Didactical Interpretation of Log Data: The MOCLog Model

Marco Bettoni
Swiss Distance University of Applied Sciences (FFHS)
Überlandstrasse 12
Brig - Switzerland
+41 27 922 3916
marco.bettoni@weknow.ch

Jetmire Sadiki
Swiss Distance University of Applied Sciences (FFHS)
Überlandstrasse 12
Brig - Switzerland
+41 27 922 3916
jetmire.sadiki@ffhs.ch

Riccardo Mazza
eLab – eLearning Lab
USI – Università della Svizzera Italiana
Via Buffi 13
Lugano - Switzerland
+41 58 666 6505
riccardo.mazza@usi.ch

Luca Mazzola
eLab – eLearning Lab
USI – Università della Svizzera Italiana
Via Buffi 13
Lugano - Switzerland
+41 58 666 6505
luca.mazzola@usi.ch

ABSTRACT
The lack of a didactical perspective puts a serious dent in the activity of analyzing the user interaction traces of an LMS in a useful way. As a contribution to overcoming this difficulty, we have developed MOCLog, a monitoring system that interprets the logfile data based on a suitable didactical model. This paper essentially presents this model of logfile analysis and at the end gives a short overview of the tool in which it has been implemented.

Categories and Subject Descriptors
H.1.m [**Models and Principles**]: Miscellaneous

General Terms
Measurement, Performance, Experimentation, Theory.

Keywords
Student tracking, log analysis, logfile analyzer, learning analytics, educational data mining.

1. INTRODUCTION
When teaching and learning is supported by learning management systems (LMS), then the logfiles (user interaction traces) of the LMS offer opportunities for understanding the activities of students and teachers; this understanding then provides a good basis for devising ways of improving the quality of teaching and learning. Unfortunately, the logfiles provided by a LMS are seldom used, mainly because it is difficult to interpret and exploit them [13]; the obstacles to interpretation and exploitation fall into four main categories:

1. certain types of usage data are not logged;
2. the data that are logged may seem incomplete;
3. users are afraid that they could draw unsound inferences from some of the data;
4. data are not aggregated from a didactical perspective.

Our attention was particularly attracted to the fourth category of these obstacles to interpretation, because the lack of a didactical perspective puts a serious dent in the activity of improving the quality of teaching and learning by means of data analysis. As a contribution to overcoming this difficulty, we have developed MOCLog, a monitoring system that helps to analyze the logfiles of the LMS Moodle by interpreting the data based on a suitable didactical model that we call "MOCLog model".

This paper essentially presents this model of logfile analysis and at the end gives a short overview of the tool in which it has been implemented[1]. Our didactically oriented model is necessary when one wants to interpret the logfiles in terms of relevant learning processes, learning activities and learning outcomes (performances). The model has been used in designing and programming the MOCLog tool for analysing the logfiles and creating visualizations for various use cases (scenarios) that are relevant to our stakeholders.

2. THE MOCLog APPROACH
Software applications need to record every activity performed and every interaction with the user(s), essentially for technical reasons. As pieces of software, LMS keep logs of their activities and interactions, using different methods. Moodle, for instance, uses its own database to register different kinds of event data. As a web based system, Moodle relies on a web server that has its own logging method. Moodle recently implemented some preliminary and in some aspects complementary information in the "Site-Administration" reports (Moodle-Add-Ons for Activity Reports, Live Logs, etc.). What Moodle does not offer is a "higher level" of interpretation of the logfiles generated by the users, one which is relevant from a didactical point of view.

[1] A detailed description of the MOCLog tool can be found in [10]. It explains the different visualisations and how depending on the use case data are obtained and displayed.

The basic idea of the MOCLog approach to logfile analysis is to combine a systematic understanding of eLearning quality with data about LMS activities by users (logfiles, user interaction traces) within a useful tool. In a similar way to a detective who tries to make sense of the clues of a crime based not only on case experience but also by using forensic science (Sherlock Holmes is famous for this), in developing the MOCLog model of logfile analysis we also used eLearning science and especially its insights into the quality of the learning process; this understanding, supplemented by eLearning good practice (experience of the stakeholders), has been our guide and has supported our work of interpreting the traces that learning activities leave in the LMS system in the form of logfile entries.

