



THE LEARNING ANALYTICS APPLICATION LEMO – RATIONALS AND FIRST RESULTS

Margarita Elkina ¹⁾, Albrecht Fortenbacher ²⁾, Agathe Merceron ³⁾

¹⁾Hochschule für Wirtschaft und Recht Berlin, Germany
margarita.elkina@hwr-berlin.de, <http://www.hwr-berlin.de>

²⁾Hochschule für Technik und Wirtschaft Berlin, Germany,
albrecht.fortenbacher@htw-berlin.de, <http://www.htw-berlin.de>

³⁾Beuth Hochschule für Technik Berlin, Germany,
merceron@beuth-hochschule.de, <http://www.beuth-hochschule.de>

Abstract: LeMo is an open source application for learning analytics, which collects data about learners' activities from different platforms. This article describes design principles of LeMo in the context of creating an efficient tool for learning analytics. Focus is on the LeMo system architecture, user path analysis employing algorithms of sequential pattern mining, and visualization of learners' activities implemented in the current version. A case study shows first results. Copyright © Research Institute for Intelligent Computer Systems, 2013. All rights reserved.

Keywords: learning analytics, visual analytics, explorative visualization, sequential pattern mining, user path analysis.

1. INTRODUCTION

The Horizon Report 2011 [1] identifies learning analytics as an emerging technology, which most likely will have a significant impact on higher education within the next years [2]. On the first LAK conference 2011 [3], learning analytics was defined as “the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs”.

In 2011, the research project “LeMo: monitoring of the learning process on personalizing and non-personalizing learning platforms” started at three universities of applied sciences in Berlin [4]. It aims at developing an analytics and visualization tool, to facilitate learning analytics for teachers, E-learning providers and researchers. Basic requirements were connectivity to various platforms for online learning, powerful analysis and mining of data obtained from the platforms, and an intuitive visualization of analytic results, providing insight into the learning process.

The following section describes the methodical approach of LeMo, which guided requirements analysis and refinement throughout the project. Then, basic design principles of the LeMo application are explained. The following sections focus on user path analysis and visualization. In a

case study at HWR Berlin, LeMo was used in a regular class, showing applicability of LeMo and its usability for teachers. The last section concludes the findings of the LeMo project, and refers to future work.

2. METHODOLOGY

Requirements analysis in LeMo is based on a survey amongst stakeholders (project partners), representing the three target groups teachers, E-learning providers and researchers. This generic qualitative method was targeted at supporting teaching with a variety of didactic methods and different degrees of technology support, e.g. E-learning as a supplement to face-to-face teaching, or online learning as a basis for distance learning.

The result of the survey was a catalogue of about 80 questions and assumptions regarding students' learning behavior and the use of online media. Some of the questions were beyond the scope of an analytics tool which relies on learners' activities obtained from a learning (management) platform.

The catalogue of questions was segmented into six categories representing different areas of interest. The first two categories contain questions about users and groups of users, as well as learning objects and type of learning objects. Interactions with the learning environment are modeled through the categories “usage analysis”, “user path

analysis”, “communication and collaboration”, and “performance”.

Based on a catalog of questions provided by the LeMo project partners, 27 indicators were derived, that provide data analysis to help answering the referred questions. The indicators “activity / time” and “activity / learning object” are associated with the category “usage analyses”, visualizing how students access learning material. The indicators “frequent paths” and “activity graph” belong to the category “user path analysis”, where visualization shows user activities ordered by time.

The survey showed a considerable interest in analyzing the sequence of user interactions. This interest can be contextualized and supported by the theoretical approach of connectivism. According to G. Siemens, learning is organized in a network – internally as a neural network and externally “as a network of nodes, representing a specialization and aggregates as actions concerning the environment” [5]. In a first step, this network is visualized as an “activity graph” in LeMo.

Consequently, the four indicators mentioned above were the first indicators realized in LeMo. Whereas “usage analysis” is quite common in comparable tools, the indicators “frequent paths” and “activity graph”, together with intuitive visualization, are unique characteristics of the LeMo tool.

3. THE LEMO APPLICATION

3.1. SYSTEM ARCHITECTURE

LeMo is designed as a 3-tier application, which facilitated distributed development among the three universities [6]. The first tier (data) contains a data model as well as functionality for data management, which includes connectivity to different learning platforms. Data analysis takes place in the second tier (data). Results of data analysis are passed to the third tier (presentation), where the data are visualized.

Filtering data, e.g. selecting male or female students within a course, is presently done in the data tier, but eventually should be performed in the third tier, thus reducing the number of requests to the data tier.

