

A logging mechanism for acquisition of real data from different collaborative systems for analytical purposes

F. Babič^{**}, J. Wagner^{*}, S. Jadlovská^{**}, P. Leško^{**}

^{*} Centre for Information Technologies, Technical University of Košice, Slovakia

^{**} Department of Cybernetics and Artificial Intelligence, Technical University of Košice, Slovakia

{ Frantisek.Babic, Jozef.Wagner }@tuke.sk

{ Slavka.Jadlovaska, Peter.Lesko }@student.tuke.sk

Abstract - This paper presents design, implementation and experimental verification of logging mechanism for collaborative systems that contains separate database for storing the logs of events from the system, communication services between database and examined system and some simple analytical services to provide basic summarized information about saved data. The proposed approach was tested within four selected collaborative systems, namely KP-Lab System, Claroline, Moodle and Dokeos. The main motivation behind described solution is to obtain real data for different analytical purposes.

I. INTRODUCTION

Analyses of user behavior during collaborative activities in virtual working or learning environment can brought some interesting findings or new knowledge. These analyses identify an interesting relations, interactions and incentive factors before making decisions or performing a next step. Results of them can be used for customization of collaborative system, creation of new innovative practices or elimination of problematic approaches, peoples or objects. Source data for this type of analyses can be obtained by monitoring changes in user environment and saving into separate database based on predefined format of logs.

Proposed solution requires tight interaction with examined collaborative system even though is implemented as independent package. In this paper we show how our logging system can be integrated with selected collaborative systems. This mechanism was designed and implemented within EU IST project KP-Lab¹ so in the beginning we tested its potential with KP-Lab System and then we wanted to test usability with other systems. Because the architecture and log format were designed as a generic framework, it is possible to adapt it to different conditions.

The paper is organized as follows: the first section introduces the whole paper, describes the motivation of the paper and provides short state of the art. The second chapter describes in detail the whole architecture of proposed logging mechanism and defined format of logs.

The third section is devoted to four experimental cases that have tested cooperation of designed approach with selected collaborative system. The fourth section presents possible analytical approaches based on obtained data from collaborative system and the last section offers some concluding remarks.

A. State of the art

Awareness in collaborative system is amongst the very important features during collaborative working or learning activities. Awareness provides all necessary information about performed activities and chances in virtual user environment in selected ways, e.g. notification messages, email notifications or RSS². We can describe a simple example as a new user access the environment and other users that have been logged here receive a notification message about this event. This procedure can also be used in a situation where some responsible person makes additions and changes into a shared document and publish a new version of this document. All relevant authors of this paper will be then informed about this new version. The second important aspect of monitoring is creation and building of a database with historical data that can be used for simple or advanced analyses of previous user activities in order to identify the positive or negative aspects of selected approach.

Moodle as representative of wide adopted collaborative system provides monitoring and saving features to offer basic statistical information about existing courses or performed actions relevant to each user [1], see Fig. 1.

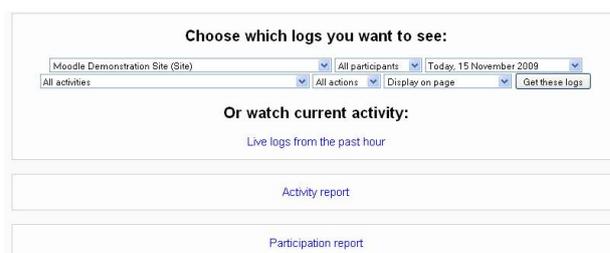


Figure 1. Example of Moodle report

¹ <http://www.kp-lab.org/>

² <http://en.wikipedia.org/wiki/RSS>

Activity report covers the list of all performed changes in selected part of virtual user environment. Participation report enables filtering logs of performed changes based on selected user or timeframe.

Open source system called Claroline³ offers possibility to visualize basic statistical information that is divided into several relevant groups as connectivity of the users, course access, access to the tools, forum usage, documents and their number of downloads (see Fig. 2) [2].



Figure 2. Example of Claroline statistics [2]

These two examples show situation that traditional collaborative system provides basic information about activities in user environment, but if you want to make an advanced analyses based on obtained data, users have to use some third party tools, e.g. SPSS. This approach has some disadvantages where the most important is to provide export data from examined collaborative system in the format that is required by analytical application.

