AIED 05 WORKSHOP 1

Amsterdam, 18-22 July, 2005

Usage Analysis in Learning Systems

AIED 05 WORKSHOP

Amsterdam

12th International Conference on Artificial Intelligence in Education, Amsterdam, the Netherlands

TABLE OF CONTENTS

FULL PAPERS

POSTERS

FOREWORD

The AIED community is increasingly interested in analysing student data collected electronically in technology supported learning activities. Understanding and taking into account usage of learning systems is now a growing topic of AIED Community, as recent events (such as ITS2004 workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes) have shown. This is a relatively new field of research with many pressing questions to address. People usually agree that it seems like a good idea to analyse student usage of learning systems, but there are no clear guidelines of what to analyse exactly, how to perform this analysis, what to look for, and how to take findings into account in the overall learning and teaching process. The purpose of this workshop is to gather a community around this exciting topic, look at concrete examples of experiments, and try and provide some directions to this research.

For instance, in the Kaleidoscope European Network of Excellence, the team DPULS[1] focuses on the definition of Design Patterns for recording and analysing usage in learning environments. The main assumption is that there is a common core of typical tracking foci, such as: detecting teaching rituals, anomalies or breakdowns in students' activity, detecting the emergence of leaders in groups, detecting students playing with the system or zapping from one activity to an other, collecting information to help teachers to manage a class and individuals in the class. Describing these problems, linking them with solutions adopted in existing systems, and sharing the results should be valuable to enhance the expertise of the designers' community. This workshop contributes to the achievement of DPULS in a way of sharing experiences and results of both the project team and the wider research community involved in tracking and analysing usage of learning systems.

We received 23 contributions and were pleased to accept 13 full papers and 7 posters for publication in these proceedings. All papers were fully refereed by at least two members of our program committee.

Papers and posters covered a wide scope of issues: Barre et al. and Kumar focus on the different perspectives of usage analysis for the different actors involved. Avouris et al. and Heraud et al. combine and analyse multiple sources of observations to provide a richer understanding of learning and discover new learning scenarios. Gibert-Darras et al., Stevanov and Stefanova, and Duval et al. propose Design Patterns for recording and analysing usage of elearning systems, based on their experiments. Bratitsis and Dimitracopoulou propose a forum tool allowing data to be recorded for subsequent analysis. Muehlenbrock also presents strategies for organising recorded data as well as an approach for automatic analysis of logged data. Feng and Heffernan and Le Calvez et al. present each their system with various student tracking functionalities to keep the teacher informed of students' knowledge. Mazza and Milani explore the use of improved graphical interface to present raw data about individuals and groups. Iksal and Choquet propose a meta language for usage tracking analysis that links observations to expectations of student usage. In the domain of algebra, both Nicaud et al. and Sander et

[1] web site: http://www.noe-kaleidoscope.org/pub/patterns/index.html

al. discuss a sophisticated student modelling approach based on a local rule-based diagnosis which is then computed into a lattice of conceptions. Hulshof's suggests to perform usage analysis across different platforms in order to obtain generic information on learning styles. Totter and Grote provides a classification schema for CSCL usage analysis whilst Mbala et al.'s system supports tutors' activity in a CSCL environment. Nogry's usage analysis serves the purpose of evaluating an ITS. Winter et al. tackle the problem of detecting topics and experts in message boards.

As can be seen, these papers form a stimulating basis for discussion at the workshop. We look forward to an exciting workshop on Usage Analysis in Learning Systems at the 2005 Artificial Intelligence in Education Conference in Amsterdam, Netherlands!

We thank all the members of the program committee for all their help, support and reviewing tasks. We are also very grateful to the three panel leaders (Nicolas Balacheff, Ulrich Hoppe and Judy Kay).

Christophe Choquet, Vanda Luengo and Kalina Yacef
Co-chairs

WORKSHOP SCIENTIFIC COMMITTEE

Program Chairs:

- Christophe Choquet (Co-Chair), *LIUM, University of Maine, France.*
- Vanda Luengo (Co-Chair), *IMAG, University of Grenoble, France.*
- Kalina Yacef (Co-Chair),*School of Information Technologies, University of Sydney, Australia.*

Committee Members:

- Nicolas Balacheff, *IMAG, University of Grenoble, France*
- Joseph Beck, *Carnegie Mellon University, USA*
- Peter Brusilovsky, *School of Information Sciences, University of Pittsburgh, USA*
- Elisabeth Delozanne, *CRIP5, University of Paris 5, France*
- Angelique Dimitrakopoulou, *Aegean University, Greece*
- Ulrich Hoppe, *COLLIDE, University Duisburg Essen, Germany*
- Judy Kay, *School of Information Technologies, University of Sydney, Australia*
- Jean-Marc Labat, *AIDA, Paris 6 University, France*
- Frank Linton, *The Mitre Corporation, MA, USA*
- Agathe Merceron, *Leonard de Vinci University, Paris, France*
- Tanja Mitrovic, *University of Canterbury, Christchurch, New Zealand*
- Jack Mostow, *School of Computer Science, Carnegie Mellon University, USA*
- Ana Paiva, *INESC, Lisboa, Portugal*.
- Richard Thomas, *University of Western Australia, WA, Australia*
- Pierre Tchounikine, *LIUM, University of Maine, France*
- Felisa Verdejo, *UNED, Madrid, Spain*

Logging of fingertip actions is not enough for analysis of learning activities

Nikolaos AVOURIS, Vassilis KOMIS, Georgios FIOTAKIS,
Meletis MARGARITIS and Eleni VOYIATZAKI
Human-Computer Interaction Group, E&CE Dept.,
University of Patras, GR-26500 Rio Patras, Greece

Abstract. In this paper we discuss key requirements for collecting behavioral data concerning technology supported learning activities. It is argued that the common practice of collecting machine generated logfiles of user actions is not enough for building a thorough view of the activity. Instead more contextual information is needed to be captured in heterogeneous media like video, audio files, snapshots, etc in order to re-construct the learning process. A software environment (Collaborative Analysis Tool ColAT) that supports inter-relation of such resources in order to analyse the collected evidence and produce interpretative views of the activity is described.

1. Introduction

Collection of usage data by registering users' operations in the form of logfiles has become mundane during technology supported learning activities these days. Many researchers assume that cognitive processes can, in principle, be inferred from studying and comparing this recorded behavior. Logfile analysis can be used when the purpose is to infer the cognitive processes of persons who interact with learning tools. Subsequent analysis can then be performed in a number of ways, for example by examining the frequency with which different operations are carried out or by focusing on the sequence in which operations occur [1]. Analysis of a learning activity is important for understanding the complex process involved, improve effectiveness of collaborative learning approaches and can be used as a reflection-support mechanism for the actors involved.

Tools to support interaction and collaboration analysis have been proposed in the field of learning technology design and human-computer interaction [2]. In the education field, analysis of collaboration and interaction between the actors (students, tutors etc.), the artefacts and the environment is a process that can support understanding of learning, evaluate the educational result and support design of effective technology [3].

In this paper we describe first the typical characteristics of Synergo, a tool that records users' operations and then supports their analysis during the activity and off line. In the second part of the paper, we argue further that while this approach is useful, more contextual information is needed to be interrelated to the logfiles. So an innovative analysis tool (ColAT) is presented that can be used for effective analysis of interrelated data that may be collected during technology supported learning activities.

2. Logfile-based analysis of learning activities

In this section, we describe the functionality of a typical environment for analysis of group learning, called Synergo (www.synergo.gr), associated to a synchronous collaboration-support environment, which permits direct communication and problem solving activity of a group of distant students, manipulating a shared graphical representation [4]. Synergo incorporates tools for analysis of usage logfiles. Through them the researcher can playback the recorded activity off-line and annotate the produced solution, while various indices and views of the logfiles can be produced.

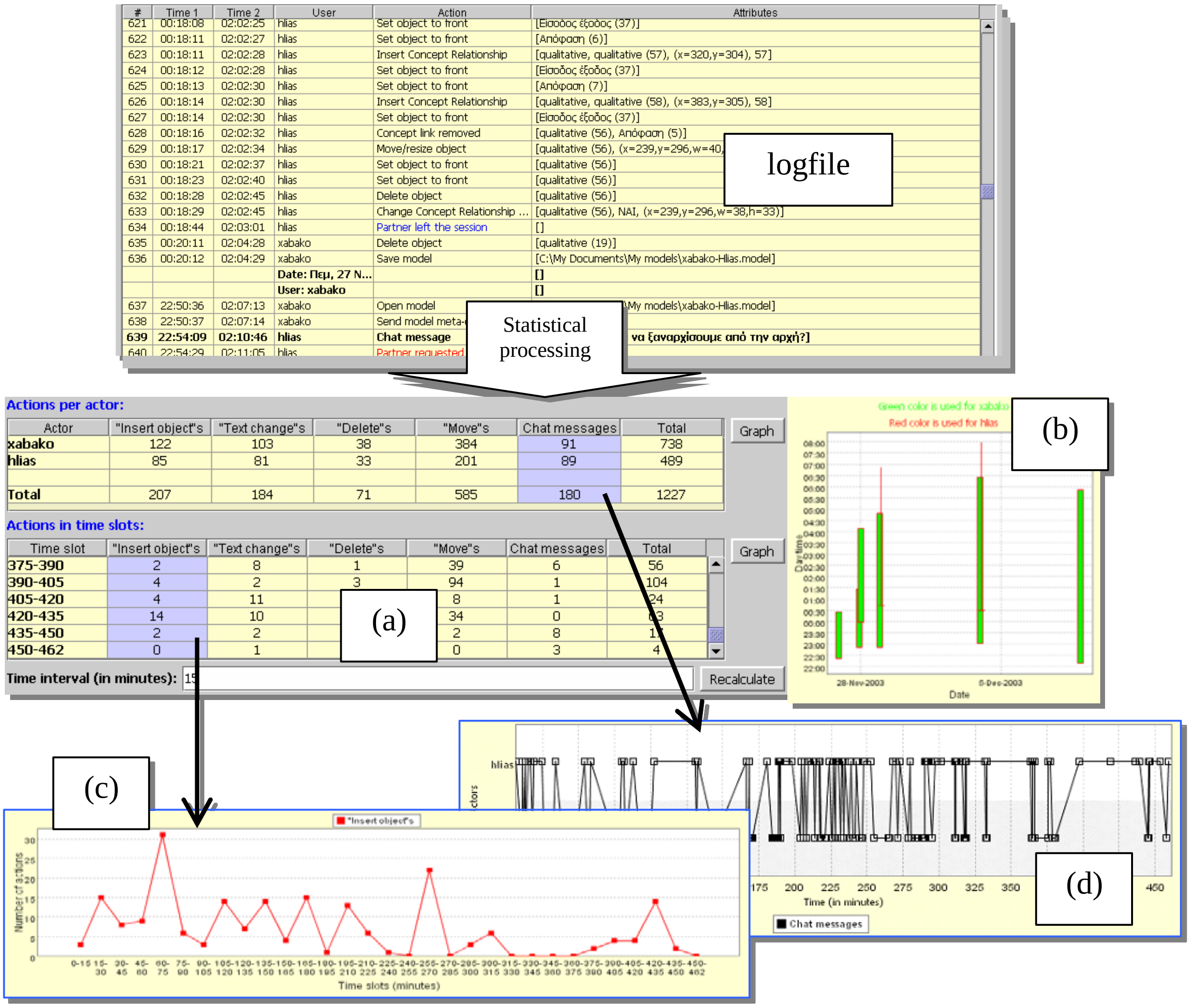


Figure 1. Snapshots from the Synergo analysis environment: The logfile (top of the picture) is used for producing statistical measures across various dimensions (type of event, time slot, actor), shown in (a). Also the extended of group sessions over time are shown in (b), while in (c) and (d) the statistical measures of (a) are drawn vs time.

In a typical synchronous collaborative learning situation in which Synergo is used, two or more actors, supported by networked equipment, collaborate at a distance by communicating directly and by acting in a shared activity board. A graphic representation of a solution to a given problem may appear in this shared board. This activity is typically monitored through logging of the main events of the actors in the shared activity board and of the dialogue events in text form. The Synergo analysis tool is used for presentation and processing mainly of the logfiles, produced during collaborative learning activities. These logfiles (see an example at the top of fig.1) contain time-stamped events, which concern actions and exchanged text messages of partners engaged in the activity, in sequential order. These events have the typical structure <time-stamp>, <actor>, <event-type>, <attributes>, <comments>.
The <event type> attribute categorizes the recorded event. This categorization can be done by interpreting one by one the logfile events manually. The Synergo environment facilitates this

tedious process, by allowing association of the events, automatically generated by the software, to classes of annotations. So for instance, all the events of type “Modification of textual description of concepts” in a concept-mapping tool are associated to the “*Modification*” type. So statistics and visual views concerning the activity can be automatically generated. For instance in figure 1 some of the views automatically generated by the Synergo analysis environment can be seen. This is an extract from a logfile that was generated by a pair of two students of a distance learning course who interacted for over 7 hours (462 minutes of interaction spread in 8 sessions). In fig. 1(a) the recorded events are grouped by user and type of event in the top table and by time interval and type of event in the second. The analyst can observe the value of various indexes, like the number of events of type “insert new object in the activity space” per time interval, shown in fig 1(c), or an interaction diagram indicating the activity per partner of a specific type of event, like chat messages between two partners in fig 1(d). Finally other views relate to length of sessions in fig 1(b). These representations can have some value for a trained analyst or teacher, or they can be used as self-awareness mechanisms for students as they can be presented to them during collaborative activities.

Not all recorded events however can be automatically annotated in this way, while important events are not captured at all by the logfile as they do not occur as a result of user-tool interaction (i.e. user fingertips activity). For instance, face to face dialogues have to be captured through other media, and interpreted by the analyst and after establishing their meaning and intention of the interlocutor, to be annotated accordingly. There are various ways of interaction, for instance, a suggestion of a student on modification of part of the solution can be done either through verbal interaction or through direct manipulation of the objects concerned in the shared activity board. Additional more complex indices may be generated, like the graph of evolution of the Collaboration Factor (CF), discussed in [4], the Collaboration Activity Factor suggested in [5], etc.

3. Interrelation of the logfile to other observational data

It should be observed that a typical logfile, like the one discussed in section 2, takes usually the form of an ordered list of events occurred at the user interface of a software tool. It contains a record of the activity of one or more learning actors, from the rather restrictive point of view of their fingertip actions. However a lot of contextual information relating to the activity, as well as results of the activity in print or other forms, dialogues among the actors, etc., are not captured through this medium. So in this section we present an analysis environment that permits integration of multiple media collected during learning activities. In section 3.1 we present examples of studies in which crucial role was played by these additional media.

The *Collaboration Analysis Tool (ColAT)* is the environment that is used for building an interpretative model of the activity in the form of a multilevel structure, following an Activity Theory approach [6], incorporating pointers and viewers of various media. ColAT permits fusion of multiple data by interrelating them through the concept of universal activity time. The analysis process during this phase, involves interpretation and annotation of the collected data, which takes the form of a multilevel description of the activity.

The ColAT tool, discussed in more detail in [7], uses the form of a theatre’s scene, in which one can observe the activity by following the plot from various standpoints. The *Operations view* permits study of the details of action and interaction, as recorded by a logfile, while other media like most typically video and audio recordings, capture dialogues, other behavioural data of actors (posture, gestures, facial expressions etc.), while media like screen snapshots, PDF files etc record intermediate or final outcomes of the activity. The automatically generated logfile can be expanded in two ways:

- First by introducing additional events as they are found in the video and other media, and by associating comments and static files (results, screen snapshots etc.) to specific time stamped events.

- Second, more abstract interpretative views of the activity may be produced: the *Actions-view* permits study of purposeful chunks of action, while the *Activity view* studies the activity at the strategic and motivational level, where most probably decisions on collaboration and interleaving of various activities are more clearly depicted.

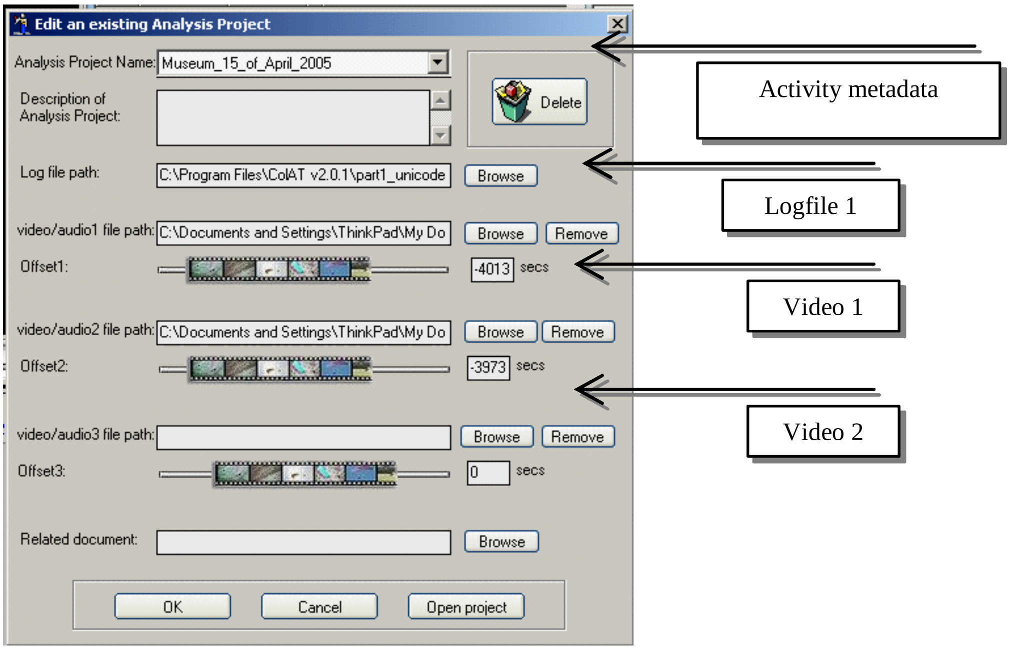


Figure 2. The ColAT environment: Project definition in which multiple logfiles and video/audio sources are synchronized by defining their corresponding offsets.

This three-level model is built gradually: the first level, the *Operations level*, is directly associated to log files of the main events, produced and annotated, and is related through the time stamps to the media like video. The second level describes *Actions* at the actor or group level, while the third level is concerned with *motives* of either individual actors or the group.
In fig. 3 the typical environment of the ColAT tool for creation and navigation of a multi-level annotation and the associated media is shown. The three-level model is shown on the right side of the screen, while the video/audio window is shown on the left-hand side. Other features shown in fig.3 include a toolbox for defining viewer filters, through which a subset of the activity can be presented, related to specific *actors*, *tools* or *types of events*.
The original sequence of events contained in the logfile is shown as level 1 (*Operations level*) of this multilevel model. The format of events of this level in XML, is that produced by Synergo, ModellingSpace, CollaborativeMuseumActivity and other tools that adhere to this data interchange format, while definition of a common format that includes requirements of other learning tools logfiles, like those generated by CoolModes [12, 13] is in progress. Thus the output of these environments can feed into ColAT, as first level structure. A number of such events can be associated to an entry at the *Actions level 2*. Such an entry can have the following structure: <ID, time-span, entry_type, actor(s), comment > where ID is a unique identity of the Action, time-span is the period of time during which the action took place, type is a classification of the entry according to a typology, defined by the researcher, followed by the actor or actors that participated in the execution, a textual comment or attributes that are relevant to this type of action entry. Examples of entries of this level are:" Actor X inserts a link ", or "Actor Y contests the statement of Actor Z".

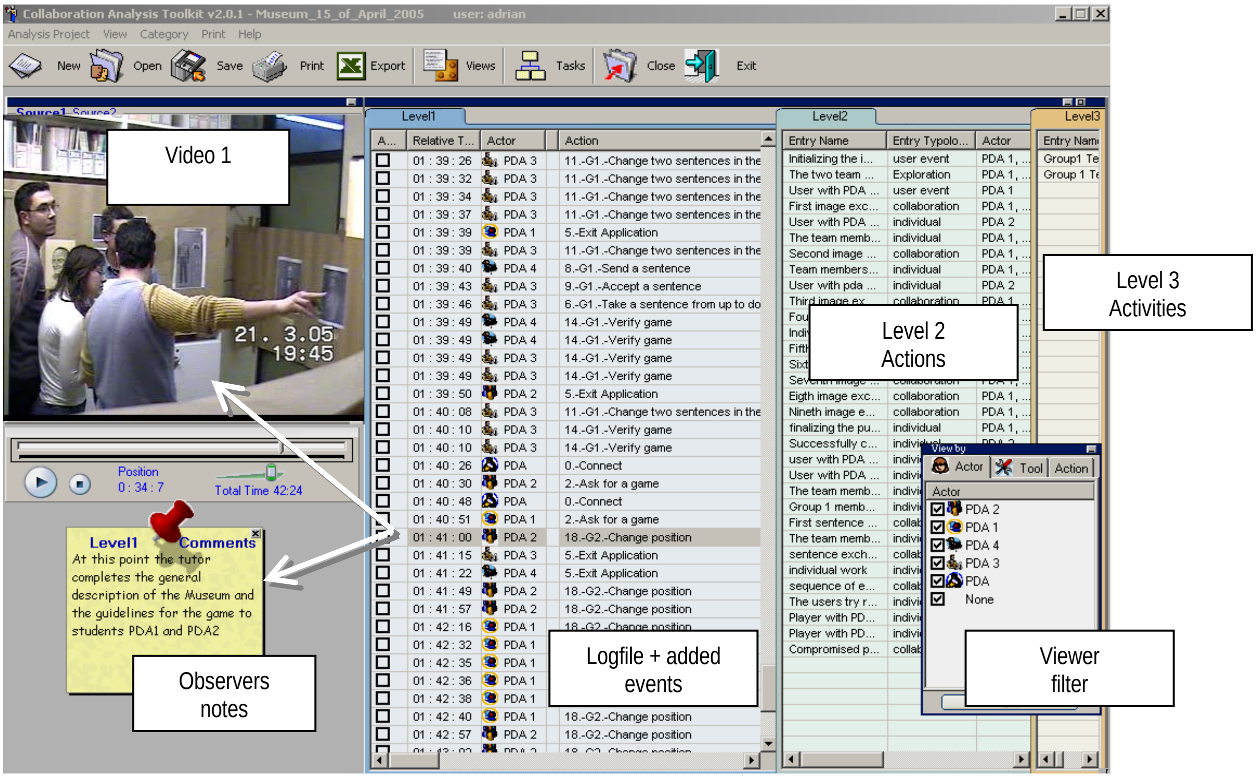


Figure 3. The ColAT environment: Multi-level view of problem solving activity

In a similar manner, the entries of the third level (*Activity level*) are also created. These are associated to entries of the previous *Actions level 2*. The entries of this level describe the activity at the strategy level as a sequence of interrelated goals of the actors involved or jointly decided. This is an appropriate level for description of plans, from which coordinated and collaborative activity patterns may emerge. In each of these three levels, a different typology for annotation of the entries may be defined. This may relate to the domain of observed activity or the analysis framework used. For entries of level 1 the OCAF typology [8] has been used, while for the action and activity level different annotations have been proposed. In figure 4 the dialogues for definition of annotation scheme for actions and identity of actors in ColAT is shown.

The various media, like video or audio, that can be associated to logged events through ColAT can be played from any level of this multi-level model of the activity. As a result, the analyst can decide to view the activity from any level of abstraction he/she wishes, i.e. to play back the activity by driving a video stream from the *operations*, *actions* or the *activity* level. This way the developed model of the activity is directly related to the observed field events, or their interpretation.

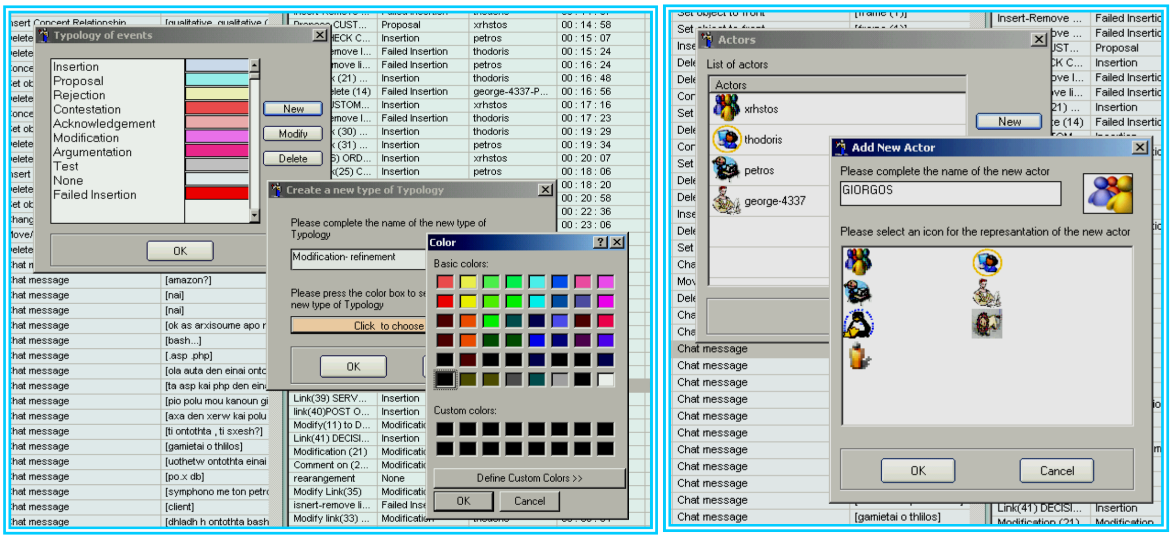


Figure 4. Definition of (a) typology of actions including the scheme that relates each type of event to a specific color code and (b) identity of actors that relates an avatar to each specific actor in ColAT

Other media, like still snapshots of the activity or of a solution built for a given problem, may also be associated to this multilevel model. Any such image may be associated through a timestamp to a point in time, or a time interval, for which this image is valid. Any time the analyst requests playback of relevant sequence of events, the still images appear in the relative window. This facility may be used to show the environment of various distributed users during collaboration, tools and other artefacts used, etc. Also observer comments related to events can be inserted and shown in the relevant window, as shown in the SW corner of fig.3.

The possibility of viewing a process using various media (video, audio, text, logfiles, still images), from various levels of abstraction (operation, action, activity), is an innovative approach. It combines in a single environment the hierarchical analysis of a collaborative activity, as proposed by Activity Theory, to the sequential character of ethnographic data.

3.1 Validation studies

The discussed tools have been used in a number of studies that involved effective analysis of collected evidence of technology supported learning activities in various forms. For instance in the study reported in [9] data were collected of groups of students interacting through the *ModelsCreator3* environment. Interaction between distant group members was affected through a chat tool and between the group members that were located in front of the same workstation through direct conversation. The first was captured through the ModelsCreator3 logfile that conforms to the ColAT format, while the latter through audio recording. By associating the two data sources, valuable information on comparison of the content of interaction that was done through the network and the dialogues of the group members was performed.

In [10] a study is discussed of activities that took place in a computer lab of a high school, using *Synergo*. The logfiles of Synergo were analysed along with contextual information in the form of video recording of the classroom during the activity and with observers' notes. These where interrelated and through this the verbal interventions of the tutor where identified and the effect of these on the students problem solving process was studied.

In Komis et al. [11] evaluation of the effectiveness of the concept mapping environment *Representation2* in the educational process is discussed according to various dimensions, like group synthesis, task control, content of communication, roles of the students and the effect of the tools used. In these studies, various features of the presented here analysis tools have been

used. First tools have been used for playback and annotation of the activity, while statistics and estimation of the collaboration factor have been produced. Subsequently, the produced video and sequences of still images, along with the logfiles of the studies were inserted in the ColAT environment through which the goal structures of the activities were constructed and studied.
Finally recently, collaborative game playing in a Museum using PDAs has been studied [14]. A logfile of the Museum server was studied in relation to three streams of video from different angles together with the observers' notes. It was found that various events related to interaction of the students with the exhibits and verbal interactions of the students between them and with their tutor/guide were captured in the video streams and were interrelated with actions at the user interface level of the PDAs. In this particular study it was found that the additional information conveyed through the posture of the users, their spatial location etc, was important for studying and understanding the activity, while the limited size of the portable devices and the technical limitations of monitoring the PDA screens during the activity, made the video streams and interrelated logged events at the side of the server most valuable source of information.
In the four studies discussed here the common characteristic was that in order to analyse effectively the studied activities and test their hypotheses the analysts used additional evidence in various forms, mostly video and audio. This was added to logfiles generated by the tools used (chat messages exchanged, actions on concept mapping tools etc.) and interrelated to them. The analysis environment ColAT that was used in these cases facilitated and supported effectively the analysis and evaluation task.

4. Conclusions

In this paper, we outlined the main features of two tools that facilitate analysis of complex field data of technology mediated learning activities, the Synergo Analysis Tool and ColAT.
The first one, is based on logfiles of events at the user interface of the *Synergo* Collaborative Learning environment. So playback and solution annotation were used in order to re-construct the problem solution and to visualize the partners' contribution in the activity space. However it was found that often this approach is not adequate for a complete reconstruction of the learning activity.
The second approach supports building a multilevel interpretation of the solution, starting from the observable events, leading to the cognitive level. This is done by using combination of multiple media views of the activity. Through this, a more abstract description of the activity can be produced and analysed at the individual as well as the group level.
It should be observed that the two presented tools are complementary in nature, the first one, used for building annotation of the problem solving at the user interface level, while the second one leading to more interpretative structures, as it takes into account additional contextual information. The result of the first phase can feed the second one, in which case the annotated logfile is just one source of information. The two presented tools are quite independent, since their use depends on the available data. The Synergo Analysis Tool is mostly related to the Synergo synchronous problem-solving environment, while the ColAT tool is more generic and can be used for studying any kind of learning activity, which has been recorded in multiple media.
In the extracts of four studies, that were discussed in section 3.1, it was demonstrated that there are many issues, relating to analysis of interaction, that necessitate multiple perspectives. So, analysis tools, like ColAT that interrelate logfiles and contextual information in various forms are needed to support and facilitated such studies.

Acknowledgements

The research reported here has been supported by projects TELL2003-4721/001-001 EDU-ELEARN (*Towards Effective network supported coLLaborative learning activities*) funded by the European Commission and Project *Methods & Tools for CSCL* funded by the Basic research program Herakleitos of the Hellenic Ministry of Education.

References

[1] Hulshof, C. D. (2004), Log file analysis, in L.Kimberly Kempf (ed), Encyclopedia of Social Measurement. Academic Press, 2004.

[2] Dix A., Finlay J., Abowd G, Beale R., (2004), Human-Computer Interaction, 3rd ed., Prentice Hall

[3] Gassner K., Jansen M., Harrer A, Herrmann K, H.U. Hoppe, (2003), Analysis methods for collaborative models and Activities, In B. Wasson, S. Ludvigsen, U. Hoppe (eds.), Designing for Change in Networked Learning Environments, Proc. CSCL 2003, pp. 411-420, Kluwer Academic Publ., Dordrecht.

[4] Avouris N., Margaritis M., and Komis V., Modelling interaction during small-group synchronous problem-solving activities: The Synergo approach, 2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction, ITS2004, 7th Conference on Intelligent Tutoring Systems, Maceio, Brasil, September 2004

[5] G. Fesakis, A. Petrou, A. Dimitracopoulou, Collaboration Activity Function: An interaction analysis tool for Computer Supported Collaborative Learning activities, Proc. ICALT 2004, Joensuu, FI, September 2004.

[6] Bertelsen O.W., Bodker S., (2003), Activity Theory, in J. M Carroll (ed.), HCI Models, Theories and Frameworks, Morgan Kaufmann, 2003.

[7] N. Avouris, V. Komis, M. Margaritis, G. Fiotakis, (2004), An environment for studying collaborative learning activities, Journal of International Forum of Educational Technology & Society, 7 (2), pp. 34-41, April 2004 .

[8] Avouris N.M., Dimitracopoulou A., Komis V., (2003a) On analysis of collaborative problem solving: An object-oriented approach, J. of Human Behavior Vol. 19, (2), pp. 147-167.

[9] Fidas C., Komis V., Tzanavaris S., Avouris N., Heterogeneity of learning material in synchronous computer-supported collaborative modelling, Computers & Education, 44 (2), pp. 135-154, February 2005.

[10] E. Voyiatzaki, Ch. Christakoudis M. Margaritis, N. Avouris, Teaching Algorithms in Secondary Education: A Collaborative Approach, Proceedings ED Media 2004, AACE publ, Lugano, June 2004, pp. 2781-2789.

[11] Komis V., Avouris N., Fidas C., (2002), Computer-Supported Collaborative Concept Mapping: Study of Synchronous Peer Interaction, Education and Information Technologies, 7:2, 169–188.

[12] L. Bollen, A. Harrer, H. U. Hoppe (2004), An Integrated Approach for Analysis-Based Report Generation, Proc. ICALT 2004, Joensuu, FI, September 2004.

[13] N. Pinkwart (2003), A Plug-In Architecture for Graph Based Collaborative Modeling Systems, Porc. AI ED 2003.

[14] A. Stoica, G. Fiotakis, J. Simarro Cabrera, H. Muñoz Frutos, N. Avouris, and Y. Dimitriadis, (2005), Usability evaluation of handheld devices, Proc. PCI 2005, Volos.

Re-engineering of collaborative e-learning systems: evaluation of system, collaboration and acquired knowledge qualities

Vincent BARRE, Hassina EL-KECHAÏ, Christophe CHOQUET
LIUM / IUT de Laval
52, rue des docteurs Calmette et Guérin
53020 LAVAL Cedex 09 (France)

Abstract. This paper relates an experimentation of a collaborative e-learning system. In this kind of system, tracks arising from communication tools allow to build useful indicators for all system actors. We show how tracks are analyzed and how this analysis is useful for reengineering purposes.

1. Introduction

The desynchronization of the two major teaching roles – course design and tutoring – in distance education, penalizes iterative optimization of the system quality by not taking into account uses with a reengineering objective. That's why in [1] we have proposed an extension of IEEE-LTSA (Learning Technology Systems Architecture) meta-architecture model [2]. This proposition explicitly integrates a step dealing with the observation and comportment analysis of distance learning systems and the learning process actors in an iterative process, guided by design intentions. We underline, in particular, the need for a formal description of the design point of view of the scenario, called prescriptive scenario, as well as assistance in uses analysis by comparing descriptive scenarios (an a posteriori scenario that effectively describes the learning situation's sequence [3]) with the predictive scenario. This produces information, significant for designers from a pedagogical point of view, when they perform a retro-conception or a reengineering [4] of their systems. In the framework of REDiM (Reengineering Driven by Models) project, we are particularly interested in supporting the implementation of those designers two main roles: (i) to establish the predictive scenario of a given learning situation, and (ii) to anticipate descriptive scenario construction by defining situation observation needs allowing the effective evaluation of the learners' activity.
In this paper, we will focus on a particular collaborative e-learning system: Symba. More precisely, we will observe the effective use of a pedagogical scenario in the context of a collective activity supported by collaborative tools. Our experiment thus consists of a project management collective activity, and more specifically, of a web-project management activity (specification and implementation of a web project). From our pedagogical reengineering point of view, many interesting information can arise from this experiment. In particular, we can be interested in comparing descriptive scenarios with predictive ones. Nevertheless, in a collaborative context, another interesting advisability is to compare roles emerging from activity to those anticipated by designers. In our experiment, desirous of putting ourselves in a normalization context, we have used pedagogical model arising from IMS consortium' Learning Design [5] in order to describe learning activities and to explicit pedagogical

scenarios. Nevertheless, we only use IMS LD as a means for designers to express their intentions, and not in an implementation perspective.

2. Presentation of *Symba* experiment

We have used an experimental CSCL support environment called SYMBA [6]. This environment is a Web-based system, developed by the LIUM laboratory in the framework of a Ph.D. study, in order to support Collective Activities in a Learning Context. It was designed following a double objective: (i) allowing students to explicitly work on their organization and (ii) providing tailorability [7] features allowing students to decide about tools and resources they want to be accessible in order to achieve tasks they have defined. Students have to develop a dynamic web-site using previously taught web project management methodology. According to our theoretical framework, students have first to collectively work (and agree) on project organization (such as what to be done, who does what, when tasks have to be finished, which tools are necessary for a particular task…) before beginning the second step, consisting in collectively performing tasks they have defined, according to their organization.