The LeMo application consists of a *data management server*, which basically comprises the first and second tier, and an *application server*. For each user interaction, a request is generated by the application server, which is passed via a web service interface to the data-management server. Both servers are realized as web apps, which could be run by a single Tomcat server, but also, for performance and security reasons, be distributed within a network.

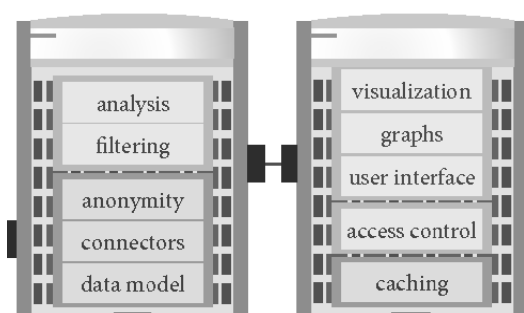


Fig. 1 – Two-server architecture.

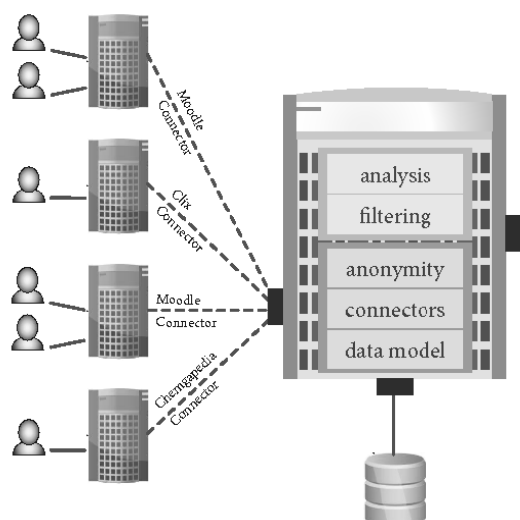


Fig. 2 – Data management server.

The data management server may connect to more than one LMS (learning management system). Presently, connectors for different versions of moodle, clix and Chemgapedia are implemented.

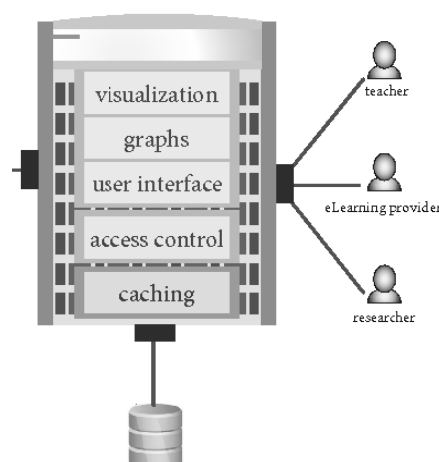


Fig. 3 – Application server.

The application server provides for visualization and user interaction. Using a JavaScript framework, this is done locally on the client, reducing server load and network traffic.

3.2. DATA MANAGEMENT

In [7], three factors for the development of learning analytics are listed: online learning, big data, and political concerns. This list is reflected by the design principles of LeMo.

Data analyzed by the LeMo tool are basically user activities obtained from platforms for online learning. LeMo connects to various platforms, including the learning management systems (LMS) Moodle¹ and Clix², as well as the online encyclopedia Chemgapedia³. The LeMo connectors for Moodle and Clix are implemented to access the underlying databases directly. In case of Chemgapedia, which is a web application, data stem from the server log file.

During the ETL (Extraction – Translation – Load) phase, data are imported into a database with a unified data model. While LMS provide detailed information on content and on users, server log files just contain page visits. Here, session information is needed to identify anonymous users (students). The LeMo data model contains entities for learning objects, e.g. courses, resources, wikis, or tests, and associations between learning objects, e.g. resource X belongs to course A. There exist students and teachers related to courses. Degree programs can group courses, a department is responsible for different degree programs, and an institution consists of several departments. The data model also includes “user activities”, which basically are represented as tuples $[u,l,t,a]$ (user, learning object, time stamp, action). An “action” may be one of “view”, “download”, “modification”, “creation”, “attempt”, or “submit”. And finally, platforms are represented in the data model. Given an element e , which was imported from platform x , a unique primary key for this element can be constructed from the pair $[e,x]$. Thus, it is possible to aggregate user data from different platforms into a single data model.

In the case of Chemgapedia, some data are not available, examples being department/faculty, or a student's enrollment in courses. This leads to sparse data, and eventually to a reduced number of available analyses.

Big data denote huge sets of, generally unstructured, data, which typically originate from social networks, and which are processed using analytic methods (“social media analytics”). Recording activity data on platforms for online learning, over a long period of time, a huge amount of data is produced. For interactive analytics applications, it is critical to handle big data

efficiently, which can be achieved via data compression, efficient data retrieval and efficient mining algorithms.

