How did the e-learning session go? The Student Inspector[1]

Oliver SCHEUER and Claus ZINN [2]
German Research Center for Artificial Intelligence (DFKI)

Abstract. Good teachers know their students, and exploit this knowledge to adapt or optimise their instruction. Traditional teachers know their students because they interact with them face-to-face in classroom or one-to-one tutoring sessions. In these settings, they can build student models, *i.e.*, by exploiting the multi-faceted nature of human-human communication. In distance-learning contexts, teacher and student have to cope with the lack of such direct interaction, and this must have detrimental effects for both teacher and student. In a past study we have analysed teacher requirements for tracking student actions in computer-mediated settings. Given the results of this study, we have devised and implemented a tool that allows teachers to keep track of their learners' interaction in e-learning systems. We present the tool's functionality and user interfaces, and an evaluation of its usability.

Keywords. Methods, tools and techniques for effective evaluation of cognitive, meta-cognitive and affective issues, distance-learning contexts, log data analysis.

1. Motivation

There is an increasing use of technology in education. Industrial-strength e-learning systems provide teaching material that can be accessed anywhere and anytime, especially outside classroom sessions. While e-learning systems can offer many advantages – the presentation of high-quality multimedia content, the self-paced exploration of learning material, the engagement of learners in interactive exercises, the opportunity to communicate and collaborate with other learners *etc.* – there are also some dangers. Learners may not be able to use the new technology effectively, potentially by spending more time and mental resources in struggling with the technology than with the actual learning process. Also, parts of the technology could be abused, *i.e.*, the misuse of the communication and collaboration tools for extensive off-topic chit-chat. The absence of teacher authority may in particular harm learners with limited abilities to self-direct their learning.

Only teachers that are aware of these and other dangers can intervene – if only there were a STUDENT INSPECTOR that would allow teachers to getting to know their learners in distance learning contexts. A tool that could capture learners' individual learning paths (and dead-ends), could keep track of learners' knowledge, higher-level competences, or the lack thereof, and that could list their problems with the subject domain taught.

[1]The work presented in this paper is supported by the European Community under the Information Society Technologies programme of the 6th Framework Programme for RTD – project iClass contract IST-507922.

[2]Correspondence to: Oliver Scheuer, German Research Center for Artificial Intelligence (DFKI), 66123 Saarbrücken, Germany, Tel.: +49 681 302 5377; Fax: +49 681 302 2235; E-mail: oliver.scheuer@dfki.de.

We are devising a STUDENT INSPECTOR that analyses log data gathered from e-learning environments and that presents the results of these analyses to teachers. In this paper, we present a prototype implementation that incorporates teachers' needs by combining state-of-the-art log data processing and presentation with AI-based analyses.

2. Background

Some e-learning systems support the tracking, analysis, and reporting of learner-system interactions. Commercial course management systems like WebCT track, *i.e.*, the date of the first and most recent sign-on for each student in the course; the usage of the conference tool (number of articles read, number of postings and follow-ups) to measure involvement and contribution; the use of other tools (*e.g.*, notes, calendar, e-mail); and visits to the content material (page visits, view times) to monitor course usage, usefulness, and quality. Data is then aggregated and visualised in tables or with bar charts [3,7].

The ASSISTment system is a web-based tutoring system in which students work on a set of test items. Correct answers lead to the next item, incorrect ones open a small tutorial session where the original problem is broken down into a series of scaffolding questions aimed at guiding the student. The system offers reports on students' performances. For enabling a fine-grained assessment of students' abilities, problems and scaffolding questions are annotated with the skills they require. The generated reports present, for instance, the time spent in the system, the number of items worked on, the number of items solved, the number of hints requested, strongest/weakest skills of a class or of an individual student, and change of performance over time [2].

