

Process-driven Approaches to Knowledge Transformation

Ján Paralič¹, František Babič¹, Marek Paralič²

¹ Department of Cybernetics and Artificial Intelligence,

² Department of Computers and Informatics,

Faculty of Electrical Engineering and Informatics,

Technical University Košice

Letná 9, 04200 Košice, Slovakia

E-mail: jan.paralic@tuke.sk, frantisek.babic@tuke.sk, marek.paralic@tuke.sk

Abstract: In this paper we focus on specific approaches to knowledge transformation within the educational domain. Our approaches can be briefly characterized as process-driven, because the core concepts are educational processes and semantic representations of them. In this paper we present two alternative ways of using process models for knowledge transfer in educational domain. First one is deductive approach, or top-down approach, where knowledge is captured from the very beginning and continuously upgraded with the repeated runs of educational processes. The second one is inductive approach, or bottom-up approach, where process logs are analyzed with the aim to derive useful knowledge patterns. We build on our experiences from more research and educational projects, where we have designed and developed information systems and services supporting these types of knowledge transformation.

Keywords: knowledge transformation; process model; ontology; process mining

1 Introduction

Knowledge transformation has been identified in [7] as the core approach when new knowledge is created. Nonaka and Takeuchi in their book identify four basic knowledge creation processes, whereas all of them are based on some form of transformation between two basic types of knowledge: explicit and tacit. The concept of tacit knowledge was first introduced by Polanyi in [10]. Nice article describing broadly area of knowledge management in the higher education context appeared in this journal [24], but we go much deeper into the educational processes proposing two different approaches to knowledge transformation.

The importance of process context is understood especially in the business area. Raghu and Vinze in [11] claim that the core of knowledge is defined in the

business process context and the knowledge is managed within the cyclical set of phases: knowledge storage and retrieval, knowledge sharing and knowledge synthesis. There is no doubt that process models are very useful and powerful means for knowledge capture, analysis and improvement of existing business processes. The question remains, to what extent created process models may be operationalized. Today's technologies (e.g. semantic technologies, workflows, various execution languages) provide quite straightforward tools for execution of well defined process models, which is often used in business [19], [20] but also in public sector [12], [3] where semantics is used to support automation of business processes. But what is the situation in educational processes? This article provides some possible answers on this question. In last seven years we had the chance to address the issues of knowledge creation and transformation in a couple of research and development projects. We designed and applied two different approaches for knowledge transformation, each of them being more suitable for different types of educational processes.

First approach, which we call deductive or top-down, is more suitable for relatively stable processes and is quite similar to the approaches used in business or public sectors mentioned above. But we also show how our operational environment provides enough flexibility to cope with process changes and improvements which are very typical in educational domain. Second approach, called inductive or bottom-up is aimed for very loosely-structured processes, typically of collaborative nature. Here the knowledge transformation is more about discovery of interesting knowledge patterns, using the information logged from the supporting educational (collaborative) information systems.

The rest of the article is organized as follows. We first present current state of the art in relevant areas in Section 2. Sections 3 and 4 create the core of this contribution. Our deductive approach to knowledge transformation is described in Section 3 and the inductive approach in Section 4. The paper concludes with a summary of main contributions of proposed approaches to knowledge transformation and recommendation for their usage.

2 Related Work

Semantic technologies have been used in various technology enhanced learning projects with different aims. Semantic models themselves are very good form of knowledge representation, which may be used also to achieve some level of operationalisation in educational processes. One of the first projects using this approach was PALETTE [21], where the solution is based on an expandable set of electronic services. Integration and interoperability was achieved by so called Cross Awareness Knowledge Base providing synchronization and searching functionalities for underlying services. Ontology models services, resources and

their actions. Similar recent approaches are e.g. [22] and [23]. Ontologies, methodologies, and other tools supporting technology enhanced learning applications were elaborated also within Kaleidoscope, a Network of Excellence project¹. Semantic knowledge middleware infrastructure to support knowledge creation processes (also known as triological learning), was designed and developed in the integrated FP6 project KP-Lab [6]. Our approach described in the following Section 3 is to some extent similar, however, we handle the service integration by means of a more transparent and flexible approach of semantic process models.

Learning scenarios provide a basic structure (very similar to process-driven models) to support educational processes [4], [25]. The main challenge is to create an effective scenario reflecting actual education conditions, course's objectives, student's experiences and knowledge, teacher's expectations, available technical tools, etc. With the goal to help teachers in creating collaborative learning scenarios an intelligent authoring tool CHOCOLATO was implemented [4] that use ontologies for representation of relevant knowledge about various learning strategies and practices. Another ontological framework to support collaboration and interaction analysis is described in [13], which provides also means to automatic analyses of performed processes in groupware systems. Collaboration and interaction analysis represents a relatively new research area bringing new methods to investigate how users interact in virtual collaborative environment, supporting in such a way the knowledge transformation processes based on loosely structured collaborative processes or external examination of performed collaborative processes [14]. This approach was used also in [15] to design methods for enabling observation in CSCL (Computer-supported collaborative learning) environment in order to collect data for complex analyses of performed collaborative processes, focused on validity of used approaches to solve defined problems.