II. ARCHITECTURE

Designed logging mechanism is based on fact that each collaborative system provides some type of monitoring so we need firstly services to exchange the data from monitoring services and store it within required format, see Fig. 3. Awareness repository is implemented on top of the MySQL database in order to provide fast and simple access with sufficient scalability.

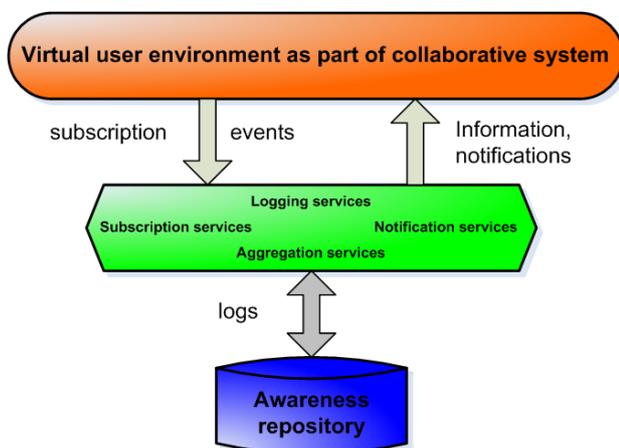


Figure 3. Architecture of the integration with logging mechanism

³ <http://www.claroline.net/>

A. Log format

Data stored in the awareness repository are represented by following format:

[*ID, GID, Time, Subject ID, Subject Type, Object ID, Object Type, Action, Comment, Custom properties, Custom data*]

- *ID* - unique identifier of the log entry;
- *GID* - group identifier that is used for identification of event collections, each collection contains data based on existing relations in-between, e.g. if there will be a change on subtask, there will be change on relevant parent task and relevant part of virtual environment too, so all of them will have same *GID*;
- *Time* – time when the event was logged into database (represented in the following format: year-month-day HH:MM:SS);
- *Subject ID* – unique identifier of actor that performed given event;
- *Subject Type* – the role of the user that is related to the relevant part of the user environment;
- *Object ID* – unique identifier of the shared object that motivates given event;
- *Object Type* – shared objects in user environment can be divided into different types, e.g. document, task, virtual meeting, link, etc.
- *Action* – a type of performed actions, e.g. creation, modification, deletion, etc.
- *Comment* – detailed specification of performed actions that can be provided by the original tool or user;
- *Custom properties* – this parameter is used when end-user application stores properties that are typical for this application, e.g. identifier of relevant part of user environment;
- *Custom data* – it is similar to custom properties, original end-user tool can store within this parameter custom data relevant for its purposes.

This basic log format can be further extended by additional parameters that are important for analytical purposes, e.g. versioning information. Actual log format version was designed based on several identified requirements as to store a rich historical data that will describe performed activities and changes in details a will be effectively used for future analytical tools. These tools will extract potentially interesting information from obtained logs of events in order to present e.g. summative statistics of performed activities or time-line based analyses of learning or working sequences.

B. Methods

The necessary communication and data management within proposed logging package is realized through web services with following defined methods:

- `AddEventResult addEvent (Event[] events)` - this method stores new events in awareness repository.

- Event[] getRecentChanges (String objectid, TimeRange timeRange) - returns the list of recent changes for a specified object.
- Event[] getRecentActivities(String subjectid, TimeRange timeRange) - returns the list of recent activities for a specified user.
- Event[] getRecentEventsFiltered (TimeRange timeRange, Map<String, String> filter, String includeNull) - returns the list of recent events. Filter is set of key value pairs which will be compared with events custom properties.

Notification and subscription services are implemented as an extension of basic logging mechanism in the same layer. Aggregation services are part of analytical package that is implemented over the logging mechanism.

III. EXPERIMENTS

A. KP-Lab System

KP-Lab System provides modular, flexible and highly integrated system for collaborative realization of different working or learning activities. This system consists of two main parts: platform and user environment. Our logging mechanism was connected on middleware layer to obtain data from user environment in cooperation with monitoring services on this level. The result of this first test is documented on Fig. 4.

