

[ID, Type, Actor, Actor Type, Actor Name, Entity, Entity Type, Entity Title, Belongs to, Time, Custom properties, Custom data]

Figure 2 presents a simple example of existing logs, in this case two events are described, one creation and one opening action.



Showing max 100 activities starting from **115101.** Number of all activities in database is **118466**
Navigate: previous next | first 1146 1147 1148 1149 1150 **1151** 1152 1153 1154 1155 1156 last

ID/Type	Actor/Type/Name	Entity/Type/Title
115128 **creation**	http://www.kp-lab/ontologies /mapit#MapItAgent:_090116-1310-6349523e-4691-4638-a118-cc7bdcb73001 - -	http://www.kp-lab.org/system-model/TLO#Task:_100826-1320-14003527-0be7-4e26-b7b0-005a3f8fa4b2 task perhetyö
115129 **opening**	http://www.kp-lab/ontologies /mapit#MapItAgent:_090116-1310-6349523e-4691-4638-a118-cc7bdcb73001 - -	http://www.kp-lab.org/system-model/TLO#Task:_100826-1320-14003527-0be7-4e26-b7b0-005a3f8fa4b2 task - -

ID/Type	Belongs To/Refers To/Used Tool	Time/Duration	Custom Data	Properties	
				Key	Value
115128 **creation**	http://www.kp-lab.org/system-model/TLO#CollectiveSpace:2009-12-15T16:20:44.910-b0db1f68-db91-478c-b5bc-def9d7faedee - - - -	2010-08-26 10:21:30.0 - -		description act_desc	perhetyö task was created
115129 **opening**	http://www.kp-lab.org/system-model/TLO#CollectiveSpace:2009-12-15T16:20:44.910-b0db1f68-db91-478c-b5bc-def9d7faedee - - - -	2010-08-26 10:21:48.0 - -		act_desc	task was opened

Figure 2: A simple example of the log entry

- *ID* – unique identifier of the log entry;
- *Type* – a type of performed actions, e.g. creation, modification, deletion, etc;
- *Actor* – unique identifier of actor that performed a given event;
- *Actor Type* – user role that is delegated according to a relevant part of the user environment;
- *Actor Name* – user name obtained from user management based on his system logging information;
- *Entity* – unique identifier of the shared object that motivates given event;
- *Entity Type* – type of shared objects, e.g. task, document, link, wiki page;
- *Entity Title* – concrete title of related shared objects
- *Belongs to* – unique identifier of relevant part of user environment where this event was performed;

- *Time* – time when the event was logged into database (represented in the following format: year-month-day HH:MM:SS);
- *Custom data and properties* – these parameters are used in a situation when end-user application will store some properties or data that are typical for it.

These logs are stored in a separate repository implemented in MySQL relational database in order to provide scalable and responsive solution. By the time of writing the repository contains more than a hundred thousand logs from KP-Lab System that represents a main testing environment for the proposed services. This system has offered collaborative environment for several pilot cases that are held in different countries as Finland, Austria (described in more detail in chapter 5), The Netherlands, Israel, Sweden and Norway. These cases are mainly oriented to learning practices, but two of them are organized within a Swedish hospital in order to analyze implemented working procedures. The proposed patterns can be used for their improvement and more effective realization.

3.2 TLBA features

The main functionalities provided by the TLBA are the following:

- Sequences of performed events in chronological order are visualized via defined (one or more parallel) timeline(s). By placing events, i.e. user's actions carried out with the virtual collaborative environment onto timelines, TLBA users get a historical overview of the activities that were performed in a collaborative space and are able to discern who performed those actions.
- Visualization of all interactions and relations between selected relevant elements based on user requirements. By linking together events involving the same object, TLBA supports objects' path trajectories both in terms of the intensity of manipulation and its manifestation in users' actions. All available trajectories are placed into the view so that users can explore and highlight events that are of particular interest to them. Object trajectory path, actor created objects (with their path trajectories) along with events or on-fly mouse time stamp are possible ways of focusing user's interest in some object.
- Filtering upon type of object or type of activity is possible within the view to ease users' perception of data shown along with the ability to comment upon event or group of events, it brings the opportunity to share thoughts about performed activities.
- The important functionality of this tool is the possibility to define and store interesting patterns (pattern is a suitably generalised set of selected events or elements) from the timeline. These patterns are well formalized projections of interesting practices or parts of the whole evolution process.
- Basic timeline visualization consists of automatically collected events that are performed in the monitored collaborative system. In some situations, it is necessary to include elements called external events (performed outside monitored system) that are relevant to analyzed practices and this operation

is performed manually by the user. This functionality is also supported by the proposed application.