Different but interesting approach is described in [16]. Authors proposed a fuzzy expert system for evaluation of virtual collaboration and task implementation. This system is based on variables extracted from virtual collaborative system as fuzzy rules inputs that are further evaluated by predefined hierarchical fuzzy rules. The rules were created by experts with the objective to respect the subjective looks of involved experts and relative vague understanding of extracted characteristics. These fuzzy rules can be seen as a one possibility how to represent the transformed knowledge for loosely structured processes.

Another form of representation can be rules reflecting patterns of usage [9]. We present in Section 4 below here, what types of patterns can be used for knowledge transformation in loosely structured educational processes, how they can be defined and searched for.

¹ <http://www.noe-kaleidoscope.org>

3 Deductive Approach to Knowledge Transformation

Our deductive approach to knowledge transformation is based on carefully designed semantic models and their suitable operationalization, which is described in this section. In the center of deductive approach to knowledge transformation applied within the IT4KT – Information technology for knowledge transfer (ITMS project code: 26220220123) project are educational processes as they are conveyed during the university study. Altogether 11 courses and group of courses from the mathematical and computer science area at the Faculty of Electrical Engineering and Informatics, Technical University of Košice were analyzed. Small groups of teachers, process modelers and ontology engineers were created and crucial phases of the educational processes were formalized in the form of BPMN (Business Process Model and Notation) 2.0 process models. Created models represent an interesting combination of traditional learning scenarios and tacit teacher's knowledge acquired during previous instances of learning courses. This creation can be understood as complex transformation including all four knowledge creation processes from Nonaka and Takeuchi model:

- Externalization of tacit teacher's knowledge into relevant BPMN elements and relations between them.
- Combination of explicit knowledge representing identified traditional learning scenarios in the form of simple workflows.
- Internalization of collected explicit knowledge about various possible procedures, tools, methods, and data sources into teacher's tacit knowledge to create an effective structure (process model).
- Socialization representing a tacit to tacit knowledge transfer during face to face meetings of created groups.

By the selection of the modeled educational process phases beside importance from the knowledge transfer point of view, the perspective utilization of ICT was one of the most influential factors.

3.1 Modeling Particular Educational Processes

In the first phase, when the groups worked independently, more than 50 first and second level processes were designed. In order to begin the generalization process that could result in general schemes of knowledge transformation process, TEL (Technology-enhanced learning) ontology was applied to all process elements. TEL ontology created also within IT4KT project uses Activity, Actor/Agent, Role, Knowledge Artefact, Tool, Event and Condition elements. Relationships between these elements can be briefly characterized by the following sentence: the Actor of a given Role is using the Tool to create/manipulate a Knowledge Object within an Activity with specific characteristics given by Event and Condition concepts. More about the TEL ontology can be found in [8].

The following Table 1 summarizes the (by individual groups) identified processes and number of their occurrences either in the definition or usage form by the independent groups corresponding to a course or to a group of courses.

Table 1
Number of definitions and utilizations of identified process within modeled courses

Nr.	Name of the process	Nr. of def.	Nr. of usages
1.	Preparation of study materials for the course	4	3
2.	Generating and solving of tasks within and outside of the lab	3	4
3.	Generating, elaboration and evaluation of tests	5	1
4.	Automatic testing of programming tasks within LMS tests	5	1
5.	Lecture – including support of interactive materials	3	2
6.	Originality control of an assignment	1	4
7.	Correctness control of an assignment	1	4
8.	Management and control of assignments	3	1
9.	Controlled self-study	3	1
10.	Self-testing	2	2
11.	Support for semantic linking of lectures content, practical assignments and exam tests	1	3
12.	Introductory tests	1	3
13.	Support for creating lab materials based on practical programming assignments and their testing	3	0
14.	Exams – theoretical and practical, oral and written	2	1
15.	Questionnaires	2	1
16.	Management of individual and team based project	2	1
17.	Automatic update of university information system from LMS after finishing the test	1	2
18.	Utilization of on-line tools within interactive materials from LMS	2	0
19.	Design and publishing of scenarios	1	1
20.	Analysis of the incremental student's work at assignment	1	1
21.	Support for design and distribution of course packages containing the learning materials (teacher&student version)	1	1
22.	Support for design, maintenance and distribution of virtual engines specialized for given topic	1	0
23.	Collaborative commenting of scenarios	1	0
24.	Dynamic knowledge testing	1	0
25.	Management of programming assignments within LMS	1	0
26.	Creating of paper equivalents of electronic tests	1	0
27.	Integration of information systems involved in educational processes	0	1