The MOCLog system wants to support four stakeholders that are involved in the monitoring of online courses. The reason for drawing a distinction between these 4 groups is that they have different but equally relevant needs and roles to play in the improvement of learning quality:

- Student: focus on self-monitoring the self learning process
- Teacher: focus on monitoring the students' learning process and on didactic-methodical aspects of the online course
- Study program manager: focus on monitoring all courses of the study program and on the usage of the LMS by teachers/students, partially didactic aspects
- Administrator: focus on monitoring all online courses of the institution and mainly technical aspects, as well as general didactic aspects

3. THE VIEW FROM RESEARCH

To understand the whole concept of our MOCLog model, it is important to know that first of all, our model should provide the required foundations for designing and implementing the MOCLog tool; secondly, that we want the MOCLog tool firmly based on insight into the quality of the learning process. As a consequence, we began the development of our model with a theoretical reflection about learning.

The main reason for explicitly searching for a didactical theory orientation and other support for our model development is that a purely technological approach to eLearning would impose some essential limitations to the logfile analysis, with the risk that the monitoring tool would not deliver the information needed by its users for accurately interpreting the LMS activities in terms of learning.

3.1 eLearning Theory

About ten years ago, Garrison highlighted the problem of the lack of a coherent understanding of distance education practice and complained that education was more focused on the fiscal implications than on educational issues [7 p. 1]. In his view, the challenge was to provide a theory that would explain and anticipate distance education practices. As a way to meet this challenge, Garrison suggested that it is the nature of a *sustained educational transaction* at a distance that must be described, understood and abstracted [7, p. 11]. Specifically, he proposed that the theory should describe a *collaborative approach* to distance education which includes, for example, asynchronous collaborative learning based on written communication and making use of the unique characteristics of text-based communication [7, p. 10-13].

An important contribution to a didactical theory of the kind envisioned by Garrison was proposed six years later by Gabi Reinmann [5] in the form of a convincing problem statement and related solution hypothesis about a *learning-oriented approach to eLearning*. Reinmann's problem statement complains of the lack of didactical concepts in eLearning practice:

- "*Although there are many good didactic concepts for utilising technological potential (see, for example, Niegemann et al., 2004), these are far too rarely found in practice or in the minds of decision-makers. Instead, it appears to be above all technical possibilities that provide the framework within which learning too is now to become faster, more effective, cheaper and more fun*." [5, p. 6].

The problem is not new. For finding a solution, Reinmann suggests a hypothesis which claims that:

- "*one must first understand learning per se in order to be able to promote e-learning*" [5, p. 1].

Such an understanding of "*learning per se*" appears for example in a heuristic framework model for designing eLearning environments, by means of which Reinmann wants to show "*that technical decisions come at the end of a chain of decisions that relate, or should relate, primarily to education and teaching.*" [5, p.1]. The framework model proposed by Reinmann (see Fig. 1) distinguishes between three levels of abstraction of e-learning environments: didactic scenarios (structural level), patterns of didactic interaction (process level) and technical tools (technology level).

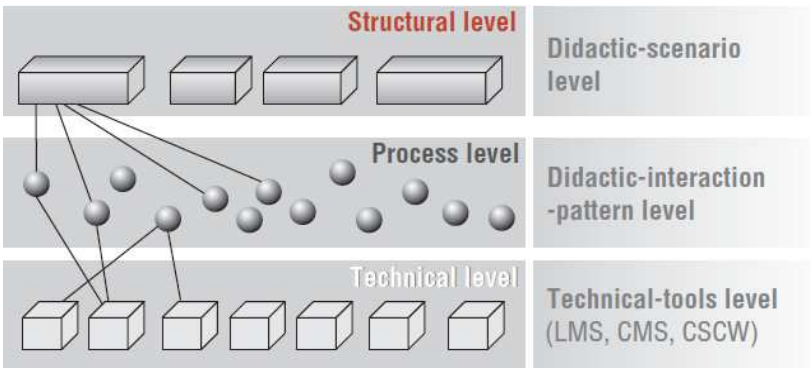

Figure 1. Heuristic framework model
(from [5]).

3.2 Theory of eLearning Functions

Because eLearning can refer to a wide range of applications which may use many kinds of new media, Reinmann [6] saw the danger, that discussions about eLearning would easily become very vague and concluded that there is a need for an orienting model which helps with keeping a clear view of the field. The model she suggested, based on Back, Seufert and Kramhöller [8], distinguishes 3 general functions of new media, each of which enables a specific way (method) of learning [6, p. 31-35]:

- the media function of "distributing information" enables the learner to learn by means of the method "Learning from information"

- the media function of "interaction between user and system" enables the learner to learn by means of the method "Learning from feedback"
- the media function of "collaboration between students" enables the learner to learn by means of the method of "Learning by collaboration".