Design and development of these functionalities were lead by the first design goal, where openness of shown data (means to filter, highlight, comment, etc.) brings new possibilities to infer information and make sense of the collected data. As it is outlined in the second design goal, TLBA users are able to add external events to the timelines both before object creation in collaborative environment to indicate that some outside event has taken place which triggered this creation, and to add external events that happened during the object's lifetime. TLBA encourages users to design and share their own analytic queries by building, searching and sharing event 'patterns', by which the third design goal is met.

4 Patterns

The proposed patterns usually represent sequences of events that have critical impact on the overall activity. Such kind of patterns might also represent conceptually interesting practices that emerged within a particular context – either being positive (something like good practices), or negative (bad practices).

Collaborative activities realized within a suitable virtual user environment cannot be fully described by some well-defined, rigid process structure. They will change in time depending on actual conditions, used procedures or approaches. As the users practices evolve in time, it is necessary to provide a dynamic environment for their modelling and realization. Especially in the case of learning processes, it is difficult to create a stable process model that will be used from the starting point to the end. The traditional approach in this case represents the collection of all possible learning scenarios as templates for future exploitation like the ITIL[2] library that contains best practices for IT services. The other approach is covered by the IMS Learning Design (IMS LD) specification[3] that offers flexible language for various learning scenarios definition based on comparison of many pedagogical approaches and related learning activities. Our approach can be understood as a combination of the two approaches mentioned above, i.e. users have the possibility to create their own learning scenarios or paths within some predefined elements represented by ontologies. This issue is covered by another application called KPS (Knowledge Process Service) [Babič and Wagner 2007], which is an integrated part of the KP-Lab System. The second aspect similar to IMS LD is represented by patterns that offer possibilities to identify interesting practices that can be used for improvements and possible building of library of best practices.

Traditional approaches to the process modelling assume that the process can be clearly defined. However, the processes we try to describe are complex, often unique, ill defined and not easily formulated. Applying methods from traditional branches of process modelling would not lead to the better understanding of such processes. Models of these processes must be able to cover ill-defined cases and to dynamically

[2] http://www.itil-officialsite.com/home/home.asp

[3] http://www.imsglobal.org/learningdesign/index.html

build, modify and customize the description of the process. This however does not solve the problem of process discovery and identification. It is not possible to explicitly define a process where its instances are different in some parts and are always unique.

The uniqueness of each instance of knowledge practice was one of the motivations for our solution. To cope with this problem, we have designed a structure called pattern in order to generalize this type of processes. These patterns would not completely describe the knowledge practices as such, but they are able to formally and explicitly define at least some of their parts. Formal pattern description is used for searching occurrences of such practices in logged data, representing performed practices/activities/processes in virtual user environment. The goal of this search is to discover other occurrences of the defined pattern, which serves as a supporting analytical feature for users, teachers or researchers in order to analyze and comprehend events in the virtual environment.

The patterns in TLBA can be defined either from scratch, or based on any subset of events presented on the timeline in TLBA with the possibility to relax some of the attributes of selected events, stating in such a way a set of constraints. The constraints for defining a pattern (which have direct implications on searching for similar patterns in the history) can take one of the following alternative forms:

- Equality or inequality of properties of different events (e.g. different users performing event 1 and event 2, the same user performing events 2 and 3)
- Multiple occurrence of events (e.g. at least 2 comments have to be posted, for example by any user).
- Sequence of events (in given order e.g. first event, second event …).
- Specification of a timeframe between events (e.g. there should be a comment at least 48 hours after the creation of a document).

