

Understanding the Learners' Actions when using Mathematics Learning Tools

Paul Libbrecht¹, Sandra Rebholz², Daniel Herding³,
Wolfgang Müller², Felix Tscheulin²

¹ Institute for Mathematics and Informatics, Karlsruhe University of Education,

²Media Education and Visualization Group, Weingarten University of Education,

³Computer-Supported Learning Research Group, RWTH Aachen University
Germany

Abstract. The use of computer-based mathematics tools is widespread in learning. Depending on the way that these tools assess the learner's solution paths, one can distinguish between automatic assessment tools and semi-automatic assessment tools. Automatic assessment tools directly provide all feedback necessary to the learners, while semi-automatic assessment tools involve the teachers as part the assessment process. They are provided with as much information as possible on the learners' interactions with the tool.

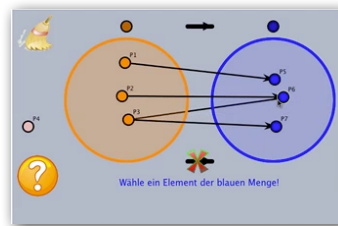
How can the teachers know how the learning tools were used and which intermediate steps led to a solution? How can the teachers respond to a learner's question that arises while using a computer tool? Little is available to answer this beyond interacting directly with the computer and performing a few manipulations to understand the tools' state.

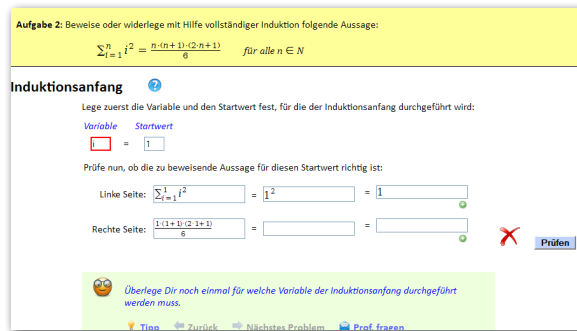
This paper presents SMALA, a web-based logging architecture that addresses these problems by recording, analyzing and representing user actions. While respecting the learner's privacy, the SMALA architecture supports the teachers by offering fine-grained representations of the learners' activities as well as overviews of the progress of a classroom.

1 Learners' Actions and the Perception of Teachers

Interactive learning tools offer rich possibilities to students when learning mathematics. On the one hand, they offer unprecedented possibilities to explore dynamic representations of the mathematical concepts: they allow students to discover the domain's objects and the rules that hold between them as much or as little as they want. Examples

of such learning tools include function plotters and dynamic geometry systems. Another example is pictured on the right: the tool Squiggle-M [Fes10] is used to explore relations between finite sets in order to learn about functions, injectivity, and surjectivity.





On the other hand, interactive tools for learning mathematics support the automated corrections of a broad range of typical exercises which allow learners to train and obtain the routine that stabilizes their mastery of the concepts. Trainers of this sort include

the domain-reasoner powered exercises of ActiveMath [Gog09] and many of the cognitive tutors [KAHM97]. Another example is pictured on the left, that of ComIn-M [RZ11]: it allows the learners to apply the classical workflow of proof by induction to proofs about number-theoretical formulæ.

Whenever computer tools are used in learning scenarios, the teacher plays a central role: he¹ explains the concepts by using representations and operations that the learner can also find in the learning tools. He invites the learners to use the learning tool: For example, he can indicate that a few exercises of the assignment sheet are to be done using the learning tools which they can find in the learning management systems. These usage incentives are not able to ensure the quality of the usage that enhances what the learners have acquired. Indeed, all sorts of risks appear when learning tools are used:

- An overwhelming cognitive load when introducing the learning tools (caused by the amount of technical details, for example).
- Too steep a learning curve to use the tool (when the students need to concretely apply the mathematical knowledge).
- Choice of exercises that lead to frustration due to unachieved exercises (for mathematical, technical, or other reasons).

These risks are challenges to teachers – the issue is underlined as insufficiently addressed in the report [AI11]: *The recurrent difficulties [...] call into question both the design of resources and the processes of their dissemination.*

To assess these risks, teachers must understand the students' usage of the learning tools. But providing detailed logs of the students' actions is not sufficient since the details may be overwhelming. The vast amount of data that could be produced by logging usually prevents the teachers using such a source to deduce any useful information about the students' learning processes.

User Requirements Approaches and tools are required that allow the efficient analysis and understanding of such log data. Learning analytics is the domain investigating methods and tools to support this.²

¹ In this paper, we shall use the feminine for the learner and the masculine for the teachers even though we mean both genders for both roles.

² Learning analytics is defined by G. Siemens as “the use of intelligent data, learner-produced data, and analysis models to discover information and social connections,

Teachers should be able to capture the overall progression of the learners in a class; being informed on the successes and failures for each exercise and each skilled acquisition aimed at. This has the potential to guide the lecture's content for such adjustments as revising a conceptual error commonly found or demonstrating a manipulation in more detail with the same concepts and representations of the learning tools.