In Fig. 2, these three media functions and the three general learning methods they enable are displayed on the axes of a xy-diagram. By connecting the media functions with their associated learning methods, Reinmann obtained a clear definition of three general e-Learning functions:

- eLearning by distributing
- eLearning by interacting
- eLearning by collaborating

These three types of learning with new media place different demands, first on the design of eLearning environments (and therefore also on the ability of media designers and teachers) and secondly on the learning processes (and thus on the requirements that the learners need to comply with). In section 5.1 we will explain our criteria for selecting a learning theory suitable to our needs and what the Reinmann over any other approach can bring to our model.

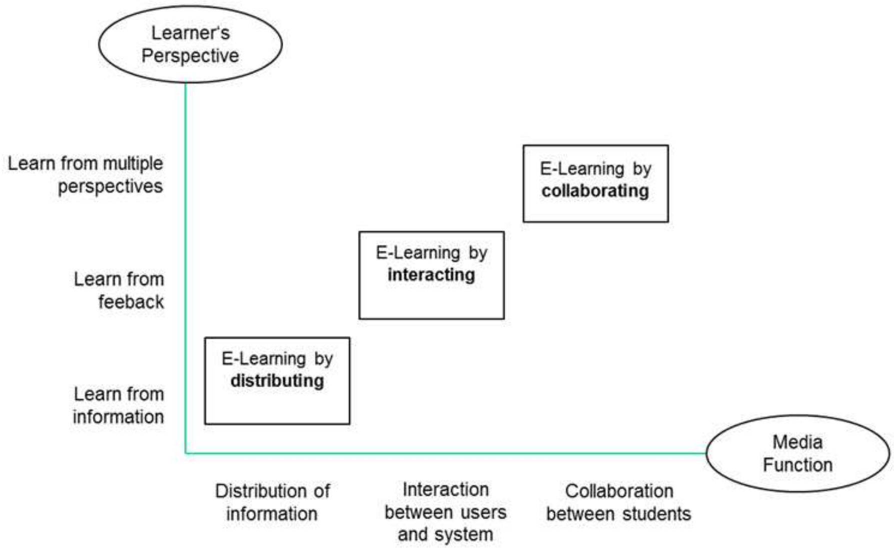

Figure 2. Three different kinds of eLearning.
([6], chapter 7.1, p. 33, partially in English)

4. THE VIEW FROM EXPERIENCE: A CASE STUDY

As previously mentioned, for interpreting the traces of learning activities, we need to complement a theoretical understanding of eLearning quality with eLearning good practice, i.e. the experience of our stakeholder. This is the role that stakeholder requirements play in our MOCLog model: they represent the view from experience in the form of needs and benefits expected from the MOCLog tool from the different types of stakeholders involved (next is a first list of them).

The requirements analysis performed at the beginning of the project was conceived as a case study, not as a representative statistical inquiry; our case study has pointed to some important needs and benefits for our stakeholders, while others were derived from the experience of our team members as researchers in eLearning, as teachers that use LMS systems and as contributors to the management of study programs and to the administration of Moodle sites. In order to elicit requirements from our stakeholders, we conducted an analysis with a number of individuals from each category. The main results of our analysis are summarized here.

Students spend an average of one hour per week on Moodle. The LMS platform is used mainly for test exam preparation, knowledge testing, and assignments. The most important monitoring functions that the students would like to have are: presentation of their test and assignment results, information about the areas in which they have weaknesses, indication about resources visited/unvisited and presentation of their test and assignment results related to the average of their class.

For **teachers,** content related issues are more important than interaction related issues. Overall, the major aspects of high importance to the teachers related to the course content are: students' course access, resources' visits, resources uploaded by students, course interaction (in which time frame do students upload the assignments, etc.)

Study program managers want different information about teachers e.g. identification of heavy and weak users. A diagram with the division of the log files into the three elements (distribution, interaction and collaboration) would be useful. In our case study, we interviewed 6 study program managers and found out that for them, the most important indicators are: the course access (how often teachers access a course), resource usage (see how often they are added or updated), news forum usage (how often teachers post messages) and an overview of the study program (what a teacher does, in which courses and how, distinguished in the 3 categories: distribution, interaction, collaboration).