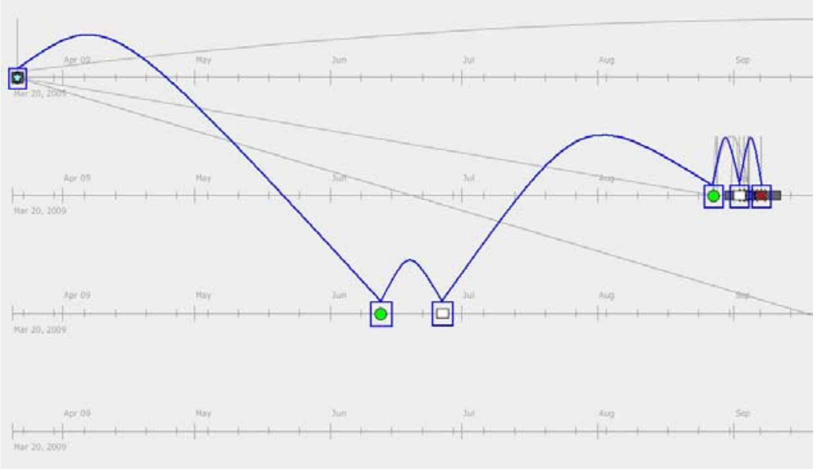


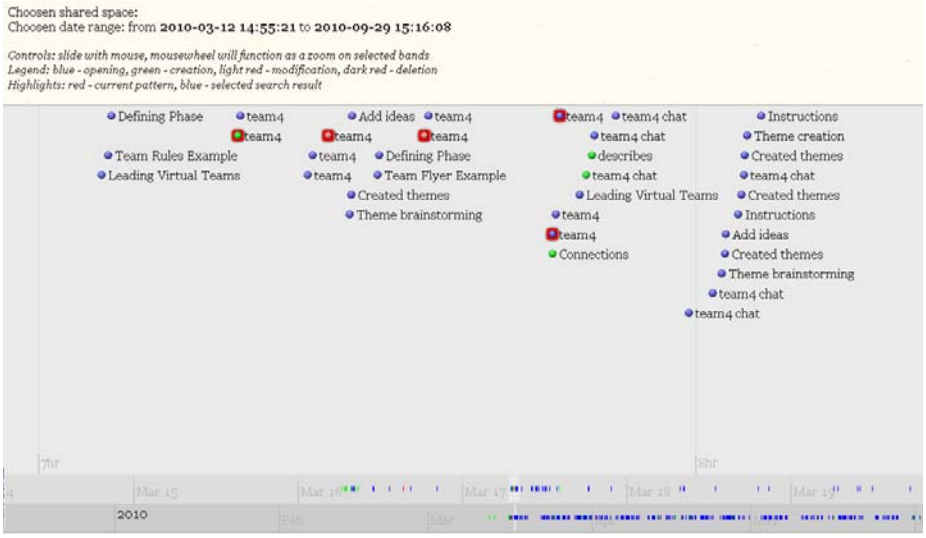
Figure 3: Pattern definition through TLBA – the first possibility

4.1 Pattern representation

Formal representation of the proposed patterns can be extended e.g. with the following requirements, which could be provided by patterns service:

- Generalization of the examined user actions or their parameters
- Ability to define relationships between actions on the level of relationships between their generalized parameters
- Beyond simple sequence, ability to define multiplicity and intervals between actions
- Ability to define weights of importance for the parts of a pattern, with the intention of more accurate pattern definition and matching

We have two possible visualization techniques for patterns definition - see Figure 3 and Figure 4. The pattern elements of a current pattern are highlighted. The whole process is interactive and iterative, where user interface helps users to express their intentions by providing friendly environment to define, customize and apply patterns.