In addition, we consider it important that the teachers complement the automatic assessment capabilities of the learning tools, making it a semi-automatic assessment [BHK⁺11]. Teachers should be able to see the detailed inputs a learner has made and the precise automatic feedback she received when requested to help. In the case of requesting help while using the learning tools, the learners should be able to formulate a request for help linked to the list of events that occurred until they needed assistance so that the teacher can analyze what the learner did and suggest effectively what actions to take next (in both computer-technical and mathematical terms).

In order to realize these objectives, teachers need a logging infrastructure which is the focus of this contribution.

Technical Requirements The purpose of this research aims at serving teachers in universities to support the use of richly interactive learning tools, typically of client-based applet-like tools which have not been designed with action logging in mind. We aim to insert an architecture for logging into the widespread infrastructure of learning management systems (LMS): those university-central systems that are used for coordination of courses and which each student regularly visits. We aim our development to not require an LMS change; indeed we have often met such a desire to be impossible to satisfy in university wide learning management systems.

Thus, one of the basic technical challenges is that of enabling teachers, who are privileged users of the LMS, to provide their learners with methods to start the learning tools from the LMS so that identified logs are received.

Other technical challenges revolve around the display of relevant information about the learning actions to the teachers in a way that is easily accessible and navigable. The learning tools, the LMSs, and the log views should be web-based and allow the servers to recognize the identity of learners and teachers.

At the Edge of Privacy A major concern of learning analytics is the set of regulations about the users' privacy. Indeed, the usage of such a monitoring tool may be turned into a powerful watching tool if not used carefully. Moreover, we acknowledge that a part of the students we have met are bothered using a tool where each of the attempted solution paths are always visible.

In comparison to log-views that irreversibly show all steps of the problem solving process, the classroom based usage where teachers and assistants can come and see the current state of the learning tool generally allows the student to cancel (and thus hide) erroneous steps that are irrelevant to a teacher question.

and to predict and advise on learning." in <http://www.elearnspace.org/blog/2010/08/25/what-are-learning-analytics>.

Thus, to support some free choice of disclosure of the students, we set forth the following principles that respect Germany's and EU's laws on privacy:

- The log of a session is only associated to an identifiable person when that individual expressly consents to being identified, i.e. when requesting help.
- The students always have the possibility to opt-out of the log collection.
- The information recorded is transparent to the learner.

These principles do not prevent all sessions of the same learner being grouped together, as long as it remains impossible for a teacher to associate a person with the log view of a session. As we shall describe below, this will be addressed by presenting the learners' *pseudonym*, a barely readable number derived from the name. We acknowledge that teachers would still be able to track regularly by remembering the pseudonym (or by many other means), but we explicitly warn the teachers that such is the start of illicit monitoring.

These principles respect the privacy laws in a same manner as the widespread Twitter or Facebook widgets in web-pages: they do not require a supplementary privacy agreement by the students since the log information that is collected and made available does not contain personal information.

Again similarly to these services, the privacy disclosure is agreed upon when the learners explicitly decide to do so: in the case of such services, this implies registrations, in the case of SMALA, it is when requesting help.

2 State of the Art in Logging User Actions

Logging users' actions can be done in multiple ways; in this section we outline existing approaches that are described in the current research literature. They revolve around two axes: the methods to integrate learning tools in learning management systems and the log-collection and log-view approaches.

2.1 Standards for Integrating Learning Tools

The widespread SCORM packaging standard³ allows makers of learning tools to bundle a sequence of web pages that use an API for communication with the LMS. A more recent derivative of SCORM is Common Cartridge, which extends it with IMS Learning Tools Interoperability IMS-LTI⁴: this specification allows the teacher of a module in an LMS to publish enriched forms of links which carry the authentication.

Both SCORM and Common Cartridge provide basic infrastructures for the web integration of learning tools. However, their logging capabilities in terms of collecting and analysing usage data is very limited. Supported log views typically show tables or counts inside the LMS. For a teacher to be able to evaluate the progress of one learner, he needs to see a less abstract view, more resembling the view the learner had when performing a learning activity.

³ The Shareable Content Object Reference Model standard emerged from ADLnet about 10 y. ago. See <http://www.adlnet.gov/Technologies/scorm/default.aspx>.

⁴ The Learning Tools Interoperability specification is an emerging standard, see <http://www.imsglobal.org/toolsinteroperability2.cfm>.

Multiple other standards have been realized to provide single-sign-on infrastructures: [GRNR09] describes many of these infrastructures and conclude with the proposal of yet another approach to authentication.

2.2 Research Around Log Collections

There are various approaches for collecting logs of user activities and making them available via suitable views. In the following, we will characterize the logging approaches by the level of detail of the collected data, the semantic content that is available, and the analysis capabilities that are offered.