DataShop (`www.learnlab.org/technologies/datashop`) is a central repository that offers learning scientists services to store and analyse their data. Generated reports present, *e.g.*, learning curves by student or skill and detailed data for a given problem step (submitted answer, assessment of answer, system response *etc.*).

Merceron and Yacef pursue a data mining approach to student activity tracking [5]. Their tool for advanced data analysis in education (TADA-Ed) offers, for instance, an association rule algorithm that can be used to identify sets of mistakes that often occur together, or a decision tree algorithm to predict final exam marks given students prior usage behaviour. Usage of the tool requires basic knowledge in data mining (and machine learning as its technological basis), and therefore, TADA-Ed is a tool that is primarily directed at educational researchers rather than professional teachers.

Interaction analysis tools are usually seen as an extension to e-learning environments; hence, their reusability is very limited. Moreover, the quality of their analyses crucially depends on the richness of the data logs that e-learning systems produce, which in turn depends on their content representations and the metadata describing them.

Besides the opportunities and limitations of given system characteristics, there are also the desiderata, priorities and preferences of teachers for tracking, analysing, and presenting learner-system interactions. In summer 2006, we conducted a study to better understand teachers' requirements for a tool that helps them understanding their students in distance-learning contexts. Their answers to this questionnaire revealed a clear preference for pedagogically-oriented measures, namely, overall success rate, a learner's mastery levels and typical misconceptions, the percentages of exercises tackled, and amount of material read. Those indicators are quite traditional, and naturally, easy to interpret and work with. Historical information, *i.e.*, past course coverage and activities, learners' navigational style, and social data attracted less interest, and were judged as too cumbersome to inspect or too time-intensive to interpret (for details, see [8]).

The design of our STUDENT INSPECTOR also profited from the experiences we gained with ActiveMath [4], a learning environment that aims at delivering tools for personalised learning in mathematics. To inform the selection of content, ActiveMath has access to learner modelling and a rich repository of learning objects that are annotated along various dimensions (*e.g.*, the competences they train, the domain prerequisites that they require, their difficulty). Many learner-system interactions are logged, and we have gathered rich data sets that we can analyse along various lines of research.

The true potential for action analysis was realised, however, within the iClass project [6]. iClass aims at empowering teachers with tools to better monitor and manage the progress of their classroom students. iClass also wants to empower learners through self-regulated personalised learning: learners should take more responsibility for their learning, and for this, they need to be supported by tools that inform their decision making (*e.g.*, identification of learning goals, choice of learning scenarios, time scheduling). In this respect, the STUDENT INSPECTOR will be of use for teachers *and* learners.

3. Architecture

Fig. 1 depicts the STUDENT INSPECTOR's architecture. Its organisation into layers minimises dependencies between components and enhances the connectivity of e-learning systems. There are four layers. The **database layer** serves to persistently store interac-

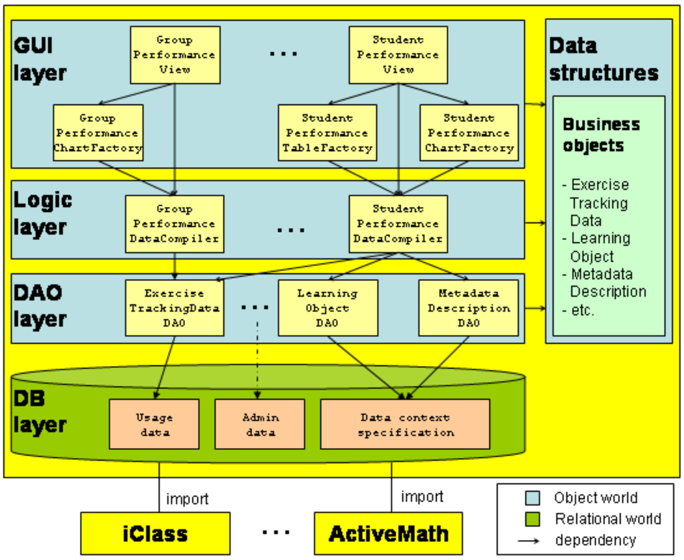

Figure 1. The STUDENT INSPECTOR's architecture.