2.1 Presentation of actors

This experimental system is used by four distinct categories of actors. First category is made of fifty-six ***learners*** in higher education, from the Laval Institute of Technology, University of Maine (France). They were associated in small groups of 5 and they where working either at the University center or at home using tools offered by *Symba*. Those proposed tools are centered about description, organization and perception of the activity, but learners must also use the environment in order to explicit organization of their work, with a sharable plan and tasks editors. Activity proposed to the learners lasts for four weeks (35 working hours per week) and a predictive pedagogical scenario implying a collaborative learning was proposed, even if students are free to adopt or modify it. One can notice that this predictive scenario may involve concepts that are not yet taught to learners.
The second category of actors is made of ***instructional designers***. They specify the predictive pedagogical scenario and the uses of the learning system to be observed, they also analyze effective use of the Learning System in order to improve it (reengineering process).
A third category is made of three kinds of ***tutors***. We have *moderator tutors* whose role is to monitor activity within the learning session and to fill reports to evaluating tutors (i.e. *assessor tutors*) in charge to evaluate learners' activity in order to measure knowledge they have acquired. Lastly, *domain experts* are in charge of assisting learners in their tasks by helping them to solve specific problems connected to their expertise domain.
The last actors category is made of two kinds of ***analysts***. *Observed uses modelers* are building tracks with collected raw data, either from the Learning system or not, whereas *observed uses analysts* are analyzing the observed uses in order to synthesize information.

2.2 Different motivations in data analysis

In our experiment, some actors want to (and are interested in) analyze data. *Instructional designers* want to verify if roles they have predicted are well taken by learners and to detect unforeseen new roles. They are also interested in understanding the effective progress of a session in order to discover inconsistencies in it, for reengineering purposes. *Observed uses modelers* are interested in finding new techniques in order to improve their analysis abilities,

whereas *observed uses analysts* are interested in finding new patterns in order to improve their analysis abilities.
A part of *moderator* tutors job is to make reports for assessor tutors on learners abilities to collaborate and to work in group. *Assessor tutors* want to evaluate knowledge acquired by learners in Web projects management by verifying if produced organization is coherent with taught method during web project management courses. Lastly, *domain experts* are also involved in analyzing data, whilst they do not currently analyze data since this analysis cannot be done during the learning session (manual analysis), but they would be interested in analyzing data in order to understand what learners have done previously when they ask them for help.

2.3 What kind of data is being analyzed?

From a reengineering perspective, we will use some raw data (either recorded by the learning system or not) in order to generate some new data that will be useful for system actors. We will also need some additional data, such as the predictive scenario for the activity, and content data, that is, outcomes produced by learners during their activities. We will now detail most important data helping us to formalize emerging roles arising from learners activity.

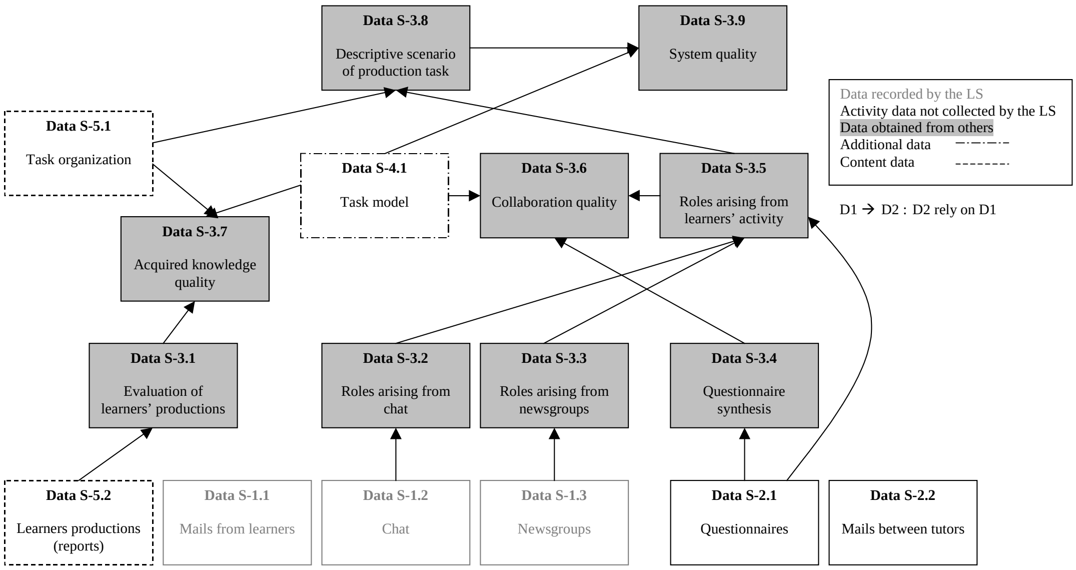


Figure 1. Dependencies between data

In this paper, we will focus on role emergence and we will not detail comparison between descriptive scenario and the predictive one. We will first detail raw data, either recorded by the learning or not. Please note that many of our raw data deal with communication tools tracks. In original tracks messages are written in French, they have been translated into English for insertion here.
Data S-1.2 (data arising from chat) corresponds to the transcription of all communications exchanged between learners via the chat service. A partial transcription of such messages can be found in Figure 2.

02/06/2004
10:45:33 *Arnaud* : I accept to begin working on functional guidance, but I will need some help.
10:46:02 *Myriam* **:** Yes, all going well, I should have finished soon and I would help you after that.
11:10:59 *Arnaud* **:** Do you need some help for the database ?

Figure 2. Excerpt of messages exchanged on the chat service

Data S-1.3 (data arising from newsgroups) corresponds to the transcription of the entire set of messages posted on newsgroup services. An example of such a message can be found in Figure 3.

41
31/01/2005, 17H43
Myriam
noemail
Re : IMPORTANT : WORK DISTRIBUTION
I agree to make the graphical charter, but I rather would work with *Arnaud* since I have already began this work with him and it will be simpler to continue together rather than with another people. Moreover, we are in the same class group and therefore it is easier to meet ourselves.

Figure 3. Excerpt of a message exchanged on the newsgroup tool

Data S-2.1 (data arising from questionnaires) consists of questionnaires, whose main goal is to evaluate group functioning by measuring some parameters such as participation, collaboration, organization… Answers to questionnaires are measured with a Likert scale, graduated from 1 to 5 (strictly disagree, disagree, not agree nor disagree, agree, completely agree). An example of such a questionnaire can be found in Figure 4.

I- PARTICIPATION

1- You always felt yourself integrated in your team because you were widely sharing information.

Completely agree: from a functionality viewpoint, as well as from a graphical viewpoint, we always concert all together before taking a decision. Discussion was therefore always privileged.

2- You always felt yourself integrated in your team because all decisions were taken after considering all opinions.

Completely agree : we have always take into account everyone's positions in order to make project progress in the good direction, with a good collective spirit.

Figure 4. Excerpt of a completed questionnaire

We will now detail data obtained by combining other data (either raw data or more synthetic data).
Data S-3.3 (data related to collaborative communication tools, i.e. *role emergence from newsgroups)* is derived from the transcription of all communications exchanged on newsgroups. Emerging roles are extracted from communication tracks using pragmatic markers [8] (see Figure 5 for an example). This data consists in a list of roles arising from observed communications. One can notice that those roles can be identical to those arising from other communication tools (e.g. chat service, data S-3.2) and are expressed using IMS/LD (see Figure 6 for an example).

Arnaud
So, I propose the following planning for the project:

June 3 -> legal aspects
June 3 to June 4 -> XHTML and PHP structures definition
June 3 to June 12 -> Overall, we can do the following : conception, content structure

[…]

June 23 : final tests and presentation preparation

Myriam
Re : Project Planning
I approve your project planning

Figure 5. pragmatic makers identifying a 'functional leader' role in newsgroups

Data S-3.2 (data related to collaborative communication tools, i.e. *role emergence from chat)* is derived from the transcription of all communications exchanged with chat service. Emerging roles are extracted from communication tracks using pragmatic markers [8] (see Figure 5 for an example). This data consists in a list of roles arising from observed communications. One can notice that those roles can be identical to those arising from other communication tools (e.g. newsgroups, data S-3.3) and are expressed using IMS/LD (see Figure 6).

```
<imsld:roles>
   <imsld:learner identifier="R-learner">
   ...
      <imsld:title>Learner</imsld:title>
      <imsld:learner identifier="R1-learner">
         <imsld:title>Socio-affective leader</imsld:title>
      </imsld:learner>
      <imsld:learner identifier="R2-learner">
         <imsld:title>Debate participant</imsld:title>
      </imsld:learner>
      <imsld:learner identifier="R5-learner">
         <imsld:title>Debate opener</imsld:title>
      </imsld:learner>
   </imsld:learner>
```

Figure 6. roles arising from chat analysis

Data S-3.4 (data related to questionnaire synthesis) is made of answers to questionnaires (data S-2.1) synthesized in percentages and reported with an evaluation grid summarizing this information for each question.

Data S-3.5 (data related to new roles arising from learners' activity). Study of interactions done with Symba communication tools (data S-3.2 and data S-3.3), as well as answers made to questionnaires (data S-3.4), allow to evaluate collaborative process from a cognitive and socio-affective viewpoint. Then, as for data S-3.2 and S-3.3, observed roles in a learning session are transcribed in an IMS/LD formalism.

```
<imsld:learning-activity identifier="LA25">
   <imsld:title>Integration</imsld:title>
   <imsld:learning-objectives>
      <imsld:item identifierref="" identifier="LA25-obj">
         <imsld:title>
            To know integrating all previously made pieces to the Web site
         </imsld:title>
      </imsld:item>
   </imsld:learning-objectives>
   <imsld:environment-ref ref="S2-mail-service" />
   <imsld:environment-ref ref="S3-chat" />
   <imsld:environment-ref ref="S4-newsgroups" />
   <imsld:environment-ref ref="LO18-outcomes-ress-detailedConception.zip" />
   <imsld:activity-description>
      Consists in integrating all previously realized pieces
      (graphical design, videos…)
   </imsld:activity-description>
   <imsld:complete-activity>
      <imsld:user-choice />
   </imsld:complete-activity>
</imsld:learning-activity>
```

Figure 7. Predictive task organization (excerpt)

Data S-3.6 (collaboration quality) corresponds to an evaluation of the quality of the collaboration between learners. This evaluation is made by comparing emerging roles from activity (data S-3.5) to predicted roles presupposed by designers (data S-4.1) and consists in a textual report.

We will lastly describe one additional data which is used in order to highlight synthesized data.

Data S-4.1 (task model specified by instructional designers) corresponds to the task model as anticipated by designers. That is, an indication of the activity sequence that learners are supposed to produce using organization workplace from Symba. This task model is expressed using IMS Learning Design (and, technically, it is an XML file conforming to IMS/LD XML-Schema definition, see Figure 7 for an example).

2.4 Who analyses data, how and when?

Analysts, and sometimes tutors, analyze data in order to synthesize information they contain. This analysis then produces new data (obtained from others) that can, in turn, be analyzed in order to produce new data.

Moderator tutors analyze data S-3.6 (*collaboration quality*) at the end of a learning session. Their analysis is made using report from the analysis sub-system (data S-3.5) and the task model produced by designers (data S-4.1) in order to verify the concordance between predicted roles and observed ones (at a per learner level).

All other analysis are made by the observed uses modelers (analyzing raw data) and observed uses analysts (making analysis from analysis reports made by observed uses modelers). We will first detail analysis made by *observed uses modelers*.

Analysis of data S-1.2 (*data arising from chat*), that is, tracks produced by learners, by way of their interactions through chat system is currently done manually, at the end of a learning session, using pragmatic markers [8] in order to identify emerging roles. This analysis will be automated in a next step. Analysis of data S-1.3 (*data arising from newsgroups*) is very similar: tracks are produced by learners, by way of their interactions through newsgroups. Currently, those data are manually analyzed with pragmatic markers [8] at the end of the session, and this analysis will also be automated in a further step.

Analysis of data S-2.1 (*data arising from questionnaires*) is made at the end of a learning session. Answers to questionnaire are synthesized in percentages and reported with an evaluation grid. Currently, this analysis is done manually, but will be automated in a next step. Since data S-2.1 contains detailed answers to questionnaires, it can be also used in order to make an analysis report concerning collaboration inside the group. This report is made at the end of a learning session and consists in a textual report.

Analysis of data S-3.2 (*role emergence from chat*) is made at the end of a learning session. Roles identified by pragmatic markers are formatted under an IMS Learning Design format. Currently, this analysis is done manually, but it will be automated in a next step. Analysis of data S-3.3 (*role emergence from newsgroups*) is done in a very similar way: at the end of a learning session, roles identified by pragmatic markers are formatted under an IMS Learning Design format. Currently, this analysis is done manually, but will be automated in a further step.

We will now detail analysis made by observed uses analysts. Analysis of data S-3.4 (*data related to questionnaires*) is made at the end of a learning session. Synthesis of questionnaires is analyzed by the human analyst in order to highlight collaboration quality (through a report on learners abilities to collaborate and to work in group). Questionnaire answering allow to evaluate variables such that: participation, collaboration, communication, work atmosphere, leadership, …

Data S-3.5 (*data related to new roles arising from learners' activity*) is produced by analysis sub-system, relying on data S-3.2 and S-3.3, relative to communication tools tracks and data S-3.4, relative to learner behavior obtained by analysis of questionnaires. Those three data are merged in one data, roles arising from multiple sources are reported only once. This unified data is then analyzed in order to find a matching between new roles and learners. Currently, this data originates in a manual analysis initiated at the end of a learning session. A computerized tool is under development, and should allow data analysis during a learning session.

2.5 Who uses the results of the analysis, how and for which kind of action?

Results of the analysis are used by many actors of our e-learning system. Analysts use them in order to produce new analysis, tutors use them in order to evaluate learners and designers use them in order to improve their predictive scenario (in a reengineering cycle).
We will first detail how analysts (both observed uses modelers and observed uses analysts) use results of previous analysis in order to build new data. First, *observed uses modelers* are in charge of formatting roles identified by analysts using pragmatic markers on chat messages (data S-1.2) and newsgroups messages (data S-1.3). They format those analysis results using an IMS/LD formalism. This is currently a manual transformation, that will later be automated.

Those two IMS/LD XML files then constitute, respectively, data S-3.2 and data S-3.3. Which, in turn, are used in order to produce data S-3.5 by merging those two XML-file in one containing emerging roles (editing out doubles). The last task for observed uses modelers is to format data arising from analysis of data S-2.1, i.e. questionnaires (percentages calculated by analyst) using a synthesis grid. This work is manually done at the end of a learning session and will be automated in a second step.
The textual report on learners abilities to collaborate and to work in groups arising from analysis of data S-2.1 (questionnaires) is then used, at the end of a learning session, by *observed uses analysts* in order to update learner binding to new roles and to clarify collaboration problems. This is a manual use.
We will now evoke uses made by tutors, and more particularly by *assessor tutors*. Their interest in analysis results is to attribute a grade to learners. They are therefore interested in moderator tutors reports made for data S-3.6 (evaluation of collaboration quality). They use those reports in order to attribute a grade concerning collaboration to learners.
Lastly, *designers* are also interested in using result of analysis. From a reengineering point of view, they are particularly interested in result of analysis of data S-3.5 (related to new roles arising from learners' activity). After a learning session, they use those analysis results in order to evaluate training that is bind to learner's collaborating capacities. It mainly consists of verifying that designer's predicted roles are well taken by learners and detecting new roles coming to the fore. When our analysis sub-system will be automated, those data would eventually be used by moderator tutors during a session, in order to regulate project progress (by adjusting role definition or pushing learners to adopt presupposed roles).

3. Conclusion

In a collaborative e-learning system, tracks arising from communication tools allow to build useful indicators for all system actors. Indeed, some indicators like 'collaboration quality' (data S-3.6) can, at once, be used by tutors in order to evaluate learners, by analysts in order to build other indicators and by designers in order to evaluate relevance of their pedagogical scenarios. From this last point of view, we have shown, in this paper, that considering emerging roles arising from communication tools tracks can be useful for reengineering purposes. For example, in our experiment, we have unfolded a first reengineering cycle, and this first cycle had allowed us to enrich predictive scenario made by designers by adding socio-affective roles arising from learning session tracks analysis. Role emergence was one key point of our reengineering process, and was in keeping with comparison of predictive scenarios and descriptive ones enriched with emerging roles.
Another interesting point is that proposed indicators can be used in a more general framework than the one of our experiment. Indeed, role mining from communication tools tracks can

help to enlighten effective use of the collaborative system and to push collaboration quality indicator forward, whatever the collaborative experiment may be. Moreover, in order to support production of such generic indicators, we have defined software tools [9] that, once developed, will allow the analysis of the collected data depending both on the predictive scenario and the formal description of elements to be observed. They will produce formal representations of user comportment, based on observation needs, and thus form a useful help in order to implement reengineering process.

Acknowledgments

This work has been done in the framework of the DPULS JEIRP project, funded by Kaleidoscope Network of Excellence supported by the European Community.

References

[1] Corbière A., Choquet C., « Designer integration in training cycles : IEEE LTSA model adaptation », *International Conference on Computer Aided Learning in Engineering Education* CALIE'04, Grenoble (France), feb 16-18, 2004, p. 51-62.

[2] IEEE P1484.1/D9, 2001-11-30 Draft Standard for Learning Technology — Learning Technology Systems Architecture (LTSA), IEEE Computer Society, 2001.

[3] Lejeune A., Pernin J-P., « A taxonomy for scenario-based engineering », *Cognition and Exploratory Learning in Digital Age* (CELDA 2004), Lisboa (Portugal), dec. 2004, p.249-256.

[4] Chikofsky E. J., Cross II J. H., « Reverse engineering and design recovery: A taxonomy », IEEE Software, 1990, 7(1), p. 13-17.

[5] IMS Global Learning Consortium. IMS Learning Design v1.0 Final Specification, Retrieved from http://www.imsglobal.org/learningdesign/index.cfm, 2003.

[6] Betbeder M-L., "SYMBA: un environnement malléable support d'activités collectives en contexte d'apprentissage", PhD Thesis, Université du Maine, France, 2003.

[7] Morch A. & Mehandjiev N. D., "Tailoring as collaboration: The Mediating Role of Multiple Representation and Application Units", CSCW'2000, Philadelphia, Pensylvania, USA, December 2-6, 2000, pp75-100.

[8] Cottier P., Schmidt C.T., "Le dialogue en contexte: pour une approche dialogique des environnements d'apprentissage collectif", In: Colloque ARCo 2004, December 8-10, 2004, Compiègne (France).

[9] Iksal S., Barré V., Choquet C., Corbière A., « Comparing prescribed and observed for the re-engineering of e-learning systems », *IEEE Sixth International Symposium on Multimedia Software Engineering* MSE, dec. 13-15, 2004, Miami (USA).

Data Recording and Usage Interaction Analysis in Asynchronous Discussions: The D.I.A.S. System

Tharrenos BRATITSIS & Angelique DIMITRACOPOULOU
LTEE Laboratory ,University of the Aegean,
Democratias 1 str, 85100 Rhodes, Greece
bratitsis@aegean.gr; adimitr@rhodes.aegean.gr; *www.rhodes.aegean.gr/LTEE*

Abstract. This paper describes the development and the implementation of a new Asynchronous Discussion Forum Software, called DIAS. While evaluating several corresponding software, we came to the conclusion that they seem to be inadequate to support the use of this activity (asynchronous discussion) as a substantial part of the learning process. Most actual forums, that incorporate interaction analysis functions, support mainly administrators, or teachers, while offer only basic awareness functions to the learners.

The DIAS system is mainly developed in order to offer extended interaction analysis support, by providing a wide range of indicators jointly used in various learning situations, to all discussion forae users (individual students, groups, teachers or even researchers), appropriate for their various roles in different learning activities. An additional goal is to provide a customizable, flexible and interoperable environment.

The present paper briefly describes the reasons that enforced the need to construct this new tool, examining the aspect of CMC (Computer Mediated Communication) assessment. The description of the system's architecture and functionality follows. The implemented technique of recording, storing and analyzing forum data is pointed out, justifying the choices. Finally the description of the on-going case studies of the systems' usage in real conditions and the expected results.

1. Introduction: Asynchronous Discussion Forae

1.1 General Overview

The past few years we witness an increased mobility in research concerning tools for analysing and supporting learning activities (by distance in particular). Several theories and techniques have evolved, using elements deriving from the CSCL and the CSCW domain. Recent developments in learning theory have emphasised the importance of context and social interaction. In this vein, the notion of a learning community is gaining momentum.

In the present paper, our center of interest is Asynchronous Discussion Forae. It is a substantial component of every Learning Community, as it provides means of communication and implicitly learning process management.

CMC can be defined as the exchange of messages among a group of participants by means of networked computers, for the purpose of discussing a topic of mutual interest [9],[2]. Such exchanges of messages can be carried out either synchronously or asynchronously. According to Groeling [8], facilitating asynchronous discussion has the potential to improve the teaching and learning experiences in traditional classroom formats, as well as in distance learning [2].

Asynchronous online discussion allows records of a participant's written messages to be kept in the virtual electronic 'space' for long periods of time [7]. Participants in such a forum need not be online at the same time [20]. They can respond to the messages posted at any time they prefer and view them many times and long after they have been posted. In this way, asynchronous online discussion can resemble written communication [7],[2].

Considerable amount of work has been done concerning interaction and argumentation analysis in discussion forae. Nevertheless most of the solutions are carried out by researchers off line and usually in scheduled time intervals analysis [6]. Indicators of quantitative as well as qualitative analysis have been introduced.

Stepping backwards and examining large Distance Learning Systems, we will find out that the forum module contains little information available for the users. For example in WebCT [23] (one of the most complete systems available), the information available for forum usage is limited to: Session Information (number of sessions, session length and mean session time), Activity Information (number of messages posted and read) and a few statistical indicators (most and least busy day, etc). Other, forum specific software like WebWiz [22] and PhpBB [19] provide additional information, like: Online users, number of messages per day, number of unread messages, etc. We consider all of this minimal information, which supports forum usage only as a subsidiary tool for the Learning System.

Furthermore, we can find qualitative analysis methods which are applied after the discussion termination or at scheduled time intervals. These methods are consisted of message content analysis, usually off line and take into account parameters like syntax, subject and content appropriateness (staying on topic), argumentation analysis (for example IRF and IDRF approach) [13].

Finally, more advanced interactions analysis methods have been developed and applied only by researchers to a few systems that partially involve forum, and are context specific in a more global learning environment [e.g. DEGREE system] [1].

1.2 In Need of a New System

Literature points out the difficulties of students participating in Forums: lack of active participation, poor quality of argumentation, behaviors that do not contribute to a collaborative activity, etc. The 'presented' reasons are often just the lack of confidence, the poor quality of their texts, the anxiety, etc. [17],[18]. We consider that serious supplementary reasons may be that: (a) students ignore how to proceed (the didactical contract is not established, the deontology is not assumed, the expected behavior is not known), (b) they suffer from an overflow of low level information, (c) they cannot create an 'image' of their own action/activity in comparison of those of others participants, (d) moreover, they cannot create an image of the whole activity as a group.

Our approach aims to offer direct assistance to students participating to a discussion forum, that could support them in the level of awareness of their actions or behaviors as well as those of their 'collaborators', in order to activate their metacognitive processes, thus allowing them to auto-regulate their own activities. In parallel, we aim to support the persons that monitor forum discussions (eg. teachers) in order to 'identify', the difficulties during learning situations, and regulate them via appropriate interventions.

Given the above considerations, and taking into account the fact that the actual forum systems offer a limited interactions analysis support or that they are context specific, we were led to the decision of designing and developing a new Asynchronous Discussion Forum System called DIAS (Discussion Interaction Analysis System).

In order to design DIAS, three central design principles were specified, concerning its interaction analysis component:

[a] *Take into account the totality of the users that are involved in a 'learning activity', as well as their cognitive systems that may form* [5], students as individuals (in various roles), but also as members of one or more groups or even communities, teachers in different roles according the category of learning activity, etc.

[b] *Provide a rich range of interaction analysis indicators*: The analysis of interactions, in terms of indicators, seems to be an appropriate framework that offers different points of view of the learning activity process, its quality, as well as its product. Different indicators may be more appropriate during different time periods of the learning, for different learning task, as well as for different profiles of forum participants [4].

[c] *Create a flexible, customizable, and interoperable system*: Forums are tools that can be used in a number of contexts, and for a variety of discussion based learning activities categories. Furthermore, forum participants take various roles and they have different needs according to their discussions subjects, the available time, etc. Thus, it is significant to create customizable, flexible and interoperable systems.

As aforementioned, minimal information is provided in order to support discussion forae as environments that promote learning [16],[12],[14],[2]. As Henri [10] mentioned, there are five (5) complementary dimensions on which we can evaluate CMC:

- *participative:* at a minimum, is it used? This can degenerate to the counting of numbers and lengths of messages. It is easy to measure (at least in CMC, it is harder face-to-face), but says nothing about the quality of what is going on.
- *social:* this is clearly important, since the social dimension provides some of the motivation for people to make use of the system. It describes the ability of learners to project themselves socially and affectively into a community[21]. However, this dimension says nothing about the quality of learning taking place.
- *interactive:* it is possible to measure the interactions, responses and commentaries taking place, seeing in detail how particular events or statements lead to particular responses. Such analyses, by whatever techniques can give useful insights into how to improve conversations (through facilitation, for example), but do not tell us much about the type of learning that has gone on.
- *cognitive:* people use skills connected to reasoning which uses critical thought. It is this dimension which is of big interest to educators, considering as common goal to encourage critical thinking,.
- *metacognitive:* reasoning about reasoning and self-awareness.

Our aim is to find ways of providing measurable elements of an Asynchronous Discussion Forum, in order to produce the means of evaluating this kind of CMC as a cornerstone of Distance Learning processes[11].

2. The D.I.A.S. System

2.1 Technical Overview

Our main goal was to develop an independent forum tool, which would be flexible and easily customizable as well as interoperable. This lead us to the selection of web based open source technology, making it easy to share with the academic community. The system is developed using asp code and java applets, making it easy to alter at will its functionality (customizability, flexibility).

The use of a Data Base Management System was decided, in order to achieve more efficient Data manipulation (storage, access, retrieval). One of our goals in the near future is to build data input filters, in order to use our system with discussion data derived from any other Forum Software (interoperability). Another aspect of its interoperability and independence is that it can be used integrated in an Distance Learning system as an add-on, being purely Web Based.

2.1.1 System Architecture

The system architecture is shown in Figure 1.

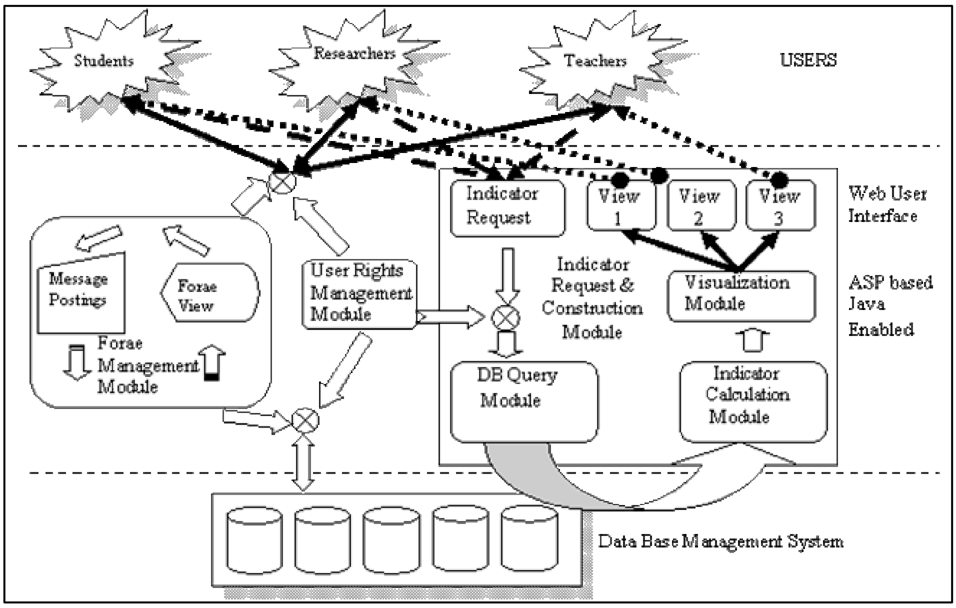


Figure 1: D.I.A.S. System Architecture

The D.I.A.S. system is consisted of two major components. The *Data Base Management System* is used for storing the data recorded throughout the forum usage. The *Web User Interface*, which is developed on ASP and Java Technologies, is constituted of three basic modules. The *Forae Management Module* allows the users to insert data into the forum system through message posting (including a number of formatting methods) and to access the data already stored. The *User Rights Management Module* which is responsible for controlling the data a user can insert, alter and view. The *Indicator Request & Construction Module* extracts data from the DBMS using queries (*DB Query Module*) according to user needs (*Indicator Request*), in order to analyze it (*Indicator Calculation Module*) and feed the *Visualization Module* which is responsible for presenting the query results in a tangible manner (Indicators' Views).

2.1.2 System Functionality

The D.I.A.S. System is a fully supported Asynchronous Discussion Forum with Interaction Analysis and Awareness Tools. The core of the system is the underlying DBMS, used for storing and managing the recorded data.

Users can write messages using a WYSIWIS editor which embodies many formatting functionalities. They are exhorted to choose the type which corresponds to their message content, facilitating guided conversations. The message types are in the teacher's discretion. By altering a few simple parameters, we can omit this functionality, thus implementing free discussion.

There are three levels of classification; Thematic *categories*, which may contain many *forae*. Each forum may contain many discussion *threads*. Users can be assigned different access rights to forae, varying from "no-access" to "full permission", depending on group membership. Furthermore the teacher is given the ability to choose the message types allowed to be used in every forum, at any time.

The capability of creating printer friendly, flat format of the forae also exists, giving the opportunity to the users to study the content and the evolution of the discussion at a glimpse.

2.2 Data Capture and Recording

We have analyzed many of the existing Forum Software in order to determine the best way of recording and storing usage data. We concluded that the most efficient way is the use of a RDBMS as it provides a plethora of ways to retrieve query-based data, leaving raw data intact. Furthermore it provides good integration with internet browsing software and consequently, convenience in choosing open source, ASP and Java components while building our system. The actual database schema is shown in the Figure 2.

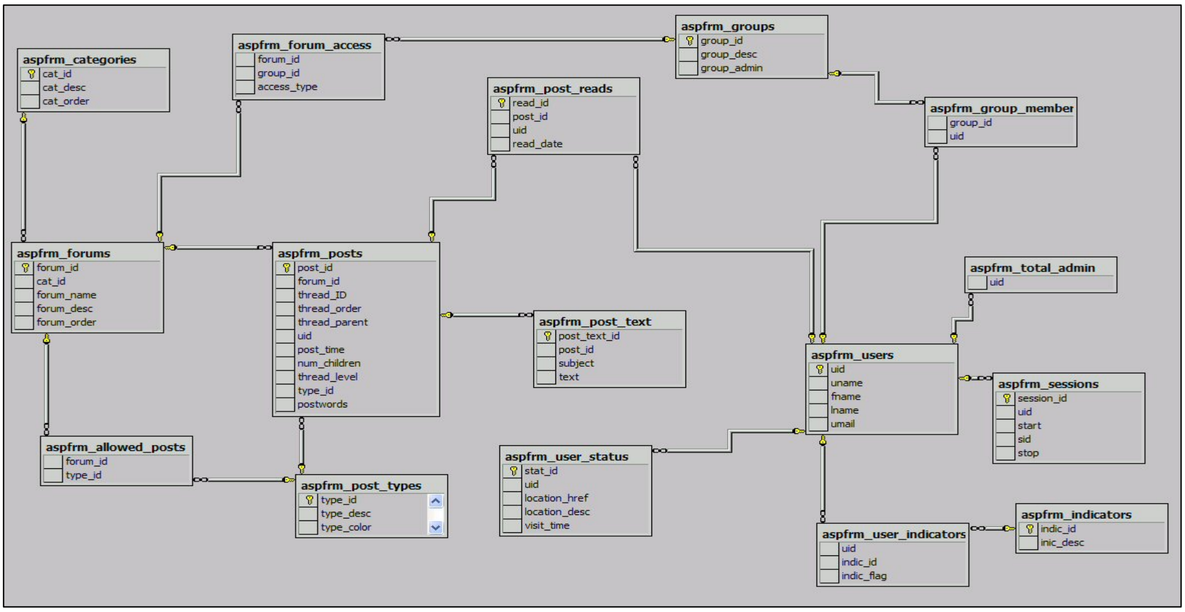


Figure 2: Database Schema

The set of raw data we find compulsory for detailed and sufficient forum usage analysis contains:

- User data (User-id, Name, Surname, e-mail address)
- Post data (Subject, Message Content, Post Type, Author, Date written, Reading access by users)

Also another set of data is necessary for better forum utilization and further usage analysis is constituted of:

- Forum structure (Forum categorization, Forum name and description, thread separation, appearance order)
- User rights management (Forum access rights, User grouping, Post type permission per forum)
- Indicator and information access rights manipulation
- Session information (logon hours, page visitation)

2.3 Data Analysis

We access the raw data by applying the appropriate set of queries to the DBMS. Calculation of the retrieved information subsequently occurs, passing results to the visualization system.

Partial set of the data is used for providing awareness information to the users, such as:

- *Participation*. Information about active participation (message writing) and passive participation (message reading)
- *Relative participation*. Information about the user's participation ratio, regarding specified time tables and group activity.
- *Discussion evolvement*. Information about the progress of the discussion regarding to time.

Another portion of the data is used to produce interaction analysis indicators. Some reveal user activity, evincing active and passive users, ones who need encouragement or coaching, etc. Others relate to discussion and interaction assessment. Some of the data recorded is not yet used in computations (message length for example). It can be accessed for further, qualitative analysis of the discussion.

Fifty two (52) indicators, divided in three sets are currently produced; Individual, Group and General Indicators. In Table 1, we can see a categorization, including a description of the basic functions used to produce them as well as the information type they embody.

Information Type	FUNCTION	Number of Indicators		
		Individual	Group	General
A_P	$N_P = F(t,f)$	3	2	---
A_P	$N_{PT} = F(t,f)$	2	2	---
A_P	$N_P = F(t,T,f)$	3	3	---
A_P,A_R,I_P,I_S,I_C	$A_B = F(u,T,f)$	2	2	---
A_P,A_R,I_C	$A_T = F(u,T,f)$	1	1	---
A_R,I_M,I_S	$R = F(u,f)$	1	1	---
A_E,I_M	Thread Depths, Thread weight	1	1	2
A_R,I_M, I_S	Classification Indicator = $F(R, N_P)$	1	1	---
A_R,A_E,I_S,I_M	Relative Activity Indicator = $F(N_P,T,S,t)$	1	1	---
A_R,A_E,I_P,I_M,I_S	Activity Indicator = $F(N_P,R,T)$	1	1	---
A_P,A_R,A_E,I	Contribution Indicator = $F(N_P,S,T)$	1	1	---
I_C,I_I,I_M	Group Interactivity Indicator = $F(A_B, A_T,u)$	---	2	2
A_P,A_R,A_E,I_I	Answered Contributions = $F(u,f, A_B)$	---	---	2
A_E,I	Contributions Answered by Others = $F(u,f, A_T)$	---	---	2
A_E,I_C,I_M	Follow-Up Contributions = $F(u,f, A_T, A_B)$	---	---	2
A_P,A_R	Average(N_P) = $F(u,f)$	1	2	2
I_S,I_C,I_M	SNA → DL Full Matrix, Agna Matrix	---	---	2

Information Type

A_P → Awareness (Participation)
A_R → Awareness (Relative)
A_E → Awareness (Evolvement)
I_P → Interaction (Participative)
I_S → Interaction (Social)
I_C → Interaction (Cognitive)
I_M → Interaction (MetaCognitive)
I_I → Interaction (Interactive)

Function Variables

t = Time Intervals
N_P = Number of Posts
N_{PT} = Number of Type T Posts
T = Post Types
A_B = Answers By User
A_T = Answers To User by others
f = Forum
R = Post Reads
S = Thread Initiating Posts
u = Users

Table 1: Indicator Functions and Classification

We have implemented a wide range of visualized information and decided to provide different sets for the teacher – researcher and the students. The main idea is to give the teacher the opportunity to choose the appropriate set of indicators for the students, depending on the learning scenario and its evolution (set variation as the scenario evolves). This would give us the opportunity to evaluate the appropriateness and utility of the various tools, regarding their acceptability by the users and their participation in usage support and enforcement.

Almost all the indicators are displayed in graphical format, using Bar Graphs, Polar Diagrams, XY Charts, Pie Charts and Scatter Charts. For the SNA diagrams production, text files in well known format are produced (Ucinet DL format and Agna Format) [11]. Some of the information available, mainly awareness related, is displayed in text format. Everything is produced on the fly, except the SNA diagrams, which have to be passed on to another software (NetDraw or Agna).