Jacareto: The most detailed approach for log collection is to record each of the user's actions and present these as a *replay* of the learning tool's user interface. This approach has been investigated in a software project called Jacareto [SGZS05]: apart from input events such as mouse clicks in the learning application tool-specific *semantic events* are stored in a recording file. Once the teacher obtains the record file, he can analyze the solution process qualitatively – either by looking at a hierarchical view of the events, or by replaying the events and observing the learning application. However, quantitative analysis of a large number of recordings is not supported, and remote logging is not yet available.

FORMID: The research project FORMID [GC06] aims at applying learning scenarios and view logs based on these scenarios. Each scenario is implemented as a script that specifies a sequence of learning activities to be followed by learners. Additional monitoring facilities enable the teacher to observe learning activities and provide support in real time. In this approach, the semantic content of the logging data is quite rich, and sufficient analysis capabilities are offered for assessing the learners' progression but only within the given scenario.

LOCO-Analyst: The approach of LOCO-Analyst [JGB⁺08] intends to support the teacher in analyzing learning processes in order to optimize and revise content elements in online learning courses. For this reason, LOCO-Analyst focuses on online learning activities such as text reading and obtaining scores when solving multiple-choice exercises. Based on semantic web technologies, LOCO-Analyst enhances logging data by semantic annotation and provides various analysis services and graphical representations of the collected data. As opposed to Jacareto, the level of detail of the logging data is quite low: it mostly considers high-level events such as page views or successful exercise solutions.

Log Repositories: Learners' activities logs are the basic input for the PSLC DataShop⁵ and for a similar initiative in Kaleidoscope [MMS08]. These initiatives are infrastructures for collecting logs of learning for their massive evaluations for such purposes as data-mining to experimentally measure e-learning theories. These logging repository initiatives are researcher-oriented: they bring together quantities of logs in order to formulate hypotheses about the learners' actions. They are not applicable for teachers, and often do not work on live streams of data.

⁵ For the PSLC DataShop see <https://pslcdatashop.web.cmu.edu/>.

2.3 Conclusions from Literature Review

From this review of the existing literature we conclude that no current standard nor widespread learning management system offers sufficient support for the deployment of logging-enabled learning tools run on the client or even for any interaction-rich learning tools.

We also conclude that the approaches proposed by state-of-the-art research initiatives provide interesting logging features but only few approaches are flexible enough to log semantically-rich events in an adequate level of detail. Customizable views that offer an efficient display of the logging data enhanced by flexible analysis capabilities are still subject to ongoing research.

Figure 1 summarizes how the existing logging approaches can be assigned to the dimensions *amount of detail*, *semantic level*, and *analytics support*. As shown in the diagram, the SMALA architecture that we present in the following chapter aims at high values in all three dimensions, with an intent to cover different amounts of details, and thus represents a new category of logging system.

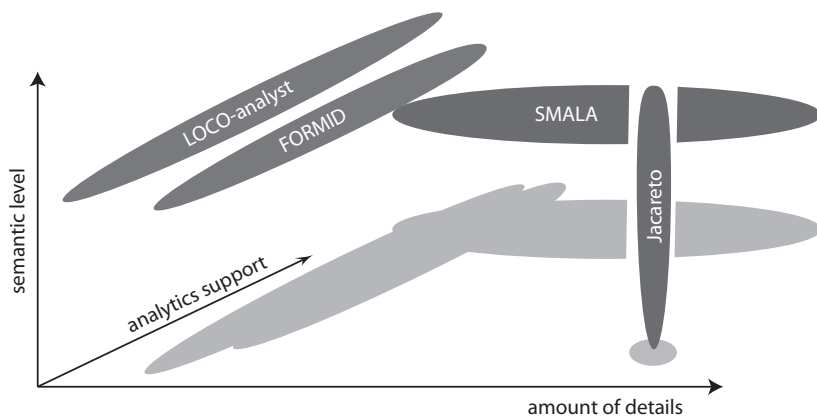


Fig. 1. The various logging systems along three dimensions

3 The SMALA Architecture

The contribution of our paper is an architecture for logging and analyzing learners' activities and solution paths. SMALA, SAIL-M's Architecture for Learning Analytics, responds to the requirements stated above and is realized as a service-oriented web application. Figure 2 depicts the SMALA system architecture where the sequence of components in the creation, deployment, initialization, usage, assessment, logging, and log observation is numbered.

SMALA's core component is the SMALA logging service that is responsible for receiving events from the learning tools and storing them persistently in the logging database. Authorized teachers can retrieve the recorded events from the SMALA web server via suitable views and can analyze the learning activities performed by individuals and groups of learners.