3.2 Identification of Generic Processes

In the second phase all created processes were analyzed in detail and groups of processes were identified, in order to make the experiences, skills and practices of individual processes as generic as possible. Similarly the hierarchy of processes was established, because different groups by focusing on different aspects of the pedagogical process used different entry points to overall process structure. The results of the second phase are the following three generic processes, whereby the most of identified particular processes listed in Table 1 are covered within the generic process (those with higher number of definitions/usages were prioritized):

- A. Preparing study materials for the course
 - 1) Lecture – materials including the interactive ones together with integrated on-line tools
 - 2) Labs – the process of task and assignment generation as well as their solving during the labs and at home
 - 3) Support of semantic interconnection of lectures content, practical examples at labs and test questions at exams
 - 4) Controlled self-study
 - 5) Creation, publication and collaborative commenting of scenarios
 - 6) Questionnaires – semantically aided feed-back
 - 7) Support and distribution of packages with learning materials and virtual machines
- B. Support for student assignments
 - 1) Correctness control of the assignment solutions and detecting of solution plagiarism
 - 2) Assignment solutions storage and maintenance
 - 3) Support for individual and team based term projects
 - 4) Analyses of incremental work during the process of solving the assignment
 - 5) Support for material and test generation of practical labs based on annotated sample solution
- C. Generation, developing and evaluation of tests including programming tasks
 - 1) Entry tests
 - 2) Self-testing
 - 3) Automatic control of programming tasks inside and outside of an LMS
 - 4) Dynamic tests of practical skills (e.g. SQL and relational algebra)
 - 5) System support for programming assignment variations
 - 6) Paper equivalents of electronic tests

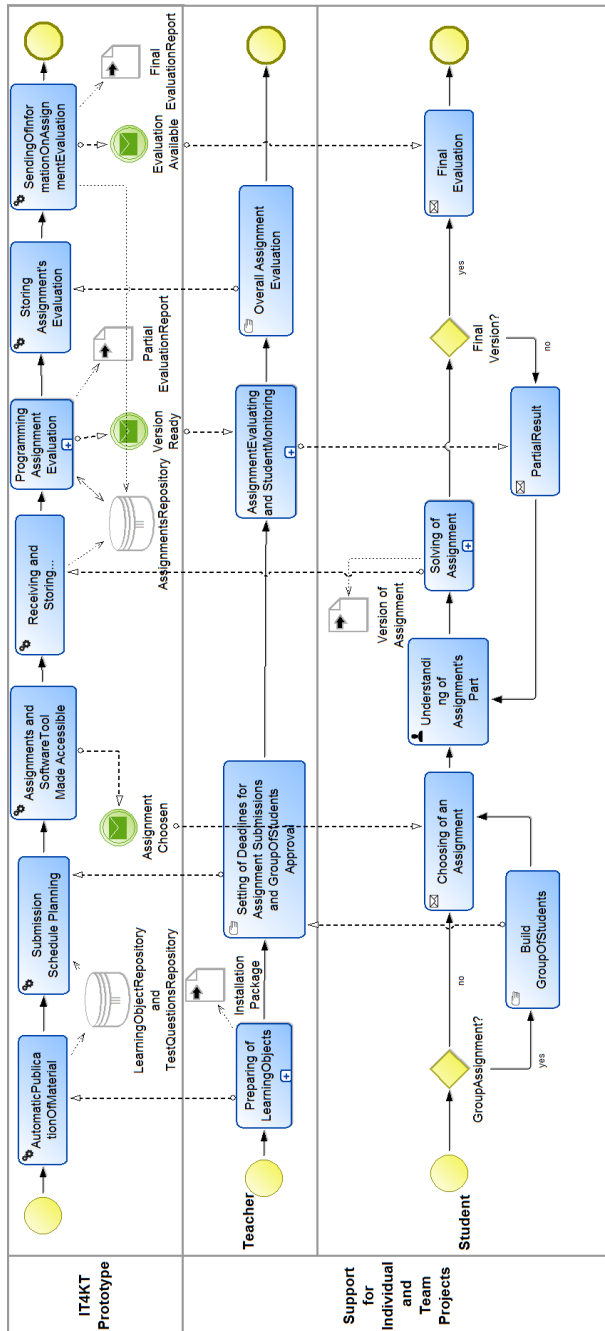


Figure 1

Top level process example - Support for Individual and Team Projects

Figure 1 depicts example of resulting generic process model at first level of modeling – Support for Individual and Team Projects (B3). It includes number of second level processes – e.g. Programming Assignments Evaluation or Preparing of Learning objects. All of them utilize concepts of TEL ontology [8] in order to describe an activity, repository or learning objects as input/output data.