Reviewing usage statistics on the Chemgapedia web server, it became apparent that log data have to be preprocessed in order to exclude activities not corresponding to learners (e.g. traffic generated by web crawlers). To solve this problem, an optional functionality was added which excludes all log entries with one of the following characteristics: multiple accesses per second, frequent repetition of time intervals between accesses, and many accesses to the same page. The use of this filter reduced the number of log entries by about 47 %, while excluding just 5 % of all users [8].

With the actual LeMo prototype, data are stored in a relational data base. First studies show that this is sufficiently efficient for institutions with about 10,000 students. Non-standard ways of storing data, e.g. in a NoSQL data base, will be investigated in the future.

3.3. DATA PRIVACY

One political concern of learning analytics is data privacy. Traditionally, personalized interactions and user modeling have significant implications on data privacy. Personal information about a user is collected and analyzed, which might not be in the interest of the user. Recently, collection of user data in social networks, and possible violation of data-privacy legislation, have been published frequently. As a result, even more strict data-privacy regulations at universities are established, which in turn limits the use of learning analytics.

The LeMo approach towards data privacy aims at achieving a high level of anonymity. The activity sequence, i.e. the “learning path” of a particular student may be analyzed, but personal data, which could serve to identify the student, must not be exposed. This is done by omitting all personal data in the ETL process, the only exception being gender information which is valuable for learning analytics. Furthermore, small data samples must not be used for a specific analysis, if this leads to the identification of a particular student (k-anonymity [9]). An example is a course with just one female (male) participant; this student could be identified easily using a gender filter.

For most of the analyses, courses must be identified by their name. Also, the title of learning objects like learning material (e.g. powerpoint presentations), the name of threads in a course-related forum, or results of a test provided in the course, are essential for efficient analysis, but must not allow for identification of learners (students).

1 <https://moodle.org/>

2 <http://www.im-c.de/>

3 www.chemgapedia.de/

To implement a role “teacher”, courses taught by a given teacher must be identified. This cannot be done using some unique attribute, like name, personal ID or login name (user ID). All these attributes are omitted during the ETL process. Instead, a mapping from teachers to courses can be realized by a pseudonym (hash value) derived from a unique attribute during ETL. If the LeMo application uses the same authentication scheme as the connected platform (e.g. authentication via LDAP, the light way directory access protocol), the pseudonym is derived from the login name. When a teacher logs in, she or he sees the list of “his” or “her” courses.

4. USER PATH ANALYSIS

Computing frequent paths in LeMo is based on two algorithms, namely BIDE [10] and Fournier-Viger [11]. Implementations for both algorithms are provided by the SPMF (sequential pattern mining framework) library⁴.

Given a course with a set of learning objects L , and a set of users (students) U , we define a path as a sequence of learning objects, allowing multiple occurrences of the same learning object, and L^* as the set of all possible paths (in terms of formal languages, each path is a word over the alphabet L). By $P(u)$ we denote the sequence of all learning objects accessed by user u , in the correct order implied by the time stamp of the corresponding activity.

$$p_1 = s_1 q_1 s_1 q_1 f_1 s_2 q_1 \quad (1)$$

As an example consider a course containing the following learning objects: two sets of slides s_1 and s_2 , two quizzes q_1 and q_2 , and one forum f_1 . Let p_1 be a path and u be a User with $P(u) = p_1$. This corresponds to the following behavior: The user u first accesses the slides s_1 , then quiz q_1 , then again s_1 and the same quiz q_1 . Then she visits the forum f_1 , accesses slides s_2 and finally again quiz q_1 (1).

A path $p = l_1 \dots l_m$ is *contained* in another path $p' = l'_1 \dots l'_n$, if it can be embedded into p' , preserving the order of the learning objects:

$$p \sqsubseteq p' \Leftrightarrow \exists 1 \leq i_1 < \dots < i_m \leq n: l_j = l'_{i_j} \forall 1 \leq j \leq m \quad (2)$$

A path $p = l_1 \dots l_m$ is a *subsequence* of another path $p' = l'_1 \dots l'_n$, if it can be embedded into p' directly, preserving the original sequence of p :

$$p \subseteq p' \Leftrightarrow \exists 0 \leq k \leq n - m: l_j = l'_{j+k} \forall 1 \leq j \leq m \quad (3)$$

Let's look at two paths

$$p_2 = s_1 s_2 q_1 \quad (4)$$

$$p_3 = s_1 q_1 f_1 s_2 q_1 \quad (5)$$

The path p_2 could reflect the teacher's intended “learning path”, i.e. first accessing slides s_1 , then s_2 , and then trying to pass quiz q_1 . p_2 is contained in p_1 (at positions 1, 6, 7 or 3, 6, 7), but p_2 is not a subsequence of p_1 . The path p_3 could reflect the following learning activities: read slides s_1 , go to quiz q_1 (and fail), go to the forum f_1 and ask for help, read slides s_2 and pass the quiz q_1 . p_3 is a subsequence of path p_1 , starting at position 3.