Finally, **administrators'** requirements are driven by annual report and eLearning strategy. The use cases are often very complex. Their main goal is to switch from the current manual and time-consuming process to an automated, complete, comprehensive and consistent, no media break and standardized process. We found that administrators are interested in: compiling reports, identifying usage trends, identification of "good", "active/inactive" or "not used" courses – based on parameters, selection, and the analysis of individual courses with the aim of offering individual coaching and to analyze why something happens (e.g. which teacher action generates which student reaction?).

5. A MODEL OF LOGFILE ANALYSIS

Based on the research review and requirements analysis presented above, in this section we will develop the MOCLog model, a combination of a concept map, a solution map and a set of use cases. We want our model to act as a guide indicating the goals and the direction in which the tool implementation should go; moreover it should indicate what to pay attention to when implementing those goals.

5.1 Concept Map

As a first step on the way to defining a didactically oriented model, we need to clarify in terms of learning theory the monitoring questions by students and teachers. From our research review, the best approach that we have found for this task is constituted by the aforementioned theory of eLearning functions by Reinmann [5,6]. Why Reinmann? We wanted a model that would be: a) up-to-date, b) simple, c) comprehensive, d) suitable for logfile analysis, e) easily, directly applicable to logfile analysis. Requirement b) and c) are a big challenge, since comprehensive models are usually rather complex, and vice-versa simple models are often too much limited to a specific aspect. But the biggest challenges are in requirements d and e, because many good models are theoretical frameworks that are either not suitable for being used in logfile analysis or cannot be easily and directly applied to practice. For example, from the list of 16 e-learning models described in a relevant review of pedagogical frameworks for elearning [12] only Salmon's e-tivities approach seems suitable for satisfying our requirements approximately as well as Reinmann's approach. Another example can be found in [11] which presents a theoretical model of online learning, the Community of Inquiry (CoI) framework, which would require a research project by itself for making it applicable to the MOCLog design needs.

What can we learn from Reinmann's theory and how can we transfer it to our model of logfile analysis? To answer these questions, we will do two things:

a) describe Reinmann's model in terms of a concept map
b) extend that concept map to fit it into our monitoring task (connect it with logfiles and with requirements).

The concept map gives us a better understanding of logfile analysis by clarifying the concepts involved and their relationships. We see on the left end of the map the "learning success" disc (where "success" means reaching the learning objectives) and on the right side the "logfile" disc (meaning the collection of LMS activity traces). Between the two, a sequence of 3 elements and their relationships connects the two discs like a bridge, a conceptual bridge on which to walk to go from one end to the other. This answers the monitoring questions:

- which way of eLearning enables us to reach the given objectives?
- by which means (functions, tools) does the LMS enable these ways of learning?
- how is the use of these means traced in the logfiles (activity log codes[2])?