3.3 Operationalization

Modeling of processes on one hand contributed to reveal tacit knowledge hidden behind individual teacher experiences and identify common services that could be supported by information and communication technologies (selected subset of identified services is shown in Table 2). On the other hand, the formalized form of processes as BPMN2.0 models that could be straightforward transformed into executable processes, contributed also to the architectural design of the IT4KT platform itself.

One way how to deal with such systems is to perceive them as process-driven service-oriented architecture (SOA) based systems. For such systems is important connecting process-oriented kernel with backend systems in flexible, scalable, maintainable and changeable way.

Table 2
Selected identified generic services which become electronic services

Generic process	Service
Preparing study materials for the course	Annotation of study materials
	Register of student profiles
	Evaluation and register of test results
	Generation of the list of practical exercises
	Verification of results of practical exercises
	Management of polls
Support for student assignments	Support for collection of assignments and projects
	Support for originality check
	Management of assignment registry
	Automatic system for validation of programming assignments
Generation, developing and evaluation of tests including programming tasks	Management of test bank, incl. programming tasks
	Evaluation of test task in LMS
	Evaluation of test task by external program
	Checking syntax of a program code fragment
	Statistical processing of test results
	Publishing of test results

In educational environment that we are focusing on, common backend systems are CMS systems (Content Management System) e.g. Joomla² or MediaWiki³ and LMS systems (Learning Management System) like Moodle⁴. In order to support different technologies that should be connected to common platform and build SOA solution, the architectural pattern Process Integration Architecture [2] can be applied. According to [2] the following tiers could be distinguished:

- *Macroflow Tier* – educational processes can be hosted by different macroflow engines – in our case they should be able to run the BPMN 2.0 processes (e.g. Activiti BPM platform)
- *Macroflow Integration Tier* – contains one integration adapter per each macroflow engine. It integrates the process activities with the technical functions provided by services.
- *Microflow Tier* – a number of services is provided as well as the support for service orchestration
- *Backend Integration Tier* – integration adapters for needed backend systems
- *Backend Systems Tier* – systems that perform functions needed for running of educational processes – e.g. CMS, LMS.

4 Inductive Approach to Knowledge Transformation

Some types of educational processes (e.g. collaborative processes [4], or knowledge creation processes [17]) cannot be fully described by well-defined static process model, as presented by the deductive approach above. They change in time based on actual conditions, changes in participant's list, used procedures or methods and this dynamic aspect represents a very challenging task for modelling an implementation techniques.

The aim of our inductive approach is to transfer the hidden knowledge in educational processes to the relevant users (learners, teachers, students) in suitable and easily understandable form. Such creative educational processes are usually supported by some suitable collaborative system offering various related end-user functionalities. In the KP-Lab project⁵, the whole virtual collaborative system was mediated by ontology to support objects' semantic representation and software interoperability across the middleware layer [1]. The process part of ontology contained the basic elements as e.g. process, task, milestone, subject, deliverable

² <http://www.joomla.org/>

³ <http://www.mediawiki.org/wiki/MediaWiki>

⁴ <https://moodle.org/>

⁵ www.kp-lab.org

and object creating a basic process model [5]. This structure can be extended by another elements from common KP-Lab ontology in order to model the complex educational processes with all characteristics [1].

Typical example of collaborative and loosely-structured educational process is work on team student assignment. The result of such an assignment can be understood as some new knowledge created in the process of knowledge transformation as combination of tacit knowledge of the students collaborating within the team and explicit knowledge represented by manuals, books, papers and other knowledge sources shared e.g. in some virtual space. Relevant activities are mediated by suitable information and communication solution, as students need collaborative space to share their inputs and contributions, to display their progress and to communicate and comment published versions of the assignment. The whole knowledge creation process consists of several phases and each of them requires interaction between participants based on their roles, experiences and theoretical background. Our inductive approach to knowledge transformation is based on tracing performed steps and visualizing them in suitable form (e.g. in form of some type of timeline) in order to provide historical overview in chronological order. Such visualization represents a historical projection of the realized educational process with all related objects (docs, tutorials, manuals, instructions, demos, etc.), involved subjects (students, teachers, instructors, etc.), created connections (between subjects, between objects or between both of these categories), etc. This type of projection can be used for collaborative analysis and reflexion to identify the key persons, important flows of information, interesting inputs for decisions or significant steps forward, etc.