For both relations “is contained” and “is a subsequence”, we can define the set of frequent user paths:

$$FP_{\subseteq}(s) \Leftrightarrow \{p \in L^*: |\{u \in U: p \subseteq P(u)\}| \div |U| \geq s\} \quad (6)$$

$$FP_{\sqsubseteq}(s) \Leftrightarrow \{p \in L^*: |\{u \in U: p \sqsubseteq P(u)\}| \div |U| \geq s\} \quad (7)$$

With a *support* value of 0.5, the set of frequent paths consists of all learning object sequences, which are followed by at least 50 % of all users.

Given four users with learning paths

$$P(u_1) = s_1 s_2 q_2 q_1 \quad (8)$$

$$P(u_2) = q_1 q_2 f_1 s_1 s_2 \quad (9)$$

$$P(u_3) = f_1 \quad (10)$$

$$P(u_4) = s_1 q_1 s_1 q_1 f_1 s_2 q_1 \quad (11)$$

we get the following frequent paths:

$$FP_{\subseteq}(0.7) = \{s_1, s_2, q_1, f_1\} \quad (12)$$

$$FP_{\sqsubseteq}(0.7) = \{s_1, s_2, q_1, f_1, s_1 s_2\} \quad (13)$$

In this example, all paths are followed by at least 3 users.

The BIDE algorithm computes frequent paths with respect to the relation “is contained in” (6). Given a constant support value, the complexity of BIDE is $O(n_1 n_2 n_3)$, where $n_1 n_2 n_3$ is the product of the number of users $|U|$, the average and the maximal length of user paths $P(u)$. Since BIDE uses an apriori approach to determine reoccurring sub-paths, the processing time increases dramatically when the number of candidate sequences is large [10].

The Fournier-Viger algorithm computes frequent paths with respect to the relation “is a subsequence of” (7). It was developed for a specific application domain, namely learning with tutorial systems for

4 <http://www.philippe-fournier-viger.com/spmf/>

robotics, and extends the well-known PrefixSpan algorithm [12]. Fournier-Viger obtains significantly less paths than the BIDE algorithm. Since the number of candidate sequences is low (compared to BIDE), the algorithm can be easily applied to low support values, which is not the case for BIDE.

Fig. 4 and Fig. 5 illustrate the difference between these two approaches, both in the results they deliver and in their performance. Both algorithms have been run on exactly the same data set with the same minimal support of 50 %. The result of the BIDE approach is shown in Fig. 4: 20 frequent paths of length 3 have been found. Different learning objects are shown with different colors, making it easy to identify the same learning object within one or several paths. Clicking on a path (Fig. 4: the left path) shows the names of the learning objects. At least 50 % of the students followed the path: *resource1.1.1.6 resource1.1.1.6 resource1.1.1.5*.



Fig. 4 – Frequent paths with BIDE.

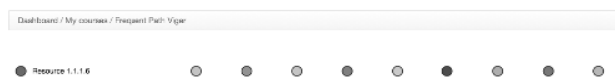


Fig. 5 – Frequent paths with Fournier-Viger.

The result of the Fournier-Viger approach is shown in Fig. 5: For a support value of 50 %, 10 paths have been found, among them the paths (with one learning object) *resource1.1.1.5* and *resource 1.1.1.6*. The maximal length is 1.

Combining the two results, we conclude that all students who accessed *resource1.1.1.6*, later again *resource1.1.1.6* and later *resource1.1.1.5* have all accessed different learning objects in between. Lowering the support to 10 % gives the results shown in Fig. 6 obtained by the Fournier-Viger algorithm, while after 10 minutes the BIDE algorithm is still calculating.



Fig. 6 – Fournier-Viger with lower support.

The drastically increased processing time of BIDE for low support values poses a serious problem to the usability of the LeMo application. As

a solution, a “default value” for minimal support is determined, depending on parameters “number of user paths”, “average length of user paths”, and “number of learning objects” within a given course.

A typical users’ behaviour is to try out low support values (which might cause BIDE to run “for ever”), and then increase the values until a result is achieved within a reasonable response time. This might start a couple of BIDE algorithms running as parallel threads, and eventually bring down the system. To avoid this, the algorithm must be terminated when a user requires the next analysis. Thus, in a multi-user environment, the number of running BIDE algorithms is limited to the number of active users. Furthermore, after a predefined lapse of time, the algorithm is terminated, which does not leave hung threads in the system.