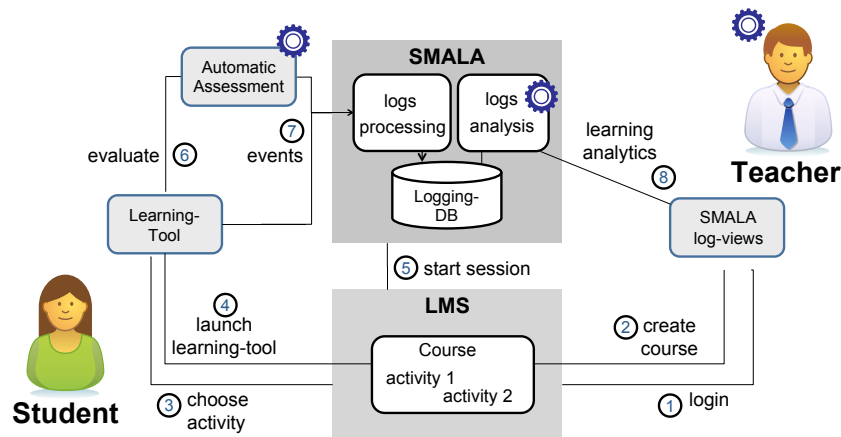


Fig. 2. The SMALA architecture. Gear wheels indicate actors and components with analytical reasoning.

Being a generic logging infrastructure, SMALA offers various interfaces for integrating concrete learning tools into the architecture. These interfaces are related to the authentication, definition, and handling of tool-specific semantic events. The following sections provide a detailed description of these interfaces and their usage.

3.1 Software Components and their Interactions

The *SMALA server* is a web server which, on the one hand, stores and displays log event streams and, on the other hand, is able to deliver the learning tools to the learners. In order to enable a learning tool to use the SMALA infrastructure, the tool must be registered with one or more *learning activities* in the SMALA environment. This means SMALA knows about the tool and the corresponding activities via XML configuration files and is able to verify that it is allowed to use the SMALA logging service by an application specific key and a whitelist of allowed host URLs. In principle, every learning tool based on web technologies, such as Java applets or AJAX web applications, can be deployed and make use of SMALA's logging facilities.

Typically, SMALA-enabled learning tools are made available to the learners using an LMS (see Section 5) and thus, can be accessed as part of the learning materials available from the LMS course page. In order to use a learning tool, course participants only have to log into the LMS using their usual LMS user account and launch the tool. When starting up the tool, the LMS requests the creation of a new SMALA session for the current LMS user. This session is associated with all subsequent communication requests that occur between the learning tool and the SMALA logging service. As soon as a valid SMALA session is established, the learner can work with the tool in her local web browser. While doing so, all relevant interactions are wrapped as semantic events and

sent to the logging service to store them in the database. Relevant interactions include both the user actions, the assessment results, and the feedback provided by the learning tools as a reaction to the users' activities. Identifying the relevant interactions is the responsibility of the learning tool designers and requires pedagogical expertise and a thorough understanding of the tool's application domain. Based on the SMALA log objects knowledge structure as described in the next section, the tool developers can model custom event objects as needed and add logging functionality to their tools by simply sending these events to the SMALA logging service.

The *SMALA logging service* is implemented as a Java servlet and accepts all logging requests containing a valid SMALA session identifier and a serialized event object. The obtained event object is deserialized by the SMALA logging service and persisted to the SMALA logging database. Event objects have a *type* and are associated with the current *user session* and *learning activity*, i.e. a learning tool configured for a specific LMS course. All learning activities have to be registered with SMALA using XML configuration files and can have so-called *trigger classes* attached to them. If incoming events for these activities match the event type specified by these trigger classes, the triggers are activated and can handle the events appropriately. Currently, our semi-automatic assessment tools make use of this functionality for handling manual feedback events. When the SMALA logging service receives a manual feedback event object, it activates the configured *Send Mail trigger* which sends an email to the designated teacher or tutor, containing the learner's question, her email address, and the associated event data. So as to ensure the pseudonymity of the data, the email address is, then, removed from the event.

Feeding events into the SMALA logging service is completely transparent to the end users of the learning tools. The learners' main focus is on using the learning tools and getting direct feedback from the tool's automatic assessment component. Teachers, however, can make use of SMALA's log views and optional tool-specific summary views on the recorded data (see section 5). These log views and summary views are presented using Java Server Pages (JSP) that query and render the event data from the SMALA logging data base. By allowing the deployment of tool-specific event renderer classes, not only *general event data* such as the timestamp and a textual event description is displayed, but also *tool-specific event data* such as rewritings, error messages or screenshots from the tool's user interface can be shown (see figure 3). In the same way, tool-specific summary views and corresponding analyzer components for processing the semantic event data can be plugged into the SMALA infrastructure and integrated into the standard SMALA log views. The analysis done by these summary views offer the teacher a usable view to support his analysis and decision making.

SMALA offers a flexible *authorization mechanism* to configure roles and access rules per activity: an LMS provenance and a signature enables an authorized *user source*; tutors (that can view log and deployment instructions) are authorized by obtaining identity from external identity providers such as Google and Facebook.