tion data. Three kinds of data can be distinguished: (i) data describing individual learning objects by means of their metadata profile, the composition of learning objects into course structures, and specifics of the e-learning system (*e.g.*, use of tools); (ii) actual usage data reflecting learners' activities with the e-learning system (*e.g.*, page views, exercise steps, glossary queries) and system's evaluation of these activities (*e.g.*, exercise success rate, diagnosis of learners' misconceptions); and (iii) administrative data (*e.g.*, learner groups, task assignments). The **data access object layer** converts relational database entries into business objects, and *vice versa*. The Hibernate persistence framework (`www.hibernate.org`) enables this mapping, and also offers other persistence-related services. The **logic layer** defines the logic for the aggregation and compilation of objects, *i.e.*, the analysis of tracking data at the object level, given their intended presentations. The logic layer has also access to machine learning algorithms for more sophisticated analyses. The **GUI layer** consists of views, each of them encapsulating a functionality (*e.g.*, views for overall group performance, student performance along topics). This layer is supported by a program library implementing, *i.e.*, diagrams, charts, and tables.

4. Functionalities

The Student Inspector consists of three major components: a browser to explore data, an admin module to manage students and student groups, and to assign tasks to them, and an AI-based analyser to perform more sophisticated data processing. In this paper, we focus on the modules for browsing and AI-based analyses.

4.1. The Browser

The teachers from our study mostly care about learners' performance, misconceptions, and topic coverage [8]. The Student Inspector implements those and other aspects.

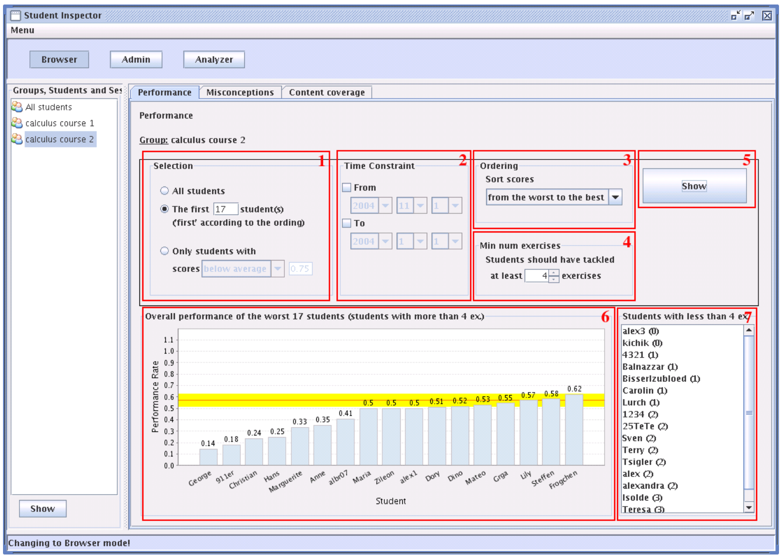

Figure 2. Identifying your classroom's performance.

Performance measurement. Fig. 2 depicts the STUDENT INSPECTOR's presentation of individual and group performance scores. The individual performance of a learner is computed by averaging over her test or exercise scores. Performance values are computed for learners that surpassed the threshold for the minimum number of exercises. Users can use this view to identify the average performance of a group, and consequently those learners that perform below or above the average. Users can focus their performance research on learners (see 1), or on the time of the learning session (see 2). Results can be ordered by student name or performance scores (see 3), and influenced by the exercise threshold (see 4). When these parameters are submitted (see 5), two pieces of information are computed: a chart displaying individual performances and the group's average (see 6), and a list of students not assessed given their insufficient exercise trials (see 7).