5. VISUALIZATION

The design of the user interface in LeMo follows the explorative metaphor “overview first, zoom and filter, then details-on-demand”, which assists analysis of learning data in an explorative way [13]. Especially when displaying large amounts of data, this approach supports a highly intuitive reception of the results of analyses.

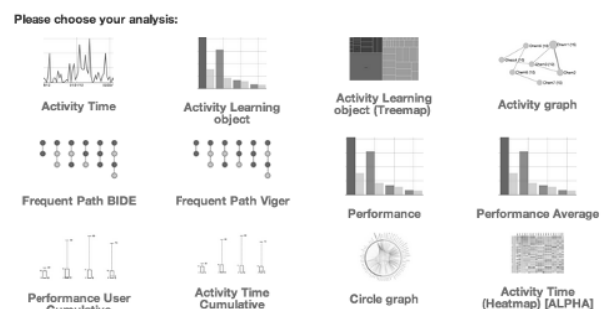


Fig. 7 – Different analyses / visualizations.

LeMo provides a variety of different visualizations, which are suitable for analysis of specific learning situations and processes. For example, the performance of students based on self-tests can be visualized. Alternatively, students’ performance could be used as a filter for other analyses, exploring correlations between performance and navigation [14].

Visualization was designed according to the following guidelines [13]:

- **pre-attentive perception:** use of visual attributes, e.g. shape, size, color and position
- **provide details on demand:** detailed information on a specific learning object is available (context sensitive), without overloading the initial visualization

- **facilitate exploration:** the interface provides the possibility to customize the visualization, to best fit the personal expectations, through features like translation, rotation, filtering, and zooming

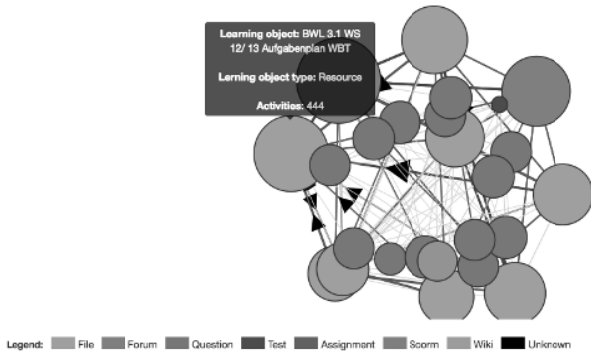


Fig. 8 – Navigation graph.

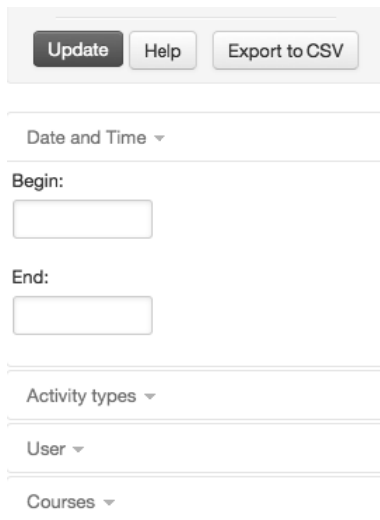


Fig. 9 – Filters.

Zooming and filtering can be applied to various visualizations, e.g. “activity / time”, “activity / learning object”, “treemap”, “frequent paths” or “activity graph”. Also, results of a self-test (quiz) can be used to filter user activities. The following example shows the application of the filters “learning object type” and “students”, as well as zooming into areas of interest.

Activities over time show a pattern, which corresponds with the concept of the course (blended learning).

Zooming provides more insight into a peak of activities around 2013-11-26.

Selecting arbitrarily (anonymous) students gives hints on different learning behaviour.

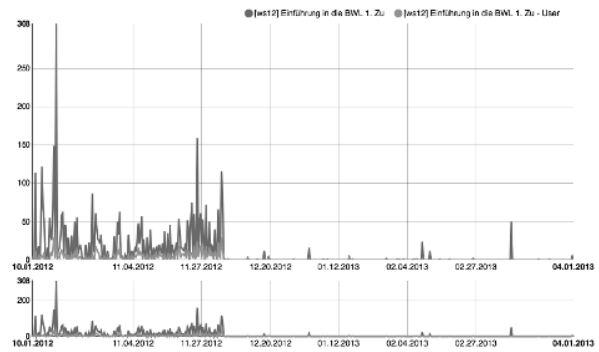


Fig. 10 – Course “Betriebswirtschaft”, 2012/13.