3.2 Log Objects Knowledge Structure

Log-event objects form the basic information entity that describes the users' actions. The semantic-event-based data sets are grouped by *activity* and by *session* and form the starting point of the analytical process.

An activity is SMALA's concept for a learning tool that is offered from within a given course. It is configured with an authorization realm for teachers (a few external accounts) and for students (an integration method into an LMS). Each activity contains sessions which are a stream of log-events for a user.

Log-event objects have been designed with extensibility in mind since their semantics are very tool specific. The basic types include question events, image events, and action events. Common attributes of these basic types include the session-identifier, the timestamp, as well as the exercise name. Based on the exercise name the teacher can identify different parts of the learning activity. Tool makers can refine their types for each of the learning tools, extending the basic types described above. The events can thus include the user's input (such as the OpenMath representations in the case of ComIn-M), the exercise state (such as the formula of the function being plotted in Squiggle-M), or even a URL to reconstruct the exercise state.

The serialization of the events transmitted to the server uses a generic XML format that is able to contain the tool specific logging information in form of key-value pairs, strings, numbers, dates and binary blobs, making the learning tools able to build their custom log event streams out of these datatypes. The events sent from the client are decoded on the server where they are validated and stored in the database using the Java Persistence Architecture.⁶ This form of storage, close to the Java objects nature, allow sophisticated queries to be formulated so that log-views are carefully engineered to be relevant to the analysis of the students' activities for each learning tool.

3.3 Availability

SMALA is publicly available from http://sail-m.de/sail-m/SMALA_en. Its source code (under the Apache Public License), technical documentation, and a link to the production and development servers with some demonstration parts are also linked there.

4 Making the Learning Tools Available

SMALA relies on a continuous identification of the users of the learning tools. The natural way to do so is to integrate the learnings tools in the LMS within the course page where students find their learning materials.

Each activity is established by the SMALA team: it is made by configurations of the roles, deployment explanations, and authorizations. A teacher then simply

⁶ The Java Persistence Architecture is an abstract API for object-relational mappings, see <http://jcp.org/aboutJava/communityprocess/final/jsr317/>.

adds a link to the learning tool to his course page using code fragments suggested on the deployment explanations. As a result, the learning tools can be started one or two clicks away from the main course page in the LMS, and thereby allow teachers to easily promote the tool during the lectures or on assignment sheets.

5 Types of Log Views and their Usage

Access to the logs starts with a *dashboard view* giving an overview of the recent activity with the learning tools. From there, one can get a list of users and a list of recent sessions. The users are not presented by name, but by their pseudonym, which we expect to be a number impossible to remember by teachers; each of them links to the list of sessions of that user, then to the detailed session-view. From the dashboard, one can also access overview pages giving a global analysis of the class performance.

5.1 Session Views

Sessions are displayed as a chronological sequence of user interactions in all detail. Each of the interactions' types are displayed with a custom representation.

This enables the session log view to display, for example, each input the learner has made, and each fragment of feedback she has received. The purpose of this session view is to document the steps that were performed by the learner in the problem solving process and make it reproducible for the teacher.

This works well with learning tools that are made of dialogs where the learner submits an input and receives feedback. An example is shown in the figure above: it shows a logging session using the learning tool ComIn-M where each of the induction proof constituents are presented. Such detailed log views enable lecturers to reproduce all actions a learner has made during the problem solving process. However, this approach works less well for learning tools oriented towards direct manipulations such as Squiggle-M: log views then become long lists of small actions whose textual rendering is hard to read; e.g. *created point P1 in domain 1, created point P5 in domain 2, ..., linked P1 to P5*).

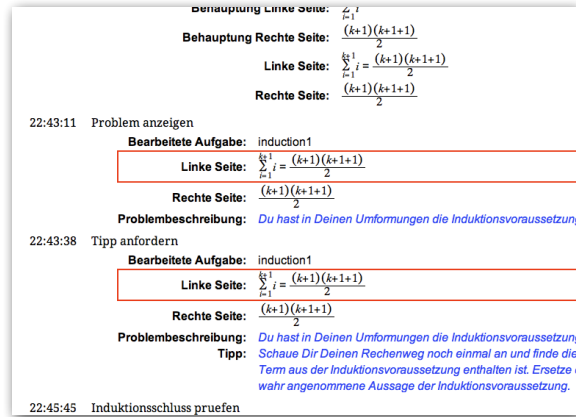


Fig. 3. The log view of a ComInM session.