We designed and implemented several necessary services to support the inductive approach, i.e. for logging, data management, extraction and visualization. Semi-automatic character of this approach can replace the often used manual methods of user behaviour evaluation that are much more time consuming and tedious for teachers or researchers. Reasons for that are e.g. necessary collection of all materials from students or laborious analysis of their communication channels, which causes difficulties by identification of the real involvement of each student. Suitable visualisation of automatically collected data with the possibility to define constraints based on users' needs provides easier approach, mainly in the case of large students' groups.

4.1 Data from Educational Processes

Source data recorded for historical projection represent actions/activities performed during educational processes. Each action or activity is monitored and logged in predefined format, e.g. if a user attaches a new tutorial to relevant task in order to help other students with this tasks; this action is logged and stored as new event in the log repository. Each event is described by specified parameters (log format) that provide complex data structure for visualization or analytical

purposes, i.e. timestamp, subject identifier, action type or related shared object. More details about used log format are presented in [9].

The implemented logging services were tested with three different collaborative systems to evaluate their adaptability and success in obtaining the required quality of data [18]. Moodle⁶ was selected as typical representative of LMS in the conditions of Slovak education; KPE⁷ as actual new software output of KP-Lab project offering an innovative look on knowledge creation processes called triological learning [6]; and Claroline⁸ as potentially interesting candidate for execution of the various learning processes. The experiments realized within Moodle covered design and implementation of a new web service responsible for cooperation with internal Moodle logging system to transfer event logs from internal Moodle repository. In the case of KPE, logging procedure was integrated on the middleware layer to obtain data from user environment in cooperation with monitoring services on the GUI (Graphical user interface) level. The last experiment was not successful because it would require changes in internal Claroline logging API (Application programming interface), which was not accessible to our project team. Despite this fact, based on the two successful results mentioned above we can conclude that the integration of our inductive knowledge transfer tools with another collaborative educational systems is possible and straightforward if suitable API of original system is available.

Once the data from a collaborative or other educational supporting information systems is available, our approach offers a set of tools to their analysis supporting knowledge transfer. Simple examination of collected historical data from overall, quantitative point of view, can be performed with supporting tool offering a basic summary overview, see Figure 2.

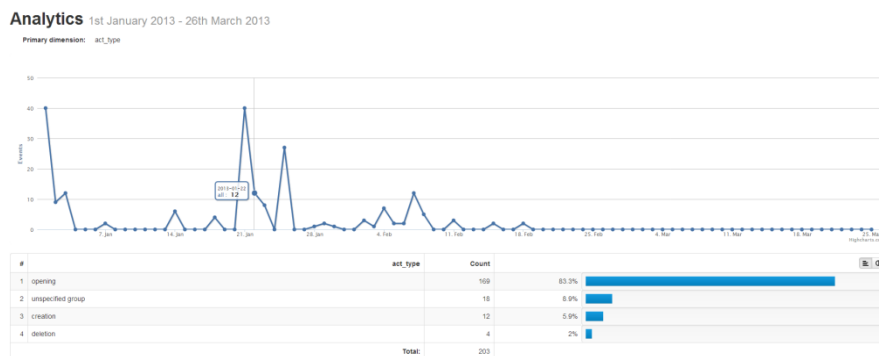


Figure 2

Example of simple summary visualization of collected historical data

⁶ <https://moodle.org/>

⁷ <http://www.kp-lab.org/tools/knowledge-practices-environment>

⁸ <http://www.claroline.net/>

4.2 Tool Supporting Knowledge Transformation

The Timeline-based Analyzer (TLBA) is an application that visualizes selected historical events in chronological order, which is a natural way how to represent a collaborative or knowledge creation processes. TLBA in such a way gives the possibility to focus on potentially interesting sequences of activities and to reflect on the existing practices and in such a way to support knowledge transformation from loosely-structured educational processes.

The main functionalities provided by the TLBA are the following:

- Timelines visualize sequences of performed events in chronological order. User can visualize more parallel timelines represented particular participants in the educational process. E.g. in case of team assignment each member of the team is represented by one timeline (see Figure 3).
- Visualised timelines contain all events from particular educational process (or its part). Authors of events are distinguished by particular parallel timelines, different types of events are distinguished by type of graphical icons used, e.g. star = modification or circle = opening. Consecutive events performed on the same knowledge object are connected with lines. In such a way all crucial information is placed into the view so users can explore and highlight events that are of particular interest to them.
- Basic timeline is constructed from automatically collected events which are stored in the event logs. If user needs to include important action/activity performed outside monitored virtual environment and relevant for investigated process, such an event can be inserted manually and stored in the log repository as so called external event and of course visualised on the timeline.
- Several supporting functions as commenting and filtering were implemented to support simple orientation and comprehensibility of visualized information for users.
- Reflexion over collected historical data is provided by definition of patterns representing a suitably generalised set of selected events or elements from timeline. These patterns are well formalized projections of interesting practices or parts of the whole educational processes (transformed knowledge).