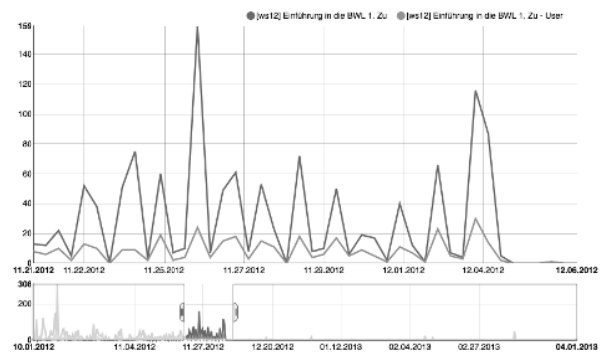


Fig. 11 – Zooming in.

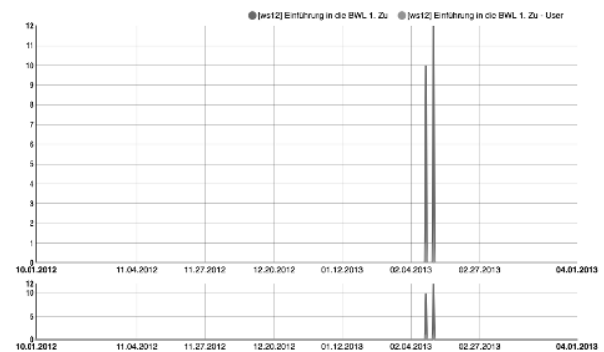


Fig. 12 – Selecting a single student.

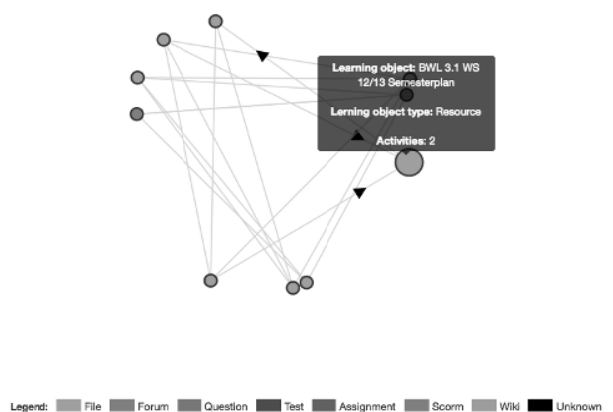


Fig. 13 – Single students' navigation.

The analysis “activity graph” provides a lot of information about the accumulated navigation behavior of all students, which in cases proves difficult to analyze and interpret. Selecting the same student as above, a selective usage of learning objects is displayed.

The results of an analysis “activity graph” are visualized by a navigation network. This analysis can be used to obtain information on the degree of cross-linking between learning objects. The activity graph is visualized by a graph, with learning objects as nodes and navigation steps as edges. Nodes are color-coded to support the visual perception of content type transitions. Learning objects are adjusted in size to encode the absolute number of user requests. Edges are weighted and color-coded to encode the amount of navigational steps.

Detailed information on specific elements of the graph can be made visible by interacting with the visualization, including tool tips with information on learning objects, and rearranging the graph, focusing on nodes of interest.

For the visualization of frequent paths (results of the BIDE algorithm, or Fournier-Viger algorithm), the following premises are important:

- A path is visualized by a sequence of nodes (learning objects) and edges (navigation steps). Learning objects are color-coded to allow a visually easy correlation of navigational patterns across different paths.
- Detailed information on specific elements of the path is visible through user interaction.

In some cases, LeMo provides even different “visual styles” for the same analysis, an example being “activity graph”. The first version features a volatile visualization of the navigation graph, which facilitates user interaction (Fig. 8). The second version is a more static visualization, which, as an overview, just displays learning objects, where the size of the circles corresponds to the number of related user activities. Edges are omitted - selected edges appear, when a specific learning object is selected. The two visualizations support different ways of perceiving information, allowing users to develop their personal style of doing learning analytics [15].

6. A CASE STUDY

The case study is based on a bachelor degree course in informatics, which used Moodle as a learning platform. In addition to the standard file upload (slides, documents), the Moodle course contained quizzes and a wiki for individual work. The questions for the quizzes, and the assignments in the wiki were formulated by the students: each student provided one question or one assignment.

The forum in Moodle was not used frequently, since this course assumed the presence of students in the lectures. The goal of this case study was an analysis of the usage of quizzes and wiki, which extends the regular possibilities provided by Moodle.

6.1. ACTIVITY / TIME

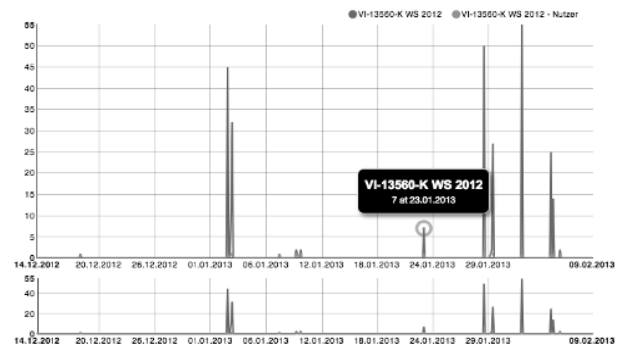


Fig. 14 – Activity / time (quizzes selected).