5.2 Requests for Help

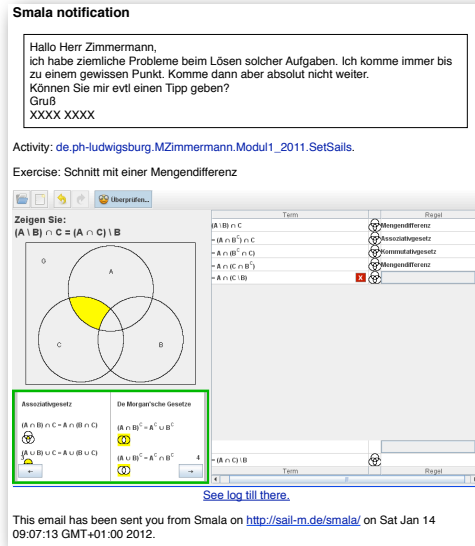


Fig. 4. An actual tutor request email when using SetSails [HS11] received during the evaluation.

that a function has been repeatedly used wrongly) therefore being able to guide her accordingly. The semi-automatic feedback paradigm of [BHK⁺11] can be applied fully, enabling teachers to help students use the tool effectively and provide hints that go beyond the automated guidance of the learning tools. An example tutor request email is shown in figure 4.

5.3 Summary Views

The two log views of the previous sections support teachers to help individuals or get an idea of sample solution paths, but they are much less useful when trying to obtain a broader overview of a whole class's advances in the usage of the learning tools.

The most basic summary view is a list of all events of a given type. These allow teachers to find interesting sessions or detect a lack of achievement. Another summary is provided by a dashboard with a timeline graph of the usage activity.

A special type of interaction realizes the semi-automatic assessment approach [BHK⁺11]: that of a request to the tutor formulated from within the learning tool. This interaction submits a special event to the SMALA server, containing the request of the student and a snapshot representing the current state of the tool (e.g. a screen copy). As soon as the SMALA server receives this event, the request is forwarded to the responsible teacher by email along with a URL to access the session until the moment of submitting the request.

These tutor request e-mails allow the teachers to grasp what the learner did (e.g. observing

Bearbeitete ComIn-M-Aufgaben:

Anzahl an bearbeiteten Aufgaben: 105
davon richtig gelöst: 97
davon falsch gelöst: 8

Aufgabe	Gesamtanzahl	richtig	falsch	Lösung Prüfen	Tipps	Tutoranfragen
induction1	33	21	12	177	31	3
induction10	5	4	1	21	3	0
induction2	21	16	5	99	24	1
induction3	7	12	5	39	3	1
induction4	6	11	5	46	3	0
induction5	10	14	4	75	14	2
induction6	5	7	2	25	4	0
induction7	6	3	3	23	0	1
induction8	8	0	8	58	10	1
induction9	4	9	5	25	1	0

Fig. 5. A table of progress of the learners.

Another summary view is the display of the *success* or *failures* in the usages of the learning tools for each of the exercises that make up a learning tool. An example in figure 5 shows a high involvement in attempting the first few exercises but clear difficulties for the remaining exercises as well as lack of attempts. The teacher could, from such a display, analyze and conclude a lack of understanding of the concepts, lack of engagement, or technical challenges; each of these hypothetical diagnoses could be answered by an in-class demonstration of success in one of the advanced exercises.

6 Evaluation of Learning Tools using SMALA-Logging

This research has been made in parallel to several evaluations in which both the learning tools' qualities and the logging approach based on SMALA were evaluated.

An early evaluation was run in the summer of 2011 in order to get first feedback from the students and teachers at Karlsruhe University of Education: about forty students brought their laptops in two successive lectures and launched the MoveIt-M learning tool [Fes10] for exploring plane isometries and their composition. This first evaluation showed that obtaining a reasonable network bandwidth for launching the tool (20 MB download) may represent a challenge to other environments. Missing bandwidth may stimulate students to find workarounds such as choosing offline versions of the learning tools. Nonetheless, this first phase showed that transmitting logs is technically easy.

Summative evaluations were run in the winter term of 2011-2012. In the basic mathematics courses of math teachers education three learning tools were deployed and used: Squiggle-M, SetSails [HS11], and ComIn-M. The logs witness the usage of the learning tools by 156 users having run 965 sessions yielding 24655 events. The logs have been complemented by a questionnaire filled out on paper and teacher interviews.

The evaluations have also shown that the transparent integration in which the learning tool is started by the learning management system and the logs are collected under pseudonym works fairly well and is acceptable to the students' privacy requirements. Indeed, only a small portion (less than 2 %) insisted on having no log collection. Teachers that collaborated with us found it acceptable to expand the privacy circle when deploying such learning tools by enabling the learning management system's pages run by learners' browsers to transmit the user-information to the SMALA server. They made this decision because they acknowledged the usefulness and clarity of SMALA's mission.

6.1 Feature Usage Statistics

The central aspect of our evaluation was to measure the relevance of the semi-automatic assessment paradigm. The usage of direct tutor requests from within the learning tools has been low (only 11 requests). Asked why they did not use the tutor request feature, many students responded that the automatic feedback had been sufficient (26%). Most stated that they preferred asking peers (36%)

or tutors (26%) directly (thus waiting till they had the chance to ask a question personally), while a few responded that no help was necessary (11%), or that formulating the question turned to be hard (15%).