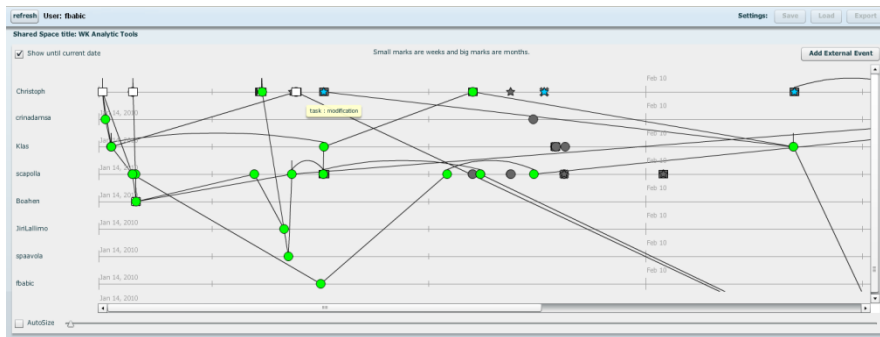


Figure 3
Example of timeline-based visualization

4.3 Knowledge Patterns

Knowledge patterns represent important mean to support knowledge transformation. It is a way how to identify interesting sequences of user actions during educational processes mediated by suitable information systems. The patterns usually describe situations that lead to some critical moments, which can mean a significant progress, discovery of new knowledge/method. However they can not only represent positive lessons learned, but on the other hand they also can reflect some negative lessons when a particular process was not successful or it finished untimely. Such kind of patterns may also conceptually represent interesting practices emerged within particular process or activity – either being positive (something like best practice), or negative (worst practices). The main issue in this case is the representation of the patterns and their identification.

Pattern can be understood as formalisation of captured tacit knowledge in the process realisation specified by user based on his experiences and own knowledge. Basically, it is some special type of knowledge transformation that connects two knowledge conversion processes from traditional model [7]: externalisation and combination. At first, user expresses her/his tacit knowledge (a kind of hypothesis) in order to verify a pattern that will be further evaluated over the collected historical data (externalisation). This pattern also represents an expected tacit knowledge hidden in investigated educational process or its part. If the subsequent evaluation of specified pattern will be successful, user verifies a new knowledge that can be further combined with existing ones to create the final transformed piece of knowledge (combination). Simple example of this transformation:

1. Teacher specified an assignment for students to solve some optimisation task and provided also necessary detailed documentation.
2. As an inspiration teacher provided to students also some examples of traditional methods.

3. One group of students solved its task successfully, but they used different approach as it was originally expected by the teacher.
4. Teacher wanted to evaluate their approach based on collected historical data representing all performed steps.
5. He visualised on timeline all activities of the specific group and looked for specific sequences of activities describing the critical sequence of actions that led to the successful result trying to generalise them.
6. Created generalized pattern was evaluated over the relevant data sample (all other groups, possibly also from previous years) and all its occurrences in previous processes were found and can be displayed for detailed inspection.
7. This complex visualization provided for teacher all necessary information to decide if performed approach can be labelled as “good practice” or “bad practice”. In the first case, teacher can update his set of possible procedures to solve this type of task for the future use. The second case indicates a potential cheating.

Based on this simple example we can propose a common workflow for our induction approach:

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. Construction of a pattern (hypothesis to be verified by its occurrences).
5. Performing search for a pattern occurrences in given data set.
6. Interpretation of results, iteration (*back to step 3 or 4*)

The pattern 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 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).

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 event's attributes. In these key-value pairs, user specifies which parts of the generalized event are important and which should be generalized. Simple example of pattern definition is presented below:

```
(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)
```

This pattern represents a case in which user “X” created a new document “Y” after reading an existing document “doc”, and then linked these two documents together. Defined pattern is matching with events stored in log repository to generate a searching tree in which obtained results are represented as leaf nodes at the lowest level of the tree, see Figure 4. In our 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. Depth-first search is used, with some optimizations which remembers environment (current parts 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).

Pattern discovery service is implemented in the Lisp language called Clojure. This dynamic programming language for JVM (Java virtual machine) provides functional approach to the programming and usage of immutable data structures. In Clojure, collections are generalized into the sequences, for which most of the operations provides lazy evaluation. For each pattern element, pattern discovery service constructs an SQL query in order to find matching events in the log.