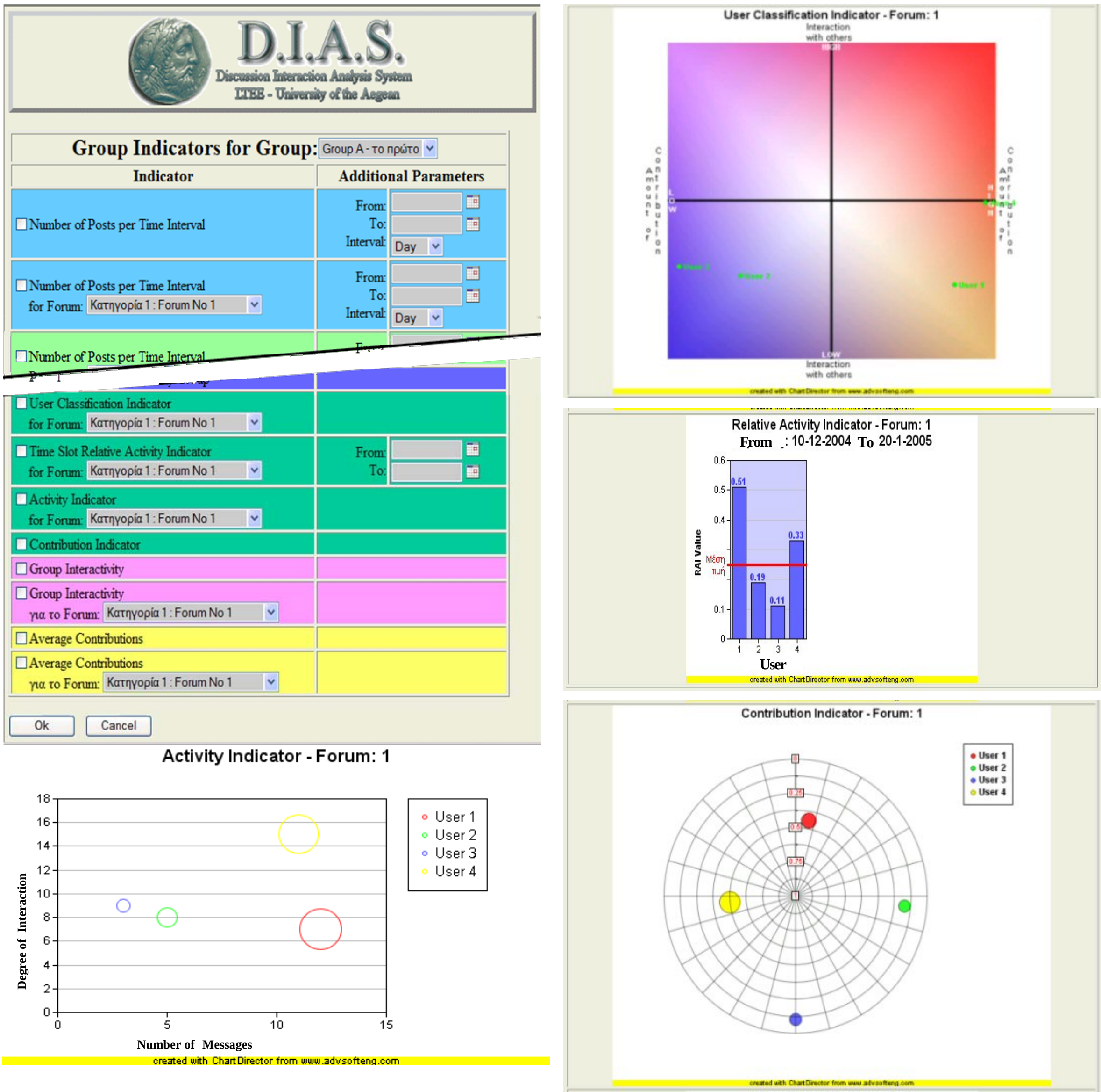


Figure 3: Screenshots

Four of the produced indicators are shown in Figure 3. These are:

- *User Classification Indicator*: It is a XY scattered chart with the X-Axis representing the amount of contribution and the Y-Axis representing the amount of Interaction by the users. The two Axis are scaled from Low to High. The X-coordinate is calculated by the contributions (messages written) of the user as a percentage of the total contributions, thus placing the lowest number at the left end of the Axis (Low) and the highest number at the right end (High). The Y coordinate is calculated as the percentage of the available messages read by a user (excluding the ones written by himself). A quick look on the chart reveals the kind of system use (balanced, mostly reading others' messages, mostly writing messages, etc)

- *Relative Activity Indicator*: A bar chart is created, showing the activity of the users for the selected time duration as a percentage of the total activity. Initiation of discussions and use of different types of messages is subsidized. The mean value of the contribution percentage for the selected time is also displayed, thus evincing the most active users. This is an indicator useful for the students as well, by providing information concerning their classification within the group activity.
- *Contribution Indicator*: A polar chart contains bullets representing the various users. The distance from the circumference of the circle is proportional to the contribution status of the user, subsidizing the initiation of discussions. The size of the bullet is proportional to the number of message types used.
- *Activity Indicator*: A XY scattered chart shows the amount of contributions (X-Axis), the amount of message types used (bullet size) and the mount of other users' messages read (Y-Axis).

Some of the indicators produced present the same kind of information using a different visualization approach. This is deliberately implemented for the investigation of the optimum method of data presentation.

3. System Testing & Evaluation

Actually, DIAS system was designed and developed, offering and supporting a wide range of possibilities, in order to allow us, in a first level to:

(a) Study the appropriateness of each indicator separately as well as indicators sets for specific interaction analysis functions users

(b) Identify the appropriateness of those indicators, regarding forum-based learning activities categories as well as context of uses.

and then, in a second level, to be able to propose to researchers and teachers (for their own work, but also for students support), specific interactions' analysis indicators' sets, for specific usages cases.

Currently the system is being tested in real settings, at the University of the Aegean. Three different case studies take place, involving the students of one undergraduate and two post-graduate programs. In the first case, 50 undergraduate students participate, and the Forum is mostly intended to support weekly discussions that enforce the work done during the traditional seminar-based course. In the second case, 40 post graduate students participate, while attending two courses for six weeks. It is a distance learning program that involve students in face to face seminars for only three times per semester (in the beginning, in the middle, at the end). During these two courses a variation of discussion activities and project management activities (for final assignment preparation) will be implemented. In the third case, 15 post graduate students participate, while attending a course involving present seminars, every week. During this test period, the teachers will be able to use most of the available indicators, in various combinations. The system administrator will assure that every day the corresponding SNA diagrams and the flat formats of each forum will be produced. It is expected that a limited set of indicators will be available to the students, varying as time goes by.

Our main goal is <u>to assess the indicators' usage</u>, considering students and teachers as users of interaction analysis component of DIAS system. More concretely, we will investigate the following:

(a) The correctness of the indicators produced is a main issue of consideration during this first testing period with real users.

(b) We intend to point out which indicators' set better present the group activity.

(c) Regarding the students, the appropriateness of each indicator as well as of indicators' set is the main issue of consideration. Furthermore, we want <u>to detect the effect</u> of the information provided by the indicators <u>in their self regulative actions. .</u>

(d) A <u>classification of user working mode and status</u> is another objective in order to nominate the corresponding <u>appropriate indicators sets</u> to each case.

(e) Finally, the possibility of <u>creating additional indicators</u> (more qualitative) is substantial during the analysis and evaluation of the project.

Future Plans include Data Input Filters, allowing to import data from other Forum Software in our DIAS interaction analysis system, in order to test the interoperability of the systems, as well as to evaluate the production of the indicators and the appropriateness of awareness and metacognitive support in a wider usages' spectrum.

4. Acknowledgments

The present work is based on previous research projects of LTEE lab, while it is actually inspired and enriched by the discussions and the work carried out in the frame of the ICALTS and Interactions_Analysis JEIRPS, of the Kaleidocsope NoE, IST-507838, [www.noe-kaleidoscope.org]

5. References

[1] Barros, B., Verdejo M.F., Read, T., Mizoguchi, R., (2002), Applications of a collaborative learning ontology, *MICAI'2002 Advances in Artificial Intelligence*. In C. A. Coello, A. de Albornoz, L. E. Sucar, O. C. Battistutti, (Eds.), Lecture Notes in *Computer Science* 2313, Springer-Verlag pp. 301-310

[2] Corich S., Kinshuk, Hunt L. (2004): *Assessing Discussion Forum Participation: In Search of Quality*. International Journal of Instructional Technology and Distance Learning. TEIR Center, Duquesne University, Pittsburgh

[3] Cramphorn C. (2004): *An Evaluation of Formal and Underlying Factors Influencing Student Participation within E-Learning Web Discussion Forums*. Networked Learning Conference.

[4] Dimitracopoulou A. (2005). Vers des outils d'Analyse des Interactions pour l'Assistance Metacognitive des Participants aux Forums de Discussion. *Symfonic Symposium :"Forum et communautés d'enseignants et d'apprentissage*,Amiens, France, Jan. 2005.

[5] Dimitracopoulou, A. Dillenbourg, P. Hoppe, U. (2004). Interaction analysis supporting participants during technology –based collaborative activities: Workshop summary, In *CSCL SIG Symposium,* Lausanne, 7-9 October, 2004. http://cscl-sig.intermedia.uib.no/lausannesymposium/, last visited 4/2005

[6] Dringus L., Ellis T. (2004). Using data mining as a strategy for assessing asynchronous discussion forums. In *Computers & Education*. (2004).

[7] Ganeva, I. (1999). *Native and non-native speakers' participation in educational asynchronous computer conferencing: a case study*. Unpublished Master thesis. Ontario Institute for Studies in Education, University of Toronto.

[8] Groeling, T. (1999). *Virtual Discussion: Web-based Discussion Forums in Political Science*. Paper presented at the American Political Science Association, 1999, Atlanta, Georgia.

[9] Gunawardena, C., Lowe, C. & Anderson, T. (1997). Analysis of a global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing. *Educational Computing Research,* 17(4), 397-431.

[10] Henri, F. (1991): *Computer conferencing and content analysis*. In O'Malley, C. (ed.) Computer Supported Collaborative Learning: Heidelberg: Springer-Verlag.

[11] Hew K., Cheung W. (2003): *Models to evaluate online learning communities of asynchronous discussion forums*. Australian Journal of Educational Technology 2003, 19(2), 241-259.

[12] Kirkley, S. E., Savery, J.R. & Grabner-Hagen, M. M. (1998). Electronic teaching: Extending classroom dialogue and assistance through e-mail communication. In C. J. Bonk & K. S. King (Eds.), *Electronic Collaborators: Learner-Centered Technologies for Literacy, Apprenticeship, and Discourse* (pp. 209-232). Mahwah, NJ: LEA, Publishers

[13] Lucas, N. (2004): Les procédés d'exposition et de developpement collectif dans un forum pédagogique : le cas Maxime, Paper presented at *Symfonic Symposium : "Forum et communautes d'enseignants et d'apprentissage,* http://www.dep.u-picardie.fr/sidir/articles/, last visited, 4/2005, Amiens, France, Jan.2005,

[14] Mason, R. (1991): *Methodologies for evaluating applications of computer conferencing in Kaye*, A.R.(ed) Collaborative Learning through Computer Conferencing: Heidelberg: Springer-Verlag.

[15] Analytic Technologies. Social Network Analysis. http://www.analytictech.com/

[16] Newman D., Webb B., Cochrane C. (1994): A content analysis method to measure critical thinking in face-to-face and computer supported group learning. Queen's University Belfast, Information Management Dept.

[17] Nonnecke, B. & Preece, J. (2001). Why Lurkers Lurk. Paper presented at the *AMCIS Conference*, Boston.

[18] Rourke L., Anderson T., Garisson D.R., Archer W. (2001). Assessing social presence in asynchronous text-based computer conferencing In *Journal of Distance Education*.

[19] PhpBB Community Building Software. http://www.phpbb.com/

[20] Romiszowski, A.J. & Mason, R. (1996). *Computer-Mediated Communication*. In D.H. Jonassen (Ed.). Handbook of Research for Educational Communications and Technology. 438-56.

[21] Rourke, L., Anderson, T., Garrison, D. R. & Archer, W. (1999). Assessing social presence in asynchronous text-based computer conferencing. *Journal of Distance Education,* 14(2).

[22] Web Wiz Guide: The Web Development Site. http://www.webwizguide.info/default.asp

[23] WebCT Learning System. http://www.webct.com/

Informing Teachers Live about Student Learning: Reporting in Assistment System

Mingyu FENG, Neil T. HEFFERNAN
Dept. of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609
{mfeng|nth}@cs.wpi.edu

Abstract. Limited classroom time available in middle school mathematics classes forces teachers to choose between assisting students' development and assessing students' abilities. To help teachers make better use of their time, we are integrating assistance and assessment by utilizing a web-based system ("Assistment") that will offer instruction to students while providing a more detailed evaluation of their abilities to the teacher than is possible under current approaches (refer to [7] for more details about the Assistment system). In this paper we describe the types of reports that we have designed and implemented that provide real time reporting to teachers in their classrooms. This reporting system is robost enough to support the 800 students currntly using our system.

Introduction

MCAS (Massachusetts Comprehensive Assessment System) is a graduation requirement in which all students educated with public funds in the tested grades are required to participate. Given the limited classroom time available in mathematics classes, teachers are compelled to choose between time spent assisting students' development and time spent assessing students abilities. To help resolve this dilemma, we are integrating assistance and assessment by utilizing "Assistment" system [7] supported by the U.S. Department of Education. The Assistments system offers instructions to students while providing a more detailed evaluation of their abilities to the teacher than is possible under current approaches. Each assistment consists of an *original item* and a list of *scaffolding questions*[1] which only show up to the students who have given wrong answers to original questions. Our supporting website "www.assistment.org" has been running for around 7 months, providing more than 100 assistments built using our online authoring tools [8] and is being used by 9 teachers and about 800 students.

Schools seek to use the yearly MCAS assessments in a data-driven manner to provide regular and ongoing feedback to teachers and students on progress towards instructional objectives. But teachers do not want to wait six months for the state to grade the exams. Teachers and parents also want better feedback than they currently receive. While the number of mathematics skills and concepts that a student needs to acquire is on the order of hundreds, the feedback on the MCAS is broken down into only 5 mathematical categories, known as "Strands". However, a detailed analysis of state tests in Texas [3] concluded that such topic reporting is not reliable because items are not equated for difficulty within these areas. To get some intuition on why this is the case, the reader is encouraged to try item 19 from the 2003 MCAS shown in Figure 1. Then ask yourself "What is the most important thing that makes this item difficult?" Clearly, this item includes elements from four of the 5 "strands" Algebra, Geometry (congruence), Number Sense (arithmetic operations) and Measurement (perimeter). Ignoring this obvious overlap, the state chose just one strand, Geometry, to classify the item,

[1] We use the term scaffolding question because they are like scaffolding that will help students solve the problem (and can "faded" later) so the scaffolds are meant to scaffold their learning. [2]

which might also be the first feeling of most people. However, as we will show below, we've found evidence there is more to this problem. The question of tagging items to learning standards is very important because teachers, principals and superintendents are all being told to be “data-driven” and use the MCAS reports to adjust their instruction. As a teacher has said “It does affect reports... because then the state sends reports that say that your kids got this problem wrong so they’re bad in geometry- and you have no idea, well you don’t know what it really is- whether it's algebra, measurement, or geometry.”

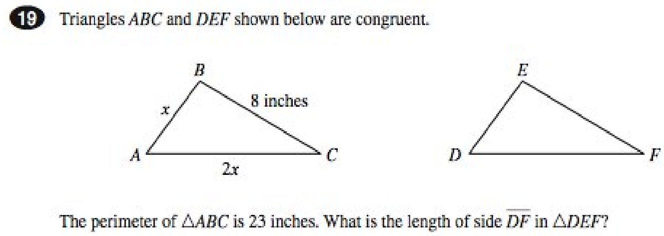


Figure 1: Item 19 from 2003 MCAS

There are several reasons for this poor MCAS reporting: 1) the reasonable desire to give problems tap-multiple knowledge components, 2) the fact that paper and pencil tests cannot figure out, given a student's response, what knowledge components to credit or blame, 3) there are knowledge components that deal with decomposing and recomposing multi-step problems, yet are currently poorly understood by cognitive science. So a teacher cannot trust that putting more effort on a low scoring area will indeed pay off in the next round of testing.

1. Data Source

The Assistment system is deployed with a completely internet savvy solution whereby students can simply open a web browser and login in to work on the problems. Our Java-based runtime system [5] will post each student's actions (other then their mouse movements) to a message server as an xml message that includes action timestamp, student ID, problem ID, student’s action type (did they attempt or just ask for help), student’s input and response, etc. The messages will be stored in the database server at WPI. As mentioned above, about 800 students of 9 teachers have been using the Assistment system every other week for about 7 months. Currently log records in our database show that about 50,000 MCAS items have been done and more than 600,000 actions made by these students. Since students are arranged to use our system regularly, our database will continually receive new data for the students. This allows our reporting system to assess students’ performance incrementally and give more reliable assessment as time goes on. These large amounts of student data also offer valuable material for further learning analysis using data mining or statistical techniques.

2. Transfer Model

A transfer model [4] is a cognitive model that contains a group of knowledge components and maps existing questions (original items and scaffolding questions) to one, or more of the knowledge components. It also indicates the number of times a particular knowledge component has been applied for a given question. It is called a “transfer model” since we hope to use the model to predict when learning and knowledge transfer will happen. Also as a predictive tool, transfer models are useful in selecting the next problem to work on. In the next section, we will show that transfer models are quite important for quality reporting.

Massachusetts Curriculum Frameworks breaks the 5 strands (Patterns, Relations and Algebra; Geometry; Data Analysis, Statistics and Probability; Measurement; Number Sense and Operations) into 39 “learning standards” for 8th grade math and tags each item with one of the 39 standards. As we have shown in Figure 1, Item 19 from Year 2003 has been tagged with "G.2.8 Congruence and similarity", the 2nd learning standard in the **G**eometry strand.

We have made several attempts of using the 39 MCAS learning standards to "code up" items, first using the state's mapping with one standard per question, and then with our own coding which allows each question to be tagged with multiple standards. However, we could not get statistically reliable coefficients on the learning standards. So we hypothesize that a finer grained model would help. Additionally, we need a more detailed level of analysis for reporting to teachers and for predicting students' responses on questions.

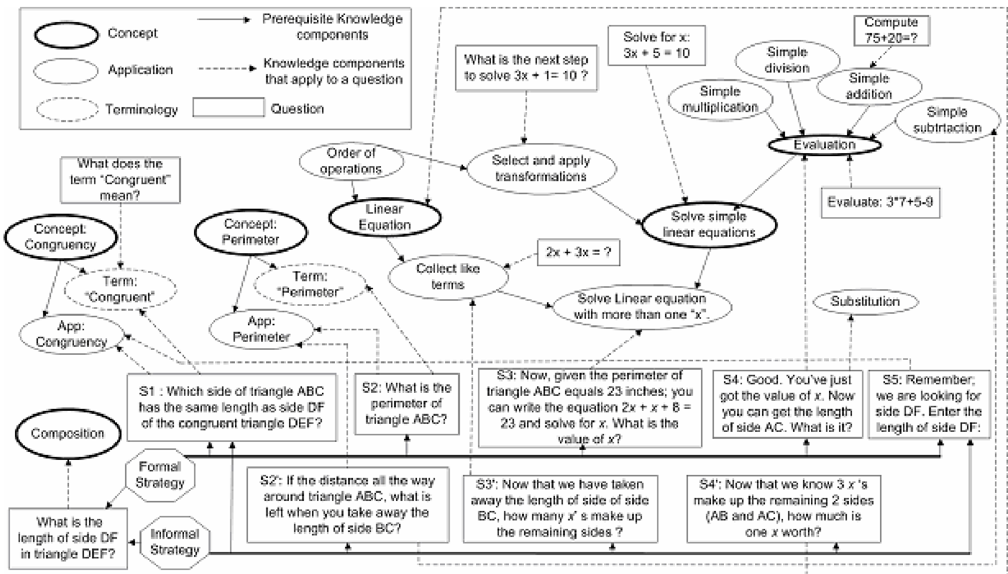


Figure2:A small piece of the WPI300 transfer model showing both how 14 questions (out of 245 in the WPI300) tap 19 knowledge components (out of 174 in the WPI300).

WPI300, which actually contains only 174 knowledge components so far, is the first model we have created. In the model, knowledge components are arranged in a hierarchy based on prerequisite structure. So far, 102 knowledge components in this transfer model have been used to tag 92 assistments (including 853 questions) in our system. Figure 2 shows 19 of the 174 knowledge components that we used to explain both a "formal" and "informal" problem solving strategy related to the item shown in Figure 1. We added a few other questions (like "What does the word 'congruent' mean?") to help define what a knowledge component means. Each of the scaffolding questions (S1 to S5) are mapped to one or two knowledge components. Tagging the scaffolding questions enable us to assess individual knowledge components instead of only overall performance. Each knowledge component might have prerequisite knowledge so that for a student to know "What does the word 'congruent' mean?" the student first needs to have mastered the "Concept of congruency" as shown by there being an arc between them.

Currently, we have been able to generate reports based on Massachusetts Curriculum Framework, as well as the WPI300 transfer model which reveals more detailed information about students' knowledge learning and knowledge components contained in problems. And we hope to be able to show that WPI300, as a finer grained cognitive model, will be more predictive. This is one subject of our current research.

3. Reporting System

3.1.1 Student Grade Book Report

Right now, we only have anecdotal information from our teachers that they find the reporting helpful. Teachers seem to think highly of the Assistment system not only because their students can get instructional assistance in the form of scaffolding questions and hint messages while working on real MCAS items, but also because they can get online, live reports on students' progress while students are using the system in the classroom.

The "Grade Book", shown in Figure 3.1, is the most frequently used report by teachers. Each row in the report represents information for one student, including how many minutes the student has worked on the assistments, how many minutes he has worked on the assistments today, how many problems he has done and his percent correct, our prediction of

Student Name	Total time before (min)	Time spent today (min)	Original Items					Scaffolding + Orig. Items				Most Difficult MA. Standard
			# Done	# Correct	% Corr.	MCAS Score*	Perf. Level	# Done	# Correct	% Correct	# Hint Req.	
Tom	34	0	15	3	20%	200	Failing	30	16	53%	15	N.1.8-understanding-number-representations (Error times: 5/6)
Dick	32	0	38	26	68%	242	Proficient	81	56	69%	4	P.1.8-understanding-patterns (Error times: 2/6)
Harry	33	0	20	9	45%	220	Needs improv.	63	28	44%	63	P.1.8-understanding-patterns (Error times: 8/10)

Figure 3.1: Grade Book on real student data

his MCAS score and his performance level[2,3] . Besides presenting information on the item level, it also summarizes the student's actions in an "Assistment metric": how many scaffolding questions have been done, student's performance on scaffolding questions and how many times the student asked for a hint. The "Assistment metric" tells more about students' actions besides their performance. For example, it exposes students' unusual behaviour like making far more attempts and requesting more hints than other students in the class, which might be evidence that students did not take the assistments seriously or was "gaming the system" [1].

In Figure 3.1, we see that these 3 students have used the system for about 30 minutes. (Many students have used it for about 250 minutes). "Dick" has finished 38 original items and only asked for 4 hints. Most of the items he got correct and thus our prediction of his MCAS score is high. We can also see that he has made the greatest number of errors on questions that have been tagged with the standard "P.1.8 understanding patterns". The student had done 6 problems tagged with "P.1.8" and made errors on 2 of those problems. Teachers can also see "Harry" has asked for too many hints (63 compared to 4 and 15). Noticing this, a teacher could go and confront the student with evidence of gaming or give him a pep-talk. By clicking the student's name shown as a link in our report, teachers can even see each action a student has made, his inputs and the tutor's response and how much time he has spent on a given problem (which we will not present here for lack of space). The "Grade Book" is so detailed that a

▶ Item 2 A-2002 (Find next term in sequence) Morph1

	Question text	Action
Find the next term in the sequence shown below: 1, 4, 13, 40, 121, _?_ A. 161 B. 242 C. 363 D. 364	Find the next term in the sequence: 1, 4, 13, 40, 121, _?_	364
	Excellent. Lets put the numbers into a diagram this way: You may notice that the differences between each two neighboring terms in the sequence also represent a sequence: 3, 9, 27, 81 and so on. What is the next term following 81 in this sequence?	HINT

Figure 3.2. Items tagged with difficult knowledge component

2 Our "prediction" of a student MCAS score is at this point primitive. The column is currently simply a function of percent correct. We might even remove these two columns related to MCAS score prediction until we feel more confident in our prediction, in another word, "rough and ready".

3 In our recent research, we have found a strong correlation between our prediction for the 68 students who have used our system May 2004 and their real MCAS raw score (r = .7) [7]. But since that is a rather small group of students compared to the number of students now (68 vs. 8000), we'll continually refine our prediction function based on this year's data.

student commented: “It’s spooky”, “He’s watching everything we do” when her teacher brought students to his workstation to review their progress.

By clicking the link of the most difficult knowledge component, the teacher can see what those questions were and what kind of errors the student made. (See Figure 3.2) Knowing students’ reactions to questions helps teachers to improve their instruction and enable them to correct students’ misunderstandings in a straightforward way. Finding out students’ difficult knowledge components also offers a chance to improving our item selection strategy. Currently, random and linear are the only two problem selection strategies supported by our runtime system. Another option could be added if we can reliably detect difficult knowledge components of each individual student, which requires the runtime system to preferentially pick items tagged with those hard knowledge components for the students so that students would have more opportunity to practise on their weak point.

3.1.2 Class Summary Report

These are 5 knowledge components your students doing well:

Knowledge Component	# Records	# Correct	Correct Rate
Application: Simple multiplication	23	19	83%
Application: Elimination	22	17	77%
Terminology: ( ) means multiplication	81	56	69%
Terminology: Square Roots	102	68	67%
Application: Compare Expressions	101	67	66%

These are 5 knowledge components your students need more practice:

Knowledge Component	# Records	# Correct	Correct Rate
Application: Simple division	110	24	22%

Figure 3.3 Class summary report for a teacher’s classes

“Class Summary” is a report we developed to inform teachers about the knowledge status of classes. Teachers can select their favourite transfer model, specify the number of knowledge components to be shown in the report. Knowledge components are ranked according to their correct rate which is students’ correct rate (demonstrated in Figure 3.3 as green bars together with percent correct as values) at the items tagged with those knowledge components. By clicking the name of a knowledge component (shown as a hyperlink in Figure 3.3), teachers are redirected to another page showing the items tagged with the knowledge components. In the new page, teachers are able to see the question text of each item and continue to preview or analyze the item if they want to know more about the item.

By presenting such a report, we hope we can help teachers to decide which knowledge components and items should be focused on to maximize the gain of students’ scores at a class level when instructional time is limited.

3.1.3 Class Progress Report

Class	Date	# Correct	# Total	# Student	Avg. Score	Std. Dev.
Period 3	2004-09-21	153	382	23	18	9.95
Period 3	2004-10-27	427	773	23	25	11.18
Period 3	2004-11-10	630	1119	24	26	11.03
Period 3	2004-12-01	879	1437	22	29	10.20
Period 3	2004-12-15	1167	1790	21	32	8.24
Period 3	2005-02-02	1341	2029	20	33	7.96
Period 3	2005-02-16	1702	2576	23	33	6.67
Period 3	2005-03-02	1972	3065	24	33	6.61
Period 3	2005-03-16	2106	3288	23	33	6.58

Figure 3.4 preliminary progress reports for a class

Since our teachers let their students using the Assistment system every two or three weeks, we thought it would be helpful if we can show to teachers students’ progress by looking at their performance at each time they worked on the assistments.

Figure 3.4 shows our preliminary progress report for a teacher’s class. In this report, we can see this class has been using our system since September 21st, 2004 and has used it as a class 9 times. The average of students’ predicted MCAS raw score increased from 18 to 33, and kept being 33 for a while. [Note, we are being conservative in calculating these

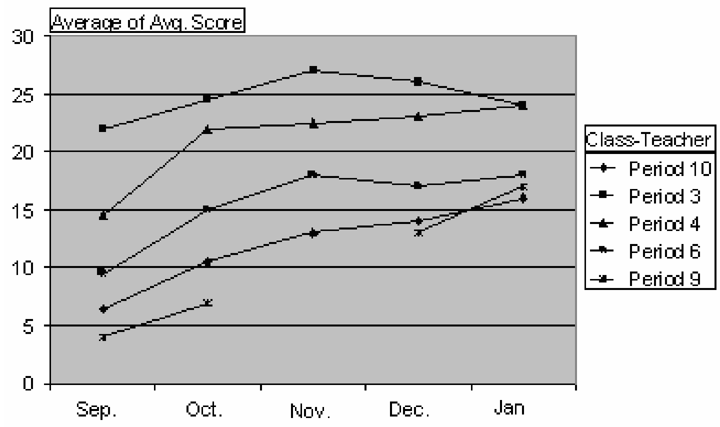


Figure 3.5 predicted MCAS Score over months

predicted MCAS scores, in that we calculate for each students their predict scores using every items them have even done in our system, instead of using only the items done on day they came to the lab.] Standard deviation of scores is also displayed as a column to help teachers see performance variance in the class.

The progress of students' predicted MCAS raw score over months is more clearly shown in Figure 3.5. Those students (all from school A) have been using our system for more than 5 month starting from Sep., 2004. We can see in this graph students' predicted MCAS score on average increase steadily with month passing (even for class "Period 9" which "left" us for two months).

3.2 Analysis of Items

A report is built to show difficulty each problem in our system. (See Figure 3.6: 5 lines of the 200+ lines that are in the report). By breaking original items into scaffolding questions and tagging scaffolding questions with knowledge components, we are able to analyze individual steps of a problem. Figure 3.7 is what we call a scaffolding report because it reports statistics on each of the scaffolding questions that are associated with a particular original item.

Item 20 N-2003 Morph (3/4 of 1 2/3)	24%
Item 20 N-2003 (2/3 of 1 1/2) Morph2	26%
Item 18 G-1998 (Angle in isosceles triangle)	27%
Item 35 G-2001 (Angle between clock hands)	27%
Item 13 D-1998 (Eiffel Tower model)	29%

Figure 3.6: Problems order by correct rate

On the first line of Figure 3.7, we see this problem is hard since only 12% of the students got it correct on their first attempt. Of the 180 students having done this item so far, 154[4] students could not get the correct answer to the original question, thus forced by the system to go through scaffolding questions to eventually

ID	Question	Correct Answer	% Correct	Hint Req.	# Attempt	Common Errors: Resp.	Common Errors: #	Common Errors: Buggy Message	WPI's Use of MA. Standard	WPI's Knowledge Components
	Triangles ABC and DEF are congruent. The perimeter of triangle ABC is 23 inches. What is the length of side DF in triangle DEF?	10	12%	56%	180	8	15	N/A	**G.2.8**, M.3.8, P.7.8	Composition, T.3, A.3, T.4, A.4, A.12, A.15, A.17
						16	13	N/A		
						23	8	N/A		
1	Which side of triangle ABC has the same length as side DF of triangle DEF?	ac	23%	50%	154	ab	13	Side AB corresponds to side DE of triangle DEF, not DF. Try again, please.	**G.2.8-congruence-and-similarity**	Term: "Congruency", Appl: Congruency
						DF	6	N/A		
2	What is the perimeter of triangle ABC?	2x+x+8	39%	20%	148	2x + 8	69	No. It looks like you have added just two of the sides of triangle ABC. Perimeter is the sum of all the sides.	M.3.8-using-measurement-formulas	Term: "Perimeter", Appl: Perimeter
3	Now, given the perimeter of triangle ABC equals 23 inches, you can write the equation **2x + x + 8 = 23** and solve it for **x**. What is the value of **x**:	5	25%	52%	147	15	13	N/A	P.7.8-setting-up-and-solving-equations	Appl: Solve linear equation
						13	10	N/A		
						8	10	N/A		
4	Remember, we are looking for side DF. Enter the length of side DF:	10	30%	43%	143	5	26	N/A	G.2.8-congruence-and-similarity	Appl: Congruency
						2x	2	N/A		
						8	3	N/A		

Figure 3.7: A scaffolding report generated by Assistment reporting system

[4] You may notice that 154 is less than 88% of 180, which should be about 158. And the number of attempts on later scaffolding questions went down more. That's because students could log out and log back in to redo the original question to avoid going through all scaffolding questions. This problem has been solved.

solve the problem. 56% of students asked for a hint, telling you something about students' confidence when confronted with this item. (It is useful to compare such numbers across problems to learn which items students think they need help on but don't, and vice versa). Remember that the state classified the item according to its “congruence” (G.2.8) shown in bold. The other MA learning standards (M.3.8, P.7.8) are the learning standards we added in our first attempt to code using the MCAS 39 standards. We see that only 23% of students that got the original item incorrect can correctly answer the first scaffolding question lending support to the idea that congruence is tough. But we see a as low percent correct 25% on the 3rd question that asks students to solve for x. The statistics result gives us a good reason to tag “P.7.8-setting-up-and-solving-equations” to the problem.

Teachers want to know particular skills or knowledge components that cause trouble to students while solving problems. Unfortunately the MCAS is not designed to be cognitively diagnostic. Given the scaffolding report can provide lower level of cognitive diagnosis, our cooperating teachers have carefully designed scaffolding questions for those tough problems to find out the answer. For example, one teacher designed an assistment for (“What's ¾ of 1 ½?”), item 20 of year 2003 8th grade MCAS. The first scaffolding question for the assistment is “what mathematical operation does the word ‘of’ represent in the problem”. This teacher said, “Want to see an item that 97% of my students got wrong? Here it is… and it is because they don't know ‘of’ means they should multiply.” The report has confirmed the hypothesis. 40% of students could not select “multiplication” with 11 of them selecting “division”.

The scaffolding report has helped us to develop our tutors in an iterative way. For each question, the report shows top common errors and corresponding “buggy” messages. When building the Assistments, we have tried to “catch” common errors students could make and give them instructive directions based on that specific error, such as correcting students' misunderstanding of question texts or knowledge concepts. But given that students may have different understandings of concepts, assistments may give no messages for some errors, which means our tutor lost chances to tutor students. Also, students may feel frustrated if they are continually being told “You are wrong” but get nothing instructive or encouraging. As shown in Figure 3.7, the wrong answer “15” to the third question has been given 13 times, but the assistment gave no instructive messages. Noticing this, the assistment builders can improve their tutor online by adding a proper “buggy” message for this error.

We also display a table that we call “Red & Green” distribution matrix as shown in Table 3.1 in the scaffolding report. Numbers in the cells show how many students got correct (indicted by green number in un-shaded cells) or wrong (indicated by red in shaded cells) on a

Table 3.1: “Red & Green” distribution matrix

<table>
<tr><td>Original</td><td colspan="16">154</td><td>22</td></tr>
<tr><td>Q1</td><td colspan="8">119</td><td colspan="8">35</td><td rowspan="4">N/A</td></tr>
<tr><td>Q2</td><td colspan="4">85</td><td colspan="4">34</td><td colspan="4">12</td><td colspan="4">23</td></tr>
<tr><td>Q3</td><td colspan="2">72</td><td colspan="2">13</td><td colspan="2">21</td><td colspan="2">13</td><td colspan="2">8</td><td colspan="2">4</td><td colspan="2">18</td><td colspan="2">5</td></tr>
<tr><td>Q4</td><td>45</td><td>8</td><td>5</td><td>7</td><td>15</td><td>6</td><td>3</td><td>10</td><td>6</td><td>2</td><td>1</td><td>3</td><td>15</td><td>3</td><td>1</td><td>4</td></tr>
</table>

question. We split the number as the questions' sequence number grows so that it also represents how those students have done on previous questions. In this example, we see that 4 students who have answered the original question wrong went through all of the scaffolding questions correctly. Given that, we tend to believe those students have mastered the knowledge components required by each step and but need instruction on how to “compose” those steps. It's also worth pointing out that there are 8 students who answered original question wrong but answered correctly to the last question, which asks the same question as the original one. Since the assistment breaks the whole problem into scaffolding steps and gives hints and “buggy” messages, we would like to believe those students learned from working on the previous steps of this assistment.

3.3 Performance evaluation

Our reporting system was used in May, 2004. In the early stage, it worked well and most reports at the class level could be generated in less than 10 seconds. And it took 10 to 20 seconds to generate a scaffolding report at "system" level. The performance went down when the number of recorded student actions increased past 1 million. In particular, we have seen the "Grade Book" report took more than 2 minutes, which we consider unacceptable as a live report. We then switched to Oracle database which provides mechanisms, such as view, stored procedure, to improve query performance. We also updated the approaches we used to generate the reports. Now we can generate the "Grade Book" report in about 7 seconds on average. The time required to generate the system level scaffolding report for Item 19 (See Figure 3.7) is about 5 seconds.