ID	Object ID	Object Type	Action	Comment	Properties	Value	Custom Data
2895 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2896 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2897 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2898 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2899 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2900 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2901 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2902 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2903 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2904 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2905 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2906 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2907 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2908 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2909 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
2910 (deleted)	2009-12-04 19:04:16.4	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	

Figure 4. Logs in awareness repository from KP-Lab System [3]

B. Moodle

Moodle represents open source course management (CMS) system based on modular architecture that allows design and implementation of new modules with additional functionalities based on user expectations. Moodle provides simple monitoring and logging services [4] as it is described in state of the art. We tried to integrate our solution with these existing services.

The first attempt was based on exploitation of internal Moodle method called Event handler that is responsible for logging defined events. This approach showed problem with implementation where each event which we want to log has to have its own handler, so if we want to log as many events as possible for analytical purposes, we have to define appropriate instances of Event handler method.

The second attempt covers the design and implementation of new web services that is responsible for communication with internal Moodle logging system to obtain data from its internal database based on required format of awareness repository. This attempt brought

successful cooperation and results are documented on Fig. 5.

ID	Object ID	Object Type	Action	Comment	Properties	Value	Custom Data
331 (331)	2009-12-06 19:04:25.0	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
332 (332)	2009-12-06 19:04:25.0	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
333 (333)	2009-12-06 19:04:25.0	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	
334 (334)	2009-12-06 19:04:25.0	http://www.kp-lab.org/moodle/mod/quiz/attempt.php?id=1234	attempt	attempt	attempt	attempt	

Figure 5. Logs in awareness repository from Moodle

C. Claroline

Claroline is an open source eLearning and eWorking platform allowing users to manage learning and collaborative activities on the web. Monitoring and logging in Claroline is strong oriented on basic summative statistics, as numbers of user access, number of documents or number of documents downloads, etc. This information was not enough for us and the next step contained modifications in Claroline logging API to extend existing functionalities. This approach was not successful by reason of low support level of this API, insufficient documentation of whole system and necessary changes in source code e.g. start monitoring services required modifications in many source files.

D. Dokeos

Dokeos represents open source learning management system (LMS) with strong support of SCORM⁴ standard. Awareness in this system is oriented on visualization of basic statistics too, as number of passed exams, accesses to the platform, etc. Monitoring services provides insufficient information for our purposes, so we tried possibility to add a new event handler for monitoring purposes. This approach requested modifications in source codes (e.g. main_api.lib.php, events.lib.inc.php), but brought potentially useful results, see Fig. 6.

ID	Object ID	Object Type	Action	Comment	Properties	Value	Custom Data
107	0	#	#	5	2009-12-04 04:49:24.0	dokeos_answer_class.phpjsave	First step
108	0	#	#	5	2009-12-04 04:49:24.0	dokeos_answer_class.phpjsave	Second step
109	3	#	#	0	2009-12-04 04:50:23.0	dokeos_api_item_property_update	ItemID
110	3	#	#	0	2009-12-04 04:50:38.0	dokeos_api_item_property_update	ItemID
111	0	#	#	6	2009-12-04 04:50:38.0	dokeos_answer_class.phpjsave	Lack of Vitamin A
112	0	#	#	6	2009-12-04 04:50:38.0	dokeos_answer_class.phpjsave	Lack of Calcium
113	0	#	#	6	2009-12-04 04:50:38.0	dokeos_answer_class.phpjsave	adfasdf
114	3	#	#	1	2009-12-04 04:50:46.0	dokeos_event_access_tool	
115	3	#	#	1	2009-12-04 04:50:52.0	dokeos_api_item_property_update	ItemID
116	3	#	#	0	2009-12-04 04:50:52.0	dokeos_api_item_property_update	ItemID

Figure 6. Logs in awareness repository from Dokeos

⁴ <http://en.wikipedia.org/wiki/SCORM>

IV. ANALYTICAL APPROCHES

The main motivation behind described logging mechanism and performed tests is to obtain data for analyses of collaborative user behavior, effective management of collaborative tasks and shared objects, identification of relevant interactions and relations, etc.