Learner Misconceptions. State-of-the-art e-learning systems like ActiveMath can diagnose learner input to exercises to identify typical errors or misconceptions. Fig. 3 displays the STUDENT INSPECTOR's analysis and presentation of such logs for a given learner group. With the pie chart teachers get access to the most frequent errors their learners

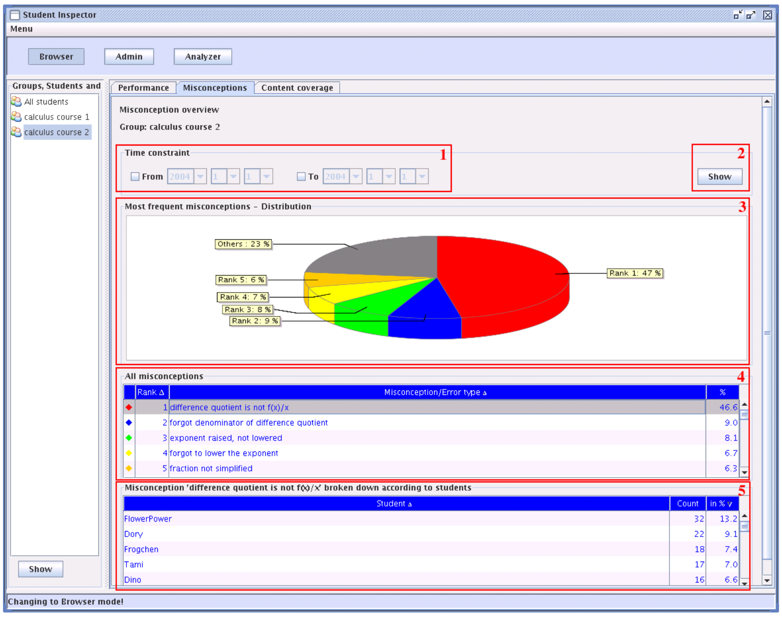

Figure 3. Browsing the misconceptions of your classroom.

showed when working with the e-learning system. In learner group "calculus course 2", *i.e.*, almost half of the learners had difficulties with the difference quotient (see 3). A table (see 4) gives access to a complete and sortable listing of errors as identified by the e-learning system. Moreover, teachers can select a misconception, *i.e.*, a table row, to obtain a list of all learners that made the corresponding error (see 5).

Topic Coverage. E-courses usually cover various topics. A calculus course, *i.e.*, might be about sequences, series, differentiation, and integration. Fig. 4 displays the STUDENT INSPECTOR's analysis and presentation of topic coverage, given a learner's exercise per-

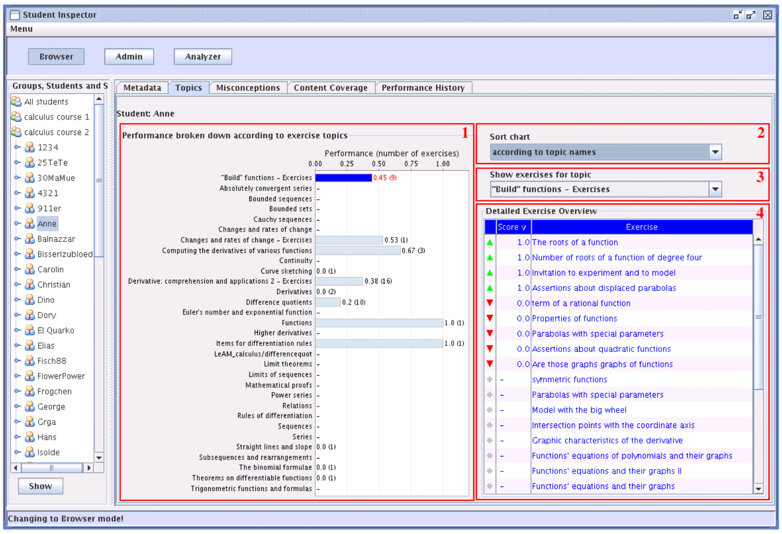

Figure 4. Browsing individual performances via topics.

formance for each of the topics. This view allows teachers to find, *i.e.*, weak and strong topic areas for a given student, here Anne (see 1, 2), and gives also access to the exercises Anne has tackled to achieve her performance score for a selected topic (see 3, 4).