4. Conclusions

In conclusion we feel that we have developing some state-of-the-art online reporting tools that will help teachers be better informed about what their students know. Our implicit evaluation is that we have made it possible for all these reports to work live in the classroom. We feel we have a lot to do in automating yet further the statistical analysis of learning experiments. We have done some learning analysis with this year's data set environing over 800 students and 30 Learning Opportunity Groups. In particular we see students are about 5% on their second opportunity and this was statistically significant [7]. Also since doing learning analysis by hand is both time consuming and fallible, another aim of our reporting system is to automat learning analysis process. Our long term vision is to let teachers create content, and send them email automatically when we know that their content is better (or worse) than what we are currently using in the assistment systems. We feel we have taken some stops in that direction.

References

[1] Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of 7th International Conference on Intelligent Tutoring System*s, 2004, Maceio, Brazil

[2] Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates.

[3] Confrey, J., Valenzuela, A. & Ortiz, A. (2002). Recommendations to the Texas State Board of Education on the Setting of the TAKS Standards: A Call to Responsible Action. At http://www.syrce.org/State_Board.htm

[4] Croteau, E., Heffernan, N. T. & Koedinger, K. R. (2004) Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model, *Proceedings of the 7th International Conference on Intelligent Tutoring System*, 2004, Maceio, Brazil

[5] Nuzzo-Jones, G., Walonoski, J.A., Heffernan, N.T., Livak, T. (2005). The eXtensible Tutor Architecture: A New Foundation for ITS. *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education* 2005, Amsterdam, Poster paper, to appear.

[6] J. Mostow, J.E. Beck, R. Chalasani, A. Cuneo, and P. Jia. Viewing and Analyzing Multimodal Human-computer Tutorial Dialogue: A Database Approach. *Fourth IEEE International Conference on Multimodal Interfaces* (ICMI 2002), October, 2002.

[7] Razzaq, L, Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T., Upalekar, R., Walonoski, J., Macasek, M., Rasmussen, K., Koedinger, K., Junker, B., Knight, A., Ritter, S. (2005). The Assistment Project: Blending Assessment and Assisting. *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education* 2005, Amsterdam, to appear.

[8] Turner, T., Macasek, M.A., Nuzzo-Jones, G., Heffernan, N.T. (2005). The Assistment Builder: A Rapid Develoment Tool for ITS. *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education* 2005, Amsterdam, Poster paper, to appear.

Towards a Design Pattern Language to Track Students' Problem-Solving Abilities

Fabrice Gibert-Darras[1], Elisabeth Delozanne[1], Françoise Le Calvez[1], Agathe Merceron[3], Jean-Marc Labat[4], Fabrice Vandebrouck[2]

[1]CRIP5 Université René Descartes
45 rue des Saints Pères, 75270 Paris Cedex 6 France
[2]DIDIREM Université Paris VII
2 Place Jussieu, 75 251 PARIS Cedex 5 France
[3]ESILV-GI, Département d'informatique
92916 Paris-La Défense Cedex France
[4]LIP6, Université Paris 6
12, rue cuvier 75005 Paris France

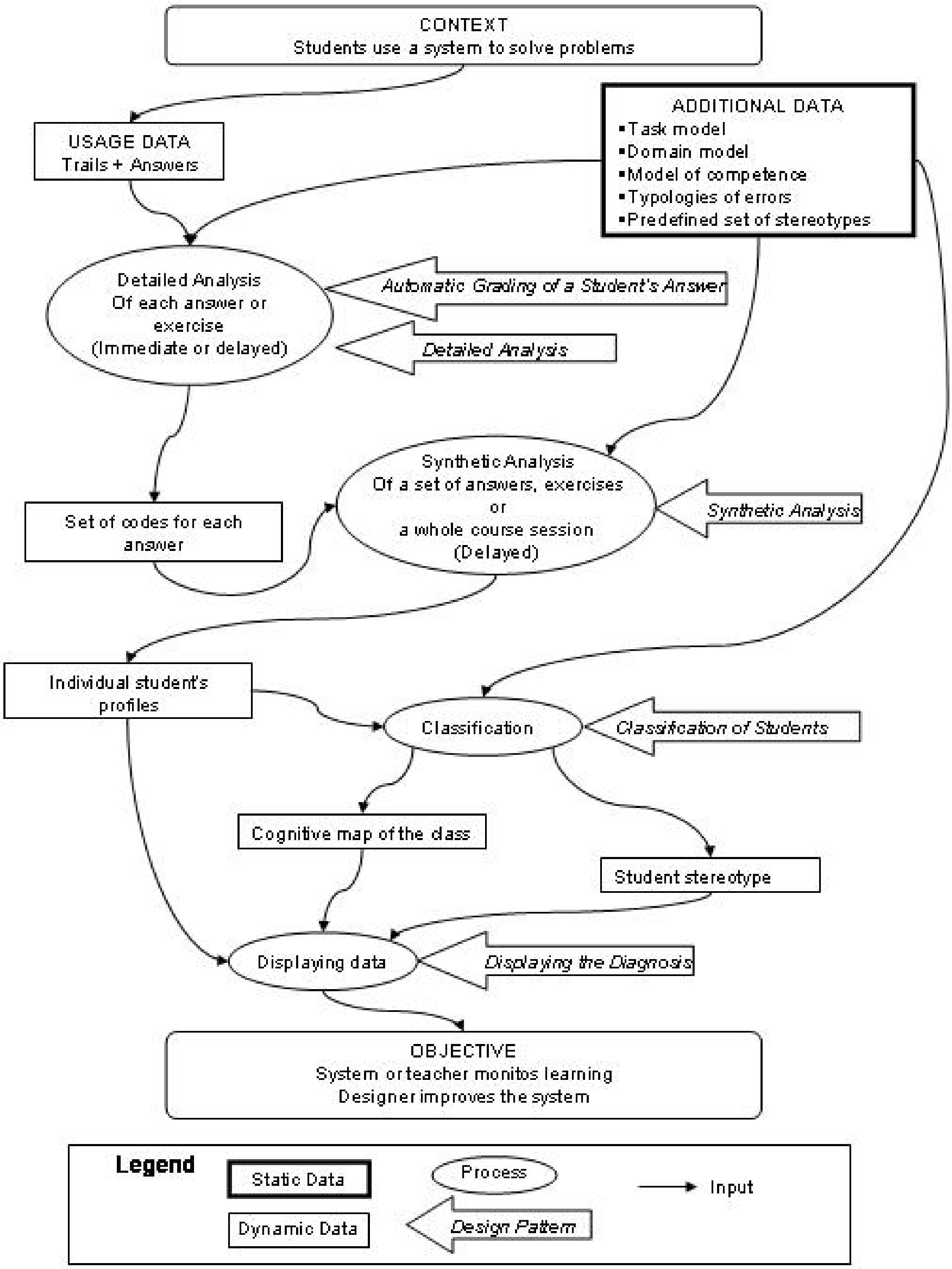
This paper is a first contribution towards a set of Design Patterns to track Usage of Learning Systems that is the focus of the DPULS JEIRP in the context of Kaleidoscope network. In our multidisciplinary team AIDA we are particularly involved in problem based learning environments and cognitive diagnosis. We implemented various systems for different domains, different types of students and institutions. BSMod [1] is a generic system using Bayesian Networks for student modeling; so far it has analyzed a sample collected by Pepite [2]. Combien? [3] is a software to learn problem-solving methods in combinatorics at high school level, it collected six hundred students' answers to exercises. Diane [4] is a diagnosis system for arithmetical problems in elementary schools, it has worked on one thousand two hundred students' tracks. Java Course [5] is an on-line introductory course on Java programming for second year university students, it has analyzed one hundred students' logs and answers. Pepite is a diagnosis system to support teachers in monitoring algebra learning in secondary schools, so far it has collected and analyzed a sample of exercises solved by three hundred secondary school students. We also experienced track analyses from four hundred logs of students connected to Wims [6], a server of math exercises at university level.

For each experiment, researchers analyzed usage tracks and problem-solving answers or performance. These tracks are diverse and the recorded data are different. According to experienced systems, a detailed analysis of students' answers is processed on-line and the result is recorded in the tracks or it is processed after the student session. The analysis is fully automatic or human supported.

We present a first draft of design pattern language to track students' problem-solving abilities to be discussed in the AIED 2005 workshop based on our diverse experiment. Design patterns were already proposed for the practice of teaching and learning [7], [8] and in the field of the Computer Learning System [9], [10], [11], [12], but there is not a real work yet on design patterns about the student tracking. We used the Alexander format of design pattern description [13].

Figure 1 is a synthetic view of the design pattern in relation with the process of track analysis. Each design pattern and its dependence diagram are presented.

Figure 1: Track synthetic process Design Patterns



Name	*Automatic Grading of Students' Answers*
Context	Your system records students' answers to problems where students can build their own solution. A task model is available.
Problem	How to automatically grade the student's answer to a problem? Or if it is not possible, how to support human grading?
Motivation	You want to know if the student answered the questions correctly. You want an automatic ranking (as much as possible).
Forces	Answers expressed by students are very different according to domain diversity and to students' cognitive diversity. The more freely the answer is expressed, the more difficult is the grading Except on very specific exercises, it is impossible to predict all types of answers
Solution	Several approaches can be used and combined. If you can define a set of predefined solutions, you can use a pattern matching approach. If the answer is expressed in a formal language, you can use specific software to assess its correctness such as a compiler, a theorem prover, a Computer Algebra System (CAS), SQL engine etc.). If the answer is expressed in natural language, you can use natural language processing tools but very often, you will need a human assessor.
Examples	Diane, Wims, Java Course, Combien, Pépite.
Actors	Teacher, Tutor, Researcher, Student (auto-evaluation).
Related Patterns	*Detailed Analysis*

Name	*Detailed Analysis*
Context	The student answered a question or solved a single exercise. The answer is recorded as well as usage data (time spent, incurred actions, help requests, etc) A task model, a model of competence, typologies of exercises and typologies of errors are available
Problem	How to analyze the answer of a student to a given question?
Motivation	You want to analyze, correct, comment on or classify the student's answer.
Forces	The answer can take a multitude of forms and its interpretation can raise many problems (typing errors, incomprehension of the system). Analysis is a very complex process. It is necessary to pay attention to bad interpretations or errors of manipulation.
Solution	This detailed analysis is immediate or delayed. An immediate assessment is carried out when the answer is given. In this case the system often gives an immediate feedback to the student and this feedback is recorded along with the answer, its assessment, the time, etc. This delayed detailed analysis occurs when, during the problem solving session, the system collects students' answers and usage data but these are analyzed after the session. The analysis depends on the actors' objectives. Actors may need success rates, grades or might wish for a more precise analysis on resolution strategies or on how much time the student spent before validating his/her answer, etc. Answer analysis can be automated, human, or supervised. From the different models (competence, task), and predefined typologies (exercises and errors), you build grids of analysis for each exercise and implement procedures or heuristics to carry out a full or partial analysis. Then, you can build a system using these heuristics to analyze the strategy of resolution. Thus you determine competences that were used by the student to answer. But you have to remember that hesitations and errors hold a significant part in the analysis of the student, it is possible to see strong points and weaknesses to propose remedial actions. The teacher carries out the human analysis. The analysis is supervised when the teacher or a researcher is needed to complete, correct or modify the software analysis.
Examples	Diane, Logic-Tutor, Combien, Wims, Pépite, Java Course.
Actors	Teacher, Tutor, Researcher.
Related Patterns	*Automatic Grading of Students' Answer, Synthetic Analysis.*

Name	*Synthetic Analysis*
Context	You collected the student's answers to a set of questions, exercises or to a whole course. The detailed analysis of each answer was carried out. A domain model, a model of competence, and typologies of errors and exercises are available
Problem	How to get an overview of the student's whole activity?
Motivation	Fine-grained analysis does not facilitate decision-making. Strategic decision-making requires a high level description of the student's activity: To monitor learning or to improve their course, teachers need a synthetic view on the student learning activity and an account of the student's evolution. Thus, you want to define the main features of the student's competence.
Forces	Trade offs have to be found between synthetic view and account of the diversity, complexity and evolution of the student's activity Main dimensions of analysis are different for different actors A pedagogical, didactical, or cognitive model is required to define the main dimensions to synthesize the analysis
Solution	The synthetic analysis is always delayed because it requires detailed analyses of every answer during the whole session. From the domain model or the model of competences, you have to define main dimensions to account for clusters of abilities or success rates or errors. For example, you could decide to determine the student's evolution on a special competence during the course or on his/her strong points and weaknesses, etc. Then you will find recurring abilities or errors in detailed analyses using a typology of exercises or errors. For instance, if your dimension of analysis is "to solve difficult exercises" you will calculate how many difficult exercises the student solved or tried to solve. If your dimension of analysis is "usage of algebra", you will find that a student does use algebra to prove relation either sometimes or always. Finally, you may classify the student in comparison with others (his/her class mates, a priori classification: see pattern *Classification of students*)
Examples	Bsmod, Wims, Pepite, Diane.
Actors	Teacher, Tutor, Researcher, Student.
Related Patterns	*Detailed Analysis, Displaying the Diagnosis, Classification of Students*

Name	*Classification of Students*
Context	Your system built a synthetic analysis of learning activity for every student in a whole class or course. Teachers or educational researchers defined a set of student stereotypes.
Problem	How to draw up an assessment of the class or a representation of the whole class activity? How to categorize types of students in the class?
Motivation	You want a cognitive map of the class built from the data collected on individuals. To monitor learning or to improve a course, teachers need to classify the usage of learning materials and to group students according to their performance or abilities.
Forces	There are different individual characteristics which must be preserved. There are many different ways of grouping students according to actors' needs.
Solution	You have to specify a classification of students. By a statistical, empirical or theoretical analysis you can define clusters of students and characterize these clusters. Thus, each cluster defines a stereotype. Stereotypes can be very simple (low achieving, regular, high achieving students), multidimensional (ranking students on a multidimensional scale) or describing usage (player, systematic learner, butterfly etc.) Then, you classify each student in a stereotype. Several techniques are available: decision trees, rates and thresholds etc. Finally, you display a map of the class. For instance, you display each stereotype with the students' names if the teacher wants a grouping of his/her students or you display charts with the number of students by stereotypes.
Examples	Pépite, Java course, Wims.
Actors	Teacher, Tutor, Researcher.
Related Patterns	*Synthetic Analysis*, *Detailed Analysis*, *Displaying the Diagnosis*

Name	*Displaying the Diagnosis*
Context	You have already conducted a requirement analysis with the different categories of actors. Your system analyzed the student's tracks Your system now has rich information about the student's problem solving skills and you wish to make the results of the analysis available for the different types of actors.
Problem	How to display the analyses of the tracks? How to make them usable and understandable?
Motivation	To improve learning activity, actors need a multidimensional point of view on a student's features, a report of his/her strong and weak points in a preset categorization of competence, on overview on the whole class competence.
Forces	There are many and diversified data to display. The different actors don't have the same requirement. The displaying must suit the expectations of the users.
Solution	Each type of user needs an adequate presentation. First, you have to structure the results from the different actors' point of interest. For example, a teacher may need to access the student's grades (success, failure or partial failure) and the success rates on his/her various abilities. Another teacher may want to see the learning strategies and the cognitive map of the class. A researcher may wish to display the all catalogue of answers to a specific exercise or statistics about success rates on a question. A student may need to display his/her grades. Second, you have to choose an adequate presentation or interaction model. You can represent the student's characteristic weaknesses and strong points with a representation by text, histogram or diagram. The interaction model could include browsing functionalities to verify or modify the automatic diagnosis. This could be made by using data base queries. It is important to let users adapt the display to their needs, but it is also important to set default parameters for occasional users. In our experiments, users need views on the student activity with different levels of abstraction. It is important to give an overview and leave the possibility to have more details or explanations on how this overview was build. The overview is needed for strategic decision-making. Details are required to understand better the synthetic view, to correct it or to take tactical decision.
Examples	Diane, Wims, Combien, Pépite, Java Course.
Actors	Teacher, Tutor, Students, Researcher.
Related Patterns	*Synthetic Analysis, Classification of Students*

This paper has presented a set of design patterns on the student tracking. However, it is a work in progress. We are currently mining for recurrent problems that we know how to solve. The goal is to define and write more design pattern on this subject. We want to draw up a pattern language capturing our experience in recording and analyzing usage of learning systems. The next improvement of the pattern language will be to propose a design pattern browser.

References

[1] Hibou M., Labat J-M., Spagnol J.-P.. (2003), Automatically generated exercise pages. Proceedings of Artificial Intelligence in Education, Sydney, Australia, IOS Press, Amsterdam, 492-498.

[2] Delozanne E., Grugeon B., Prévit D., Jacoboni P. (2003). Supporting teachers when diagnosing their students in algebra. Proceedings of Artificial Intelligence in Education, Sydney, Australia, IOS Press, Amsterdam, 461-470.

[3] Le Calvez F., Giroire H., Duma J., Tisseau G., Urtasun M. (2003), Combien? a Software to Teach Students How to Solve Combinatorics Exercises. Proceedings of Artificial Intelligence in Education, Sydney, Australia, IOS Press, Amsterdam, 447-453.

[4] Hakem K., Sander E., Labat J-M.. (2005). DIANE, a diagnosis system for arithmetical problem solving. Proceedings of Artificial Intelligence in Education 2005, Amsterdam, Holland (to appear).

[5] Duval P., Merceron A., Scholl M. and Wargon L. Empowering Learning Objects: an experiment with the Ganesha platform, ED-MEDIA 2005, Montreal, Quebec, June 27-July1, 2005.

[6} Vandebrouck F., Cazes C. (2004), Exploitation des journaux de traces (log) en mathématiques à l'université. Actes du colloque TICE 2004, 250-256

[7] Fricke A. and Völter M.,(2000), SEMINARS, A Pedagogical Pattern Language about teaching seminars effectively, presented at EuroPLoP 2000, Irsee, Germany

[8] Eckstein J., (2000), Learning to Teach and Learning to Learn Running a Course, presented at EuroPLoP 2000, Irsee, Germany.

[9] Avgeriou P., Papasalouros A., Retalis S., and Skordalakis M., (2003), Towards a Pattern Language for Learning Management Systems, Educational Technology & Society, vol. 6, 2,. 11-24,

[10] Goodyear P., Avgeriou P., Baggetun R., Bartoluzzi S., Retalis S., Ronteltap F., and Rusman E., (2004), Towards a Pattern Language for Networked Learning, in Proc NLC'04 (Networked Learning Conference). Lancaster, UK

[11] Steeples C. and Zenios M.,(2004) , Organisational Patterns for E-Learning Centres, in EDMEDIA'04.

[12]Rune B., Ellen R., and Caterina P., (2004), Design Patterns For Collaborative Learning: From Practice To Theory And Back, in EDMEDIA'04.

[13] Alexander C., Ishikawa S., Silverstein M. (1977), A Pattern Language - Towns, Buildings, Constructions, Oxford University Press.

Web sites:
http://pepite.univ-lemans.fr
http://www.math-info.univ-paris5.fr/combien
http://www.pedagogicalpatterns.org
http://www2.tisip.no/E-LEN

Helping the Interpretation of Web Logs: Application to Learning Scenario Improvement

Jean-Mathias Heraud*, Jean-Charles Marty**, Laure France*, Thibault Carron**
*ESC Chambéry, **University of Savoie*
{jean-mathias.heraud|jean-charles.marty|laure.france|thibault.carron}@univ-savoie.fr

Abstract. *Our objective is to observe learner behavior within a Web-based learning environment and to understand it. In some cases, this comprehension can lead to improve the learning scenario itself. In this paper, we first show how to gather observations from various sources. Next, we propose to compose these observations into a trace. This trace reflects the activity of the learner and will be used as a basis for interpreting his/her behavior. We illustrate our approach through an experiment carried out on a dedicated Web-based learning environment.*

1. Introduction

Recent Web-based learning environments allow designers of learning activities to define their own learning scenarios. The rules of a learning activity are thus clearly defined. To make the scenarios operational, they are described formally, most of the time in IMS-LD [hr1]. New perspectives are, for instance, the possibility to observe collaborative activities in distance learning. The observation can be performed by multiple approaches: qualitative techniques using interviews carried out at the end of activities; quantitative techniques based on analysis of the logs representing the various actions of the users.

Why is this observation essential? First of all, it is used for the regulation of the activity. In a class, a teacher is unceasingly searching for signs which enable him/her to perceive the state of comprehension of learners. Then s/he adapts his/her scenario by possibly adding an explanatory sequence, or adding a complementary exercise, or leaving an exercise for a next session. In this case, the observation permits the acquisition of useful information to adapt the course, throughout the activity. Second, the observation produces a great quantity of information useful for the persons in charge of the teaching activity. We can for example examine, at the end of the learning session, the traces of the activities of some learners to know if they have followed the recommended learning scenario. Finally, the observation is an important factor for the management of the quality of the learning scenario. As mentioned in [5], the quality of learning scenarios can be evaluated in the same manner as software processes, for example with the CMM model [6]. The idea is to find out how to modify the scenario in order to take advantage of what really happened. In that case, the fact that some recommended activities were systematically ignored by the users is an important hint: Are these activities necessary? Are they introduced at the right time?

Within the framework of this paper, we briefly present the experimentation and the goals we want to reach. We then highlight the diversity of sources to be observed. The assistance provided to the user on the interpretation of collected information is illustrated through an example. Finally, we draw conclusions from this experimentation with respect to the goals we fixed.

2. Goals and experimentation

2.1 Goals

The first goal that we want to reach is to help the teacher to understand the activity of learners. The teacher provides a learning scenario and the learner's activities are then compared to the recommended scenario. The second objective relates to the trace analysis once the learning activity has been completed. It is interesting to compare the traces of a picked sample, for instance the students who validated the session, in order to understand the reasons of their success. The third objective that we set in this experimentation relates the improvement of learning scenarios. The possible re-use of these scenarios leads us to revise them. As a result, an a posteriori analysis of the traces is necessary to identify the redundant, missing or useless steps in the scenario.

2.2 Description of the experimentation

The experimentation proceeded with a public of 3rd year students at the Graduate Business School of Chambéry (France). It lasted one hour and a half, and took place at the same time in two different rooms. There were eighteen students, a teacher and a human observer in each of the classrooms. During this session, the students had to carry out a suggested learning scenario consisting of two independent activities. All the necessary resources (course documents, original images) as well as communication tools (webmail, forums) were available on the experimentation platform. Students were not allowed to communicate except through computers. The teacher communicated with students using an instant messenger and validated the work completed. The observer took notes about the students' and the teacher's behaviors on a special grid.

3. What do we need to observe?

3.1 Diversity of the observed elements

Our objective is to observe learners' behavior in order to compare it with a learning scenario prescribed. According to the diversity of the observed elements, the trace resulting from this observation will be more or less interpretable by the teacher. For example:

- If the observed elements are at a low level (i.e. close to the elementary actions of the machine or the software used), then the trace is not easily interpretable by a teacher. A trace made up of hundreds of lines such as ***"192.168.105.65-[24/Sep/2004:15:09:01] +0200 GET /course/index.php?module=trace&type=adm&fnct=view http/1.1"*** would not be directly exploitable by a teacher.
- If the observed elements are abstract and represent activities with clear semantics, then the trace will be more easily interpretable by a teacher. A trace of that type, presented in figure 1, suggests that reading "document 1" helped the learner to succeed exercise 1.

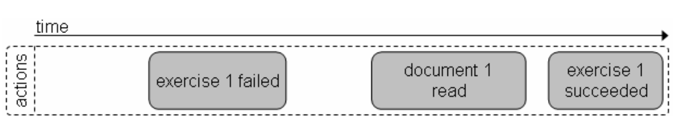


Figure 1: observation of the scenario

A first approach thus would encourage us to instrument the Web-based learning environment in order to observe abstract elements. This approach is unfortunately too restrictive because it reduces the diversity of the observed elements while forcing us to retain only those for which semantics can be found a priori. Thus, the teacher can observe the omission of scenario's steps by one learner (for example a course document was not read) but will not be able to observe the activities not envisaged and useful for the learner to finish the scenario (for example the search for a solution to the exercise in a forum). To avoid this disadvantage, we propose an approach consisting of observing and then composing observable elements from several sources.

3.2. Observation Sources

We retained three observation sources related to the machine: the learning scenario, the server and the learner's station. A fourth source relates to external events that cannot be traced by the machine. Each of these sources requires a particular instrumentation.

3.2.1. Observation of the learning scenario

By instrumenting the learning scenario activities, we obtain a trace whose level of observation corresponds directly to the level of restitution to the teacher. We had already explored this type of observation within the project Pixed [3] where the visualization of traces helped the teacher to understand the activities carried out by learners. In the present case, the learning scenario was implemented with the module "pscenario" [hr3]. This module, easy to integrate in our technological platform, offers the advantage of being easily instrumentable. During the experimentation, all beginnings and endings of activities of the scenario were recorded within "pscenario".

3.2.2. Observation of the server

The logs of the software used on the server constitute another source of observation. Our experimentation platform was located on an Apache server [hr2] file. This file contains the whole actions carried out on the server. The creation of the traces starting from these logs is a complex process which requires many operations (e.g. cleaning, recombining in sessions) [2][7].

3.2.3. Observation of learners' station

Let us suppose that during an exercise a learner converses with a friend using an instant messenger. This interaction is not observed on the server. However, this dialog can be a major element of explanation of the learner's activity. We thus propose to instrument learner stations in order to observe all the learners' interactions.

3.2.4. Other observations

In spite of the quantitative richness of the traces resulting from the learners' stations, some crucial interactions to model the learners' behavior might be lacking. Thus, if the observed course proceeds in a class where a teacher is present, none of the previously observed sources can give a report of an oral explanation from the teacher or an oral dialog between students. In the case of distance learning, the problem is identical: it is impossible to know that the learner is in front of his/her machine. Complementary sources of observations could consist in using video [1] or human observers during the course. We adopted the latter solution which obviously cannot be considered outside the framework of experimentation, in order to better locate the limits of the instrumentation selected.

4. Obtaining an interpretable trace

In this section, we will show how a person will be able to compose an interpretable trace by annotating the raw observations. We call this person the "trace composer". This role could be played either by a student who wishes to keep a trace of his/her course, or by a teacher who wishes to compare several traces in order to improve his/her scenario.

4.1. Raw observations

Figure 2 presents the three observed sources in parallel on the computers: learning scenario activities, events on the server and events on the learner's station. The learning scenario activities observed correspond to several observed elements on the server. For example three elements observed on the server correspond to "exercise 1 failed": (a) the recovery of the statement, (b) the recording of the file containing the proposed exercise solution, (c) then consultation of the negative validation from the teacher. There are observed elements on the server which do not match any activity envisaged in the learning scenario. For example, the learner posted a message in a forum (d).

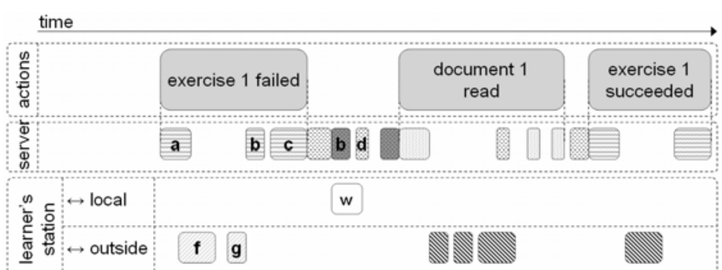


Figure 2: raw observations

The observations on the learner's station can be divided into two categories: On one hand, local interactions with software on the learner's station. For example, the learner can use Microsoft Word (w) to compose a text. On the other hand, interactions with other computers on the network. For example, communications using an instant messenger (f) and (g). Among these latter interactions, those with the server can be easily identified and removed because they already appear in the server log.

4.2. Comparison with the recommended learning scenario

In this section, we compare the activity carried out to the recommended activity in order to measure the differences between the achieved scenario and the recommended learning scenario (c.f. 4.2.1.). This comparison enables us to estimate the comprehensibility (c.f. 4.2.2.) of our observation. In order to help the trace composer, we propose a graphic representation of this estimate (c.f. 4.2.3.).

4.2.1 Comparison between achieved activity and recommended activity

There are many dimensions to measure the variations with the recommended learning scenario: for instance, quantity of documents used and produced by the learner or exchange richness with the others. Within the framework of our experimentation, it was important for the teacher that the learner ends the learning scenario in a given time. We thus chose to compare the duration of the activity carried out with the duration recommended by the designer of the activity.

4.2.2 Intuitive estimate of the clearness of the observation

We define the comprehensibility of a zone as the probability that no ignored activity took place for this duration. The comparison system enables us to represent the observation of an activity with a duration close to the one recommended by the teacher with a strong comprehensibility. On the contrary, if we observed an activity with a duration definitely higher than what was recommended, then the probability that another activity (even not related to the training) was missed is strong. We thus consider that our observation was incomplete in this shaded area. We therefore propose observations available from other sources to the trace composer to let him/her complete the trace.

4.2.3. Graphic representation of the comprehensibility: the shadow bar

The shadow bar is a graphic representation of the comprehensibility of an observation. The color of each area is determined by the estimate of comprehensibility: clear if the observed activities are in connection with their recommended durations; dark if the observed activities exceed the recommended time for their achievements; and even completely black if no observation explains the elapsed time. Figure 3 presents the shadow bar corresponding to an observation session. In this example, only observations of the activities of the learning scenario appear in the trace. (a) is the time recommended for exercise 1. (b) is the time exceeding the recommended duration. In the black zone (c), no activity was observed.

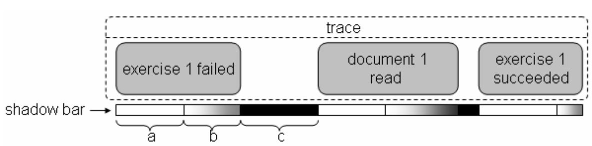


Figure 3: the shadow bar

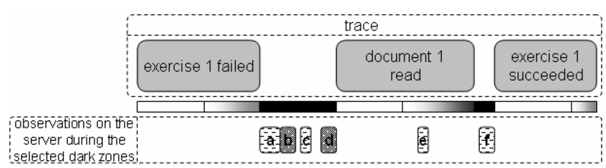


Figure 4: choosing observation on the server

4.3. Interactive construction of the interpretable trace

In this section, we show how the shadow bar helps the trace composer to choose and annotate the observations among the various sources. Figure 4 shows the initial situation presented to the trace composer. Only the observations of the activities of the learning scenario are presented. The shadow bar is black during times when no activity was identified and darkens when an activity exceeds the duration recommended by the teacher. When the trace composer selects shaded zones, the observations on the server corresponding to these zones are posted in the lower part of the chart. The trace composer can retain observed elements that s/he considers significant within the framework of the observed learning task. For instance, in figure 4, (a), (c), (e) and (f) are of comparable nature: forum consultation. The trace composer selects these four elements, annotates them as “use of forum”. The result obtained is presented in figure 5: a new observation was added to the trace that makes the shadow bar clearer in the corresponding zones. If the trace composer considers observations not significant for the learning task, s/he can indicate them as explanatory elements of the time passed without including them in the trace. For example in figure 5, if s/he indicates (b) and (d), the shadow bar on the corresponding zone is cleared up on figure 6 with no observation being added. If shaded zones persist in the bar, observations on the learner’s station can be integrated in a similar way. For example in figure 6, the dialogs of the zone (a) were initiated by an external person and are taken into account by the trace composer. On the other hand, the communications with another student of the course in (b) and (c) were annotated as “dialog” in the figure 7.

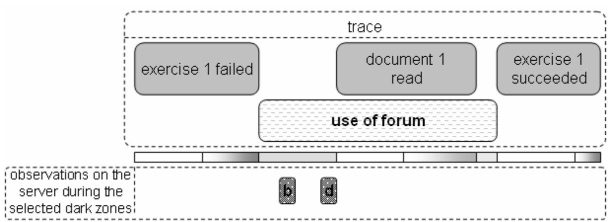


Figure 5: adding observations from the server

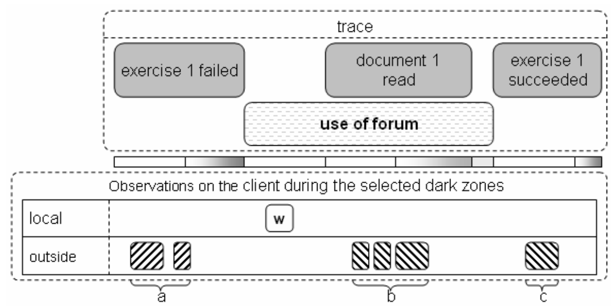


Figure 6: choosing observation on the learner’s station

4.4. Result

Figure 7 presents the trace obtained at the end of the explanation process of the shadow bar. This trace contains only observations of activities related to the observed learning task, according to the trace composer. The absence of shaded zones at the end guarantees the low probability of having missed the observation of a learning activity.

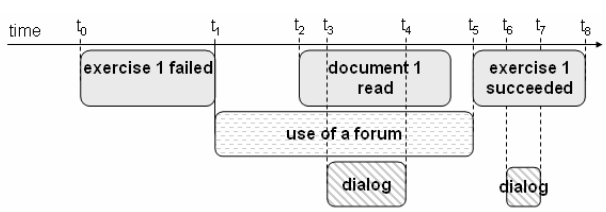


Figure 7: interpretable trace

5. Discussion on preliminary results

The amount of data recovered during our experimentation is significant and requires the development of specific tools for treatment, in particular to automate the composition of elements resulting from different sources. The complete analysis of the results of this experimentation is a work in progress. However, we manually analyzed part of the data and we present these first results here.

5.1. Contribution to the comprehension of the activity

Our first objective was to provide a help to the comprehension in the activity for the teacher. The assistance suggested consists of a graphic formalism presented in subsection 4.2.3. Moreover the degree of comprehension is indicated thanks to the concept of clearness. Starting from this formalism, we can consider a dashboard for the teacher representing the state of different learners. This tool would be especially useful within the framework of distant learning and would let the teacher supply each learner with an individualized help according to encountered difficulties.

5.2. Analysis of activities leading to a success

The learning scenario comprised two exercises and only 7 out of the 36 students finished the scenario before the time limit. We thus isolated the observations of these 7 students in order to compare them with the others. The scenarios carried out by these 7 students are rather similar. If we consider a sufficiently large unit of time (10 minutes) to take into account the various speeds of execution between these students, then the trace of the scenario activities presented on the right of figure 8 summarizes their activity.

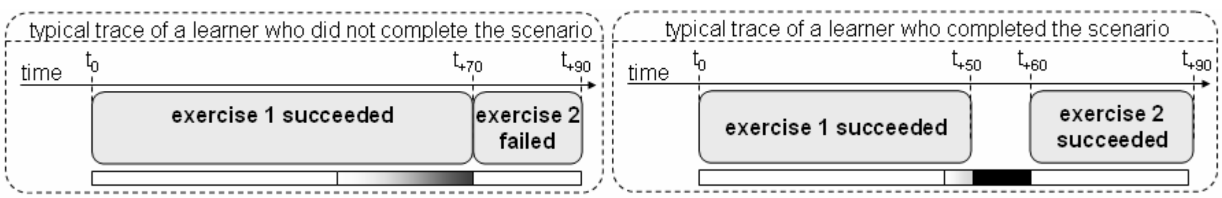


Figure 8: standard traces of students according to their final results

As an early conclusion, we supposed that the 7 students that had finished were faster than the others. A more detailed investigation, illustrated in figure 9, showed us that the true common point between these students was an additional activity, not defined in the learning scenario: all these students requested the validation of exercise 1 by the teacher present in their room.

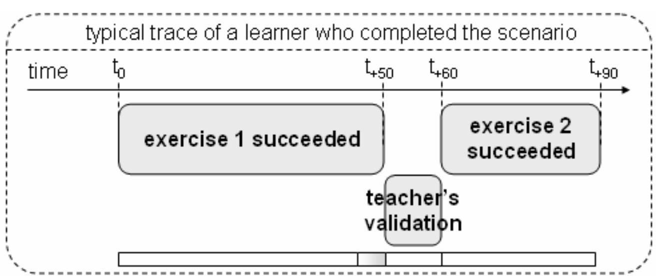


Figure 9: standard trace of a student having finished the scenario

Among the students who had not finished, some read course documents again; others restarted exercises of the preceding course or compared their results with other students before starting exercise 2. Their traces are too different to be represented in a standard trace. Consequently, it is not possible to compare them with the trace presented in figure 9.

5.3. Improvement of the scenario

In the previous section, we noted the emergence of a new task carried out by all learners that completed successfully the two exercises. It is then legitimate to revise the learning scenario and to wonder whether the insertion of this new task in the scenario is not necessary. Obviously, the system will not make such a decision. It is in general the designer of the scenario who will deliver his/her opinion on the utility of this modification. We think that this public needed to be reassured on the quality of their work before starting the second exercise. If we had to redo this course with this learning scenario, we would add an activity of validation by the teacher in the scenario between the two exercises.