Choosen shared space:
Choosen date range: from **2010-03-12 14:55:21** to **2010-09-29 15:16:08**

Controls: slide with mouse, mousewheel will function as a zoom on selected bands
Legend: blue - opening, green - creation, light red - modification, dark red - deletion
Highlights: red - current pattern, blue - selected search result

Figure 4: Pattern definition through TLBA – the second possibility

Patterns are manually selected and customized subsets of actions from the awareness repository. They are formalized as a sequence of activities, a list of pattern elements. Each pattern element represents one generalized event, which is essentially a list of key-value pairs. User can specify the element based on any of the events properties (see chapter 3.1), including custom ones. In these key-value pairs, the user specifies which parts of the generalized event are important and which should be generalized. The user can specify the element based on any of the events properties (see chapter 2.1), including custom ones. Example of pattern element:

```
{"Actor" = "http://...#MapItAgent:_080315-b5865f",
"Entity Type" = "content item",
"Type" = "creation "}
```

This pattern element represents any event that is about a creation of (any) content item object performed by actor identified by a given ID.

4.2 Pattern service implementation

Pattern discovery service finds the matches of the given pattern within the specified time range, and within the actions by a specified group of users, if given. The service returns a list of matches, each represented as an array of events, comprising the given pattern.

The current version of pattern discovery service is implemented in the emerging lisp language called Clojure[4]. This dynamic programming language for JVM realises a functional approach to the programming and usage of immutable data structures. Its unique approach to handling the structured data allows an easy interoperability with data stored in relational databases. In Clojure, collections can be generalized into sequences, for which most of the operations provide lazy evaluation.

The log data is stored in the MySQL repository. Columns in the SQL table are structured according to the event structure described in chapter 3.1. Moreover, one table is dedicated to the custom properties. For each pattern element, pattern discovery service constructs an SQL query in order to find matching events in the log.

As the pattern is a sequence of such pattern elements, the resulting matches produce a tree of matches, in which the results are in the leaf nodes with a given depth (number of pattern elements). A non-informed depth search is performed, collecting results on the way. The collected results are then sent back to the user interface, which displays the matching result to the user.

The lazy evaluation, easily achieved in Clojure, realizes only those parts of the result tree, which are actually used. This subsequently lessens the number of queries sent to the MySQL database, dramatically speeding up the whole matching process, if the user is interested only in a small number of results.

Full source code for the pattern discovery service can be found in the project code repository[5].

5 Case study

The aim of this case study is to describe how timeline based visualizations can be used to foster collaborative reflection in student project teams and how this process might be enhanced by the use of the Timeline-Based Analyzer. This case study is based on a collaborative retrospective carried out in connection to the "Project eModeration", a compulsory first-year course at the University of Applied Sciences

[4] http://clojure.blip.tv/file/982823/

[5] http://kplab.tuke.sk/trac/browser/hpa/trunk/src/clj/hpa

Upper Austria[6]. The collaborative retrospectives were facilitated by a research assistant and took place at the end of the course. The pedagogical intent of the retrospectives was to give students an opportunity to reflect on their past working practices and to derive lessons learned for future projects. The procedure described below represents previous iterations of the presented case study, without usage of suitable IT analytical solution. The motivation behind it is to show the advantages of the proposed application as a simple tool to obtain historical data, identify patterns of interesting behavior, and explore of a set of actualized events in a selected time interval.

The collaborative retrospectives lasted 60 minutes on average and were divided into six steps. Following a brief introduction on the purpose and agenda for the meeting (step 1), students were asked to recollect the main activities relevant to their project and note them down on a printed timeline individually (step 2), see Figure 5.