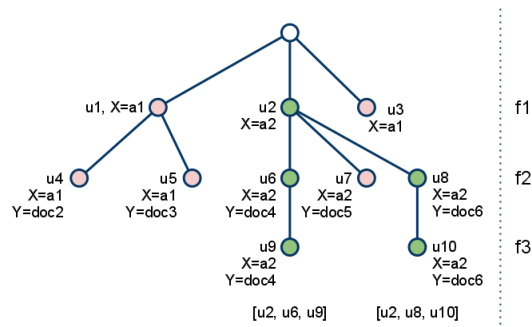


Figure 4

Resulting tree for pattern example

The lazy evaluation, easily achieved in Clojure, traverses 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 small number of results.

Proposed inductive approach was tested in real conditions within the KP-Lab project focused mainly on various educational knowledge creation processes. Figure 5 displays an example from Austrian pilot case visualizing students' and teachers' activities within a collaborative learning environment KPE 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.

Conclusions

In this article we described two alternative approaches to knowledge transformation in educational processes. Deductive approach is suitable for well structured processes, which is typically the case for more general processes like preparation of lectures, evaluation processes – tests or management of the whole lifecycle for student assignments. These processes are often used in many university courses and it is therefore efficient to design and implement supporting electronic services, which can be shared across various subjects. Proposed process-driven semantic approach makes it possible not only to model, but also support operationalization of selected parts of generic educational processes in form of electronic services.

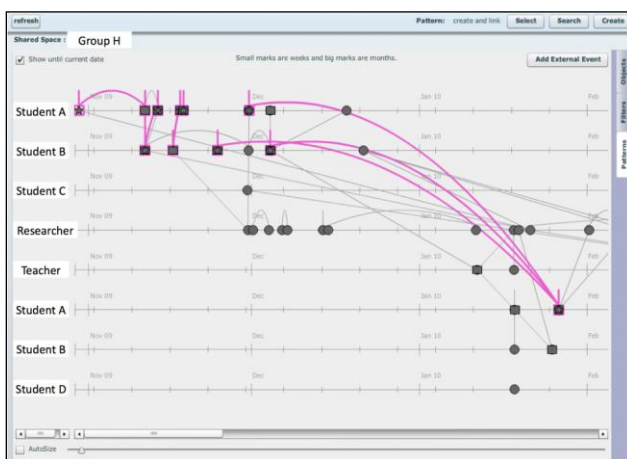


Figure 5

Screenshot of TBA displaying 27 matches of a defined simple pattern

Inductive approach is more suitable inside particular subjects, where educational processes are less structured and often highly specialised. These are e.g. various collaborative activities or knowledge creation processes. Nowadays even these

less structured processes can be traced to some extent thanks to the mediating information systems, which can generate logs of events. Proposed inductive approach provides formal as well as practical tools to represent interesting pieces of transformed knowledge in form of knowledge patterns.

Acknowledgement

This work was supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the grant No. 1/1147/12 (50%). It is also the result of the project implementation: IT4KT - Information technology for knowledge transfer (project number: 26220220123) supported by the Research & Development Operational Program funded by the ERDF (50%).

References

- [1] F. Babič, J. Wagner, J. Paralič: The Role of Ontologies in Collaborative Systems. SAMI 2008 6th IEEE International Symposium on Applied Machine Intelligence and Informatics - Proceedings, Herlany, Slovakia, January 2008, pp. 119-124
- [2] C. Hentrich, U. Zdun: A Pattern Language for Process Execution and Integration Design in Service-oriented Architectures. Transactions on Pattern Languages of Programming, Vol. 1, Springer, 2009, pp. 136-191
- [3] J. Hreňo, P. Bednár, K. Furdík, T. Sabol: Integration of Government Services Using Semantic Technologies. Journal of Theoretical and Applied Electronic Commerce Research, Vol. 6, Issue 1, April 2011, pp. 143-154
- [4] S. Isotania, R. Mizoguchi, S. Isotani, O. M. Capeli, N. Isotani, A. R. P. L. de Albuquerque, Ig. I. Bittencourt, P. Jaques: A Semantic Web-based Authoring Tool to Facilitate the Planning of Collaborative Learning Scenarios Compliant with Learning Theories. Computers & Education, Volume 63, April 2013, pp. 267-284
- [5] S. Jalonen, M. Lakkala, S. Paavola: Investigating Knowledge Creation Technology in an Engineering Course. Computers & Education, Volume: 57 (2011), Issue: 3, pp. 1930-1942
- [6] M. Lakkala, S. Paavola, K. Kosonen, H. Muukkonen, M. Bauters, H. Markkanen: Main Functionalities of the Knowledge Practices Environment (KPE) Affording Knowledge Creation Practices in Education. Computer Supported Collaborative Learning Practices: CSCL2009 Conference Proceedings, pp. 297-306
- [7] I. Nonaka, H. Takeuchi: The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation, New York: Oxford University Press, 1995, 284 p.