6. Conclusion

The work presented in this paper proposes a step of trace annotations to allow a later interpretation of it. This step was illustrated throughout the paper by an experimentation carried out with two groups of students. The discussion held in paragraph 5 gives an idea of the profits which we can draw from such an approach.

The prospects at the end of this experiment are numerous. We used the factor of comprehensibility based on the time spent to carry out certain activities. This is only one of the numerous examples of possible metrics. We are going to consider other metrics such as the degree of collaboration.

In addition, the external source of observation, although presented, has not yet been explored. In this experiment, it was simply used as validation for a certain number of assumptions. It would be necessary to consider for the trace composer a means to use this source of information as well as the others when s/he wishes to eliminate shaded areas.

In a more general way, increasing the richness of the observation level and finely clarifying the sequencing of a course are paramount stages. We already proposed some visualization tools to represent the traces in [4]. All this work will make it possible to improve the quality of the learning scenarios implemented and, in a second phase, to allow us to evaluate their level of maturity.

References

[1] J.M. Adam, B. Meillon, M. Dubois, A. Tcherkassof, "Methodology and tools based on video observation and eye tracking to analyze facial expressions recognition" in 4th Int. Conference on Methods and Techniques in Behavioral Research, 2002.

[2] R. Cooley, B. Mobasher, J. Srivastava. "Data Preparation for Mining World Wide Web Browsing Patterns" in Knowledge and Information Systems (KAIS), vol. 1, no. 1, pp. 5-32, 1999.

[3] J.M. Heraud, L. France, A. Mille, "Pixed: An ITS that guides students with the help of learners' interaction log" in Workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at the Int. Conference on Intelligent Tutoring Systems, pp. 57-64, 2004.

[4] L. France, J.M. Heraud, J.C. Marty, T. Carron, "Help through visualization to compare learners' activities to recommended learning scenarios" in ICALT2005 : 5th IEEE International Conference on Advanced Learning Technologies, 2005.

[5] J.C. Marty, J.M. Heraud, T. Carron, L. France, "A quality approach for collaborative learning scenarios" in Learning Technology newsletter of IEEE Computer Society, Volume 6, Issue 4, pp.46-48, 2004.

[6] M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber, "Capability Maturity Model - Version 1.1." IEEE Software, pp. 18-27, 1993.

[7] O.R. Zaïane, "Web Usage Mining for a Better Web-Based Learning Environment" in Conference on Advanced Technology for Education, pp. 60-64, 2001.

[hr1] IMS Global Learning Consortium: Learning Design Specification, http://www.imsglobal.org/learningdesign/.

[hr2] The Apache Software Foundation, "Apache HTTP Server Project", http://httpd.apache.org/

[hr3] pscenario, TECFA's Pedagogical Scenario Tool for PostNuke, http://tecfaseed.unige.ch/door/

Usage Analysis Driven by Models in a Pedagogical Context

Sébastien Iksal, Christophe Choquet
University of Maine – LIUM – CNRS FRE 2730
IUT de Laval, 52 rue des Docteurs Calmette et Guérin, 53020 Laval, France
firstname.lastname@univ-lemans.fr

Abstract. In the context of distance learning and teaching, the re-engineering process needs a feedback on the learners' usage of the learning system. The feedback is given by interviews, questionnaires, but in the majority of systems, it is given by log files. We consider that it is important to interpret tracks in order to compare the designer's intentions with the learners' activities during a session. In this paper, we present the usage tracking language – UTL. This language was designed to be generic and we present an instantiation with IMS-Learning Design, the representation model we chose for our three years of experiments. At the end of the paper, we develop an instance of a scenario for tracks analysis and we present the architecture of services around UTL.

Introduction

Nowadays, numerous interactive systems are available on the Web. Most of these systems need some kind of feedback on the usage in order to improve them. In the specific context of distance learning and teaching, the desynchronization between teachers' two major roles – instructional designer and tutor – brings about a lack of uses feedback. The software development process should explicitly integrate a usage analysis phase, which can provide designers with significant information on their systems' uses for a reengineering purpose [1]. Semantic Web aims at facilitating data management on the Web. It brings languages, standards and corresponding tools that make the sharing and building of automatic and semi-automatic programs easier [2]. Automatic usage analysis is often made by mathematicians or computer engineers. In order to facilitate the appropriation, the comprehension and the interpretation of results by instructional designers, who are the main actors of an e-learning system development process, we think they should be fully integrated in this analysis phase.

The research contribution we present in this paper is fully in line with our approach to the reengineering of e-learning systems, where we particularly stress the need for a formal description of the design view, in terms of scenarios and learning resources, to help the analysis of observed uses (i.e., descriptive scenarios) and to compare them with the designer's intention (i.e., predictive scenario) [3]. When designers use a formal language such as Learning Design [4] proposed by IMS Global Learning Consortium [5] to explicit their intention regarding the learners activities during a session, a set of observation needs are implicitly defined. Thus, one of the student data analysis difficulties resides in the correlation between these needs and the tracking means provided by the educational system, all the more in the case where courses designed will be broadcasted by a Learning Management System (LMS), which usually provides predefined tracking abilities. We propose in this paper a meta-language to describe the track semantics recorded by a LMS and to link them to observation needs defined in the predictive scenario. This meta-language could be instantiated both in the formal language used to describe the pedagogical scenario and in the track file format implemented by the LMS.

The next section of this paper presents this meta-language, called Usage Tracking Language (UTL). In a third part, we provide a use case which highlights the possibilities of this language. Finally, we present an open architecture for usage analysis based on the exploitation of UTL. All the examples cited in this article are taken from a number of tests we have made with our students over the last three years. The first one is composed of six activities designed for teaching network services programming skills. We used the "Free Style Learning" system [6], based on “Open-USS” LMS [7], in which students can navigate as they chose to between all the activities. Our designers have defined a predictive scenario and, each year, we have compared this scenario with descriptive ones by hand, for a reengineering purpose. The second experiment started last year. It aims at students learning the main notions of project management by a collaborative work around a real software development project.

1. A Meta-Language For Usage Analysis

All the systems which need to analyze the user behavior work with data-mining techniques [8] or by hand. These techniques are often used to build user models or to adapt the content or the layout to the user [9]. They are based on statistical or mathematical analyses [10]. In our case, we are interested in analyzing the user behavior in order to improve the pedagogical scenario and the learning materials. Our proposal consists of an analysis driven by models. That is to say, using a model to describe the learning scenario, and using the same model as a guideline to analyze the user behavior inside the Learning Management System. We consider that the result of an analysis will be better used if it has a meaning for the designer of the system and/or the content.

As already mentioned, our activity focuses on the re-engineering driven by models. We consider that each designer has his own representation model for the learning activity. In order to facilitate the comprehension of the analysis, the tools must take into account the designer model and provide the results using the same model. In our experiments, we focus on a standard model of representation: IMS Learning Design. But, in the future we want to refer to a meta-model in which all designer’s models may be described. XML-Schema is an interesting candidate because a number of models are based on this meta-language. We currently have a project on the collaborative design of a model of representation for learning scenario. In this project, we plan to develop a collaborative editor based on XML-Schema. So, one of the goals of this project is to design tools that may work on XML-Schema in order to interpret the designer’s models. Since the beginning of our experiments, we have used IMS-LD to describe learning scenario, and IMS-LD has its own description in XML-Schema.

1.1 Usage Tracking Language : UTL

Even if we are able to process the designer’s model, it is not sufficient for the automation of the tracks analysis. We will also propose a specific language for describing the track semantics according to the designer’s model. This language – called UTL, for Usage Tracking Language – is a meta-language which needs to be instantiated according to (i) the designer’s model and, (ii) the specific format of the logs. Because they have not been designed for this, existing representation models don't include tracking facilities, so UTL is proposed to link tracks and designers’ models through the semantic data. UTL is implemented in RDFS syntax [11]. Figure 1 describes the UTL part concerning the representation. This part of UTL is necessary to interpret some elements of the representation model which are observed. This section has been designed to be as generic as possible, because we want it to be compatible with the majority of designer's models. The term *CONCEPT* refers to all concepts that are defined by the designer, for instance in LD, we can have *Activity, learning object, role, … TRACEABLE CONCEPTS* are concepts from which it is possible to track something, for instance, a video player is a traceable concept from which we can track the start/stop events.

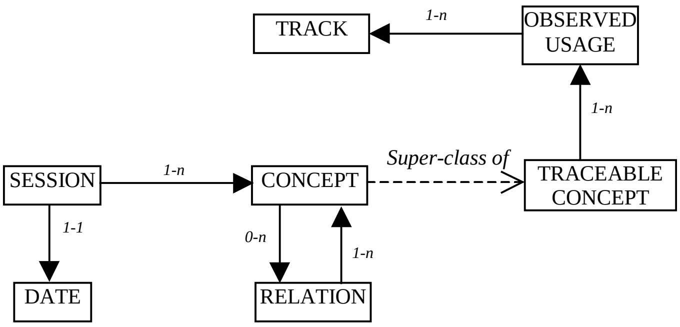


Figure 1: The meta-language UTL – Representation part

In order to work on the track itself, we need to identify it or a part of it. Thus, we have defined another section in UTL : the track representation presented in the Figure 2. The model is also generic, and we propose an implementation that should work with the majority of log formats, as the location of data may be described with a character position and/or with tokens. This section of UTL is useful for retrieving specific tracks, extracting values and bringing sense to each of them. The *KEYWORD* is used to retrieve the track, it is a word which is always present in the track. The *VALUE* depends on the learner, it may be the time spent to read, the name of the page read or the score of the evaluation exercise.

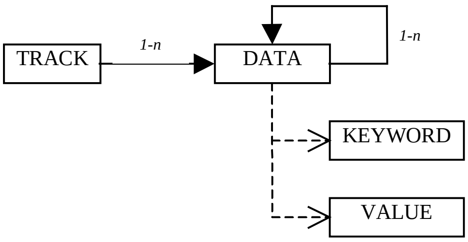


Figure 2: The meta-language UTL – Tracks part

The specific attributes for the specification of the data locations are the following :

- Type : Is used to type the data – to associate semantics.
- Begin : Gives the first character position of the data.
- End : Gives the last character position of the data.
- Delimiter : Sets the delimiter used to break down the track into tokens.
- Position : Gives the position of the token.

The data locations are used to specify the position inside the track of the keyword or the value.If we consider a prescribed scenario in IMS-LD and tracks generated by FSL, Figure 3 is an example of the instantiation of UTL. In this case, a session is identified by the student identifier, because for one session we have a set of log files which corresponds to the work of a single student. First, we describe some data that can be extracted according to these two models. In the following example, we describe a track which represent the end of a video player done by the learner.

```
<? xml version="1.0" encoding="iso-8859-1" ?>
<TRACKING
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="fslTrack.xsd"
   session="SRC1E06" date="18/12/2002">
  <ACTIVITY name="View Objectives">
    <USING>
      <LEARNINGOBJECT name="Introduction"
```

```
                  type="VideoPlayer">
         <OBSERVEDUSAGE name="functioning">
           <TRACK type="stop player">
             <VALUE type="date" begin="1" end="25"/>
             <KEYWORD type="component" begin="38"
                      end="52">FreeVideoPlayer</DATA>
             <KEYWORD type="command" begin="55"
                      end="58">stop</DATA>
             <VALUE type="duration" begin="74"
                    end="end"/>
           </TRACK>
           ...
```

Figure 3: Example of track description

This description has been used to filter the log file and to extract the following track :

```
[18/12/2002:09:45:29 +0043]          [FreeVideoPlayer] stop() currentTime=182.0s
```

We have also obtain the following data :

```
Date of the track : 18/12/2002:09:45:29  Duration of the video : 182.0s
```

In this example, we worked with a single student. In other experiments, we may have to track the activity of a group (especially in collaborative work). UTL is able to describe tracks if we have a single log file for all members – server log file –, and also if we collect a set of log files, one per member – client log file. We just have to define in the designer model the concept of “group” and “member of group”.

1.2 Instantiation of UTL in IMS-LD

In our experiments, we have used IMS-LD as a representation model for the designer. In order to manage tracks according to this language, we have instantiated a part of IMS-LD in UTL (See Figure 4).