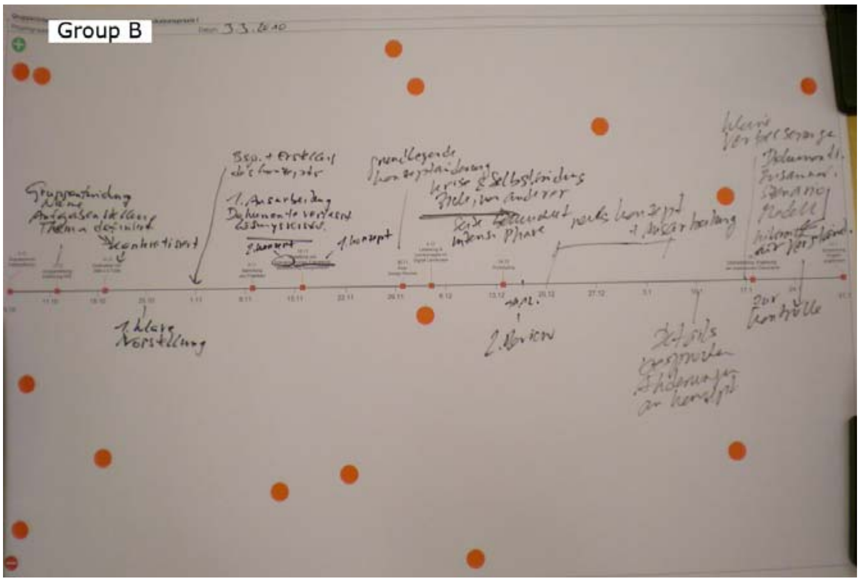


Figure 5: Paper-based timeline created during collaborative retrospective

Course meetings and short descriptions of related assignments had been added to the timeline beforehand to provide some anchoring point for the students. The students were then asked to explain the activities they had noted to the group, while the facilitator made notes on these events on large-scale printout of the same timeline (step 3). Once all activities were added and organized chronologically, the students were asked to return to their individual printouts and to mark those events they perceived as having an impact (positive or negative) on their project. Besides adding short notes on the event itself, they were also asked to rate whether the perceived impact was positive, negative or neutral (step 4). Afterwards students were asked to

[6] http://www.fh-ooe.at

make visible the events they deemed critical by adding a marker to the shared timeline, whereby the position was meant to indicate both time and perceived impact. The students were then asked to inspect the shared timeline and discuss those events the students felt most relevant for the entire team. In this phase the facilitator encouraged participants to elaborate on divergent perspectives and/or to think about possible causes of these events and how they could be avoided/triggered in the future projects (step 5). In a short wrap-up phase students had time to add further comments or ask questions (step 6).

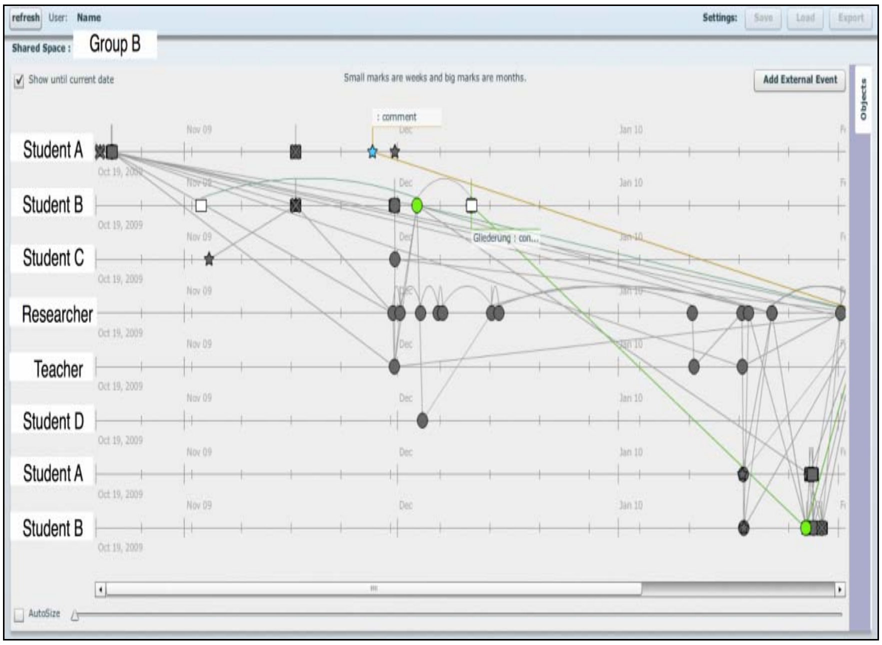
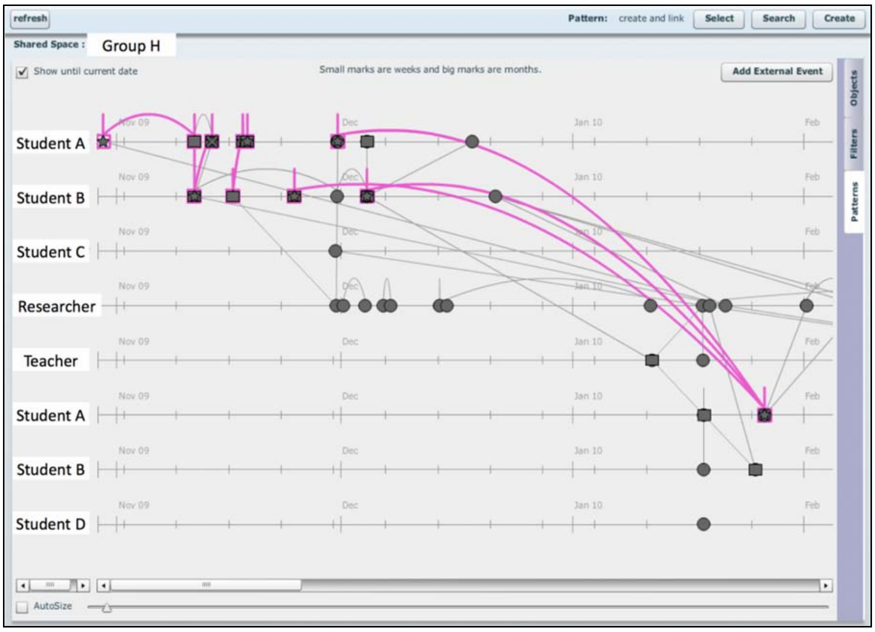