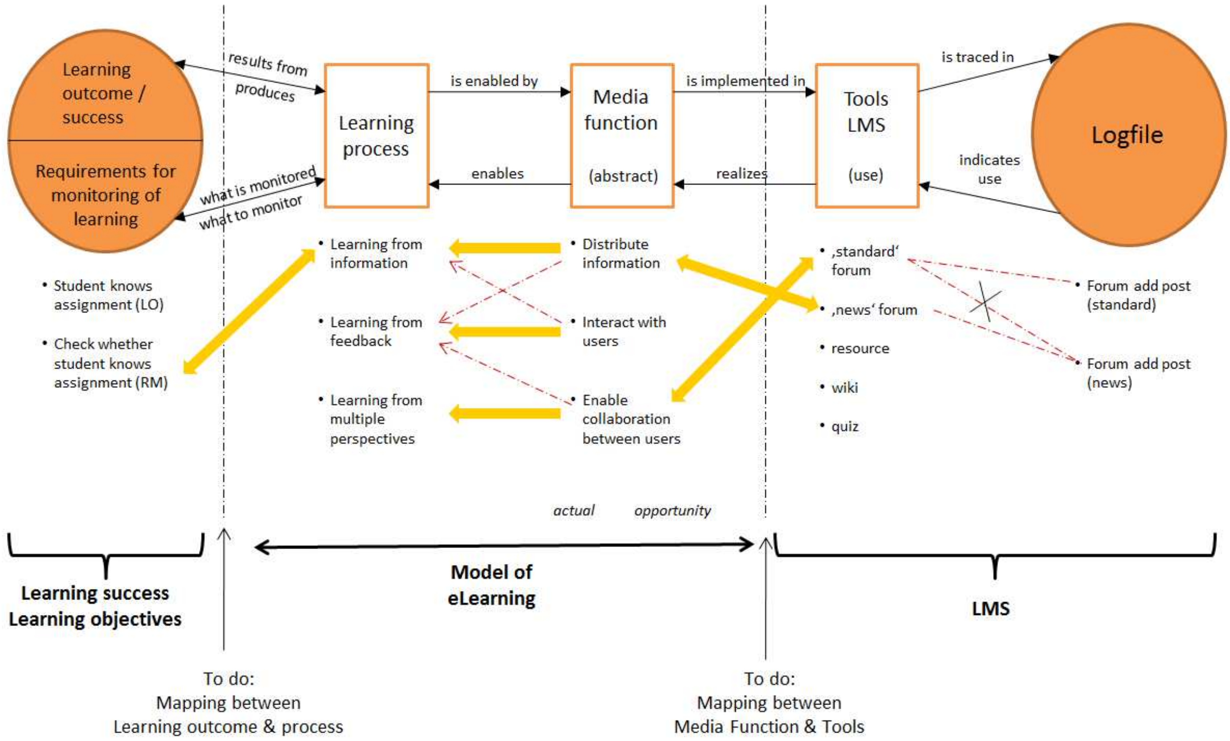

Figure 3. MOCLog concept map

[2] Designation of an action like *‚forum add post'* in a Moodle logfile entry.

5.2 Solution Map

What we need now is to go down from the abstract level of the concept map (objectives, functions, tools, log codes) to the concrete level where the MOCLog model describes specific instances:

- a specific didactical objective that needs to be achieved
- the learning process that leads to that objective and needs to be monitored
- the eLearning function which enables that learning process
- the LMS-tool which supports that function
- and finally the log code and logfile entry which indicates the use of the LMS-tool.

The MOCLog solution map shown in Fig. 4 has four columns:

1. the first column, "PDCA", specifies which of the map elements belong to the "performing learning" step and which to the "controlling learning" step
2. the second column, "Didactical objective", organizes needs (required "learning methods") in terms of the theory of eLearning functions
3. the third column, "Monitoring solution", displays another set of needs (required "desired functions") and shows with its left side connections how MOCLog could support the monitoring of a specific learning method; with its right side