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
         xmlns:utl="utl.rdfs">
  <rdfs:Class rdf:ID='Activity'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='Role'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='Environment'>
    <rdfs:subClassOf rdf:resource='&utl;Concept' />
  </rdfs:Class>
  <rdfs:Class rdf:ID='LearningObject'>
    <rdfs:subClassOf rdf:resource='&utl;TraceableConcept' />
    <rdfs:subClassOf rdf:resource='#Environment' />
  </rdfs:Class>
  ...
  <rdf:Property rdf:ID='performs'>
    <rdfs:subPropertyOf rdf:resource='&utl;Relation' />
    <rdfs:domain rdf:resource='#Role'/>
    <rdfs:range  rdf:resource='#Activity'/>
  </rdf:Property>
  <rdf:Property rdf:ID='using'>
    <rdfs:subPropertyOf rdf:resource='&utl;Relation' />
    <rdfs:domain rdf:resource='#Activity'/>
    <rdfs:range  rdf:resource='#Environment'/>
  </rdf:Property>
</rdf:RDF>
```

Figure 4: UTL file for IMS-LD

2. Scenario of Tracks analysis

Our first need on usage analysis is about track analysis. We have three years of logs on two different experiments. For each of these case studies, we have a prescribed scenario described in IMS Learning Design. We use our Usage tracking language to bring semantics to each track. The first step consists in the interpretation of tracks according to the designer model and the corresponding track semantic description. Next, the observed usage of the learning system are available for the analysis.

2.1 Tracks Interpretation

At the beginning, automatic track analysis needs an automatic interpretation of these tracks. UTL is designed to add semantics to the content of the log files. We use it to filter the content of the log files, that is to say, to keep only tracks that are considered relevant by the designer. A track is relevant if a description is given inside the UTL file. The second use of UTL consists of associating a specific type to each track and in extracting values that are representative of the learner's activity.

The result of this stage is a data structure which contains the interpretable tracks and which is shareable between all services of our architecture. The data structure is available also for each researcher who wishes to propose new services.

2.2 Usage Analysis

There are various ways to use the observed usage-interpreted tracks. Our first service retrieves patterns to find the resource usage, or to compare a learner scenario with the predictive scenario. Next, by means of the semantic description of tracks, it is possible to define services in a declarative way.

Examples of analysis results are the following: rate of use of a resource, performance of a student, emergence of a role (leader, …), extraction of an observed learning scenario, detection of a sequence of resource uses which have not been prescribed…

To present the usage analysis, we will focus on three cases : A statistical data, a result which has to be re-transcribed in the designer's model, and an intelligent information detection.

A statistical data. These data are, for instance, the rate of use of a resource, the average mark concerning the evaluation exercise, or the time spent on a particular activity (the shortest, the average, the longest). We have to filter the tracks according to their semantics and to make a small calculation on them. As an example, for the data (a) a first solution is to count students for whom we find at least one track about the use of the resource. In our experiments, we have observed that sometimes we have tracks about the use of the resource, but the student has spent less than 15s on the resource because he clicked everywhere during an exploration period. The solution adopted was to detect the duration of each period of use, and to count students who have spent a minimum of 15s on at least one period.

Retranscription in the designer's model. One of the main goals of the re-engineering driven by models is to use the same representation model for the description of the predictive scenario by the designer as for the observed scenario build with tracks generated by the learning system. In our first experiments, we worked with IMS-LD as a representation model. The interest in the use of a common model is the possible comparison between the different scenarios, that leads us to identify non-predicted usages of resources or incoherences in the sequence of activities. In one of our experiments (the one based on FSL), we observed that some students have used the evaluation exercise as a quiz at the beginning of the experiment, they just have navigated inside the list of questions in order to self-evaluate their knowledge (before the first activity of the learning session). That observation leads us to propose two facets on our exercise, one for evaluation and another for a quiz. We consider two kinds of retranscription of the observed scenario: the one generated from a single student tracks, and a stereotypical scenario that represent a combination of all student scenarios.

(a) Retranscription of one student observed scenario.
First, we have to read the representation model in order to identify the core concept, such as the activity for IMS-LD. Next, we filter tracks in order to represent this concept and all its components. The last step consists in organizing all instances of the core concept in a sequence which corresponds to the observed scenario.

(b) Retranscription of a stereotypical observed scenario.
A stereotypical observed scenario corresponds to the combination of all student scenarios. To build this scenario, we must have all the students' observed scenarios. Next, we compare the sequence of core elements (for instance, activities), and we compare in depth each element. We observe the percentages about the use or the position in the sequence of each element. A stereotypical scenario is a graph where each relation is qualified with the percentage of students which have chosen the corresponding direction.

3. An Open Architecture for Usage Analysis

We have observed that many researchers are interested in a collection of tools which may assist them in the "semantic" analysis of the learner's learning session. We propose an architecture that may create a kind of practice community around the usage analysis. Our approach is close to that of Web Services, that is to say each server has to declare itself to the system and it provides a set of services. Servers may be deployed everywhere. But in our case, a service is a collection of methods around a specific concept or domain. For instance a service may be the management of log files, and we may have one method per log format. The other major difference consists in the use of methods, all methods are available from all the servers, because with our architecture based on RMI, we don't have to know where the method is executed, we just need to know its head (name, parameter, output). We execute all methods from one server. In our approach, we share also a common data structure between each service to facilitate the sharing of complex data.

3.1 The Architecture of Services

The most important feature of the architecture is that each researcher in usage analysis must be able to add new services in order to share them. That is why we have chosen to propose an architecture based on the Java technology and the RMI functionality. The architecture is presented in the figure 5. It is a cluster of servers around a special one which is called the *Router*. A server provides various services and registers itself at the router. A service is a set of methods that may be executed in order to request or modify something. For instance, we can have a service for the importation of log files with one method per log format, or a service about the analysis of chat discussion. This architecture is open and distributed; that is to say, it is possible to plug new servers from everywhere in the cluster. The community of researchers may use this architecture for analyzing users' behavior, but also may propose new methods in order to have feedback on the use of these methods on various data repositories. The user just has to connect to one server and to ask for a service, he doesn't have to know which server is concerned nor where the process is run. He can simply use the services available in the architecture.

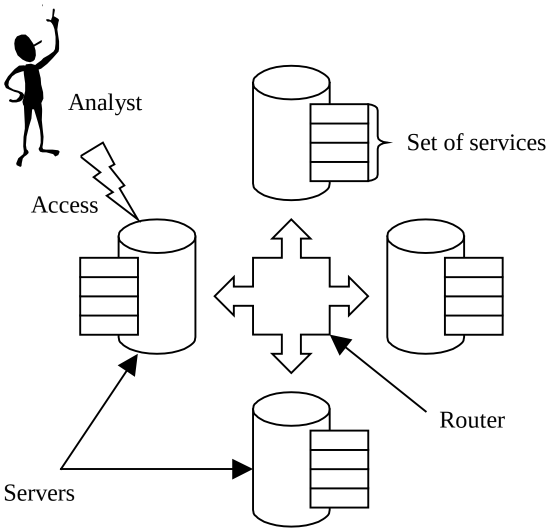


Figure 5: The architecture

3.2 Integration of Other Services

The distributed aspect of our architecture enables us to have numerous usage analysis services. We consider a data structure specific for each kind of data – tracks, chat, interviews, etc. For the moment, we are working essentially on track analysis. We choose to have some data structures instead of a database, in order to keep researchers free to use their own data management systems (databases, knowledge based systems, …) for their services.

4. Conclusion and perspectives

The meta-language presented in this paper is well suited for defining what the system has to track, based on the predictive scenario designed for a learning activity. For each traceable concept present in his scenario, the designer could define what to track (e.g. the tracking means) and why it should tracked (e.g. the semantics of the track chunk). Because of its meta-level, UTL could be also used, after usage analysis, to define and highlight semantic links between predictive and descriptive scenarios, via the association between traceable concepts and observed uses.

Works such as [12] have shown that teachers and trainers – who are the main potential designers of educational systems – have some difficulties in instructional design, especially regarding the explicitation and the technical reification of their pedagogical intentions. We are defining rules which can be inferred on the meta-model (e.g. the XML schema) of the instructional language used by a designer (for instance, Learning Design) in order to identify opportunities and observation possibilities. They reason on the structure of the instructional language (data-type, relations, etc.) and provide to the designer information on the needs of observation. These needs are relative to the concepts of the language and thus, define the traceable concepts. Using these rules with UTL could be a way to provide designers with a semi-automatic tool for decision helping purposes. Our approach of student data capture is focused on automatic techniques driven by designer prescriptions. UTL is presently without the spectrum of both existing non-automatic techniques, such as interviews for instance, and data-mining or machine learning ones. We think all these techniques, including ours, are complementary. One of our research objectives is to enlarge the spectrum and the abilities of UTL, in order to take into account results established with these other techniques. Non-automatic data capture methods are usually based on interviews and questionnaires deployed during and after the session. Questions asked to students and / or tutors are defined regarding

(i) the learning objectives and the activities proposed (e.g. the designer's intention) and (ii) the characteristics of the session (e.g. the social and technical context). All of these are known (or assumed) when the designer defines the predictive scenario. Concerning this aspect, we have started a study with researchers specialized in usage analysis (Communication Science background) of which the objective is to define when, why and how a designer has to explicit the requirements to these techniques.

References

[1] Corbière A., Choquet C., Re-engineering method for multimedia system in education, In: IEEE Sixth International Symposium on Multimedia Software Engineering (MSE), 2004, Miami (USA), p. 80-87.

[2] Berners-lee, T.. Weaving the Web. San Francisco: Harper, 1999.

[3] Lejeune A., Pernin J-P., A taxonomy for scenario-based engineering, Cognition and Exploratory Learning in Digital Age (CELDA 2004) Proceedings, p.249-256, Lisboa, Portugal, dec. 2004

[4] Koper R., Olivier B., Anderson T. eds., « IMS Learning Design Information Model », IMS Global Learning Consortium, Inc., version 1.0, 20/01/2003.

[5] http://www.imsglobal.org

[6] Brocke, J. v.. Freestyle Learning - Concept, Platforms, and Applications for Individual Learning Scenarios. 46th International Scientific Colloquium, Ilmenau Technical University, 2001.

[7] Grob, H. L., F. Bensberg, et al.. Developing, Deploying, Using and Evaluating an Open Source Learning Management System. Journal of Computing and Information Technology 12 no 2: 127-134, 2004.

[8] Mostow, J.. Some useful design tactics for mining ITS data. Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, Maceió, Alagoas, Brazil, 2004.

[9] Zheng C., Fan L., Huan L., Yin L., Wei-Ying M., Liu W.. User Intention Modeling in Web Applications Using Data Mining. World Wide Web: Internet and Web Information Systems, 5, 181–191, Kluwer Academic Publishers, 2002.

[10] Bazsaliscza M., Naim P. . Data Mining pour le Web, Editions Eyrolles, Paris, 2001.

[11] Brickley D., Guha R.V.. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004

[12] Seel N., Dijkstra S.. General Introduction. In Dijkstra, S., Seel, N., Schott, F. & Tennyson, R.D. (Eds.). Instructional Design : International Perspectives. (vol. 2). Hillsdale, NJ, Lawrence Erlbaum Associates, p. 1-13, 1997.

Usage Analysis in Tutors for C++ Programming

Amruth KUMAR
Ramapo College of New Jersey,
Mahwah, NJ 07430, USA

Abstract. We have developed and deployed several web-based tutors for C++ programming concepts. We have been remotely collecting usage data from these tutors, and analyzing them for the benefit of the students, the teachers and ourselves. In this paper, we will describe the tutors, the types of data collected, and the types of analysis we perform on the data for the benefit of the various constituencies: students, instructors and the designers of problems. Since we use learning objectives ontology to model the domain, student, and adaptive problem generation in our tutors, our analysis of usage is also in terms of learning objectives. We will illustrate with examples that usage analysis in terms of learning objectives yields results readily usable by students, instructors and the designers of problems.

Keywords: Programming tutor, Evaluation, Usage Analysis

1. Introduction

We have been developing tutors to help students learn programming language concepts by solving problems. To date, we have developed tutors on arithmetic and relational expression evaluation, selection statements, counter and logic-controlled loops, pointers in C++, parameter passing mechanisms, scope concepts and their implementation, and C++ classes. The tutors present programming problems to the learner, grade the learner's answer, explain the correct answer, log the student's actions, and determine whether the student has learned the material. Our tutors address analysis in Bloom's taxonomy [2]:

- Given a program, students identify syntax/semantic/run-time errors in it by selecting (a) the line of code, (b) the object and (c) the nature of the error, *in that order* (Please see Figure 1);
- Given a program, students predict the output of the program by entering the output generated by each line of code individually;
- Given an expression, students evaluate it by indicating the operator that is evaluated in each step and the intermediate result that is generated during the evaluation (Please see Figure 2).

Our tutors are designed as supplements to traditional programming projects, as recommended by the whole language approach for reading instruction [6].

In this paper, we will briefly describe the domain model, student model, and problem generation as relevant to our discussion of usage analysis. Thereafter, we will discuss the types of data collected by our tutors and the types of analyses we perform on the data.

2. The Domain and Student Model in terms of Learning Objectives

We identify a set of learning objectives for each programming topic. Learning objectives for a topic are concepts that must be understood in order to learn the topic. Preferably, these concepts are at a fine level of granularity so that problems can be designed to teach or assess them individually. For instance, the learning objectives for arithmetic expressions are:

- Correct evaluation, precedence, associativity and coercion for each of addition, subtraction and multiplication operators;
- Correct evaluation of integer and real division, precedence, associativity and coercion of division operator and divide by zero error;
- Correct evaluation, precedence and associativity of the remainder operator, divide by zero error and the inapplicability of remainder operator to real operands.

The above list is typical in that we identify over 20 learning objectives per topic. Typical semantic and run-time errors are also part of the learning objectives.

The domain model we use for our programming tutors is the concept map of the programming domain, enhanced with learning objectives. The concept map is a taxonomic map of the domain, with domain topics as nodes, and is-a and part-of relationships as arcs. The learning objectives for a topic serve as the children of the node for that topic. The domain model is a hierarchical tree, with domain topics as intermediate nodes and learning objectives as leaf nodes. With each learning objective, we associate measures (the minimum number of problems the student must attempt, and the minimum percentage of problems the student must solve correctly) that determine whether the learner has "satisfied" the learning objective.

We use an overlay of the above domain model as our cognitive student model. With each learning objective in the student model, we associate five terms that record the student's progress - the number of problems generated (G), attempted (A), correctly solved (C), incorrectly solved (W) and missed (M - e.g., the bugs in a program that the student missed identifying). Maintaining student progress in this raw form enables us to be flexible about how we interpret it. We use a pre-test to initialize the student model in our tutors, as has been proposed by other researchers (e.g., [1, 3]).

3. Problem Templates and Problem Generation

Limited problem set has been recognized as a potential drawback of encoding a finite number of problems into a tutor [7]. In our tutors, we generate problems as instances of parameterized templates, a scheme similar to that found in [5]. Every instance of a template is a new problem and no two problems are identical. Each problem template is indexed by learning objectives for which it may be used. E.g., the following is a template on arithmetic expressions:

Template No. 120
Learning Objective: /.Real.Correct
If Correct: /.Coercion
Template: 24.0 / <R1#integer;2<=R1<=6;#>
Type: expression

The learning objective for which the above template may be used is the correct evaluation of real division. The template contains a meta-variable R1, which is instantiated during problem generation to an integer value between 2 and 6, inclusive. For instance, any of the following problems may be generated from the above template: 24.0 / 2, 24.0 / 3, 24.0 / 4, 24.0 / 5 or 24.0 / 6. If the student answers the problem correctly, the student also gets credit for coercion of division operator. Typically, we have 20-25 such templates per learning objective in our template knowledge base.

Our adaptive tutor uses a rule-based algorithm to select the problem based on the needs of the learner. After each problem, the tutor consults the student model to identify the learning objectives that have not yet been satisfied by the student. It picks one of these learning objectives, selects the least recently used problem template for this learning objective and presents the next problem as an instance of the problem template. Therefore, the ontology of learning objectives is shared among the domain model, student model and adaptive problem generation in our tutors.

4. Tutor Stages and Data Collection

The tutors are set up to automatically administer a protocol of clinical evaluation, consisting of the following stages:

- **Pre-test** – This stage is used to assess the prior knowledge of the student. The tutors use the pre-test to initialize the student model. During the pre-test, the student is presented with a pre-determined sequence of problems. Usually, the sequence consists of two problems per learning objective. In order to maximize the information gathered about the student in limited time, the problems are ordered so that students see problems on all the learning objectives before seeing a second problem on any learning objective. The tutor does not provide any feedback during the fixed-duration pre-test.
- **Practice** – This stage is designed to help students learn from the tutor. The tutor provides detailed feedback for each problem. The tutor adapts to the student's needs in two ways:
    - It presents problems on only those learning objectives that the student did not satisfy on the pre-test.
    - For each such learning objective, it presents up to 3 problems at a time, or until the student satisfies the learning objective, whichever comes first.

  The practice session lasts for a fixed duration of time or until the student satisfies all the learning objectives, whichever comes first. Therefore, students who satisfy all the learning objectives on the pre-test are presented no problems during practice.
- **Post-test** – This stage is used to assess the effect of practicing with the tutor. The sequence of problems presented during the post-test is very similar to, if not identical to that used in the pre-test. The tutor does not provide any feedback during the fixed-duration test.

The three stages namely pre-test, practice and post-test are administered back-to-back, with no break in between. For each problem attempted by the learner during the pre-test, practice and post-test, the tutor collects the following information:

1. The unique number used to identify the template for the problem;
2. The instance of the problem generated based on the template;
3. The correct answer to the problem generated by the model-based tutor [4];
4. The answer entered by the student;
5. The grade for the answer as determined by the tutor – whether the student's answer was correct, partially correct or incorrect;
6. The total time spent on the problem by the student;
7. All the GUI events generated by the student while solving the problem, including the menu options selected by the user, the answers considered by the user, etc.

After each stage, the tutors report the collected data to a central server on the web. For analysis purposes, the reported data is separated by topic.

5. Analysis of the Collected Data

We developed a batch analyzer to analyze the data collected from each tutor. The analyzer analyzes the data for the benefit of three constituents: students, instructors and the designer of the problem templates.

Students: Our analyzer reports the fraction of problems that the student solved correctly on the pre-test and the post-test. This data may be used by the instructor to assign course credit. It also provides a per-problem summary that lists the number of steps correctly solved by the student on each problem. The number of steps is determined by the tutor:

- For debugging problems, this corresponds to the number of bugs the student correctly identifies;
- For output problems, this corresponds to the number of consecutive items of program output, beginning with the first output, that the student correctly predicts;
- For expression evaluation, this corresponds to the number of consecutive steps, beginning with the first step, that the student correctly evaluates.

The tutor uses and updates the same student model through all the three stages: pre-test, practice and post-test. Therefore, the summary of the learning objectives for the student after the post-test provides a good indication of the topics the student has mastered and the topics the student must revisit. The following learning objective summary is from an actual analysis in fall 2004 for a student using a tutor on while loops:

1. Relational expressions as condition of while loop: 1.0 / 0.0 / 0.0 / 0.0 / 0.0 /
2. The condition variable of a while loop is modified multiple times in the body of the loop: 0.5 / 0.5 / 0.0 / 0.0 / 0.0 /
3. The condition variable is updated after the action in the while loop body: 1.0 / 0.0 / 0.0 / 0.0 / 0.0 /
4. The condition variable is updated before the action in the while loop body: 0.16 / 0.66 / 0.16 / 0.0 / 0.0 /

The numbers listed after each learning objective refer to the fraction of problems solved by the student correctly, partially, incorrectly, missed and not attempted (i.e., generated – attempted). The summary indicates that the student understands the typical while loop (learning objectives 1 and 3), since the student answered 100% of the problems correctly. The student needs further practice on while loops wherein the loop condition is updated “non-traditionally”, i.e., it is updated before the action rather than after the action, or it is updated multiple times in the body of the loop.

Instructor: Two summaries are of particular interest to instructors: the average class performance on each learning objective on the pre-test and on the post-test.

The learning objective summary for the pre-test provides the instructor feedback on his/her coverage of topics in class. The following learning objective summary is from an actual analysis in spring 2005 for a class using a tutor on if/if-else statements:

1. Compound statement as the else clause of an if-else statement:

Number of students who solved problems: 18

1.0 / 0.0 / 0.0 / 0.0 / 0.0 /

2. Nested if-else statements:
Number of students who solved problems: 23
0.52 / 0.13 / 0.13 / 0.0 / 0.21 /

3. Cascading if-else statements:
Number of students who solved problems: 23
0.45 / 0.21 / 0.17 / 0.0 / 0.15 /

Once again, the numbers listed against each learning objective refer to the fraction of problems solved correctly, partially, incorrectly, missed and not attempted by the class. Note that all 18 students correctly solved all the problems on the "typical" if-else statement that uses a compound statement as the else clause (learning objective 1). On the other hand, the class would have benefited from additional coverage of nested and cascading if-else statements, since the class solved only 45-52% of the problems correctly on these learning objectives.

Similarly, the learning objective summary for the post-test provides direction on which topics should be additionally emphasized in class or in recitations. The following summary from the post-test for the same class suggests that the instructor should spend additional time on nested and cascading if-else statements. The limited duration of practice may have contributed to the marginal improvement from pre-test to post-test.

1. Compound statement as the else clause of an if-else statement:
Number of students who solved problems: 23
1.0 / 0.0 / 0.0 / 0.0 / 0.0 /

2. Nested if-else statements:
Number of students who solved problems: 25
0.55 / 0.13 / 0.12 / 0.0 / 0.19 /

3. Cascading if-else statements:
Number of students who solved problems: 25
0.57 / 0.15 / 0.12 / 0.0 / 0.14 /

Designer of the Problem Templates: Finaly, the tutor provides data that can be used by the designer of the problem templates for item-analysis. The analyzer lists the class average of the percentage correctness of student answers for each problem template, in the following format:

For template # 107:
Generated by: 24
Attempted by: 22
Solved Correctly: 11
Solved Partially: 6
Solved Incorrectly: 5
Average grade of students: 17.0/22 = 0.77
For template # 404:
Generated by: 23
Attempted by: 18
Solved Correctly: 10

Solved Partially: 5
Solved Incorrectly: 3
Average grade of students: 12.5/18 = 0.69

After correctly answering all the steps, if a student enters additional non-existent steps in the solution, the analyzer marks the answer as being partially correct, but awards credit to the student for all the correct steps. E.g., for template 107, the 6 answers that were marked as being partially correct were all awarded full credit. Hence, the raw score of 17.0 was used to calculate the average grade of students.

The designer of the problem templates can use the average grade of the students as an indicator of the level of difficulty of a template. With this information, the designer can re-design tutor-based assessment instruments (such as pre-test and post-test), in order to ensure that the instruments contain an even mix of easy and hard problems. Recall that the pre-test and post-test use the same sequence of problem templates. The designer can analyze the test-retest reliability of each problem template by comparing the performance of the students on the pre-test and post-test on the problems generated using each template. This is especially applicable to students in control group who are not expected to improve from the pre-test to the post-test.

6. Future Work

We have illustrated that analyzing usage data in terms of learning objectives yields results that are readily usable by students, instructors and the designers of problem templates. In the past, we have shared the results of our analysis with instructors. One type of analysis we plan for the future is about the thought process of the student while answering a question. We have been collecting data about the user's actions during problem-solving. For instance, the following actions were captured by the tutor on C++ pointer for a user in fall 2004:

Mouse over AnonVarModel/anonymous @ 1102617039937
Mouse over AnonVarModel/anonymous @ 1102617043796
Mouse over PtrModel/indirectPointer @ 1102617050015
Mouse over PtrModel/indirectPointer @ 1102617057031
Mouse over AnonVarModel/anonymous @ 1102617058062
Mouse over PtrModel/indirectPointer @ 1102617068468
Mouse over AnonVarModel/anonymous @ 1102617069546
Mouse over PtrModel/indirectPointer @ 1102617081937
Answer: Code OK clicked @ 1102617095034

(Mouse over PtrModel/indirectPointer refers to the student considering the pointer indirectPointer as a possible source of bug.) This information can be inspected to analyze the sequence of possible answers considered by the user before entering the final answer (in this case, Code OK). It could give valuable information about the mental model being used by the tutor. Say the correct answer is a bug on line L, on object O, and is of type T. We can represent this as a 3-tuple (L, O, T). Suppose the student considered a bug whose 3-tuple is (X, Y, Z). The semantic distance between the student's answer and the correct answer is given as the number of affirmative answers to the following three questions: 1) L ? X? 2) object-type(O) ? object-type(Y)? 3) error-type(T) ? error-type(Z)? Ideally, the semantic distance between the answers considered by the student and the correct answer should decrease monotonically to 0. A completely random sequence of semantic distances might indicate that the student is guessing. An oscillating sequence of semantic distances might indicate that the student has misconceptions that prevent the student from arriving at the correct answer.

The tutor collects sufficient data to indicate whether or not a student took the tutor seriously. For each problem, the tutor provides a "Don't Know" button. If the student clicks this button, the tutor reports that the student "did not attempt the problem." When a student exercises this option for multiple problems on the same learning objective back to back, clearly, the student is not applying himself/herself to the task. The following data reported by the while loop tutor in fall 2004 illustrates this from the practice session of a student. In it, templates 375, 376 and 377 all pertain to the same learning objective.

On problem 1, (Template 325), Solved 0/6 in 14 seconds
On problem 2, (Template 326), did not attempt problem
On problem 3, (Template 327), Solved 4/4 in 38 seconds
On problem 4, (Template 375), Solved 0/12 in 4 seconds
On problem 5, (Template 376), did not attempt problem
On problem 6, (Template 377), did not attempt problem
On problem 7, (Template 550), did not attempt problem
On problem 8,(Template 551), did not attempt problem

In the future we plan to use this and other collected data to provide affective feedback to the student. For instance, in the above example, the tutor could give the student the option to read about the topic. It could ask the student to re-examine the previous problem, and even require the student to re-attempt the problem before proceeding to the next problem.

7. Acknowledgements

Partial support for this work was provided by the National Science Foundation's Educational Innovation Program under grant CNS-0426021.

References

[1] Aimeur, E., Brassard, G., Dufort, H., and Gambs, S. CLARISSE: A Machine Learning Tool to Initialize Student Models. S. Cerri, G. Gouarderes, F. Paraguacu (eds.), Proc. of ITS 2002, Springer (2002). 718-728.

[2] Bloom, B.S. and Krathwohl, D.R.: Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain, New York, Longmans, Green (1956).

[3] Czarkowski, M. and Kay, J. Challenges of Scrutable Adaptivity. U. Hoppe, F. Verdejo and J. Kay (eds.), Artificial Intelligence in Education IOS Press, 2003, 404-406.

[4] Kumar, A.N., Model-Based Reasoning for Domain Modeling in a Web-Based Intelligent Tutoring System to Help Students Learn to Debug C++ Programs, Proc. of ITS 2002, Biarritz, France, June 5-8, 2002.

[5] Koffman, E.B. and Perry, J.M.: A Model for Generative CAI and Concept Selection. International Journal of Man Machine Studies. 8 (1976) 397-410.

[6] Mann, P., Suiter, P., & McClung, R.: A Guide for Educating Mainstream Students. Allyn and Bacon, 1992.

[7] Martin, B. and Mitrovic, A.: Tailoring Feedback by Correcting Student Answers. Proc. of ITS 2000. G. Gauthier, C. Frasson and K. VanLehn (eds.). Springer 383-392, 2000.

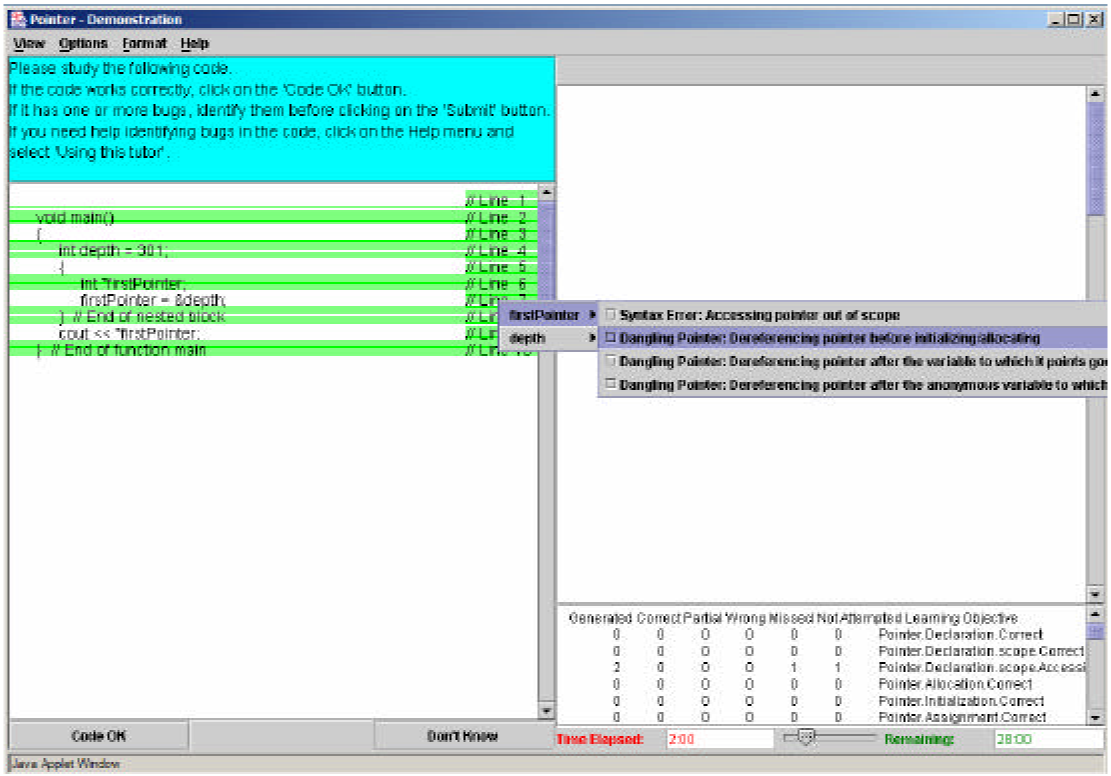


Figure 1: Tutor on C++ Pointers – Interface for selecting the bugs is shown

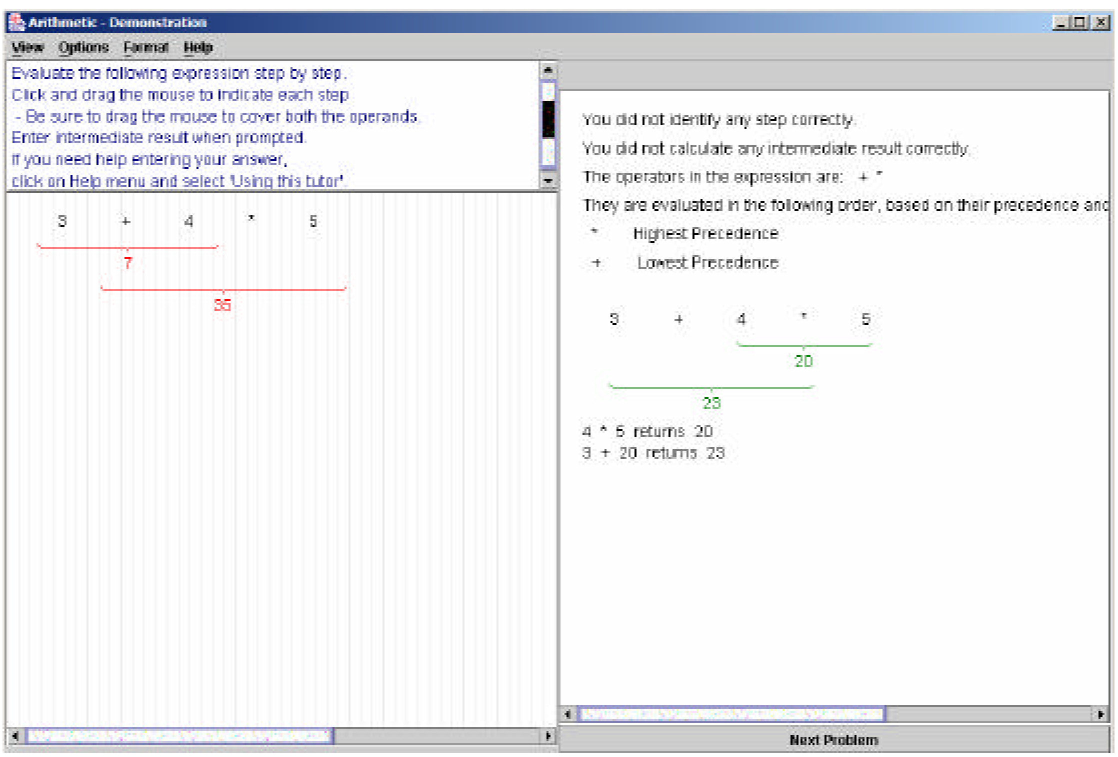


Figure 2: Tutor on Arithmetic Expression Evaluation –
User interface is shown in the left panel, feedback is shown in the right panel

Exploring Usage Analysis in Learning Systems: Gaining Insights From Visualisations

Riccardo MAZZA, Christian MILANI
Faculty of Communication Sciences, University of Lugano, Switzerland
{mazzar, milanic}@lu.unisi.ch

Abstract. This paper presents a novel approach of exploring usage analysis in learning systems by means of graphical representations. Learning systems collect large amounts of student data that can be used by instructors to become aware of what is happening in distance learning classes. Instead of being processed with techniques from user modelling, data is displayed "as it is". Techniques from Information Visualization show how useful insights can be gained from graphical representations. A system called GISMO illustrates the proposed approach. By presenting graphical representations, GISMO allows the user to visualize data from courses collected in real settings. We will show how using graphical representations of student tracking data enables instructors to identify tendencies in their classes, or to quickly discover individuals who need special attention.

1 Introduction

Today, web-based learning environments are widely used among universities around the world. They take advantage of the client-server communication on the Internet to provide instructors with a learning environment where they can distribute information to students, produce content material, prepare assignments and tests, engage in discussions, and manage distance classes [10].

In most cases, these tools are no more than sophisticated web server applications that help students and instructors in their learning activities. Even though such systems have been used in the last decade, they lack the latest functions that we can find in research prototypes. In particular, current commercial applications lack the possibility of understanding the users' needs and of adapting the content and presentation to a specific learner (adaptivity).

In a face-to-face classroom lecture, the teacher is able to perceive users' feedback, and he/she adapts the teaching according to the learners' comprehension. In a distance learning setting, because of the nature of computer-mediated communication, the tutoring lacks some specific modalities of interaction such as gestures, facial expression, direct dialogue, etc. This leads to one of the most common problems of distance learning from the tutor's perspective: the monitoring and checking of students' activities in courses delivered with distance education tools [4].

The interaction mediated by the digital media makes it difficult for the instructor to verify elements essential in didactic. There is a lack of understanding which part of the course an individual student or a group of students is working on, or the level of mastery achieved by each student for specific concepts of the course, etc.

Activities such as answering questions, monitoring and promoting discussions, monitoring the learners' progress, and testing the acquired knowledge and skills on a regular basis are essential for a good on-line tutoring practice [4][2][11]. While web-based learning environments are supposed to help tutors to accomplish these tasks, they often provide complex, confused, and useless information [6].

This research attempts to bring some advanced features to the web-based learning environments with respect to adaptivity. Particularly, we are interested in giving instructors additional functionalities to help them in their teaching activities, and adapt teaching according to individual and class activities and progress. A prerequisite for this is the instructor's awareness of what is happening to his students: Are they reading materials? Are they regularly accessing the course? Do they engage in discussions? Are there quizzes or assignments particularly problematic to the students? Are their submissions performed in due time?

Most learning environments accumulate large data logs of the students' activities, and usually provide some monitoring features to enable instructors to view some aspects of the data: e.g. the history of pages visited, the number of messages read and posted in discussions, etc. Student tracking data provided by the learning environment is a valuable source of data that can be used by the instructor not only to check students' activities, but also to improve the quality of the materials. For instance, an instructor may check which part of the course materials are most or least accessed by the student. The instructor may then perform further investigations to understand whether the students found these parts difficult to understand or not. However, student tracking data is complex and is usually organized in some form of a tabular format, which in most cases is difficult to follow and inappropriate for the instructors' needs [6]. As a result, web log data is used only by skilled and technically powerful distance learning instructors.

2 GISMO – a Graphical Interactive Student Monitoring System

In order to explore an alternative method to represent student usage data, we implemented a tool that we called GISMO. GISMO uses the students' tracking data as source data, and generates graphical representations that can be explored and manipulated by course instructors to examine social, cognitive, and behavioural aspects of distance students. It implements some of the visualizations found useful by teachers, based on our experience with the CourseVis research [7][9], within a new context, namely the Edukalibre project funded by the European Union.

GISMO uses techniques from Information Visualization [1] to build graphical representations that an instructor can manipulate, which may help him/her to gain an understanding of his/her students and become aware of what is happening in distance classes.

From a technical point of view, GISMO is an application that runs in conjunction with a web-based learning platform, and it is delivered through the Web using a Java Applet. We considered the Moodle learning platform in this work for his Free and Open Source nature. However, it can be adapted to support other learning platforms. In fact, a software API is committed to retrieve some data that is usually present in a wide range of platforms such as moodle, claroline, fle3, mimerdesk, etc. as the data points are: discussions, accesses to the course, marks students receive in quizzes, and so on. This API can be adopted to support other platforms.

Figure 1 represents the welcome page of GISMO. As you can see, there are 3 different areas in the user interface:

1. *Graph Panel*: graphs are drawn on this panel.
2. *List Panel*: contains a list of students, groups, resources, quizzes, and assignments of the monitored course. For each list the instructor can select/deselect data to visualize.
3. *Time Panel*: using this panel the instructor can reduce the selection on time and

restrict the graph to a specific range of dates.

A copy of GISMO has been installed in the local Moodle platform at the University of Lugano. This installation manages about 250 courses and more than 2.400 users. It provides a valuable source of data that can be used to analyze GISMO's graphical representations in real settings.

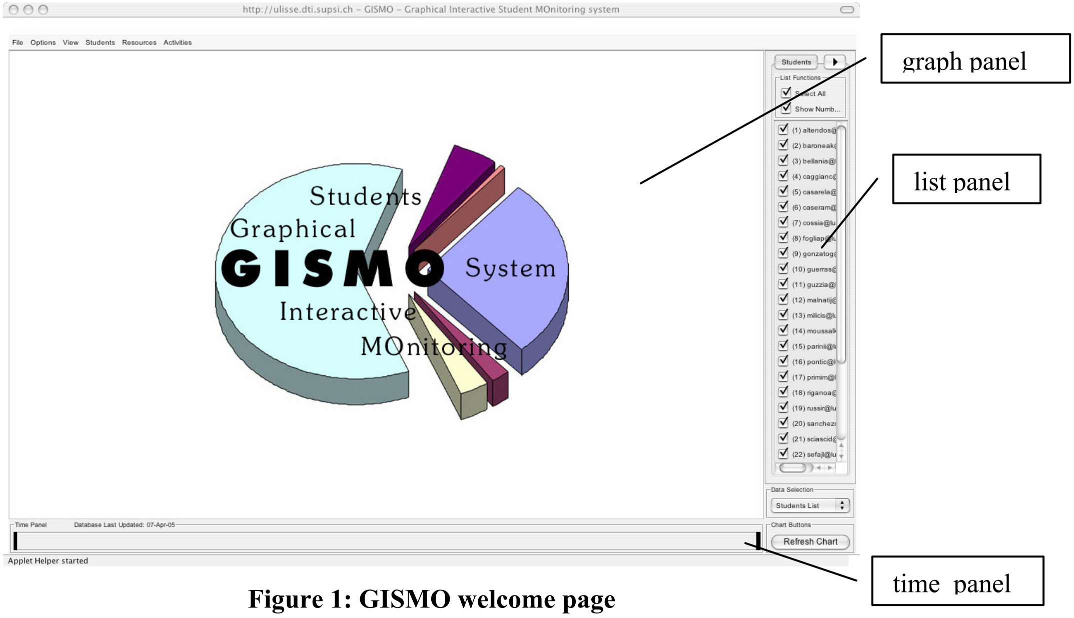


Figure 1: GISMO welcome page

In the next section we will illustrate some graphical representations of GISMO's abilities on data collected from real courses, and we will describe some insights that can be derived from representations. Representations were produced for the information regarded as interesting for instructors, that we had detected with a survey submitted to instructors involved in distance learning in a previous research [8]. That information is student attendance, access to resources, overview of discussions, and results on assignments and quizzes.

3 Students’ attendance and reading of materials

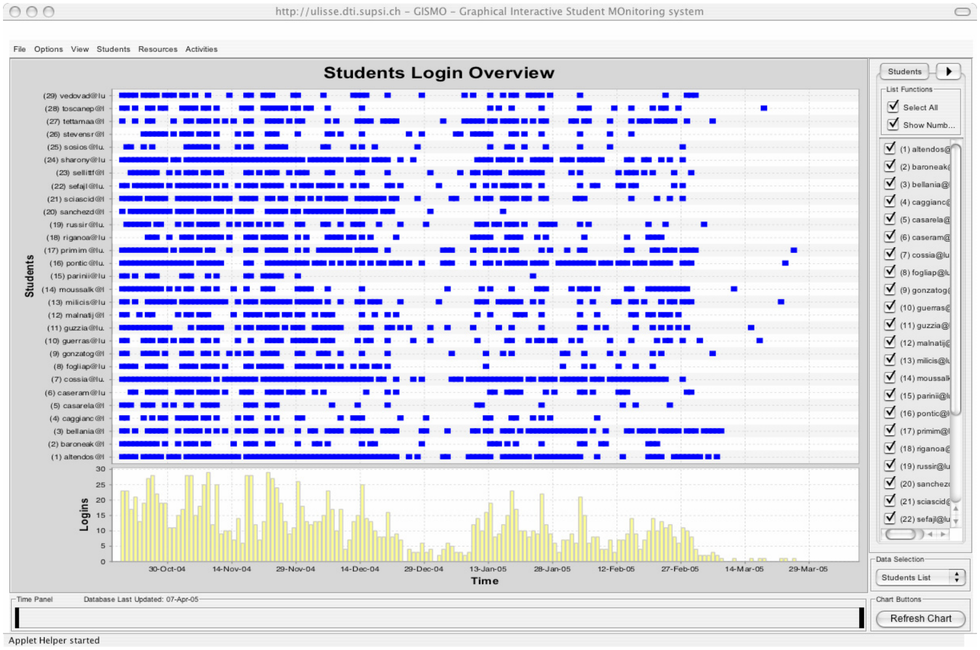


Figure 2: A graph reporting the students' accesses to the course.

Figure 2 reports a graph on the students' accesses to the course. A simple matrix formed by students’ names (on Y-axis) and dates of the course (on X-axis) is used to represent the course accesses. A corresponding mark represents at least one access to the course made by the student on the selected date. The histogram on the bottom shows the global number of hits to the course made by students on each date. With these graphs, the instructor has an overview, at a glance, of the global accesses made by students to the course with a clear identification of patterns and trends, as well as information about the attendance of a specific student of the course. For instance, we can clearly see how the attendance of the students during the first period of the course was particularly regular and uniform, with an inactive period at about the half way mark of the course (which corresponds to the Christmas period). It is interesting to notice how the accesses to the course become scattered during the second half of the course. The same image may allow the instructor to focus on about five students who were particularly persistent in accessing the course throughout (even in the second half of the course).

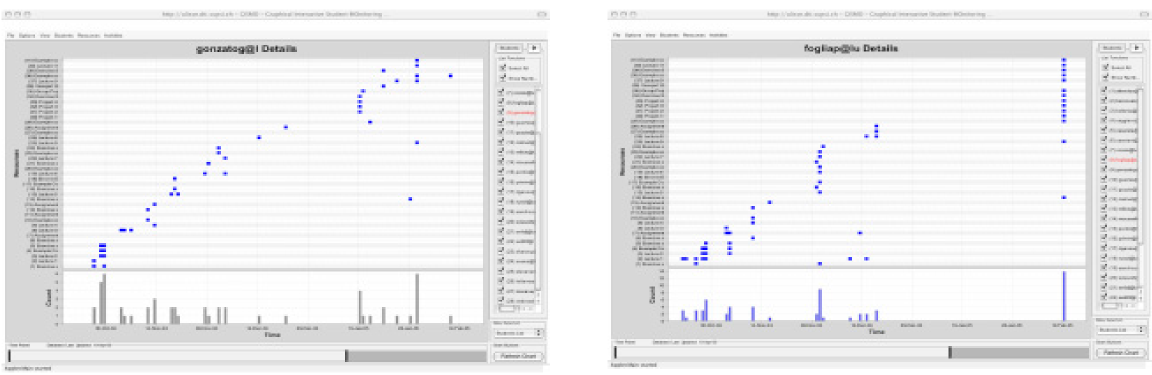


Figure 3: Two graphs reporting on the accesses performed by two different students to the course’s resources.