Figure 6: Screenshot of the Timeline-Based Analyzer displaying activities of a project team in a 15-week period for comparison with the paper-based timeline.

Even though the collaborative retrospectives turned out to be quite productive and were assessed positively by the students, the case study also revealed some limitations of the current method. In particular, the paper-based format only allowed providing students with the most general events that applied to all teams, i.e. the course meetings. Consequently students had to recollect most events and their chronology from memory, providing only limited input for the exploration and reflection on particular working practices. In comparison, tools such as the TLBA provide specific information based on the teams' actual work processes. Figure 6 provides a screenshot of the TLBA visualizing students' and teachers' activities within a collaborative working environment recorded over a period of 15 weeks. As a consequence, less time is needed to reconstruct the work processes and more efforts can be spent on the actual analysis of critical events.



Pattern: create and link

Shared Space : Group H

☑ Show until current date

Small marks are weeks and big marks are months.

□ AutoSize

Figure 7: Screenshot of the Timeline-Based Analyzer displaying matches of a simple create-and-link pattern.

While these kind of timelines often become quite messy when teams collaborate intensively, the possibility to define and search for patterns provides an important instrument to explore underlying structures in the overall flow of events. Figure 7 shows the matches of a simple create-and-link-pattern based on the logs of another project team. The create-and-link-pattern includes two events, (a) the creation of a content item and (b) the subsequent linking of this content item with another item in the shared workspace. The 27 matches are marked in magenta. Even though the create-and-link-pattern is quite simple, it is also instructive as it gives an idea of whether new resources are just added to the shared workspace or if they are also organized somehow, trying to add to the body of shared resources in some systematic manner. Figure 7 seems to indicate that this team used links among content items differently at different stages of their project. While in the beginning resources were usually linked directly after they had been added, resulting in spikes in the screenshot, the links created in the second half of the project refer to items created several weeks before. Even though the results provided by the TLBA are inconclusive in themselves, they can provide important input for the retrospective discussions with the teams involved.

Finally, the retrospective sessions with paper-based timelines revealed that the mixture of individual and collaborative reflection is quite important to give room for divergent opinions and to avoid premature consensus. Sitting around a table and working on a set of printouts participants have been quite tempted to start group discussion instead of focusing on their own ideas. Working with digital media such as the TLBA might make it easier for participants to focus on their own ideas while still allowing to easily share the results with others. First tests with the TLBA revealed that the visual representation is accessible and comprehensible for students and provides meaningful input for the kind of exercise described here.