Although only few help requests were formulated (such as the one in figure 4), the requests that were actually submitted to the teachers could be answered: the detailed log views of the sessions have been sufficiently complete for them to answer questions with a precise hint that set the students on the right path. For this central workflow, the evaluation showed a successful setup.

A teacher who had seen the logs of ComIn-M commended the *screenshots of the users' input*. In fact, the logs do not contain screenshots, but MathML representations of the user's input. This comment shows that a faithful graphical display of the entire mathematical formulas is important to understand the students' solutions.

The usage of the logging feature for each of the teachers has been almost limited to responding to individual requests. When interviewed at the end of the project, almost all teachers said they would consider using SMALA logging views more intensively if richer summary views were offered. Indeed, we focused on the log views to support individual requests, expecting overviews to be obtained by sampling a few sessions. As a result such summary views as the table in figure 5 were implemented late in the development cycle. Our interviews also indicated that the separation of statistics *by exercise* seems to be an important basis of most log views.

Analyzing the web-server logs (1813 page views) showed that the students made use of the *myLog* links displayed aside of each tool launch. This feature is offered to support the transparency requirement. It allows each student to review her activities in the same format in which a teacher would see it.

6.2 Technical Challenges

The development of SMALA towards its sustained usage in day-to-day classrooms has been iterative, based on the feedback of the teachers and the students. It has faced the following technical challenges:

Genericity The SMALA server was designed to serve both web-applications and rich-client applications and it succeeded in doing so. Both ComIn-M (a web-server that sent all its logs to SMALA) and Java Applets (SquiggleM, MoveItM, and SetSails) could start and send identified logs to the SMALA server. Although the learning tools send log-events of different types through different connection methods, the resulting log displays are delivered in a unified user interface.

Scalability The choice of persistence layers such as the Java Persistence Architecture has been an important key to ensure the scalability to several thousands of log entries. For some display methods, large amounts of events still have to be inspected and it is still a matter of a few seconds (e.g. to display thousands of log entries). However, SMALA has sometimes been at the edge of performance even though it only handled a handful of courses. For example, listing long sessions

or listing all uploaded screenshots takes several minutes during which the data is extracted from the database and converted for display. Summary views, in particular, need to strictly enforce the rule that their results are fetched using statistical queries to the database, which the tool developers enter using the JPQL language instead of iterating through numerous events.

An aspect at which the persistence libraries have shown to be too fragile is that of the evolution of the database schema of the log events. While creating an SQL schema based on the Java properties is transparent and effective (including support in IDEs to formulate queries), the persistence architecture offers almost no support to subsequently upgrade the SQL schema.

A modification such as raising the maximum size of a property (hence the size of an SQL column) needs either to be done manually in SQL or by an externalisation followed by a rebuild of the database. We ran the rebuild process at almost every deployment, the last taking about three hours. This process is, however, risky, and only thorough testing revealed that some information was lost in the process. The web nature of the log views, accessible by simple URLs allowed an easy test infrastructure: a simple web-page described each piece of information expected at each log-viewer URL. This allowed us to ensure that there are no unintended effects of the new versions' deployment.

7 Conclusion

In this paper we have described an approach to convey to practicing teachers what their students have been doing with learning tools. The approach relies on the usage of a software whose role is both to make available the learning tool in the learning management system and to display log views representing the learner's activity. This approach turns the automatic feedback of the assessment tools into more intelligent tools which can exploit the teacher's knowledge. The implemented toolset conveys broad enough information to watch the learners' overall progress, and to respond to individual queries.

Below we present research questions that this approach opens.

7.1 Open Questions

The involvement of the teachers in the usage of the learning tools remains a broad area of research which this paper contributes to. It is a subtle mix of mathematical competency, technical skills, and pedagogical sense – all of which are addressed by a teacher support tool. It is bound to the teachers' representations of the mathematical knowledge that students have in mind, the meaning they attach behind them and behind the operations with them.

The *instrumental orchestration* [DT08] is a model of the role of the teacher who uses the instruments (the tools) to let himself and the students (the orchestra) play the mathematical music. It mostly studies the usage of the learning tools in class, while students often use learning tools in the comfort of their

homes: the learning tools are there, precisely, to let the learners deepen or discover at their own pace, away from the tight rhythm of the lectures.

The research of this paper contributes to letting the teacher perceive this activity. In this paper we have proposed a few solutions relying on both detailed and summary views. Does it imply a high level of diagnostics? The survey [ZS06] indicates that higher level indicators such as the learners' mastery level and misconceptions are much desired. We note, however, that they would only be useful if they can be trusted by teachers to be complete and precise; we are not sure that this is easily achievable, and indeed the paper highlights this difficulty in its conclusion. Sticking to more concrete events, such as *typical feedback types* might be easier to implement, easier to represent to the teacher, directly linkable to steps in session actions, and thus much more trustable.