Interesting analytical approach represents time-line based visualization that provides complex visualization of relevant actions in virtual user environment with common goal, e.g. collaborative writing of conference paper. This process requires collaborative cooperation of several authors in order to produce one document based on different professional backgrounds, working procedures and approaches. Exchange through emails with particular paper parts and further paper version is not very user friendly and useful approach. Collaborative system provides possibilities to realize related actions within one virtual user environment. Every action will be monitored on the GUI (graphical user interface) level and stored into separate repository to provide log storage about performed events. Logs of events can be used for construction of a basic timeline view that represents sequences of actions (steps) from start point of the whole process creation to delivery of paper final version, see Fig. 7.

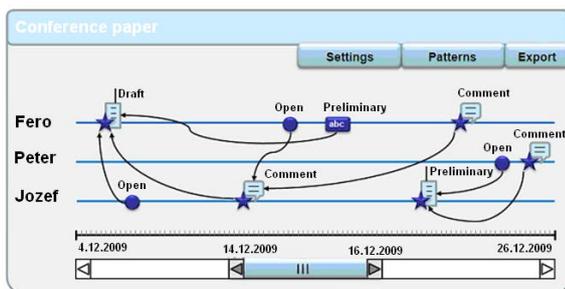


Figure 7. Example of time-line based visualization

This basic flow can be extended by additional information based on the data storage, with information about involved users, shared objects that were used as inputs, etc. Details of described analytical approach are presented in [5] and [6].

V. CONCLUSION

The domain of collaborative system covers various applications that are marked as LME (Learning Management System), CMS (Content Management System), VLE (Virtual Learning Management), systems for CSCW (Computer-Supported Collaborative Learning) or CSCL (Computer-Supported Collaborative Working) and some others. All of these systems are based on common principles as access to the virtual user environment, tools for object sharing, communication tools, etc.

Monitoring of performed activities or practices in collaborative system is an important feature which allows us to obtain historical data about changes and user's interactions with the whole system. This data can be further used for different analytical purposes, from basic approaches such as statistical information about accessing the platform, to complex information about user sequence of action with all relevant information. Relevancy and

exploitability of acquired outputs is strong depending on quality, consistency and range of the whole source data.

We designed a logging mechanism for acquisition of relevant data from collaborative system. This mechanism was primary implemented and actually used in KP-Lab System as middleware components and source of data for analytical tools. But we wanted to test its universality with some other similar systems. We choose Moodle as very well-know system that has more than 30 million users; Claroline with strong language support – more than 30 languages containing Slovak; Dokeos with more than 20 million users and wide adoption.

Results of performed tests show that monitoring and logging services don't belong between most important parts of the collaborative systems. It was necessary to make some changes in source codes to connect our logging services. The obtained data weren't consistent in each case; some values for concrete parameters were missing. But the final conclusion is that our logging mechanism can be connected with different collaborative system with respect to existing requirements, necessary changes in implementation (services or system), existing documentation and level of logging API expansibility.

ACKNOWLEDGMENT

The work presented in this paper was supported by: European Commission DG INFSO under the IST program, contract No. 27490; the Slovak Research and Development Agency under the contract No. VMSP-P-0048-09; the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic under grants No. 1/0042/10; project implementation Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020) supported by the Research & Development Operational Programme funded by the ERDF.

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

- [1] Moodle reports, available on <http://docs.moodle.org/en/Category:Report>
- [2] Claroline statistics, available on http://doc.claroline.net/en/index.php/The_course_statistics
- [3] Web interface for awareness repository, available on http://kplab.tuke.sk:8080/hpa/secure/browse_events.jsp
- [4] Moodle Events API, available on http://docs.moodle.org/en/Development:Events#Events_which_exist
- [5] Paralič, J., Babič, F., Wagner, J., Simonenko, E., Spyrtos, N., Sukibuchi, T., *Analyses of knowledge creation processes based on different types of monitored data*, In: Proc. of the ISMIS 2009. Foundations of Intelligent Systems. LNCS 5722/2009, Springer Berlin / Heidelberg, pp. 321-330, ISBN 978-3-642-04124-2
- [6] Wagner, J., Babič, F., *Design of technological framework for analyses of collaborative processes realized in virtual environment*. 4th Workshop on Intelligent and Knowledge oriented Technologies: Proceedings : November 12 - 13, 2009, Herľany, Slovensko. s. 95-101. ISBN 978-80-89284-42-9.