6 Conclusions

In summary, mirroring tools in support of collaborative reflection, not only have to provide means for the faithful and comprehensible representation of data collected but also for sustained and collaborative exploration and interpretation.

As interesting practices or activities do not have to occur frequently or regularly, typical data mining approaches (inductive learning techniques), where frequent patterns are to be discovered, appear to be of limited value. Therefore the idea of providing suitable timeline-based visualization of performed practices (with the possibility to highlight and comment on particular events as well as to add external events) and means for manual definition of patterns within this framework has been proposed.

TLBA offers chronological overview of performed actions into the user interface, which allows seeing and exploring what kind of activities were carried out in relation to which documents or artefacts. This overview offers different analytic perspectives such as the evaluation of individual practices and their contribution to overall progress; comparison of individual practices or group activities; identification of interesting sequences of actions that can be marked as ’good‘ or ‘bad’ practice for future use; visualization of shared objects evolution processes, etc.

The user-defined patterns provide an important instrument in the evaluation of collaborative processes as they unfold in time as well as in the identification of critical events in these processes. These critical events can have positive or negative nature, but have a strong impact on the examined processes in both cases. This fact was the main motivation for our solution that offers features for manual pattern definition through created timeline-based visualization that represents related users’ actions in virtual user environment.

The actual version of TLBA is available within the KP-Lab System [Markkanen et al. 2008] [Lakkala et al. 2009] and is ready for final usability evaluation with the aim to finish the whole implementation phase with a stable prototype available for various research purposes.

Acknowledgements

The work presented in this paper was supported by: European Commission DG INFSO under the IST program, contract No. 27490 (40%); the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/0042/10 (10%). 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 (50%).

The KP-Lab Integrated Project is sponsored under the 6th EU Framework Programme for Research and Development. The authors are solely responsible for the content of this article. It does not represent the opinion of the KP-Lab consortium or the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

[Aalst et al. 2007] van der Aalst, W. M. P. et al.: “ProM 4.0: Comprehensive Support for Real Process Analysis“; In: Petri Nets and Other Models of Concurrency – ICATPN (2007), Lecture Notes in Computer Science, 4546/2007, 484-494.

[Aalst et al. 2009] van der Aalst, W.M.P. et al.: “ProM. The Process Mining Toolkit“; In Proceedings of the BPM 2009 Demonstration Track, 489 of CEUR-WS.org, Ulm (2009), Germany.

[Avouris et al. 2002] Avouris, N.M., Dimitracopoulou A., Komis, V., Fidas, C.: “OCAF: An Object-oriented Model of Analysis of Collaborative Problem Solving“; Proceedings of CSCL 2002, Colorado (2002), Hillsdale: Erlbaum, 92-101.

[Avouris et al. 2007] Avouris, N., Fiotakis, G., Kahrimanis, G., Margaritis, M., Komis, V.: “Beyond Logging of Fingertip Actions: Analysis of Collaborative Learning Using Multiple Sources of Data“; Journal of Interactive Learning Research, 18, 2 (2007), 231-250.

[Babič et al. 2010] 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, Slovensko (2010), IEEE, 109-112.

[Babič and Wagner 2007] Babič, F., Wagner, J.: “Modeling of knowledge creation processes based on Activity theory“; In. Proc. of the 1st Workshop on Intelligent and Knowledge oriented Technologies, WIKT 2006, Bratislava, Slovakia (2006), 131-134, ISBN 978-80-969202-5-9

[Baker 2010] Baker, R.: “Data Mining For Education“; In: B. McGaw, P. Peterson, R.S.J.D. Baker (eds.), International Encyclopedia of Education (3rd edition), Oxford, UK: Elsevier, 7 (2010), 112-118, ISBN 978-0-08-044894-7.