The upper part (blue) displays the number of activities, the lower part (orange) indicates the number of users who started these activities. Tool tips show the exact numbers of activities and users, and the exact time. The scale of the axis is adapted to the current value range.

Fig. 16 shows activities on quizzes, which can be achieved by the filter “learning object type”. The analysis of activities indicates, that students' activities are rather high at the end of the semester. There was no examination at this time, the final examination is at the end of the second semester. It appears that the students were motivated to test their knowledge in order to evaluate themselves.



Fig. 15 – Activity / time (wiki selected).

Part of the didactic concept was to use a wiki for student collaboration. After the deadline for submission of the assignments, which had to be provided via the wiki, continuous (optional) activities in the wiki showed that students accepted this concept.

6.2. ACTIVITIES DURING THE WEEK

LeMo offers the visualization “activity time cumulative”, which displays all activities during the week as box plots. Optionally, each day can be split into four slots, with six hours each. The box plots display the span of activities during the weekdays, as well as quartiles and median of the distribution.

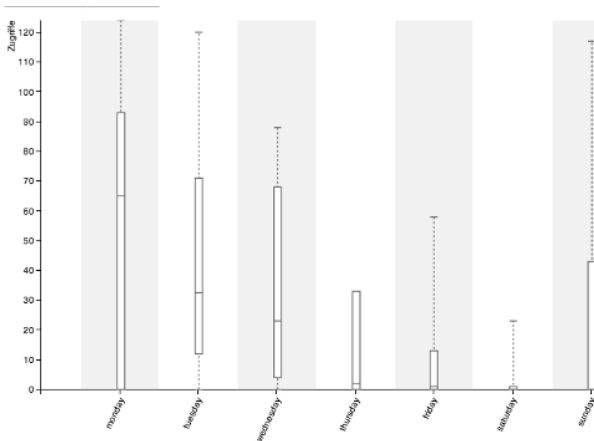


Fig. 16 – Cumulative activities.

From the visualization, it is obvious that students started to learn on Sunday, became most active on Monday and stayed rather active on Tuesday. Saturday is the day with the lowest activity. At first sight, Sunday activities were unexpected, although it can be explained logically: on Monday, students attend other courses as well. The slides for the next lecture were always made accessible on Monday morning; didactically, it would make more sense to provide access already on Saturday.

6.3. RESULTS OF THE CASE STUDY

The analysis offered by LeMo could be adapted to the particular course, so that the expected results of the learning scenarios could be tested. Interesting aspects of the learning behavior of the students have been disclosed, providing hints on how to improve of the courses. These conclusions were drawn mostly from a combination of the various visual analyses in LeMo.

7. CONCLUSION AND FURTHER WORK

The actual LeMo prototype proved to be a useful tool for teachers, giving hints on the learning behaviour of their students, and helping with the evaluation of their didactic concept. The combination of user path analysis combined with a powerful interactive visualization distinguishes LeMo from other tools for learning analytics.

Focus of future work will be on improving the LeMo application with respect to efficiency and

usability, as well as on enhancing data analytics and visual analytics. System improvements will include a standardized ETL interface, connectivity with further platforms, a (non-standard) data model for efficient data retrieval, and additional pattern-mining and data-mining algorithms.

Functionality will be enhanced by realizing more indicators derived from the stakeholders' original list of requirements (questions). Of special interest are indicators referring to “group of users”, “students' performance”, and “communication and collaboration”.

The actual prototype realizes the role “teacher”, where a teacher can analyze his/her own courses. In the future, the roles “E-learning provider” and “researcher” should be defined and implemented.

For all improvements on the functionality and usability of the LeMo tool, an evaluation study is essential and will be conducted in the immediate future.