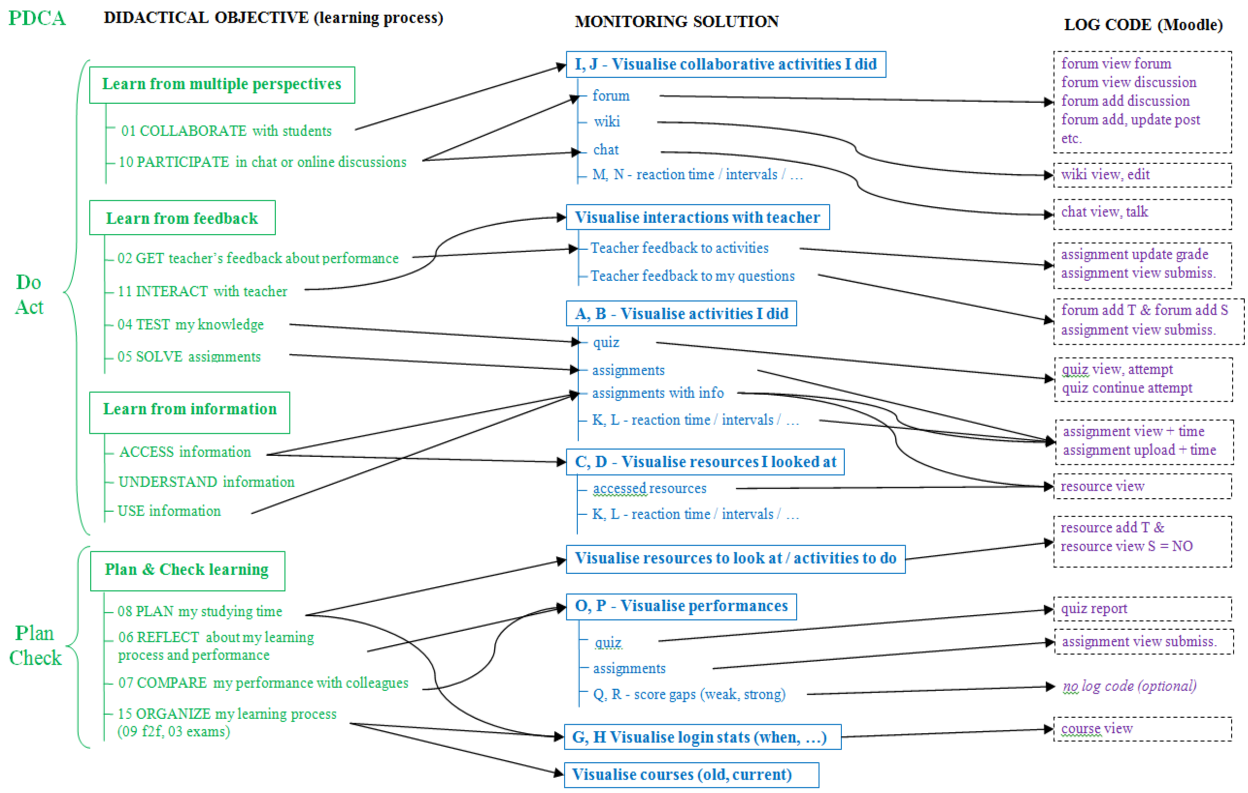

Figure 4. MOCLog solution map

In a certain sense, this is the place where we must integrate theory and practice with the aim of producing a solution:

- theory in the shape of a framework of didactical objectives and related means
- practice in the shape of stakeholders' requirements (the view from experience) and LMS metrics (log codes, logfile entry formats).

For supporting this process of integration of theory and practice, we have used two methods: the so-called "Solution Finder Model" (SFM) and PDCA cycle. The SFM is a system-oriented problem solving instrument [9] based on the idea of the unity of 3 elements: needs, objectives and solutions. The PDCA cycle (plan-do-check-act) also called "Deming cycle" is an iterative four-step problem-solving process typically used in business process improvement.

connections, this columns shows how the solutions comply with metrics available in the LMS.

4. the fourth column, "Log Codes (Moodle)", displays log codes (from Moodle) grouped in terms of the tool for which they are used in tracing the use of the tool.

The MOCLog solution map provides a mean to derive which information should be extracted and visualized from a Moodle LMS in order to satisfy a specific didactical objective. The solution map has driven the definition of use cases, which are a specification of a solution map for some didactical objectives that are requested by stakeholders.

5.3 Use Cases

At this point of model development, when looking at the solution map, the following question arises: how could MOCLog users profit from the solutions mentioned in the solution map (section 5.3)? The answer can be found by using the knowledge formalized within the solution map itself: in fact, by integrating theory (didactics) and practice (requirements, log codes), the solution map provides in synthesis all the knowledge (elements and connections) that can guide us in defining some appropriate use cases.

By the term "use case", we indicate here a specification of how to use the MOCLog tool; it is an outside view by users (as opposed to the inside view of programmers) that is guided by the concept map, the solution map, and analysis and generalization of user requirements. Specifying a use case for students and teachers in general requires the following steps:

- you begin by selecting a didactical objective (for example "learn from multiple perspectives"); the reason for this start is that we want our use cases to have a foundation in didactics;
- the selected objective has several needs associated with it (for example: "01 COLLABORATE with students" and "10 PARTICIPATE in chat or online discussions"); among these needs you then select that one, which is the most general
- next you check the other needs in the same group and see whether they can be subsumed under the previously selected need (as a part of it);
- now you follow the connection to the associated monitoring solution and further to the metrics (log codes): how can these metrics be processed for obtaining that visualization?
- Finally, for answering this question you must specify two things: a) the logic of visualization (input and output interface, GUI); b) the logic of processing (metrics, combinations of metrics, indicators, operations, etc.).

From the 10 use cases that we defined, we will present here the one called "Students' Collaboration" as an example. In this use case we want to monitor how much the student collaborates with other students; collaboration support can be provided by forums and chats for communication and by wikis for collaboratively building knowledge resources. In terms of our model, this case is defined by (see Fig. 4):

1. *didactical objective*: "learn from multiple perspectives";
2. *associated need* (requirement): "01 COLLABORATE with students";
3. *subsumed need* (requirement): "10 PARTICIPATE in chat or online discussions";
4. *monitoring solution*: "I, J - Visualise collaborative activities I did (forum, wiki, chat)".