[Baker and Yacef 2010] Baker, R., Yacef, K.: “The State of Educational Data Mining in 2009: A Review and Future Visions“; Journal of Educational Data Mining, 1, 1 (2010), 3-17.

[Bolfíková et al. 2009] Bolfíková, E., Hrehová, D., Chrenová, J.: “Analysis of satisfaction with a participation in organizations“; In: Serbian Journal of Management, 4, 1 (2009), 105-115.

[Bartaloš and Bieliková 2009] Bartaloš,P., Bieliková, M.: “Semantic web service composition framework based on parallel processing“; 2009 IEEE Conference on Commerce and Enterprise Computing, CEC 2009, 495-498, ISBN 978-0-7695-3755-9.

[Gomez-Aguilar et al. 2009] Gomez-Aguilar, D. A., Theron, R., Garcia-Penalvo, F. J.: “Semantic Spiral Timeline as a Support for E-learning“; Journal of Universal Computer Science, 15, 7 (2009), 1526-1545.

[Habala et al. 2009] Habala, O., Paralič, M., Bartaloš, P., Rozinajová, V.: “Semantically-aided Data-aware Service Workflow Composition“; In SOFSEM 2009: Theory and Practice of Computer Science, Springer (2009), LNCS 5404, 317-328, ISSN 0302-9743.

[Hardless and Nulden 1999] Hardless, C., Nulden, U.: “Visualizing Learning Activities to Support Tutors“; In Proc. CHI '99 Extended Abstracts on Human Factors in Computing Systems, CHI '99, 312-313, ISBN 1-58113-158-5.

[Kahrimanis et al. 2006] Kahrimanis, G., Papasalouros, A., Avouris, N., Retalis, S.: “A Model for Interoperability in Computer Supported Collaborative Learning“; Proc. of the Sixth International Conference on Advanced Learning Technologies (ICALT'06), IEEE Computer Society, ISBN 0-7695-2632-2.

[Markkanen et al. 2008] Markkanen, H., Holi, M., Benmergui, L., Bauters, M., Richter, C.: “The Knowledge Practices Environment: a Virtual Environment for Collaborative Knowledge Creation and Work around Shared Artefacts“; In J.Luca & E.R. Weippl (Eds.), Proceedings of ED-Media 2008, World Conference on Educational Media, Hypermedia and Telecommunications, Vienna (2008), Chesapeake, VA: AACE, 5035-5040

[Lakkala et al. 2009] Lakkala, M., Paavola, S., Kosonen, K., Muukkonen, H., Bauters, M., Markkanen, H.: “Main functionalities of the Knowledge Practices Environment (KPE) affording knowledge creation practices in education“; In C. O'Malley, D. Suthers, P. Reimann, & A. Dimitracopoulou (Eds.), Computer supported collaborative learning practices: CSCL2009 Conference Proceedings, 297-306, ISBN 978-1-4092-8598-4.

[Paavola and Hakkarainen 2009] Paavola, S. Hakkarainen, K.: “From Meaning Making to Joint Construction of Knowledge Practices and Artefacts – A Trialogical Approach to CSCL“; In C. O'Malley, D. Suthers, P. Reimann, & A. Dimitracopoulou (Eds.), Computer Supported Collaborative Learning Practices: CSCL2009 Conference Proceedings, 83-92, ISBN 978-1-4092-8598-4.

[Perera, et al. 2009] 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 (2009), 759-772, ISSN: 1041-4347.

[Sabol et al. 2010] Sabol, T., Furdík, K., Mach, M.: “Employing Semantic Technologies for the Orchestration of Government Services“; In: T. Vitvar, V. Peristeas, K. Tarabanis (eds.), Semanic Technologies for e-Government, Springer-Verlag, Berlin Heidelberg (2010), 47-74, ISBN 978-3-642-03506-7.

[Zaïane and Luo 2001] Zaïane, O.R., Luo, J.: “Towards Evaluating Learners’ Behaviour in a Web-based Distance Learning Environment“; 2nd IEEE international conference on advanced learning technologies (ICALT’01), Madison (2001), WI, 357-360, ISBN 0-7695-1013-2.