8. REFERENCES

- [1] L. Johnson et al., *The 2011 Horizon Report*, The New Media Consortium, Austin, 2011, <http://www.nmc.org/pdf/2011-Horizon-Report.pdf>.
- [2] A. Fortenbacher, Learning analytics for higher education – perspectives and challenges, in *Pratsi: Scientific, Science and Technology Collected Articles*, Issue 1(40), Odessa National Polytechnic University, 2013, pp. 184–187.
- [3] *1st International Conference on Learning Analytics and Knowledge*, ACM Digital Library, 2011.
- [4] L. Beuster et al., Learning analytics und visualisierung mit dem LeMo-Tool, in *Proceedings of Interaktive Vielfalt – DeLFI*, Bremen, 2013.
- [5] G. Siemens, Connectivism: Learning Theory of Pastime of the Selfamused, http://www.elearnspace.org/Articles/connectivism_self-amused.html. 2006.
- [6] A. Fortenbacher, L. Beuster, M. Elkina, L. Kappe, A. Merceron, A. Pursian, S. Schwarzrock, B. Wenzlaff, LeMo: a Learning Analytics Application Focussing on User Path Analysis and Interactive Visualization, *The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2013)*, 12-14 September 2013, Berlin, Germany, Vol. 2, pp. 748-753.
- [7] R. Ferguson, The state of learning analytics in 2012: a review and future challenges, Tech. rep. KMI-12-01, Knowledge Media Institute, The Open University, UK, 2012.

- [8] S. Schwarzrock, Analysis of learner navigation on web-based platforms using algorithms for sequential pattern mining, in *Pratsi: Scientific, Science and Technology Collected Articles*, Issue 1(40), Odessa National Polytechnic University, 2013, pp. 18–21.
- [9] V. Ciriani et al., K-Anonymity, in *Secure Data Management in Decentralized Systems*, Ed. by T. Yu and S. Jajodia, Vol. 33, Advances in Information Security, Springer, Berlin Heidelberg, 2007, pp. 323–353.
- [10] J. Wang and J. Han, Efficient mining of frequent closed sequences, in *Proceedings of the 20th International Conference on Data Engineering*, Boston, MA, USA, 2004.
- [11] P. Fournier-Viger, R. Nkambou, and E. Mephu Nguifo, A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems, in *Proceedings of MICAI*, Ed. by A. Gelbukh and E. F. Morales, *Lecture Notes in Artificial Intelligence*, Vol. 5317. Springer-Verlag Berlin Heidelberg, 2008, pp. 765–778.
- [12] Jian Pei et al., Mining sequential patterns by pattern-growth: the prefixspan approach, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Issue 11, 2004, pp. 1424–1440.
- [13] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in *Proceedings of the IEEE Symposium on Visual Languages*, 1996, pp. 336–343.
- [14] A. Merceron et al., Visual exploration of interactions and performance with LeMo, in *Proceedings of the 6th International Conference on Educational Data Mining*, Memphis, USA, 2013.
- [15] L. Beuster et al., Prototyp einer plattformunabhängigen Learning Analytics Applikation – fokussiert auf Nutzungsanalyse und Pfadanalyse, in *E-Learning Symposium 2012 Aktuelle Anwendungen, innovative Prozesse und neueste Ergebnisse aus der E-Learning-Praxis*, Ed. by U. Lucke, Universitätsverlag Potsdam, 2012, pp. 69–72.

versity, she was teaching at the university, then turned to industry and for over 10 years performed research in software development. She managed many German and international projects, designed graphical user interfaces and implemented different software solutions. Starting from 2011, she also belongs to the management of the project “Monitoring of learning processes in personalizing and non-personalizing learning platforms”. Currently her main research interests are Learning Analytics, Data Mining and Visualization.



Prof. Dr. Albrecht Fortenbacher graduated in computer science from the University of Karlsruhe, Germany. After one year at IBM Thomas J. Watson Research Center, Yorktown Heights, USA, where he contributed to the IBM computer algebra system Scratchpad/AXIOM, he received in 1988 his Ph.D. from University of Karlsruhe. Then, Albrecht Fortenbacher joined the IBM Scientific Center in Heidelberg, where he worked, and did research, on supercomputing and parallel computation.

In 1994, Albrecht Fortenbacher was appointed professor for distributed and secure systems at the HTW Berlin, University of Applied Sciences. His research focus is on Teaching Learning Technology, recent projects range from E-learning infrastructure (LMS), ePortfolios, virtual classroom and ad-hoc meetings, to learning analytics. Albrecht Fortenbacher is active in the German learning analytics community, with a focus on introducing learning analytics to universities.



Prof. Dr. Agathe Merceron holds a Ph.D in Computer Science of the University Paris VII. She joined Beuth University of Applied Sciences in 2006 and is the head of the online degrees Computer Science and Media, Bachelor and Master as well as head of the Online-Learning Laboratory. She teaches courses such as programming, foundation of computer science, algorithms or data mining. Her research interests presently focus on Information Systems and Knowledge Management – application to E-Learning, Technology Enhanced Learning, Educational Data Mining, Learning Analytics. She is a board member of the International Educational Data Mining Society and an associate editor of the International Journal of Educational data Mining.



Prof. Dr. Margarita Elkina is teaching the courses on informatics, program and software development at the Berlin School of Economics and Law, Berlin, Germany. After obtaining her PhD degree in applied mathematics at the Lomonosov Moscow State Uni-