4.2. The Analyser: Predicting Future Learning Outcomes

The analyser uses machine learning techniques to predict future learning outcomes. Teachers can ask the exercise recommender to suggest "more difficult" exercises to, say, challenge well-performing learners, or ask for "easier" exercises to, say, motivate below-average learners, given learners' knowledge state and competencies. Analyses can be run for student groups or individuals. To initiate the machine learning method, one first selects those metadata categories judged relevant from the list of available categories. Subsequently, a model is computed given all exercises that the student has tackled. Then, untackled exercises are classified within this model, yielding two tables: the first table shows exercises where a bad performance is expected, whereas the second table shows exercises where a good performance is expected. Expert users can display the model.

5. Evaluation

The Student Inspector has been evaluated. Similar to our first questionnaire, where we gathered teacher requirements, we have devised a second questionnaire to evaluate the implementation of these requirements. Given a series of tool screenshots that demonstrate various analyses and views of student tracking data, we have asked participants to evaluate the software along the lines (adapted from [1]): *usefulness* (does the information support your work); *ease of use* (GUI handling); *functionality* (completeness and sufficiency of information); *organisation of information* (clear or confusing presentation of information); and *terminology* (understandability of wording and labelling). Furthermore, for each of the views we asked whether they would use it in practise.

Questionnaire Participants' Profiles. 31 participants filled in the complete questionnaire, most of them rather experienced with an involvement of 4 or more years in the area of e-learning (72%). The most often mentioned roles were teacher/instructor (79%), instructional designer (57%) and course coordinator (39%). All together, participants used 17 different e-learning systems; the most mentioned ones were Blackboard (11), WebCT (5), Moodle (3) and WTOnline (3). A large majority work in a college or university context (undergraduate courses: 79%, postgraduate courses: 64%). Participants teach a wide range of topics, among others, biology, nursing, statistics, mathematics, engineering, psychology, educational science and information technology.

Questionnaire Results. Fig. 5 depicts the average assessment for the views presented in this paper as well as for the participants' overall judgment of the tool. All four views were well received, with average scores near or above 4 (out of a maximum of 5 points). Also, the tool as a whole got good marks. The positive feedback was strengthened by the fact that the large majority of participants would use each view in practise.

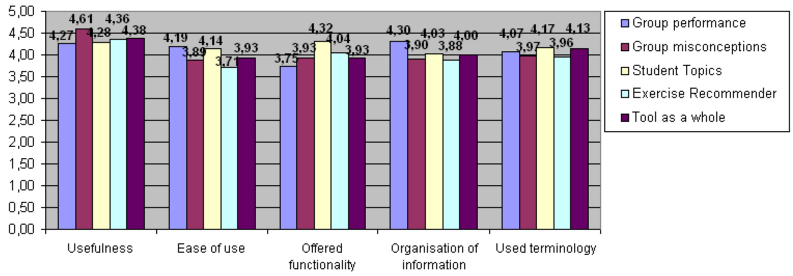

Figure 5. Average scores for selected views.

Information about a group's misconception received the best average usefulness score among all views. One teacher found the pie chart visualisation of misconceptions "quit [sic] helpful, especially with a large class", and that it allows one "to find out the most difficult problems easily". A participant noted that he could "evaluate [his] type of work better with this [misconception] view". This information was found to be "potentially very good for modifying instructions for exercises, or providing additional material". Another participant wished they had such a tool, "especially for our professors who are not very sensitive to students' needs".