We have introduced SMALA in relatively traditional didactical settings in which the learning tools are mostly used for the purposes of exercising. This is an important aspect but other didactical setups can leverage a logging architecture.

Among didactical setups is the exploitation of the *own log* feature beyond the mere transparency requirement. Displaying the individual logs of a selected learner, this feature can be used by the student as a personal log to be reminded of the competencies she has already acquired, much similar to the learning-log approach. What needs to be achieved to encourage and make possible such practice? Should links be made from a personal diary into concrete sessions?

As another example of a different didactical setup, Erica Melis suggested to display the logs of past uses of the learning tool overlaid on the *behaviour graph* of the Cognitive Tutor Authoring Tools [AMSK09] and to use this in coordination with the learning tools. This approach would constitute a discussion tool that allows the teacher to explain the possible avenues to solve the exercise, changing the state of the exercise and discussing what was correct or what was wrong. An approach and architecture such as SMALA makes such a tool possible.

Acknowledgements This research has been partly funded by the Ministry of Education and Research of Germany in the SAiL-M project. We thank our student assistants Torsten Kammer and Michael Binder for their active support.

References

- AI11. Michèle Artigue and International Group of Experts on Science and Mathematics Education Policies. Les défis de l'enseignement des mathématiques dans l'éducation de base. Number 191776 - UNESCO Archive, IBE Geneva, 2011.
- AMSK09. Vincent Aleven, Bruce McLaren, Jonathan Sewall, and Ken Koedinger. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 20, 2009.
- BHK⁺11. Christine Bescherer, Daniel Herding, Ulrich Kortenkamp, Wolfgang Müller, and Marc Zimmermann. E-learning tools with intelligent assessment and feedback. In Sabine Graf, Fuhua Lin, Kinshuk, and Rory McGreal, editors, *Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers*, page IGI Global. Addison Wesley, in press 2011.

- DT08. Paul Drijvers and Luc Trouche. *From artifacts to instruments: a theoretical framework behind the orchestra metaphor*, volume 2, chapter Research on Technology and the Teaching and Learning of Mathematics, pages 363–392. Information Age, Charlotte, NC, USA, 2008. K Heid and G Blume (Eds).
- Fes10. Andreas Fest. Creating interactive user feedback in DGS using scripting interfaces. *Acta Didactica Napocensia*, 3(2), 2010.
- GC06. Viviane Guéraud and Jean-Michel Cagnat. Automatic semantic activity monitoring of distance learners guided by pedagogical scenarios. In *Proc. EC-TEL 2006*, volume 4227 of *LNCS*, pages 476–481. Springer Verlag, 2006.
- Gog09. George Goguadze. Representation for interactive exercises. In Jacques Carette, Lucas Dixon, Claudio Sacerdoti Coen, and Stephen M. Watt, editors, *CalcuIemus/MKM*, volume 5625 of *Lecture Notes in Computer Science*, pages 294–309. Springer, 2009.
- GRNR09. J. Fontenla González, M. Rodríguez, M. Nistal, and L. Anido Rifón. Reverse oauth: A solution to achieve delegated authorizations in single sign-on e-learning systems. *Computers and Security*, 28(8):843 – 856, 2009.
- HS11. Daniel Herding and Ulrik Schroeder. Using capture & replay for semi-automatic assessment. In D. Whitelock, W. Warburton, G. Wills, and L. Gilbert, editors, *CAA 2011 International Conference*, 2011.
- JGB⁺08. Jelena Jovanovic, Dragan Gasevic, Christopher Brooks, Vladan Devedzic, Marek Hatala, Timmy Eap, and Griff Richards. LOCO-Analyst: semantic web technologies in learning content usage analysis. *International Journal of Continuing Engineering Education and Life Long Learning*, 18(1):54–76, 2008.
- KAHM97. Kenneth Koedinger, John Anderson, William Hadley, and Mary Mark. Intelligent tutoring goes to school in the big city. *IJAIED*, 8(1), 1997.
- MMS08. Erica Melis, Bruce McLaren, and Silvana Solomon. Towards accessing disparate educational data in a single unified manner. In P. Dillenbourg and M. Specht, editors, *ECTEL 2008*, volume 5192 of *LNCS*, pages 280–283, Berlin, 2008. Springer-Verlag.
- RZ11. Sandra Rebholz and Marc Zimmermann. Applying computer-aided intelligent assessment in the context of mathematical induction. In *eLearning Baltics 2011*. Fraunhofer Verlag, Stuttgart, 2011.
- SGZS05. Christian Spannagel, Michaela Gläser-Zikuda, and Ulrik Schroeder. Application of qualitative content analysis in user-program interaction research. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 6(2), 2005.
- ZS06. C. Zinn and O. Scheuer. Getting to know your student in distance-learning contexts. In *Proc. of the First European Conference on Technology Enhanced Learning (EC-TEL 2006)*, volume 4227 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2006.