For the processing logic of the implementation we need to define: 1. A set of collaborative tools (forum, wiki, chat); 2. A set of "observation" (view, read, etc.) and "contribution" log codes (add, update, etc.); 3. Two indicators: for observation and for contribution.

6. A MONITORING TOOL

In order to implement the model previously presented, a software solution called MOCLog has been implemented for use with the Open Source LMS Moodle (for details see [10]).

The choice of relying on data from Moodle is due to its adoption by the partner Universities as the main distance education platform and to our experience with analytics for Moodle [2], based on a model previously presented [4]. It is evident that the kind of log actions and the format that is adopted to collect the interaction deeply affect the software implementation; for this reason, trying to achieve a tool working with all the LMSes falls outside of the scope of the current implementation, as we did in other projects instead [1], [3].

Our actual solution is composed by two independent parts that are conformant to the plugins format required by the LMS. The first one is mainly aimed at supporting teacher and students, while the other one is directed toward the support of administrators and study program managers. The two modules are respectively called MOCLog-GISMO and MOCLog-admin. They can also be used independently of one another due to their different final objective. For extracting the required information, both modules rely on the data collected about the interaction between the users and the LMS. For the sake of scalability and responsiveness, each one of them uses some additional database tables to pre-process the data and to save them in computed form. The computation, whose objective is to extract data from logs and derive aggregated information, is run on a regular basis by the cron in background, in order to reduce the impact on the responsiveness of the normal service as much as possible.

MOCLog-GISMO uses the students' and teachers' log data from Moodle, and generates graphical representations that can be explored and manipulated by teachers and by students. It is implemented as a web application, launched from the Moodle interface. To allow a better interactivity and responsiveness of the interface, and to reduce the computational load on the server, it works on the client side, downloading the interface, updating only the needed data as a JSON string and caching it for further computation.

With MOCLog-GISMO, students can inspect their learning situation: whether they are on track and what kind of activities/resources they have alreadyread. Thus they can use it for their own individual learning analytics. What is not yet possible at the moment is that the students compare their individual progress with the progress of the entire class, as we already did in another experiment [3]. On the teachers' side, MOCLog-GISMO is also helpful because it can track the participation of the students by analyzing the activities they used and the resources they looked at, allowing them to better control and adapt their interventions based on this information.

The other part, MOCLog-admin, uses the Moodle log data to generate statistics and visualizations of users' activity, the evolution of the LMS's tools usage, and aggregated user and course activity. We developed some prebuilt configurations to evidence distribution, participation and interaction. Furthermore, it is possible to define a self-configuration for stressing one or more characteristics, by changing the value attributed to different event generated by users. In this way, it can be useful for supporting the activity of study-program managers and administrators, whether technical or not. They can see and analyze the course activity (selected courses with their number of hits on each Moodle tool), the tool activity (each tool with the total number of hits across all courses), teacher facilitation activity, students' total learning and the level of facilitation and learning across the whole course.

This implementation of the tool is released on sourceforge (see links on http://moclog.ch/) and is open to the usage and contribution by everyone interested.

7. CONCLUSIONS

In this paper, we have presented a model of logfile analysis that is based on a didactical perspective and we have given a short overview of the monitoring tool MOClog (a plugin for the LMS Moodle) that implements the model, thus allowing a didactically oriented interpretation of user interaction traces. The development of our model began with a theoretical reflection about learning which allowed us to identify a suitable theoretical foundation in Reinmann's theory of e-Learning functions [6, p. 31-35].

At the same time, by means of a requirement analysis conceived as a case study, we also obtained a view from experience, in the form of needs and benefits expected from the MOCLog tool by the different types of stakeholders involved.

Based on these two foundations, we developed the MOCLog model, a combination of a concept map, a solution map and a set of use cases. The concept map gives us a better understanding of logfile analysis by clarifying the concepts involved and their relationships. The solution map provides a mean with which to derive which information should be extracted and visualized from a Moodle LMS in order to satisfy a specific didactical objective.

Thus, by integrating theory (didactics) and practice (requirements, log codes), the solution map provides all the knowledge (elements and connections) in synthesis that can guide us in defining some appropriate use cases. By means of the use cases, the model has been then implemented in the MOCLog tool. Our next goal will be the deployment, evaluation and exploitation of the tool both in the institutions of the project partners as well as in other institutions (universities, universities of applied sciences and education that use Moodle).