The group performance view was also well received. Participants "liked especially the graphic presentation of the data", or even "like[d] this interface better than BlackBoard"; we received similar remarks also to other views. Suggested improvements for this part included a function to "email individuals highlighted", or to "drill down from here - to look at an individual students' results for all assessments to find out why their mark is high or low". The topic-specific performance view was rated of "great value!". It would "develop some useful information fore [sic] changing the course".

Participants were concerned about the AI-based analyser: it is "too complicated for a teacher to handle", presuming "heavy up-front preparation on model definition and creation", with unreliable outcomes: "not sure whether I would trust the model to give me correct answers", or whether "it shows me more than the previous, simpler screens".

6. Discussion and Future Work

Overall, the reaction to the STUDENT INSPECTOR was very positive. Teachers welcomed the abilities "to examine student performance trends and specifics", and to inform teachers "to give feedback to the students". Moreover, the visualisaton of information was judged a great benefit, the usability "easy to use". Some teachers feared, however, that the interface could scare those users which are "not in general comfortable with computers". It is clear that the use of all the features is very time-consuming, and that some information is more directed at educational researchers and content developers. Nevertheless, we had participants asking for more functionality, *e.g.*, visualise collaborative processes; keep a record of all skills achieved; use these records to inform group formation; plot learners's progress over several years; and include learners's time consumption per exercise. To satisfy these demands, we may need to customise the STUDENT INSPECTOR for different target groups or preferences, or to add some configuration management. One participant asked for a training component so that teachers could profit from a maximum of available student tracking analyses; others demanded transparency with respect to the computation of performance scores; and some participants were overwhelmed by the large variety of data and metadata, and their potential manual import, or the synchronisation of data between the STUDENT INSPECTOR and their learner management system.

One teacher noted that the STUDENT INSPECTOR is in favour of a class-instruction model, whereas some teachers may be more interested in student-centred, constructivist pedagogy. In fact, we are currently considering learners as a major target group for using our tool. This is in line with the iClass consortium that promotes *self-regulated personalised learning*. Self-directed learners need to inform their choices (*e.g.,* learning goals, styles) by tools that help them to become aware of their learning process, and that can generate personalised learning content to support this process. Teachers will still be necessary, esp. for learners with moderate abilities to self-regulate their learning. Teachers need to know these students to give them direction, and to adapt their teaching to their needs. Thus both parties could profit from the STUDENT INSPECTOR. For this, we are enhancing our tool, continuously checking with end-users that we are on track.

References

[1] J. P. Chin, V. A. Diehl, and K. L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proc. of CHI'88*, pages 213–218. ACM, 1988.

[2] M. Feng and N. Heffernan. Toward live informing and automatic analyzing of student learning: reporting in the assistment system. *J. of Interactive Learning Research*, 18(2):207–230, 2007.

[3] M. W. Goldberg. Student participation and progress tracking for web-based courses using webct. In *Proc. of the 2nd North American Web Conference*, Fredericton, NB, Canada, 1996.

[4] E. Melis, G. Goguadze, M. Homik, P. Libbrecht, C. Ullrich, and S. Winterstein. Semantic-aware components and services of activemath. *Brit. Journal of Educational Tech.*, 37(3), 2006.

[5] A. Merceron and K. Yacef. Tada-ed for educational data mining. *Interactive Multimedia Electronic Journal of Computer-Enhanced learning*, 7(1), 2005.

[6] A. Turker, I. Görgün, and O. Conlan. The challenge of content creation to facilitate personalized e-learning experiences. *Int'l Journal on E-Learning*, 5(1):11–17, 2006.

[7] WebCT. *WebCT Vista^{TM} 3.0 Designer and Instructor Reference Manual*, April 2004. Technical Communications, `http://www.webct.com`.

[8] C. Zinn and O. Scheuer. Getting to know your student in distance-learning contexts. In *Proc. of EC-TEL*, volume 4227 of *LNCS*, pages 437–451. Springer, 2006.