-
- [8] J. Paralič, K. Furdík, M. Paralič, P. Bednár, P. Butka, J. Wagner: Semantic Support for Educational IT Services. *Acta Electrotechnica et Informatica*, Volume 12, No. 4, October – December 2012, pp. 39-46
- [9] J. Paralič, C. Richter, F. Babič, J. Wagner, M. Raček: Mirroring of Knowledge Practices based on User-defined Patterns. *Journal of Universal Computer Science*. Volume 17, Issue 10, 2011, pp. 1474-1491
- [10] M. Polanyi: *The Tacit Dimension*, University of Chicago Press: Chicago, 1966
- [11] T. S. Raghu, A. Vinze: A Business Process Context for Knowledge Management. *Decision Support Systems*, Volume 43 (2007), Issue 3, pp. 1062-1079
- [12] I. Savvas, N. Bassiliades: A Process-oriented Ontology-based Knowledge Management System for Facilitating Operational Procedures in Public Administration. *Expert Systems with Applications*, Volume 36 (2009), pp. 4467-4478
- [13] R. Duque, C. Bravo, M. Ortega: An Ontological Approach to Automating Collaboration and Interaction Analysis in Groupware Systems. *Knowledge-based Systems*, Volume 37 (2013), pp. 211-229
- [14] T. Bratitsis, et al.: Supporting Members of a Learning Community Using Interaction Analysis Tools: the Example of the Kaleidoscope NoE Scientific Network, in: Eighth IEEE International Conference on Advanced Learning Technologies, 2008, pp. 809-813
- [15] R. Duque, C. Bravo, J. Gallardo, W. J. Giraldo, M. Ortega: Modelling and Capturing Users' Actions in CSCL Systems for Analysis Purposes. *International Journal of Emerging Technologies in Learning (IJET)*, Volume 4 (2009), pp. 53-59
- [16] S. Strigunaite, D. Kriksciuniene: Fuzzy Expert System for Virtual Team Collaboration and Work Evaluation. In *Lecture Notes in Business Information Processing*, Volume 127 LNBIP (2012), pp. 37-43
- [17] S. Paavola, K. Hakkarainen: The Knowledge Creation Metaphor – an Emergent Epistemological Approach to Learning. *Science and Education*, Volume 14 (2005), pp. 535-557
- [18] F. Babič, J. Wagner, S. Jadlovska, P. Leško: A Logging Mechanism for Acquisition of Real Data from Different Collaborative Systems for Analytical Purposes. SAMI 2010: 8th International Symposium on Applied Machine Intelligence and Informatics, Slovensko, IEEE, 2010, pp. 109-112
- [19] M. Laclavík, Š. Dlugolinský, M. Šeleng, M. Ciglan, M. Tomášek, M. Kvassay, L. Hluchý: Lightweight Semantic Approach for Enterprise Search and Interoperability, *Proceedings of the Fifth Interop-Vlab.It Workshop on Complexity of Systems, Complexity of Interoperability in Conjunction with itAIS 2012, Rome, CEUR-WS.org, Vol. 915, pp. 35-42*

- [20] A. Garcia-Crespo, B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, I. Gonzalez-Carrasco: Semantic Model for Knowledge Representation in e-business. Knowledge-based Systems, Volume 24, Issue 2, March 2011, pp. 282-296
- [21] J. Naudet et al, *First version of PALETTE services delivery framework (D.IMP.06)*, <http://palette.ercim.org/images/stories/DocumentPDF/d.imp.06-final.pdf>, 2008
- [22] S. Heiyanthuduwage, R. Schwitter and M. A. Orgun, "Towards Ontology-driven E-Learning Information Systems", *Advances in Ontologies*, Proc. of 7th Australasian ontology workshop, Perth, Australia, pp. 21-26, 2011
- [23] L. Romero, M. Gutierrez and M. L. Caliusco, "Conceptualizing the e-Learning Assessment Domain using an Ontology Network", *International Journal of Artificial Intelligence and Interactive Multimedia*, Vol. 1, No. 6, pp. 20-28, 2012
- [24] F. Farkas, Á. Király: What Makes Higher Education Knowledge-Compatible? Acta Polytechnica Hungarica, Vol. 6, No. 3, 2009, pp. 93-104
- [25] P. Tóth: Learning Strategies and Styles in Vocational Education. Acta Polytechnica Hungarica, Vol. 9, No. 3, 2012, pp. 197-216