8. ACKNOWLEDGMENTS

This project has been carried out as part of the "AAA/SWITCH – e-infrastructure for e-science" program under the leadership of SWITCH, the Swiss National Research and Education Network, and has been supported by funds from Federal Office for Professional Education and Technology (OPET).

We would like to thank other people who contributed to this work: Per Bergamin (Swiss Distance University of Applied Sciences in Brig), Cinzia Garcia, Achim Dannecker, and Maria Papanikolaou (Learning lab, School of Business, University of Applied Sciences and Arts Northwestern Switzerland), Gerd Josten, Michael Röthlin, and Hanspeter Zimmermann (Bern University of Applied Sciences), Marco Faré (eLab – eLearning Lab, USI – Università della Svizzera Italiana).

9. REFERENCES

[1] Mazzola L., Eynard D., Mazza R. (2010). GVIS: a framework for graphical mashups of heterogeneous sources to support data interpretation. 3nd IEEE Conference on Human System Interactions, 2010. HSI '10, pgg. 578-584.

[2] Mazzola L., Mazza R. (2011). Visualizing Learner Models through data aggregation: a test case. . Red-conference, rethinking education in the knowledge society, Monte Verità, Switzerland, March 7-10, pp.372- 380, ISBN 978-88-6101-010-9.

[3] Mazzola L., Mazza R. (2010). GVIS: A Facility for Adaptively Mashing Up and Representing Open Learner Models. "Sustaining TEL: From Innovation to Learning and Practice" Lecture Notes in Computer Science, Volume 6383/2010, pp. 554-559, Proceedings of EC-TEL 2010, DOI: 10.1007/978-3-642-16020-2_53 .

[4] Mazzola, L., Mazza, R. (2009) Supporting learners in Adaptive Learning Environments through the enhancement of the Student Model, In Human-Computer Interaction. Interacting in Various Application Domains, Springer, pgg. 166-175.

[5] Reinmann, G. (2006). Understanding e-learning: an opportunity for Europe? European Journal of Vocational Training, 38, 27-42.

[6] Reinmann-Rothmeier, G. (2003). Didaktische Innovation durch Blended Learning. Bern: Hans Huber.

[7] Garrison, R. (2000) Theoretical Challenges for Distance Education in the 21st Century: A Shift from Structural to Transactional Issues. Int. Review of Research in Open and Distance Learning, Vol. 1, No. 1 (June 2000), 1-17.

[8] Back, A,, Seufert, S., and Kramhöller, S. (1998) Technology Enabled Management Education. Die Lernumgebung MBE Genius im Bereich Executive Study an der Universitiit St. Gallen. *io management* Nr. 3, 1998.

[9] Bettoni, M., Bernhard, W., Eggs, C. & Schiller G. (2010) Idea Management by Role Based Networked Learning. In: E. Tomé, Proc. 11th European Conference on Knowledge Management, Universidade Lusíada de Vila Nova de Famalicão, Portugal, 2-3 September 2010, Vol. 2, pp 107-116. Reading: Academic Publishing Ltd.

[10] Mazza, R. Bettoni, M., Faré, M. & Mazzola, L. (2012) MOCLog – Monitoring Online Courses with log data. In: S. Retalis & M. Dougiamas (eds.) Proc. of the 1st Moodle Research Conference, Heraklion, Crete, Greece, Sept. 14-15, 2012, pp. 132-139, Download at http://research.moodle.net/MoodleCon_Proceedings_program/pdf/mag.pdf

[11] Swan, K., Garrison, D. R., & Richardson, J. (2009) A constructivist approach to online learning: The community of inquiry framework. In C. R. Payne (Ed.), Information technology and constructivism in higher education: Progressive learning frameworks (pp. 43-57). Hershey, PA: IGI Global.

[12] Mayes, T. & de Freitas, S. (2004) Review of e-learning theories, frameworks and models. Report of the JISC e-Learning Models Desk Study, Stage 2. Download at: http://www.jisc.ac.uk/uploaded_documents/Stage%202%20Learning%20Models%20%28Version%201%29.pdf

[13] Mazza R. & Dimitrova V. (2005). Generation of Graphical Representations of Student Tracking Data in Course Management Systems. In: 9th IEEE International Conference on Information Visualisation. London 6-8 July 2005. pp. 253-258.