Figure 3 reports an overview of the accesses of a student on the course's resources. By resources we mean any type of content that can be inserted into the course, such as a text page, an assignment, an external link, a Power Point file, etc. Dates are represented on the X-axis; resources are represented on the Y-axis. Resources order on Y-axis reflects the resource sequence order inside the course. The histogram on the bottom represents the total number of accesses made by the student to all course's resources. Figure 3 shows two particular students of the course with different behaviors in accessing course resources. The graph on the left shows that this student regularly accessed materials. The graph on the right depicts a student having a different behavior: he/she concentrated his study during three periods: at the beginning, on a range of dates on the middle, and at the end (after a long time of inactivity he/she accessed several pages at once on the same day).

4 Resources

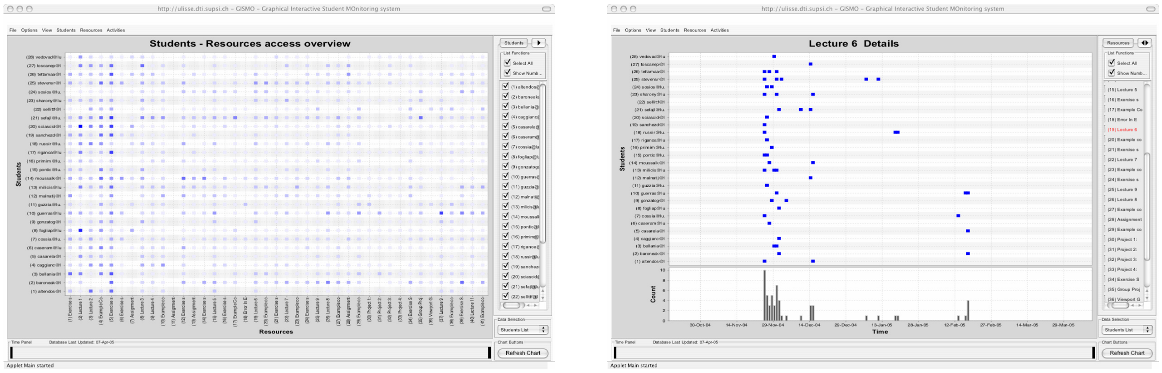


Figure 4: Two graphs reporting the overview of students' accesses to resources of the course (left), and the students' accesses to a particular resource of the course.

Instructors could also be interested in having the details on what resources were accessed by all the students and when. Figure 4 is intended to provide this information. The figure on the left reports student names on the Y-axis, and resource names on the X-axis. A mark is depicted if the student accessed this resource, and the color of the mark ranges from light-blue to dark-blue according to the number of times he/she accessed this resource. Some interesting insights can be seen. For instance, there are some resources intensely accessed by students of the course on the leftmost part of the graph (if the user puts the cursor of the mouse up one of the marks, a little tool tip appears showing the number of accesses made by the student on a particular resource). Moreover, there are some resources that had few accesses; these are easily identified by columns of the graph having few or no marks. Here the color propriety is used to provide a visual distinction between resources that had few accesses between those that had several accesses. This indication could be useful to the instructor to analyze the level of usage of the course material.

The Figure 4 on the right shows on which days the students visited a specific resource (with the graph on top) and how many global accesses they made to this resource for each day of the course (with the bar chart on the bottom). Again, student names are on the Y-axis, and dates are on the X-axis. This image can provide some insights to the instructor interested in knowing when a particular resource has been accessed during the distribution of time of the course.

5 Discussions

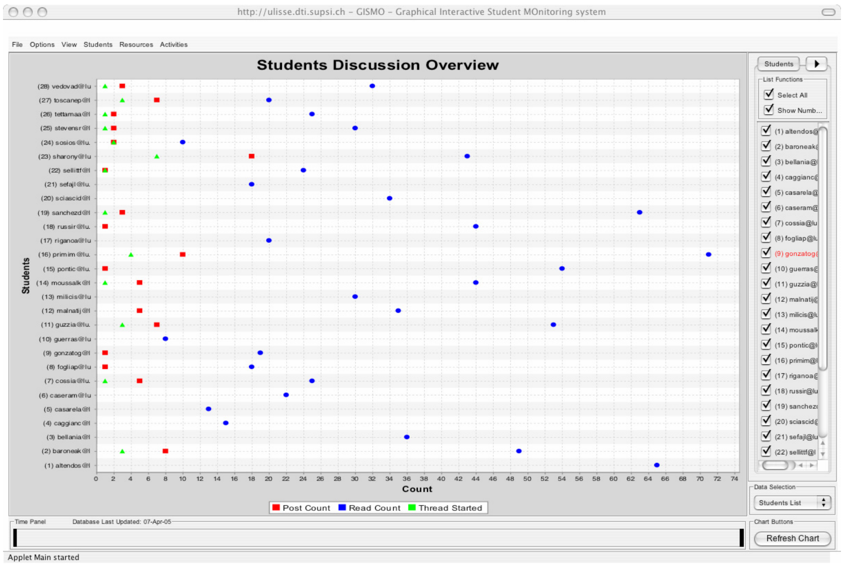


Figure 5: Graphical representation of discussions performed in a course.

Discussions are a form of social activity that several instructors consider crucial in their courses [8]. Participation in discussion boards has to be considered for a comprehensive student analysis. The discussion board is a tool that allows students to read and post messages. Each message has a sender, a date, and a topic. A set of posts on the discussions, composed of an initial post about a topic and all responses to it is called a *thread*. Discussion boards' data is mapped onto a 2-D scatterplot and the generated image is illustrated in Figure 5. In this chart, instructors have an overview of all the discussions in which students participated. For each student of the course the chart indicates the number of messages posted (with a square), number of messages read (with a circle) and finally the number of threads started by the student in the discussions (with the triangle). It can be seen in the example in the figure that the most activity represented is reading, and less activity has been done in starting new threads or replying to other messages.

6 Assignments and quizzes

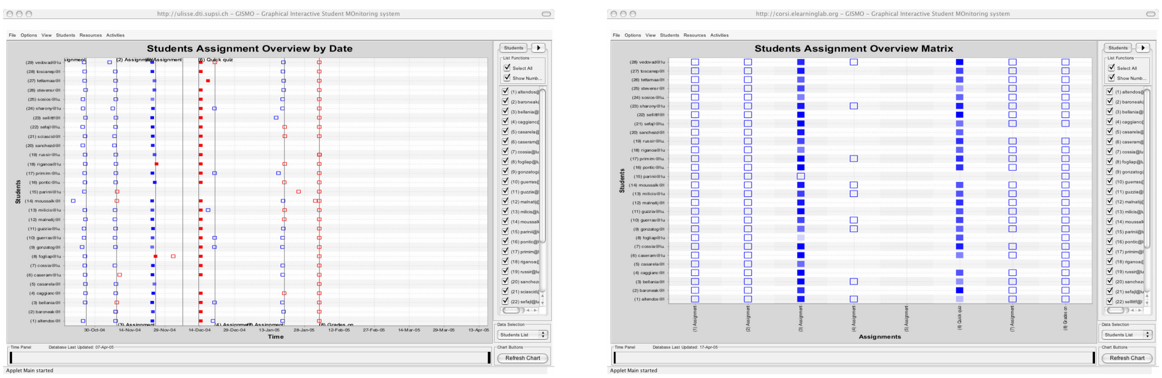


Figure 6: Two graphs reporting data from the evaluation tools. On the left instructors can see when the students submitted the assignment/quiz. On the right there is an indication of who submitted the assignment/quiz and an approximate indication on the grade. Different graphs are provided for quizzes and assignments.

Learning environments provide instructors with some tools to measure the level of comprehension achieved by students in the course's concepts. These tools are quizzes and assignments. For these two tools we produced the representations depicted in Figure 6. The tools collect data from these submissions: date and time of the submission, and, if this is available, the grade. This information is precious to the instructor who has to analyse the feedback on level of comprehension achieved by students, and then tune the teaching or the material accordingly. The left graph in Figure 6 is dedicated to visually indicate the date of submissions. Vertical lines correspond to deadlines of each quiz or assignment provided to students (represented here on the Y-axis). In this example it can be clearly seen that almost all submissions were made just in time when the deadline was almost approaching. Also, all students submitted their work late for the fifth assignment. Very few students submitted their solutions for the sixth assignment. Lines and marks have different colours for different quizzes or assignments to help the reader locate the marks corresponding to each.

Together with the submission time, the grade is another useful piece of information provided with the graph on the right. On the X-axis we mapped the assignments (or quizzes in the graphs dedicated to quizzes) and marks denote students' submissions. An empty square means a submission not graded, while a coloured square reports the grade: a lower grade is depicted with a light colour, a high grade is depicted with a dark colour. In the example, only the third and sixth assignment have been graded.

The graph allows micro and macro analysis of the students' performance. At the macro level, it can bee seen how most students submitted their solutions to each work, excepting the fourth and the fifth ones. At the micro level, the instructor may use the graph to detect problematic topics or students by comparing columns and rows. The instructor can also see how it is the performance of a particular student on a specific topic.

7 Conclusions and future work

We have presented a novel approach of using graphical representations of student tracking data collected by learning tools to help instructors become aware of what is happening in distance learning classes. A system, called GISMO, illustrated the proposed approach. GISMO has been implemented based on our previous experience with the CourseVis research, and proposes some graphical representations that can be useful to gain some insights on the students of the course.

Some forms of graphical representations have been explored in other works. Particularly, some forms of visualizing cognitive aspects of students have been explored in open student modelling projects, e.g. ViSMod [12] uses concept maps to render a Bayesian student model; UM [5] uses different types of geometric forms to represent known/unknown concepts; KERMIT [3] uses histograms to represent levels of a student's knowledge. The pictorial representations provided by such systems externalise a student model built by the system based on some Artificial Intelligence inference. Extracting student and group models can be fairly challenging, especially when dealing with large numbers of students. By contrast, graphical representations provided by GISMO merely represent data collected by CMS in a visual format with minimum data processing. In this case models are inferred in the instructor's mind, instead of being inferred by algorithms.

The GISMO is part of the research project "*EDUKALIBRE, Libre software methods for E-Education*" funded by the European Union in the years 2003 – 2005. This project aims at the translation of the uses and procedures of libre software (free/open source software) to the creation of content suitable to be used as material for education. Within this project, a pool of experts is currently evaluating GISMO with respect to the usability and pedagogical evaluations.

Acknowledgements

This work has been supported by the Swiss Federal Office of Education and (OFES), grant N. UFES SOC/03/47, and it is part of the research project "*EDUKALIBRE, Libre software methods for E-Education*" funded by the European Union in the years 2003 - 2005. (project N. 110330-CP-1-2003-1-ES-MINERVA-M). Thanks to Marcello Mazza for reviewing this paper.

References

[1] Card, K. S., Mackinlay, J. D., Shneiderman, B., 1999. *Readings in Information Visualization, using vision to think.* Morgan Kaufmann, Cal. USA.

[2] Cotton, K. (1988). *Monitoring student learning in the classroom*. School Improvement Research Series (SIRS). Northwest Regional Educational Laboratory, U.S. Department of Education.

[3] Danita Hartley, Antonija Mitrovic. *Supporting Learning by Opening the Student Model*. Proceedings of ITS 2002, Biarritz.

[4] Helic, D. and Maurer, H. and Scherbakov, N. (2000). *Web based training: What do we expect from the system*. In Proceedings of ICCE 2000, Taiwan, pages 1689–1694.

[5] J. Kay. *The UM toolkit for cooperative user modelling*. User modeling and user adapted interaction journal. Vol. 4, 1995.

[6] Mazza, R. *Using information visualisation to facilitate instructors in web-based distance learning.* Ph.D. thesis, University of Lugano, 2004.

[7] Mazza R., Dimitrova V. *Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education.* 13th International World Wide Web Conference (WWW 2004) - Educational Track. 2004 May 17-22, New York.

[8] Mazza R., Dimitrova V. *Informing The Design of a Course Data Visualisator: an Empirical Study.* 5th International Conference on New Educational Environments. Lucerne 26-28 May 2003. (ICNEE 2003). pp. 215-220.

[9] Mazza R., Dimitrova V. *CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning.* In U.Hoppe, F. Verdejo, J,. Kay (eds.) Proceedings of the International conference in Artificial Intelligence in Education. Sydney July 20-24, 2003. (AIED 2003). IOS press. pp. 279-286

[10] McCormack, C. and Jones, D. *Building a Web-Based Education System*. Wiley, New York. 1997.

[11] Ragan, L. C. (1998). *Good teaching is good teaching: an emerging set of guiding principles and practices for the design and development of distance education*. DEOSNEWS, The American Center for the Study of Distance Education, Pennsylvania State University, 8(12).

[12] Diego Zapata-Rivera, Jim E. Greer. *Exploring Various Guidance Mechanism to Support Interaction with Inspectable Learner Models.* Proceedings of ITS 2002, Biarritz.

Automatic Action Analysis in an Interactive Learning Environment

Martin Muehlenbrock
German Research Center for Artificial Intelligence DFKI
Martin.Muehlenbrock@dfki.de

Abstract. Recently, there is a growing interest in the automatic analysis of learner interaction data with web-based learning environments. The approach presented in this paper aims at helping to establish a basis for the automatic analysis of interaction data by developing a data logging and analysis system based on a standard data base server and standard machine learning techniques. The analysis system has been connected to a web-based interactive learning environment for mathematics teaching, but is designed to allow for interfacing also to other web based learning environments. The system has been tested in a five-month experiment in which four classes of a secondary school participated throughout a complete school term on a weekly basis.

1. Introduction

Recently, there is a growing interest in the automatic analysis of learner interaction data with web-based learning environments. This is largely due to the increasing availability of log data from learning environments and in particular from web-based ones. The objectives include the detection of regularities and deviations in the learners' or teachers' actions among others, and to support teachers and learners by providing them with additional information to mange their learning and teaching, respectively, and possibly suggest remedial actions. Commercial systems such as WebCT, Blackboard, and LearningSpace already give access to some information related to the activity of the learners including some statistical analyses, and provide teachers with information on course attendance and exam results. With this information already being useful, it only represents the tip of iceberg of what might be possible by using advanced technologies.

This upcoming field, i.e., addressing the automatic analysis of learner interaction data, is related to several well-established areas of research including intelligent tutoring systems, web mining, and machine learning, and can build upon results form these fields for achieving its objectives. In contrast to intelligent tutoring systems, learner interaction analysis does not rely on models of learner or domain knowledge since these are heavy to build and maintain. In this regards, learner interaction analysis is comparable to website data mining, but with a specific perspective on learning settings and with the availability of pedagogical data that usually are not available in web mining applications that are mostly based on click through data. Click through data streams only allow for a rather shallow analysis, but with the inclusion of pedagogical data also more advanced techniques can be adopted from the field of machine learning.

Although a number of open questions have already been tackled [1][2][5][6][7][10][12] , there is not yet a systematic approach in analysis interaction data from huge learner action logs. The approach presented in this paper aims at helping to establish a basis for the automatic analysis of interaction data by developing a data logging and analysis system based on a standard data base server and standard machine learning techniques. The analysis system has been connected to a web-based interactive learning environment for mathematics teaching, but is designed to allow for interfacing also to other web based learning environments. The system has been tested with a medium scale experiment in which four classes of a secondary school participated throughout a school term of five months on a weekly basis.

2. System

ActiveMath is a web-based learning environment that dynamically generates interactive courses adapted to the student's goals, preferences, capabilities, and prior knowledge [3], [4], [9]. The content is represented in a reusable XML-knowledge representation specifically designed for an educational context. ActiveMath supports individualized learning material in a user-adaptive environment, active and exploratory learning by using (mathematics) service tools and with feedback, better reusability and interoperability of the encoded content and exercises. For different purposes and for different users the learning material and its presentation can be adapted: the selection of the content, its organization, the means for supporting the user have to be different for a novice and an expert user, for an engineer and a mathematician, for different learning situations such as a quick review and a full study. Since there is no way of knowing in advance the goals, the profile, and the preferences of any user when designing the system, ActiveMath builds on adaptive course generation.

For each learner, the ActiveMath environment generates an online log that lists all user actions in the learning environment in terms of general information such as time, type of action, user name, and session number, as well as specific information including which page has been presented to the user, which item has been seen by the user, which exercise has been tackled and solved or not solved. A recent implementation of the learning environment also provides information on user actions in terms of events that other system components can subscribe to. The analysis system is comprised of three major components, i.e., the log database, the updater, and the analyzer.

- The **log database** is at the center of the action analysis system. It contains not only representations of the raw data in the user logs, but also has tables that hold the results of the analysis as well as tables for additional background knowledge concerning users or courses among others (see in figure 1, which will be detailed below).
- The **updater** receives event information on the users' actions from the learning environment, and transforms every user event into one or more corresponding database tables. Usually, the updater receives the information online from the event queue, but it can also read in files with log data that have been generated in an offline mode. In addition to updating the event information in the database, the updater also enhances and extends the event data, as will be described below.
- The **analyzer** performs data aggregation and evaluation in terms of queries to the log database as well as incorporates a number of learning methods and takes the data from the log data base as an input. If needed, adjustments and preferences are input by the user that is running the analysis. An example will be given in section 4.

The analysis system has been implemented by using standard technology such as Java and mySQL, which are available for a number of platforms and operating systems, together with the suitable drivers for database connectivity. In addition, the analyzer is based on the Weka toolkit [11], which provides tools for visualizing and exploring data as well as means for integrating machine learning functionality into applications.

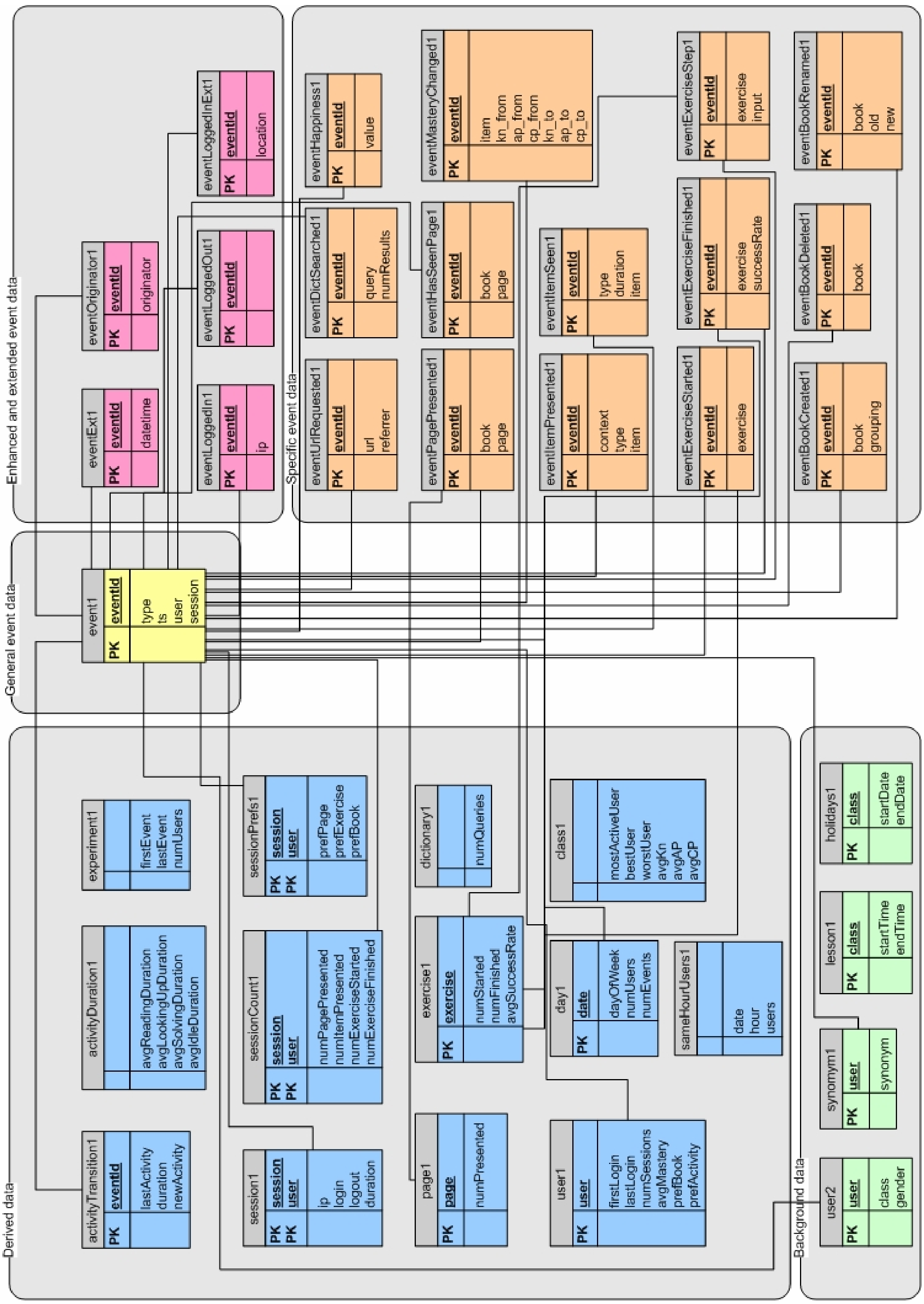


Figure 1. The schema of the log database is comprised of tables for the raw data (subdivided into general event data, specific event data, and enhanced and extended event data), for the derived data, and for background data.

The database is organized in four areas as illustrated in figure 1. The schema depicts a table for generic event information (in the center), tables for event-specific data (at the bottom right), tables for enhanced and extended event-specific data (at the top right), tables for derived data and views (at the top left), and tables for background information (at the bottom left). The event table contains information that is present in every event. In represents the backbone of the data schema related to the raw data. Event-specific tables incorporate information that is provided only by individual events. The structure of these tables has been designed closely to the events specification, since this allows for simpler updating operations when the event subsystem is changed or replaced by another system.

In order to comply with this principle, information that is derived by the activity analysis and is closely related to the underlying events is not incorporated in these tables, but further tables are created, e.g. an extension of the event table that incorporates date and time information in a readable format instead of the timestamp, and an extension of the login table that derives information on a user's location in terms of being in school/work or at home from the network address of the user's computer. In addition, frequently for some tables the information is not complete. For instance, most users do not log out of the learning system explicitly but simply close the browser or shut down their computer. In this case no event is generated concerning the logout. The corresponding logout table is enhanced by information that is derived from the other events the user created and on heuristics concerning pauses and open hours among others. This information is automatically added to the login table, but is marked as derived information in an additional table concerning the source of information.

3. Experiment

The analysis system has been tested in an experiment in a secondary school with about 70 students from three different classes that used the learning environment for a period of five months. The subject area was fractions and divisibility, and material had been prepared in terms of reading material, exercises, and dictionary entries for the ActiveMath learning environment (see figure 2). In addition, a further course of about 25 students were taught the same subject, but in the traditional classroom manner. The other three courses used the ActiveMath learning environment on a weekly basis in two-hour lessons. During the online course each class was split into two subgroups using different computer rooms. Many students already were familiar with computers, but a considerable number needed further instruction even for basic operations such as login.

A preliminary evaluation of the logged data after a first couple of sessions showed some problems in the quality of the data. For instance, instead of registering with the ActiveMath system only in the very first session and using the created user account in the sequel, a large number of students created a new account including a new user name for each session, which makes difficult the longitudinal analysis of the data. The problem was resolved by having the students create only one account and making the registration procedure inaccessible for them after that. Figure 3 provides a view on the data that shows the number of events related to the hours of the day. Clearly, the major amount of events was created during lesson hours between 9 am (9h) and 2 pm (14 h), but some events were generated earlier or later in the day. Some of them are due to a small number of students using the system off time, though most of these are due to teachers and system administrators preparing or evaluating the system.

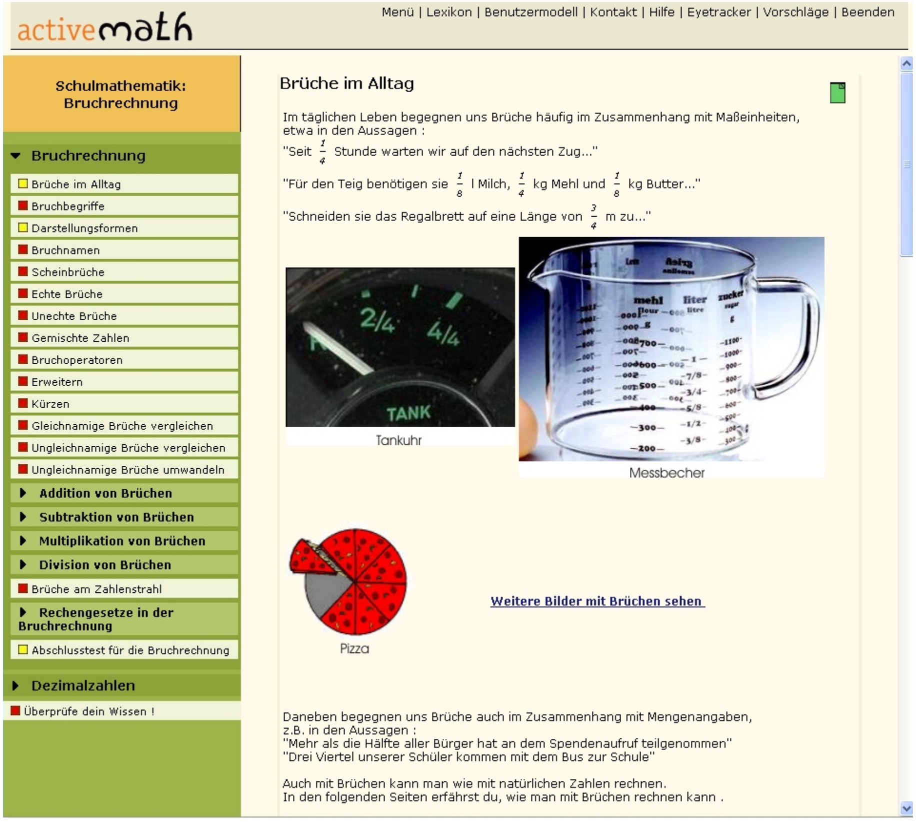


Figure 2. Sample course content on fractions and divisibility (in German).

At the end of the term, a written post test has been done with the students to assess what they had learnt. The results were added to the database manually as well as some further information on gender, teacher, etc. Further information was automatically generated by the analysis system, or more specifically by the updater component, from the log data and added to the database. For each student the information in table 1 has been gathered for further analysis. For anonymity reasons the students used arbitrary user names in the learning environments, and they were to give these user names also in the post test. However, in one course the students put down their real names on the test sheets, a fact which makes the linking to their log data impossible. Finally, 25 student records were complete and clean enough for being used in the further analysis.

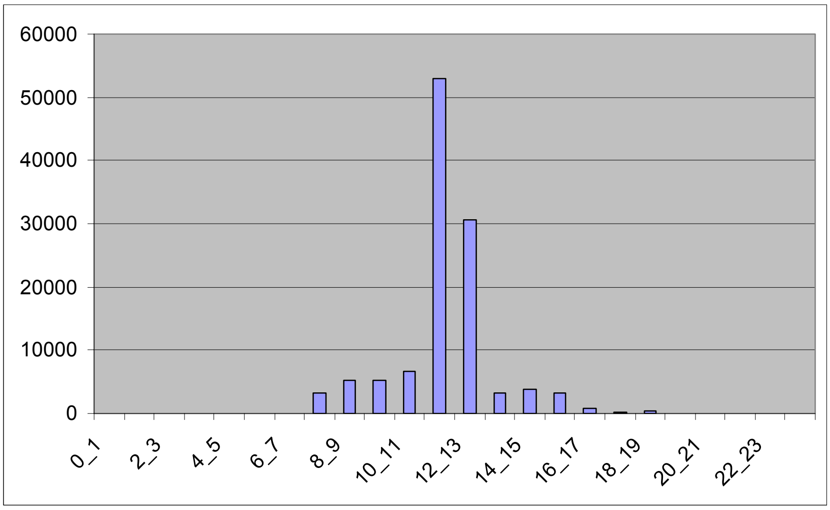


Figure 3. Number of user actions in relation to hours of the day.

Attribute	Generation	Comment
User	Manual	User name (not used for the decision tree learning)
Class	Manual	Course, each comprised of about 20 students (not used for the decision tree learning)
Teacher	Manual	Each class has been split into two subgroups, with each being taught by another teacher (not used for the decision tree learning)
Gender	Manual	Male or female
Integration pupil	Manual	Whether the student is handicapped
Post test result	Manual	Results in the post test done in writing (binned into low, medium, and high for the decision tree learning)
Ex_started	Automatic	Number of exercises started
Ex_finished	Automatic	Number of exercises finished
Num_successes	Automatic	Number of successful exercises
Avg_reading	Automatic	Average number of reading actions in a session
Avg_solving	Automatic	Average number of exercise solving actions in a session
DictUsed	Automatic	Whether the student used the dictionary for searching information
WorkedOffTime	Automatic	Whether the student accessed the learning environment beyond lesson hours, e.g. from home or during free periods
Ex_finished_rate	Automatic	Rate of finished exercises to all started exercises
Ex_success_rate	Automatic	Rate of successful exercises to all finished exercises

Table 1. Data automatically gathered and updated for each user, plus some background information added manually.

4. Analysis

The analyzer component incorporates a number of machine learning methods for automatically analyzing the data in the log data base. In addition to getting a better insight into the underlying relationships in the data, this also allows for prediction and classification of future sessions. Many machine learning methods provide their output in an intelligible, human readable form. For instance, methods for generating decision tress from data, such as C4.5 [8], allow for a tree-shaped representation of the learning results. A decision tree is constructed by the algorithm first selecting an attribute to place at the root node of the tree and make one branch for each possible value. This splits up the example set into subsets, one for every value of the attribute. The attribute is selected in a way that maximizes the information gain by the chosen attribute. This process is repeated recursively for each branch, using only those instances that actually reach the branch. If at any time all instances at a node have the same classification, the developing of that part of the tree is stopped.

Figure 4 shows the decision tree that was generated by the system for characterizing the attribute post test, after the values of the post test results had been binned into the three categories low, medium, and high with nearly equal distribution as well as nearly equal intervals. According to this decision tree, the attributes exercise success rate and exercise finished rate are the most important and second most important features to classify the post test results, respectively. On the third and forth level also the number of finished exercises and the average time spend on reading are relevant. This decision tree is based on the data from the experiments, but it can also be used to make predictions in future usages, i.e., by estimating the effect of using the learning environment on new users in terms of a potential post test.

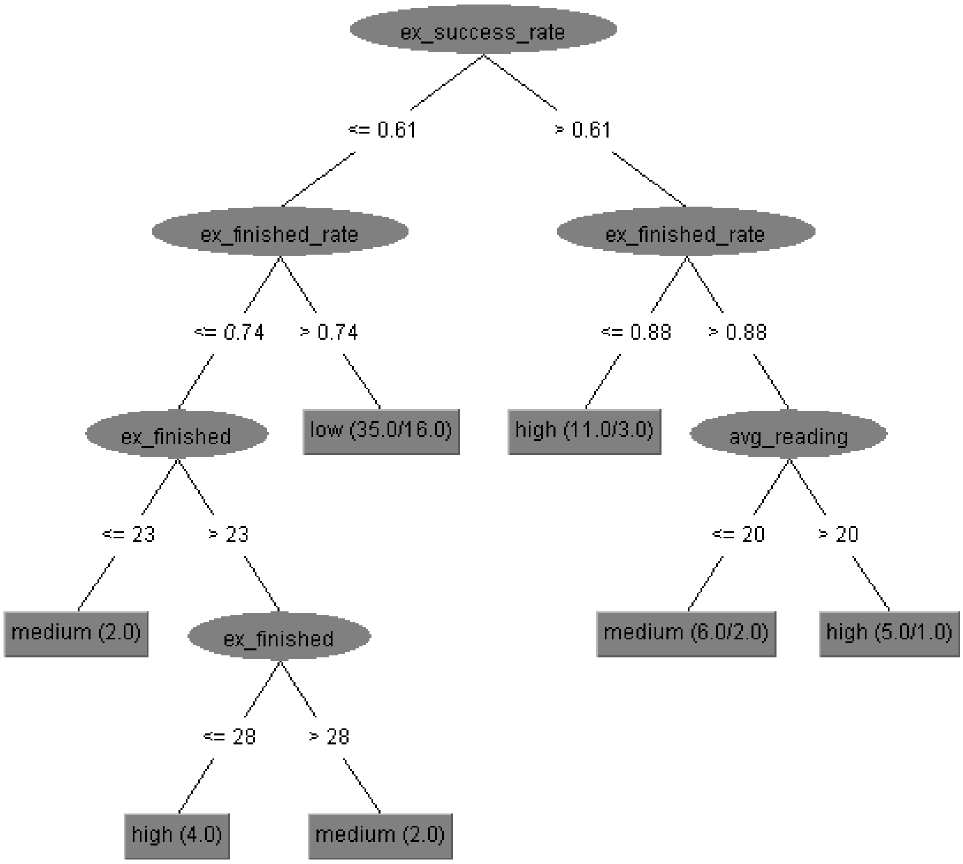


Figure 4. Decision tree for characterizing post test result.

	Low	Medium	High
Low	11	3	7
Medium	9	1	13
High	5	5	11

Table 2. Confusion matrix with tenfold cross validation for the decision tree in figure 4.

The quality of the decision tree is presented in table 2 as a confusion matrix. A confusion matrix displays the result of testing the decision tree with data as a two-dimensional matrix with a row and a column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements. Stratifies tenfold cross-validation has been used to produce this table, which means that the data is divided randomly into ten parts, in each of which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the learning scheme is trained on the remaining nine-tenth, and finally the error estimates are averaged.

The confusion matrix indicates that only for slightly more than one third of the examples the post test result is predicted correctly, for less than one half of the examples a near miss (medium instead of high etc.) is indicated, and for one fifth of the examples the classification is completely wrong.

5. Summary and further work

In this paper a system is presented for the automatic analysis of user actions in web-based learning environments. It has been tested in a school experiment with about 70 students over a couple of months. The automatic analysis of the data already produced a number of interesting results including decision trees that could also be used for prediction in further experiments as well as normal usages. The analysis components have been implemented by using Java and mySQL, which are available for a number of platforms and operating systems. This paper described work in progress, hence further experiments on school and university level will be conducted as well as further analysis methods and machine learning techniques will be investigated.

Acknowledgements

I am grateful to the members of the ActiveMath group for a number of stimulating discussions. A part of the implementation of the action analysis system has been done by Oliver Scheuer. The work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project iClass contract IST-507922.

References

[1] Arroyo, I., Murray, T., Woolf, B. P. (2004). Inferring unobservable learning variables from students' help seeking behavior. *In Proceedings of the workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, pages 29-38, Maceio, Brasil, August.

[2] Heiner, C., Beck, J., Mostow, J. (2004). Lessons on Using ITS Data to Answer Educational Research Questions. In Proceedings of the workshop *Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at ITS 2004*, pages 1-9, Maceio, Brasil, Aug.

[3] Melis, E., Büdenbender, J.,Andres, E.,Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., and Ullrich, C (2001). ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4): 385–407.

[4] Melis, E. and Ullrich, C. (2003). How to teach it - Polya-inspired scenarios in ActiveMath. In *Proceedings of the International Conference on AI in Education AIED-2003*, pages 141-147, IOS Press.

[5] Merceron, A. and Yacef, K. (2003). A Web-Based Tutoring Tool with Mining Facilities to Improve Learning and Teaching. In *Proc. of the 11th International Conference on Artificial Intelligence in Education AIED 2003*, pages 201-208, IOS Press.

[6] Merceron, A., Oliveira, C., Scholl, M., & Ullrich, C. (2004). Mining for Content Re-Use and Exchange - Solutions and Problems. In *Poster Proceedings of the 3rd International Semantic Web Conference, ISWC2004*, pages 39 - 40, Nov.

[7] Mostow, J. (2004). Some Useful Design Tactics for Mining ITS Data. In Proceedings of the workshop *Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at ITS 2004*, pages 20-28, Maceio, Brasil, Aug.

[8] Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA.

[9] Ullrich, C., Libbrecht, P., Winterstein, S., and Muehlenbrock, M. (2004). A flexible and efficient presentation-architecture for adaptive hypermedia: Description and technical evaluation. In *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies ICALT 2004*, pages 21-25, IEEE Computer Society.

[10] Oliveira, C. and Domingues, M. (2003). Data Warehouse for Strategic Management of an eLearning System, In *Proceedings of: the 2nd International Conference on Multimedia and ICT in Education m-ICTE 2003*, pages 627-631. Badajoz, Spain, Dec.

[11] Witten, I. H., Frank, E. (1999). *Data Mining*. Morgan Kaufmann, San Francisco.

[12] Zhang, J. and Lu, J. (2001). An educational data mining prototype. In *Proceedings of the 10th International Conference on Artificial Intelligence in Education AIED 2001*, pages 616-618, IOS Press.

Student's modelling with a lattice of conceptions in the domain of linear equations and inequations

Jean-François NICAUD*, Hamid CHAACHOUA**, Marilena BITTAR***, Denis BOUHINEAU*

** Université Joseph Fourier, Leibniz-MeTAH, 46 Ave Viallet, 38031 Grenoble, France*
Jean-Francois.Nicaud@imag.fr, Denis.Bouhineau@imag.fr
*** IUFM-Leibniz-MeTAH, 46 Ave Viallet, 38031 Grenoble, France, Hamid.Chaachoua@imag.fr*
**** Universidade Federal de Mato Grosso do Sul, Campo Grande, Brasil, marilena@nin.ufms.br*

Abstract. We present a student's modelling process in algebra which consists of two phases. The first phase is a local diagnosis where a student's transformation of an expression A into an expression B is diagnosed with a sequence of rewriting rules. A library of correct and incorrect rules has been built for that purpose. The second phase uses a lattice of conceptions built for modelling students more globally. Conceptions are attributed to students according to a mechanism using the local diagnoses as input.

This modelling process has been applied off-line to data gathered in France and Brazil with 13-16 years old students who used the Aplusix learning environment. The results are described and discussed. The process will be included later inside Aplusix in order to model students on-line and to provide the students' conceptions to the teachers.

Keywords: student's modelling, algebra, rewriting rules, conceptions.

1. Introduction

Since 2003, we are engaged in a research work[1] devoted to automatic student's modelling in algebra. Our general model is based on correct and incorrect *rules* allowing to interpret the local behaviour of the student, and on *conceptions* allowing to model the student in a more global way. Our methodology contains five components: (1) Gathering a large set of data concerning students' calculations; (2) Construction of a library of correct and incorrect rules; (3) Automatic diagnosis of the student's calculations in term of sequences of rules; (4) Construction of a library of conceptions; (5) Automatic diagnosis of the student's calculations in term of conceptions. This paper mainly describes the parts devoted to conceptions.

According to Artigue [1], a conception is related to a concept and is characterized by three components: (1) a set of situations which give meaning to the concept; (2) a set of significations (mental images, representations, symbolic expressions); (3) tools (rules, theorems-in-act, algorithms). This paper is mainly devoted to theorems-in-act associated to the concept of movement in (in)equations. An example of theorems-in-act is the following: *When a sub-expression is moved from one side to the other side in an (in)equation, its sign is always changed*. Note that this is correct for additive movements (e.g., $x-3 = 5 \rightarrow x = 5+3$) and incorrect for multiplicative movements (e.g., $-3x = 5 \rightarrow x = 5/3$). This theorem-in-act is not a rewriting rule: it applies to all the movement rules (and generally, several theorems-in-act apply to a given rule). In the rest of the paper, we will use the term conception instead of the term theorem-in-act for fluidity reasons. The reader needs just to remember that conception means, in the paper, a theorem-in-act part of a conception.

[1] This work is funded by the programme 'ACI, Ecole et Sciences Cognitives' of the French Ministry of Research.

Several research works have been devoted to conceptions and some conceptions have been built by hand [1, 3, 4]. However, automatic diagnosis of conceptions is a difficult problem which has been little investigated. This is the part of our research which is mainly described in this paper. Examples of previous works in this field are Pépite [7] and Baghera [10]. We will compare our work with Pépite in the last section.

For gathering the data to be analysed, we have used the Aplusix learning environment [8, 12] which allows students to freely make calculation steps, as they do in the paper environment, and which records all the students' actions in log files. Except for 6 classes, Aplusix has been used in an ecological context, i.e., by students supervised by their mathematics teachers during the normal schedule of the class. Data have been gathered in France and Brazil from about 3000 students of grades 8 to 11.

The long term goal of this work is twofold: First, we aim at building a map of conceptions covering the domain of elementary algebra (algebra for grades 8 to 11) and to diagnose thousands of students from many countries, to get a repartition of the obtained conceptions; this is our academic goal. Second, we plan to encompass the map of conceptions and the diagnosis mechanism in the Aplusix system to improve its interest by providing to the teachers the conceptions of their students, and by selecting the best exercises to give to the students according to the calculated conceptions (in particular to correct the erroneous parts of these conceptions); this is our applied goal which is similar to Cognitive Tutors [6] in a larger domain. This paper presents the ongoing work on the movement concept in (in)equations. A detailed description of the different parts of the work is available in a research report [11].

2. Modelling student's calculations with rewriting rules

Most of the calculations in formal algebra concern the application of rewriting rules, according to the "replacement of equals" principle. Given two expressions A and B, a rewriting rule R: $A \rightarrow B$ can be applied to an expression E if there is unification between A and a sub-expression U of E. The application of the rule, when possible, consists of replacing U in E by B. Rewriting rules usually come from algebraic identities provided by axioms and theorems. For example, the identity $A(B+C)=AB+AC$ produces two rewriting rules, one is the expansion rule $A(B+C) \rightarrow AB+AC$, the other is the factorisation rule $AB+AC \rightarrow A(B+C)$. Most of the rewriting rules of elementary algebra can be classified in *reduction and simplification rules* (numerical calculations, like term collection, e.g., $3x+5x \rightarrow 8x$, etc.), *factorisation rules* (which provides additional factors e.g., $AB+AC \rightarrow A(B+C)$), *expansion rules* (inverses of factorisation rules), and *rules on relations* (see below). When a student solves an exercise in formal algebra, he/she produces calculation steps. Our model interprets these steps as the application of correct or incorrect rules. For example, if the student transforms $2x(3x^2–4)$ into $6x^3–4$, we can interpret this calculation step by the application of the incorrect rule $A(B+C) \rightarrow AB+C$ followed by the application of correct reduction rules.

The main strategy for solving linear equations and inequations consists of expanding both sides, if necessary, then isolating the variable, using rules which carry out identical operations on both sides or using movement rule that move an additive or multiplicative expression from one side to the other. Examples of these rules are: $A=B \rightarrow A+C=B+C$, addition to both sides; $A+C=B \rightarrow A=B–C$, additive movement; $AC=B \rightarrow A=B/C$ ($C\neq0$), multiplicative movement. There are 12 movement rules for equations (4 additives and 8 multiplicative) and also 12 for $\neq$ inequations. For the other inequations ($<$ $\leq$ $>$ $\geq$), multiplicative movement rules are duplicated according to the semantic sign of C, because when this sign is negative, we have an inversion of the inequality (e.g., AC ≤ B → A ≥ B/C (C<0)). Therefore, there are 20 movement rules for each sort of inequations (4 additives and 16 multiplicative). In the school practice, operations on both sides of (in)equations are progressively replaced by movement rules which are the fundamental rules for solving linear (in)equations. Other rules are expansion and reduction rules, which do not apply to the entire (in)equation but to a sub-expression.

The very large quantity of movement rules (104 = 2*12+4*20) questions the relevance of the rule model. Is a student supposed to have all these detailed rules in mind when he/she solves an (in)equation? Certainly not in this form. Hence, we propose another model which emphasizes the main features and we consider at the same time incorrect movements. A movement has an argument which is the element moving from one side to the other side. In an additive movement, the argument is located in a sum or is the whole side. When the movement is carried out correctly, the argument remains additive, on the other side, with an opposite syntactic sign. In a multiplicative movement, the argument can be multiplicative within a numerator, or multiplicative within a denominator. When the movement is carried out correctly, the argument is still multiplicative in the other side, and its place changes (denominator ↔ numerator). In the case of an inequation, the orientation of the inequation changes if the semantic sign of the argument is negative. We can describe movements with only one general rule, entitled “Movement”, associated to the vector presented in table 1.

Table 1. The seven dimensions of the general Movement rule.

Dimension	Possible values
Symbol of relation	= ≠ < ≤ > ≥
Horizontal orientation	LeftToRight, RightToLeft
Vertical orientation	NumeratorToNumerator, DenominatorToNumerator, NumeratorToDenominator, DenominatorToDenominator
Initial position of the argument	InitAdditive, InitMultiplicative
Final position of the argument	FinAdditive, FinMultiplicative
Change of syntactic sign of the argument	SynSignChanged, SynSignUnchanged
Change of the inequality orientation	OrientationChanged, OrientationUnchanged

For example, the incorrect transformation 2x–4 < 5 → 2x > 5–4 is represented by a Movement with the argument “–4” and the vector (<, LefToRight, NumeratorToNumerator, InitAdditive, FinAdditive, SignUnchanged, OrientationChanged).

3. A local diagnosis algorithm

We have implemented an algorithm to diagnose student’s local algebraic transformations in terms of rewriting rules. The algorithm uses a library of 260 correct and incorrect rules. This library has been obtained from a cognitive analysis (we observed students’ actions in various situations using the replay system of Aplusix) and from an epistemic analysis (as the one carried out for the movement rule), see details in [9, 11]. The incorrect rules of the library currently covers operations on (in)equations, reductions (except operations on fractions and square roots) and expansions. Fractions, square roots and factorisations will be studied later.

The local diagnosis algorithm has three phases. The first phase isolates the sub-expressions where the transformation occurs. As a result, an expression A has been transformed into an expression B by the student. The second phase is a heuristic search algorithm which develops a tree from the starting expression A. At each step, the node of the search space which is the closest to B, according to a distance between expressions, is chosen and developed.
The development consists of applying the applicable rules of the library. When the

development produces the expression B, the goal is reached and the path from A to B in the tree is a sequence of rules that explains the transformation of A into B. The second phase stops when a chosen number of nodes have been developed or when no more rules are applicable. So this phase can fail. It can fail because of: a missing incorrect rule, an early stop of the process, a student's behaviour that has not to be understood. The third phase consists of the evaluation of the different diagnoses and of the choice of the best one. For example, the transformation of $2x–6 = 7x–8$ into $–5x = –14$ is diagnosed with 4 rules: (1) Incorrect additive move of 6 leading to $2x = 7x–8–6$; (2) Correct additive move of 7x leading to $2x–7x = –8–6$; (3) Correct additive reduction, leading to $–5x = –8–6$; (4) Correct additive reduction, leading to $–5x = –14$.

The current performance of the local diagnosis, in terms of success/failure, for classes of grades 8 and 9, in France (540 students) and Brazil (2500 students), is the following: Between 90% and 100% of success for correct transformations, depending of the class, and between 74% and 93% of success for incorrect transformations. Note that a failure is sometimes the best diagnosis (researchers don't always explain a student's transformation). The correctness of the diagnoses has been studied by three researchers for two French classes, grades 8 and 9, for incorrect expansions, incorrect reductions and incorrect transformations on in(equation): between 82% and 97% diagnoses have been considered to be correct, depending of the class and the category of rule (expansions, etc.).

4. A lattice of conceptions

We have designed a set of conceptions for the concept of movement which is organised as a lattice. Each conception which is not a micro-conception has two subconceptions and is the union of these two subconceptions. The conceptions at the first level are called "global conceptions" with respect to an aspect of the concept of movement. We have considered three aspects of the concept of movement: the ***sign aspect*** (whether the sign of the argument is changed or not), the ***inequality orientation*** for inequations (whether the orientation of the inequality is changed or not) and the ***operator evolution*** (what happens to the operator linking the argument to the (in)equation in the movement). Figure 1 shows a part of this lattice. We have defined five global conceptions for the *sign aspect*:

- CorrectSign: Correct treatment of the sign of the argument;
- AbsoluteValue: Change of the sign of the argument if and only if this sign is "–";
- SemiAbsoluteValue: Change of the sign of the argument if and only if: this sign is "–" and the argument is multiplicative; or the argument is additive;
- SaveSign: Never change the sign of the argument;
- ChangeSign: Always changes the sign of the argument.

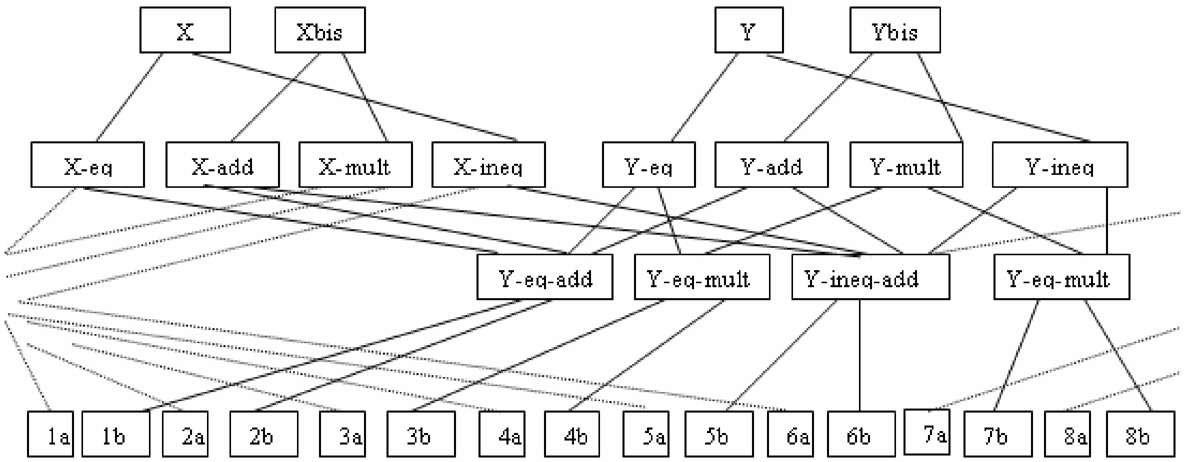
Of course, CorrectSign is the only correct conception in that list. The other conceptions produce correct or incorrect calculations depending on the context. The *inequality orientation* and the *operator evolution* aspects lead to similar decompositions in global conceptions.

Let us detail the ChangeSign global conception of the sign aspect. This global conception consists of "always changing the sign" of the argument of a movement. It can be decomposed in partial conceptions: ChangeSign-eq for "always changing the sign in an equation" and ChangeSign-ineq for "always changing the sign in an inequation". Again, ChangeSign-eq can be decomposed in more partial conceptions ChangeSign-eq-add for "always changing the sign in an equation for an additive argument" and ChangeSign-eq-mult for "always changing the sign in an equation for a multiplicative argument". In this framework, some students may have a ChangeSign-eq and not a ChangeSign-ineq conception. Those who have both ChangeSign-eq and ChangeSign-ineq conceptions have the ChangeSign conception.

For ChangeSign-eq-add (Y-eq-add in figure 1), we have two micro-conceptions named 1b and 2b. The first one is changing the sign when it is “+” in an equation and when the argument is additive; the second one is changing the sign when it is “–” in an equation and when the argument is additive.

In order to have a lattice, we add some nodes considered as abstract conceptions: three nodes containing the aspects (SignAspect, InequalityOrientation and OperatorEvolution) and having the global conceptions as direct descendants, a “Top” node having these three nodes as direct descendants, and a “Bottom” node being direct descendant of every micro-conception. The lattice has a total of 146 nodes.

Figure 1. Part of the lattice for two global conceptions of the sign aspect: CorrectSign (X in the figure) and ChangeSign (Y in the figure). The abstract nodes are not represented. X has an Xbis conception which is another decomposition of the same conception (X is first decomposed in eq/ineq then add/mult; Xbis is first decomposed in add/mult then eq/ineq). X and Xbis are active at the same time. Xbis is necessary to allow to have X-add and X-mult in the lattice.



5. Computation of the student’s conceptions

The computation of the student’s conceptions uses first the local diagnoses described in section 3, and an intermediary construction: the Local Behaviour Vector (LBV) which links movement rules and micro-conceptions. The movement rules of the local diagnoses which match LBVs are counted for each LBV. When a condition is verified for a LBV, a corresponding micro-conception is activated. Then a propagation mechanism is launched to activate other conceptions.

For building the Local Behaviour Vectors, we use some elements of the vector of the unique movement rule and some other useful elements. This is the decision of the observer, as emphasised by [3]: “Modelling behaviour requires a first level of interpretation, that of the organization of reality”. The LBVs we have chosen for the *sign aspect* is described in table 2.

Table 2. The seven dimensions of the Local Behaviour Vector for the sign aspect.

Dimensions (or variables)	Possible values
Type of the exercise	Equation, Inequation
Initial position of the argument	InitAdditive, InitMultiplicative
Sign of the argument	SignArgPlus, SignArgMinus
Change of syntactic sign of the argument	SynSignChanged, SynSignUnchanged

The first three variables define the context, the last one corresponds to the action. When the first three variables are chosen, the two possible values of the last one determine a couple of opposite LBVs. An example of a couple of opposite LBVs is the following: the context is (Equation, InitAdditive, SignArgMinus); one LBV has the action SignChanged, the other has the action SignUnchanged.

When we have a list of rules attributed to a student by the local diagnoses of a set of transformations, we match each movement rule to each LBV and count the number of occurrences. A student who has a rational behaviour, with respect to the model, would mark only one LBV of a couple of opposite LVB. However, we cannot expect to have this level of rationality. As a consequence, we have chosen the following mechanism to activate LBVs: Let LBV1 and LBV2 be two opposite LBVs. Let n1 and n2 be the respective numbers of occurrences of LBV1 and LBV2 for a given student:

IF n1+n2 = 0 THEN there is no LBV activation
ELSE IF n1 / (n1+n2) ≥ 2/3 THEN LBV1 is activated with coefficient n1 / (n1+n2)
ELSE IF n2 / (n1+n2) ≥ 2/3 THEN LBV2 is activated with coefficient n2 / (n1+n2)
ELSE there is no LBV activation

An activated LBV becomes a micro-conception which can be expressed by: "In the context of this LBV, the student generally performs the action of the LBV". Generally is expressed by the coefficient which is a sort of *Certainty Factors* [5].

The micro-conceptions are the lowest real conceptions in the lattice (having Bottom as direct descendant). After the activation of the micro-conceptions, each upper conception is determined by calculating recursively its coefficient as the geometrical average of the coefficients of its two direct descendants (if the coefficients of the two direct descendants are a and b, the result is sqrt(ab)).

A micro-conception is correct or incorrect. When it is incorrect, any behaviour that matches the micro-conception is incorrect. A conception which is not a micro-conception is correct if and only if its two direct descendants are correct. When a conception, which is not a micro-conception, is incorrect, some behaviours that match the conception are incorrect but, generally, others are correct. For example, the ChangeSign conception in which "the student always changes the sign" contains correct and incorrect behaviours.

6. Experimental study

Since 2003, we have conducted several experiments with classes in France and Brazil of grades 8, 9 and 10 and we have recorded thousands of hours of students' activities. We have recently applied the modelling process to a part of the data. It produced a description of each student in terms of a list of conceptions attributed to this student, and a summary table containing the number of occurrences of each conception.

An experiment with a group of 342 students of grade 9 was conducted in Campo Grande (Brazil) in 2004 with 20 minutes of use of Aplusix for familiarisation and 1 hour of use in the test mode where no feedback was given to the students. The modelling process has been applied to the data corresponding to the test phase. The analysis of the distribution of the conceptions with respect to the context shows what follows:

- Type of exercise: 56% of conceptions concern equations (97% correct and 3% incorrect); 44% of conceptions concern inequations (71% correct and 29% incorrect).
- Initial position of the argument: 62% of conceptions concern an additive position (95% correct and 5% incorrect); 38% of conceptions concern a multiplicative position (68% correct and 32% incorrect).

Most of the conceptions concern equations with an additive position of the argument. The high rate of correct conceptions cannot be viewed as certitude of a good result because the level of generality of the conceptions has to be taken into account, the more general ones being the "global conceptions" defined in section 4. Actually, we had only 2% of correct global conceptions (e.g., CorrectSign). At the other levels (the level is the depth in the lattice, the global conception having level 1), we have 32% correct conceptions for level 2 (e.g., CorrectSign-eq), and 66% correct conceptions concerning very specific contexts of levels 3 and 4. These results are coherent with hand analysis made for a few students and with the general opinion of the teachers of the classes. The distribution of the number of conceptions per student is a Gaussian distribution, see table 3.

Table 3. Distribution of the numbers of conceptions. Note that the ideal student has 3 conceptions: the correct global conception of each of the 3 aspects of the movement concept.

Number of conceptions	0	1	2	3	4	5	6	7	8	Total
Number of students	11	37	58	80	38	47	48	22	1	342
Percentage	3.2	10.8	16.9	23.4	11.1	13.7	14	6.4	0.3	100

The distribution of the conceptions with respect to the aspects of the movement concept is shown in table 4. There are many correct conceptions for *Sign aspect* and *Operator evolution*, but just a few of them are at level 1. There is an important amount of incorrect conceptions at level 1 for *Inequality orientation*. This is coherent with the fact that these students have had many exercises about equations and not many about inequations in the preceding school year.

Table 4. Distribution of the conceptions with respect to the three aspects of the movement concept.

Level of the conception	Sign aspect		Inequality orientation		Operator evolution	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
1	20	0	0	37	0	0
2	158	0	85	4	158	0
3	129	11	0	0	186	0
4	0	0	0	0	335	9

We have applied the modelling process to 221 French grade 10 students, obtaining similar results. The main result for both populations is the following: incorrect conceptions concern two contexts: (1) inequations, (2) multiplicative initial position of the argument.

Last, we have modelled 30 French grade 10 students who used Aplusix during the whole school year 2003-2004. The analysis data collected at the end of the school year shows:

- A total of 171 conceptions with 18% for level 1, 15.2% for level 2 and 30% for level 3;
- 70% students have the CorrectSign correct global conception;
- 40% have the SaveOrientation incorrect global conception (never change the sign of an inequality).

We note that these students, who had a longer training, have more general conceptions, and that the sign aspect is rather well acquired but the orientation aspect of the inequality is not.

7. Discussion and future work

This work is a significant step towards the achievement of the goals we have presented in the introduction. The obtained results are coherent with opinions of teachers and with analyses "by hand" of a few students' data, but a deeper study of the coherence is necessary and will be carried out. However, we need to analyse in depth the data that are not captured by the process. For example when we find that a student has 3 conceptions, we have an interesting result, but in order to achieve a completion goal, we would like to have an opinion about the behaviours of this student that do not participate to these 3 conceptions. Some may be sleeps, other random behaviour, other rational behaviour not captured by the model.

Let us summarize our work and make a comparison with Pépite [7]:

- ○ We collect very large sets of data in ecological situation; Pépite collect limited sets of data in specific situations;
- ○ Our diagnoses are based on implemented detailed knowledge which means that our system has a deep algebraic understanding of the phenomena; this is not the case of Pépite;
- ○ We are building a wide map of conceptions understandable by teachers. So does Pépite which domain is larger than ours, including other registers than the formal one.
- ○ We are preparing an operational process in which the students will learn with Aplusix in ecological situations and in which Aplusix will calculate on-line the students' conce-ptions and tell them to the teacher; this is not the case of Pépite. As Aplusix has proved to help students learn algebra, and as publishers are very interested in selling Aplusix, we hope that many students of many countries will benefit of this work in a few years.

This way of modelling students is not a dynamic way in the sense of building automatically pieces of knowledge to attribute to students like [2]. There are several reasons for that. First, we think that algebra of grades 8 to 10 is a too complex domain for that goal, when a deep modelling is expected; this complexity led us to consider two levels: the rule level and the conception level. Second, we are fundamentally interested by the map of conceptions and by a capacity to provide understandable descriptions of conceptions to the teachers.

Last, we began to think of a new goal which consists of making a benchmark for student's modelling in algebra. This means to give access to the data we collect and to ask the interested research teams to use these data to model students with their own methods and tools. Results of the researches would be published on a Website and compared. Two levels would be possible: a rule level and a conception level. Tools made by teams could be accessible on the Website.

References

[1] Artigue M. (1991). Epistémologie et didactique. Recherche en didactique des mathématique, 10 (2/3).

[2] Baffes P., Mooney R. (1996). Refinement-Based Student Modeling and Automated Bug Library Construction. Journal of Artificial Intelligence in Education, 7(1), p 75-117.

[3] Balacheff N., Gaudin N. (2002). Students conceptions: an introduction to a formal characterization Cahier du laboratoire Leibniz # 65, http://www-leibniz.imag.fr/LesCahiers/Cahiers2002.html

[4] Brousseau G. (1997). Theory of didactical situations in mathematics. Kluwer Academic Publishers, Dordrecht.

[5] Buchanan B.G. and Shortliffe E.H. (1984). *Rule-Based Expert Systems, The Mycin experiments of the Stanford Heuristic Programming Project.* Addison-Wesley Publishing.

[6] Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, p. 30-43.

[7] Jean S., Delozanne E., Jacobini P., Grugeon (1999). A diagnostic based on a qualitative model of competence in elementary algebra. AI-ED 1999, IOS Press, Amsterdam, p. 491-498.

[8] Nicaud, J.F., Bouhineau, D., and Chaachoua H. (2004). Mixing microworld and CAS features in building computer systems that help students to learn algebra, in *Int. Journal of Computers for Mathematical Learning*, Volume 9 (2), Springer-Verlag.

[9] Sander E., Nicaud J.F., Chachoua H., Croset M.C. (2005) From usage analysis to automatic diagnosis: The case of the learning of algebra. AIED Workshop on Usage Analysis in Learning Systems. Ibidem.

[10] Webber C., Pesty S., Balacheff N. (2001). A multi-agent and emergent approach to learner modelling. Proceedings of ECAI.

[11] Modélisation cognitive d'élèves en algèbre et construction de stratégies d'enseignement dans un contexte technologique. Project report of the "Ecole et sciences cognitives" research programme. Cahier #123 of the Leibniz laboratory, http://www-leibniz.imag.fr

[12] The Aplusix Website. http://aplusix.imag.fr

From usage analysis to automatic diagnosis: The case of the learning of algebra.

Emmanuel SANDER*, Jean-François NICAUD**, Hamid CHACHOUA***, Marie-Caroline CROSET**

** Université Paris 8, 2 Rue de la Liberté 93526, Saint Denis, France sander@univ-paris8.fr*
*** Université Joseph Fourier, Leibniz- MeTAH, 46 Ave Viallet, 38031 Grenoble France Jean-Francois.Nicaud@imag.fr & mkcroset@wanadoo.fr*
**** IUFM-Leibniz-MeTAH, 46 Ave Viallet, 38031 Grenoble, France, Hamid.Chaachoua@imag.fr*

Summary: User modelling is a significant part of usage analysis. We present a case study in the field of the learning of algebra that aims at producing automatic diagnosis rules, based on the analysis of tracks of students solving algebra exercises within the Aplusix learning environment. We present two experiments that were conducted among 8^{th} and 9^{th} grade students. Manual analyses performed on the data made it possible to contribute to the construction of a library of rules aiming at modelling students by hand or automatically. The automatic diagnosis, based on the use of a library of correct and incorrect rules, and on a heuristic search algorithm, reveals a high performance on some of the algebra fields and will be extended to other fields through iterative comparisons with the results of the manual diagnosis.

1. Problematic

Within the scope of a project aiming at automatic student's modelling in algebra [9], we conducted tracks analyses with a variety of students from 8 and 9 grades, in order to identify the systematic errors they commit when solving algebra exercises, and to use the identified incorrect rules [5, 7] for automatic diagnosis of the students' transformations in term of rule applications. In our view, this requires: (i) Designing relevant tasks, i.e., relevant algebra exercises; (ii) Identifying, a set of correct and incorrect rules; (iii) Designing an automatic diagnosis algorithm; (iv) Assessing the quality of this diagnosis. For gathering the data, we have used the Aplusix learning environment [3] that allows students to freely make calculation steps and records all the students' actions (Figure 1). This part of our research work is described in this paper. The diagnoses obtained are next used to model students in term of conceptions[1] [4]. In a longer term, we plan to insert our global process in the Aplusix system to be used in ecological situations where the students will learn algebra skills with Aplusix in usual school situations and Aplusix will calculate on-line the students' conceptions and inform the teacher.

Students rely on conceptions, inadequate in some contexts, that are likely to subsist despite learning [1, 2, 8]. One target of this study, due to the lack of converging and exhaustive results on conceptions in algebra, is to build a panel of exercises and to analyse the errors observed. In other words, we aim at identifying a "map" of incorrect rules and of conceptions. This project is quite ambitious since, to our knowledge, this is the first time

[1] An example of part of a conception is the following: *When a sub-expression is moved from one side to the other in an (in)equation, its sign is always changed*. Note that this is sometimes correct (e.g., $x-3 = 5 \rightarrow x = 5+3$) and sometimes incorrect (e.g., $-3x = 5 \rightarrow x = 5/3$).

that a work aims at achieving exhaustiveness in the field of elementary algebra[2] with a data driven approach. More precisely, we aim, in a first step, at providing a methodology that could be applied at a larger scale in order to provide the expected exhaustiveness. For this reason, the experimental work focuses only on semi-beginners in algebra (8th and 9th grades) and part of the study is achieved only on a subfield, namely solving linear equations.

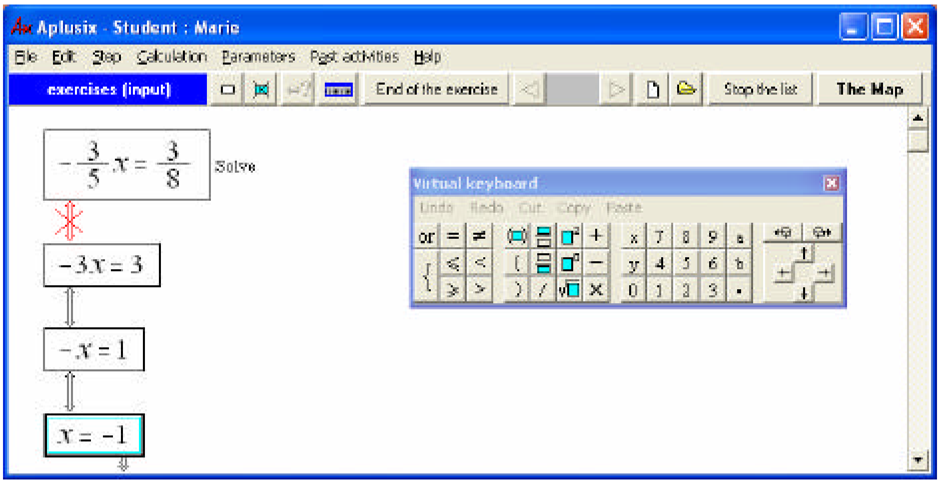


Figure 1. Aplusix problem solving interface

Beyond the incorrect rules identified, the manual analyses performed hereby provide contributions in at least two directions that address issues relevant to many cases of usage analysis involving user modelling: (i) Reaching a conclusion about the stability of the behaviours of the students in the use of the incorrect rules, measured as the repetition of the same behaviour within the same context. This stability is crucial since it conditions the relevance of the diagnosis. (ii) Evaluating the quality of the automatic diagnosis by providing a basis of comparison.

2. Methodology

2.1 Choice of the relevant exercises

We designed two experiments, with complementary purposes. The first one, designed for grade 9 students, intends to make possible gathering a large set of data covering the whole range of rules that might be applied at this level of the curriculum. It is composed of 31 exercises listed below (Table 1). This set of exercises makes possible to observe a large range of incorrect rules in order to build a library of the rules involved. However, this experimental setting reveals some limitations since the importance of the range covered is not compatible with an accurate assessment of the context of application of the rules and of their stability; an apparent lack of stability in the behaviour of a user might reveal that s/he categorizes the situation as different from the previous one despite that they are equivalent from an expert point of view. For this purpose, we designed a second experiment that investigates a subfield in a more systematic manner, namely solving linear equations.

This second experiment was designed in order to focus on a subfield and to check out the possibility of increasing the assessment of the stability of the behaviours identified through a more accurate description of the context of use of the rules. Thus, we built 15

[2] We term elementary algebra, the calculations made on polynomial and rational expressions, polynomial and rational equation and inequation up to highschool. Most of the research devoted to student's modelling in algebra concerns very beginners.

linear equation exercises listed below (Table 2) and we manipulated in quite a systematic manner the parameters that could lead to the identification of the context of application of incorrect rules. More precisely, in an equation of the form ax+b=cx+d we varied the nature of a, b, c, and d and the relations between these values along the dimensions that might be relevant from a naïve point of view; for instance, whether there value is zero or not, and if not whether they are positive or negative numbers, integers or fractions. This methodology is intended to be reproducible in other subfields, in order to encounter the whole range of exercises in algebra as they are categorized by the learners.

2.2 Gathering data and the tool for visualization and construction of numerical indicators

Each action of the student is recorded and can be transcribed by a "video tape recorder" integrated into the software. As a first step of the diagnosis and before its automation, manual analyses were performed. For each student the detail of the resolution of each exercise was looked at with the video tape recorder of APLUSIX (Figure 2) and the rules which made it possible to explain the transformation of an expression into another were identified by the analyst.

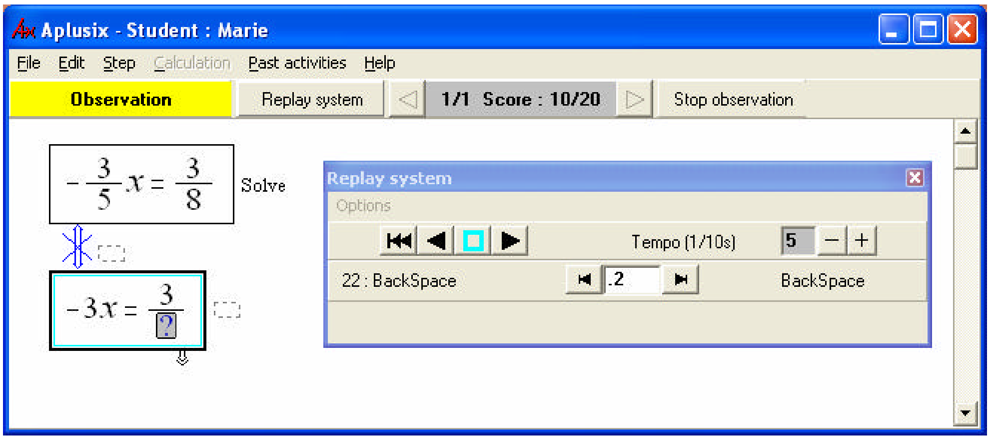


Figure 2. Aplusix's videotape recorder

We built several numerical indicators. Because of the length of these exhaustive manual analyses (several hours for one hour of one student problem solving session), we applied them only to a small part of the whole population that we tested in other studies [9]; namely 3 classes of grade 8 students and 3 classes of grade 9 students.

We measured the rate of occurrence of each incorrect rule among our population. First, this indicator makes possible to distinguish the marginal rules, known as orphan, that seldom occur and that do not deserve designing specific remediation strategies, from the dominant ones that are present in a significant part of the population and deserve to be taken into account seriously. Second, the orphan rules shall not be implemented in the automatic diagnosis in order to avoid combinatorial explosion, whether the most frequent ones have to be implemented in order to avoid diagnosis failures or psychologically implausible automatic diagnoses.

We measured the predictive character of the identified rules with a binary indicator. A rule is regarded as predictive when its use alone leads to the result given by the student. We took a restrictive criterion since we considered that the use of several rules altogether implies that none of the rules are predictive, just as calculation or copy errors interfering with an incorrect rule. This is a strict criterion that might be considered as gathering the cases for which the automation of the diagnosis appears more easily accessible.

We measured the stability of each rule for each student. One of our concerns was to test the robustness of the analyses carried out by the construction of indices on the systematic character of the use of the identified rules, being quite obvious that the diagnoses related to the protocols have interest if one observes a certain stability of the behaviours. The stability is calculated by a ratio between the frequency of use of the rule and the total frequency of cases in which it could be used.

3 Manual diagnoses

3.1. Results with 9th graders

We analysed in a systematic manner the protocols of 73 students (3 classes from a Parisian school) solving the 31 exercises of the first experiment. We organized experimental settings in small groups (from 8 to 10 students) and with as many sessions as necessary in order to solve the whole range of exercises. The list of the exercises, as well as the frequency of success, are provided Table 1.We then performed manual protocol analyses: all in all, 104 rules were identified, and gathered within a typology: (i) Power rules (P1 to P10); (ii) Priority of operators rules (O1 to O11); (iii) Factorization rules (F1 to F10); (iv) Distributivity rules (D1 to D18); (v) Sign rules (S1 to S12); (vi) Elimination of the coefficient of the unknown rules (Eu1 to Eu18); (vii) Elimination of the fixed value rules (Ef1 to Ef16); (viii) Calculation rules (C1 to C9)

Table 1. List of exercises and frequency of success for the first experiment

	Exercises		Class1/21	Class2/25	Class3/27	Sum /73
1	$5x^2+3x-7-3x2+2x+8$	Simplify and order	18	20	20	58
2	(-3-6)*(6-8)	Calculate	14	19	23	56
3	x+2=-3	Solve	20	24	24	68
4	7x+(2x-8)-(-3x+12)	Expand, simplify and order	13	12	17	42
5	(-2)*(-5)*(+3)+(-2)*(-4)	Calculate	20	24	24	68
6	9-x=12	Solve	17	17	16	50
7	2-3(-5x-5)+5(4x+8)	Expand, simplify and order	17	13	12	42
8	4x=16	Solve	20	18	26	64
9	8a+8b	Factor	21	22	23	66
10	5x=9	Solve	20	19	22	61
11	7x(3x+5)	Expand, simplify and order	18	17	17	52
12	8a+40	Factor	21	22	22	65
13	(9x-5)(-6x+2)	Expand, simplify and order	14	8	11	33
14	$12x^2-7x$	Factor	19	21	23	63
15	8x-4=3x-2	Solve	14	19	15	48
16	$10x+1-6x^2+5-3x^2+6x-6$	Simplify and order	19	15	19	53
17	-9*(-2)-7*(-6+2)	Calculate	17	13	15	45
18	10+x=-8	Solve	21	18	24	63
19	9x-(-4+5x)-(5x+10)	Expand, simplify and order	16	7	12	35
20	5x=25	Solve	19	23	26	68
21	4/3+7/6	Calculate	20	23	24	67
22	4x(-1-7x)	Expand, simplify and order	20	11	16	47
23	x/3=-7	Solve	20	16	23	59
24	2/5-1/7	Calculate	17	17	25	59
25	$10(-4x-1)-2(4x^2-6)$	Expand, simplify and order	18	12	16	46
26	-8=-7x+5	Solve	12	14	16	42
27	-10/9*-6/-5	Calculate	16	14	17	47
28	(1+5x)(2x-3)	Expand, simplify and order	13	11	17	41
29	-2x+8=3+2x	Solve	12	11	10	33
30	2-5*5-7*3	Calculate	17	12	10	39
31	7x=4/5	Solve	15	17	10	42

Thus, we achieved our goal of identifying a large set of incorrect rules that are used by students solving algebra exercises. In this sense, the usage analysis appeared to be very informative. However, if this first experimentation made it possible to identify a library of rules, the analyses of the associated numerical indicators described above, which is not detailed hereby, revealed that the stability was often quite low. We reached the conclusion that because of the variety of the exercises, the identification of the context of application of the rules is made dubious, and some indices of non stability of the rules might in fact reveal different contexts of application: conditions that seemed strictly equivalent from the expert point of view were not for the students. So it seemed necessary to carry out a more systematic analysis on a subfield, as discussed above. In order to avoid ceilings effects related to a too high level of the students, this experimentation was carried out with grade 8 students, after they already studied the resolution of linear equations (ax+b=cx+d type).

3.2 Results with the 8^{th} grades

Table 2 indicates the list of exercises as well with the frequency of success (the numbers are not always integer numbers since some intermediate 0,5 mark were attributed in some specific cases).

Table 2: List of exercises and frequency of success for the second experiment

	Exercise	Type	Class1/30	Class2/30	Class3/30	Sum/90
1	-1/4 x=6	Solve	4,5	1,5	14	20
2	$\frac{x}{7}=3$	Solve	17	19	26	62
3	7=28x	Solve	8	6	15	29
4	-4x=-27	Solve	12	6	21	39
5	12-6x=-15x-3	Solve	11	6	16	33
6	8x-11=7+10x	Solve	12	6	10	28
7	2=-x+15	Solve	20	15	18	53
8	-9=x-7	Solve	9	9	14	32
9	11-x=-12	Solve	13	11	11	35
10	-x+2=7+x	Solve	12	4	13	29
11	-3+2x=-2x-2	Solve	15	10	16	41
12	$-\frac{7}{2}x=5$	Solve	8,5	1,5	8	18
13	$\frac{3}{8}x=4$	Solve	10	3,5	15	28,5
14	$9x=\frac{27}{2}$	Solve	9	4,5	13	26,5
15	$-11x=-\frac{22}{2}$	Solve	8	4	16	28

We used the same typology of rules than with the previous experiment. Due to the systematic manipulation of the factors that might influence the choice of the rules, we were able to identify, for each student, the context of application of the rule and it’s stability.
Table 3 and Table 4 are extracted for the tables gathering the data from the students. Table 3 indicates, for a sample of the participants, the numerical indicators that were performed: percentage of rules identified relative to the number of exercises that the student got wrong, predictability relative to the total number of failures, and predictability of the identified rules (2^{nd} column divided by first column). Table 4 indicates, for a sample of the participants, the total number of incorrect rules identified for a given student and the degree of stability of these rules.

Table 3. Numerical indicators regarding rate of incorrect rules and rules predictibility

Student	Rules/Failures	Predictibility/Failures	Predictibility
Beck	62,50%	37,50%	60,00%
Bert	40,00%	40,00%	100,00%
Beri	42,86%	42,86%	100,00%
Bong	50,00%	36,36%	72,73%
Dasi	65,22%	52,17%	80,00%

Among the failures, we were able to identify an incorrect rule for 58% of the failures in average. The cases in which no rules were identified concern mostly calculation errors and unachieved exercises. The average rate of predictability among the rules identified is 67%.

Table 4. Stability of the identified rules.

Student	Stable	Intermediate	Non stable	Total
Beck	1	2	3	6
Bert			1	1
Beri	1	1		2
Bong	2	1	1	4
Dasi		4	1	5
……	……	……	……	……
total	46	162	152	360

If the rate of full stability (the rule is used in 100% of the cases) appears to be quite low (13% of the total), partly due to calculation errors, 45% show intermediate stability (the rule is used in at least 50% of the cases), which might show competition between the correct rules and the ones taught in school, as well as the use of opportunistic strategies that we identified with grade 9 students as well and which consist in using in an ad hoc manner a rule that make the problem simpler; the opportunistic rule appear to have a low stability because their context of application is not captured by the manipulated factors.

4. Local automatic diagnosis

The purpose of the local diagnosis is to automatically find a sequence of rules (correct or incorrect) that explains a transformation made by a student (e.g., 7=28x $\rightarrow$ x=28-7). The term "local" is used because we consider only one transformation at this point. Such diagnosis is achieved to be followed by other automatic treatments: (1) Calculation of the frequencies of incorrect rules used by a student or a class; (2) Attribution of conceptions to students, conceptions being more global representations of the students' knowledge, see details in [9]. We only develop the local diagnosis work in the rest of this paper.

We have implemented formal rules in a computer language and we have implemented an algorithm for providing diagnoses. At the present time, our focus is on the rules that apply to linear (in)equations and the rules for performing expansions and reduction (for other fields, like factorization and fractions, just a few rules have been implemented). As the goal is to automatically diagnose a lot of students' transformations (a class working during 2 hours with Aplusix produces about 1000 transformations to be diagnosed and we have more that 100 classes to study), we did not implement the rules that are very specific and very rare.

We have combined the above cognitive study of the students' productions and epistemic study of (in)equations to produce the rules to be implemented. This led to consider two sets of rule. First, we consider the correct fundamental operations on both sides of the (in)equations (addition, subtraction, multiplication, division) like A = B $\rightarrow$ A+C = B+C. Second we consider the correct *movement rules* that are compiled form of these rules when they are combined with reduction; there are *additive movement rules* like: $A+C = B \rightarrow A = B–C$ and *multiplicative movement rules* like $AC = B \rightarrow A = B/C$ (C?0); in these rules, C is said to be moved from one side to the other; and we consider incorrect movement rules obtained by the following processes: (1) Incorrect (un)change of the sign of the moved expression, like in $4x+5=7 \rightarrow 4x=7+5$; (2) Incorrect (un)change of the orientation of the inequality sign, like in $-4x<7 \rightarrow x<-7/4$; (3) Incorrect operator linking the moved expression to the global expression (e.g., move C from a multiplicative to an additive position like in $28x=7 \rightarrow x=28+7$). The combination of these processes and of the sort of relation (=, <, etc.) and of the orientation (left to right, right to left) lead to more that 1000 rules. We did not implement 1000 specific rules but a general rule with features that correspond to the above "processes". For example, the transformation $4x<7 \rightarrow x>7-4$ is diagnosed as the application of the movement rule with the features: <, LefToRight, NumeratorToNumerator, InitAdditive, FinAdditive, SignChanged, OrientationChanged.

The implemented rules are based on the above cognitive study but some of the rules were slightly generalized and the very specific and very rare rules were abandoned. We have not yet implemented a complete set of rules for the calculation of fractions.

4.1. The algorithm of the diagnosis

The diagnosis algorithm that we have implemented is a heuristic search algorithm of the "best first" type [6]. Such an algorithm manipulates objects or states (algebraic expressions in our case) based on the use of operators (rewriting rules in our case) and uses a heuristic in order to constrain the search; the heuristic being a function that provides a proximity measure between two objects. Initial data are composed of two objects, the algebraic expressions A and B in our case, B being the result of the transformation of A by the student. The algorithm searches a list of operators (correct or incorrect rewriting rules in our case) allowing the transformation of A into B. For achieving this purpose, it builds a search tree, A being included in the root, and develops successive nodes. Developing a node N consists of applying all the rules that are applicable to the object that it contains, and to generate a successor of N each time a new object is generated. Algebra is a difficult domain for this kind of search because of: (1) an important branch factor (number of successors of a node) coming from a large number of applicable rules; (2) the presence of cycles in the application of rules; (3) the absence of a good distance to evaluate the proximity of a produce expression with the target. For these reasons, we had to adapt the general algorithm, in particular, some rules take into account the goal: they are not applied when they are applicable if some conditions regarding the goal are not verified. This is the case of the above movement rule that can be applied for 8 expressions in $2x+3+4x+5=6x+7+3x+2$ and each time with a lot of features, and can be applied as many times in the produced nodes. Such an algorithm sometimes fails. When the algorithm does not reach the target after a chosen number of developed nodes (we often chose 30 nodes), it fails. When the target is reached, the obtained diagnosis can be considered as inappropriate because it makes a bizarre combination of rules when the analyst has a better diagnosis.

4.2. Results

Here is an example of diagnosis of the transformation of $2x–6 = 7x-8$ into $–5x = –14$. It is diagnosed with 4 rules: (1) Incorrect additive move of 6, leading to $2x = 7x–8–6$; (2) Correct additive move of 7x, leading to $2x-7x = –8–6$; (3) Correct additive reduction, leading to $–5x = –8–6$; (4) Correct additive reduction, leading to $–5x = –14$. The current application of the automatic diagnosis on recorded data of a grade 8 class (29 students) and a grade 9 (21 students) provides a good ratio of "success" presented in table 6. The "appropriateness" ratios of these diagnoses are presented in table 7 for the incorrect transformations of the two families studied (expansions, reductions and transformations on equations).

Table 6. Success of the automatic diagnosis (i.e., when it did not fail).

Class	Type	Number	Success	Ratio
Grade 8	Correct transformations	1070	1005	94%
Grade 8	Incorrect transformations	434	351	81%
Grade 9	Correct transformations	1071	985	92%
Grade 9	Incorrect transformations	155	121	78%

Table 7. Appropriateness of the automatic diagnosis for incorrect expansions, reductions and transformations on equations (i.e., when it is judged appropriate by the analyst).

Class	Type	Number	Appropriate	Ratio
Grade 8	Incorrect transformations on equations	78	76	97%
Grade 8	Incorrect expansions and reductions	126	103	82%
Grade 9	Incorrect transformations on equations	33	28	85%
Grade 9	Incorrect expansions and reductions	50	46	92%

5. Perspectives

We are now working on the process of improving the quality of the automatic diagnosis through the implementation of other fields (e.g., fractions, factorization) and iterative comparisons with the results of the manual diagnosis. The diagnoses produced by this process are used in another part of our project devoted to the production of conceptions, conceptions being more global representation of the knowledge of the student. This work is presented in another communication of the workshop.

Besides providing libraries of incorrect rules and of conceptions to the scientific community, our results will be used later for two purposes: (1) to calculate students' conceptions in the Aplusix system and present them to the teacher; (2) to build artificial tutor devoted to remediation of inadequate conceptions.

References

[1] Brousseau G. (1997). *Theory of didactical situations in mathematics*. Kluwer Academic Publishers, Dordrecht.

[2] Fischbein, E., (1989). Tacit models and mathematical reasoning, *For the Learning of Mathematics, 9*, 9-14

[3] Nicaud, J.F., Bouhineau, D., and Chaachoua H. (2004). Mixing microworld and CAS features in building computer systems that help students to learn algebra, in *Int. Journal of Computers for Mathematical Learning*, Volume 9 (2), Springer-Verlag.

[4] Nicaud J.F., Chachoua H., Bittar M., Bouhineau D.. (2005) Student's modelling with a lattice of conceptions in the domain of linear equations and inequations. AIED Workshop on Usage Analysis in Learning Systems. Ibid.

[5] Payne S. and Squibb H. (1994) Algebra Mal-Rules and Cognitive Accounts of Error. Cognitive Science, 14.

[6] Pearl J. (1984). *Heuristics*. Addison-Wesley, London.

[7] Sleeman D.H. (1983). Inferring student model for intelligent aided instruction. In Michalski, Carbonell and Mitchell (Eds) Machine Learning: An Artificial Intelligence Approach. Morgan Kaufman.

[8] Tirosh, D., & Graeber, A.O. (1991). The influence of problem type and primitive models on preservice elementary tearchers' about division. *School Science and Mathematics, 91,* 157-163.

[9] Modélisation cognitive d'élèves en algèbre et construction de stratégies d'enseignement dans un contexte technologique. Project report of the "Ecole et sciences cognitives" research programme. Cahier #123 of the Leibniz laboratory, http://www-leibniz.imag.fr

Analysis of the Usage of the Virtuoso System

Krassen STEFANOV, Eliza STEFANOVA
Faculty of Mathematics and Informatics, University of Sofia "St. Kliment Ohridski"
1164 Sofia, James Bouchier Str. 5, Sofia, Bulgaria
krassen@fmi.uni-sofia.bg, eliza@fmi.uni-sofia.bg

Abstract: We present our experiences for recording and analyzing usage of the Learning System Virtuoso at the University of Sofia. The description follows the structure that emerged from the joint work on the Kaleidoscope DPULS Project meetings held on February and April 2005. We also show one possible generalisation and one example pattern for tracking the learning problems described.

1. Introduction

Technology enhanced learning is seen as a strategic tool for enhancing the quality of education. One of most promising technologies in this field are Learning Content Management Systems (LCMS). There are lot of efforts to develop and use such systems, but the results so far are not very optimistic. So, we need to perform very careful and systematic analysis of this phenomena, in order to identify the weaknesses of current LCMS and their usage. One possible research approach is to track the usage of the LCMS, to analyse the traces, to find the more important gaps and problems, and to propose solutions which can improve the results of the usage of the LCMS.
The DPULS Project [1], part of the Network of Excellence Kaleidoscope, is trying to use such an approach. More specifically, the objectives of the DPULS are:

- to collect experiences in order to identify recurrent tracking problems and solutions
- to capitalise knowledge and experiences in order to provide instructional designers with the means to deal with tracking students' activity and analysing recurrent problems
- to create a structured set of Design Patterns for recording and analysing the usage of e-learning systems

In this article we will present our work on recording and analyzing the usage of the Learning System Virtuoso at the University of Sofia. This work was inspired from our participation in the first two workshops of the project DPULS. The description of this case will follow the common structure that emerged from the joint work on these two workshops held on February and April 2005. At the end a generalisation of the experience is proposed, and one example pattern for tracking the learning problems is given.

2. The Context of Use of the System

Virtuoso [3] delivers e-learning to more than 450,000 students enrolled in over 10,000 schools in 150 countries worldwide. Virtuoso is an innovative e-learning platform encompassing three modular components for authoring, delivering, and managing dynamic, Web-accessible learning content. Virtuoso is a complete Web-based e-learning platform with focus on assessment, collaboration, interactivity and personalized feedback. Virtuoso puts reusability into practice with repositories for course content, assessments, documents, and media. Once created, repository content can be dynamically searched, reassembled and reused, saving time and efforts. We are using Virtuoso as a regular system for the delivery of 15 different courses at four faculties from the University of Sofia, starting with introductory courses (year 1 and 2 students) and ending with MSc courses.

2.1. Users Profiles

When working with the Virtuoso system, we can distinguish the following main users' profiles:

- Learners:
  They can access all courses in which they are enrolled, can work with all learning materials and resources, can take all planned assessments and exams, can participate in collaborative activities, etc.
- Designers:
  They develop and constantly update courses and course materials, assessments, exercises. They are following the quality issues regarding the courses delivery, and are trying to respond to the issues raised by teachers/tutors and learners.
- Teachers/tutors:
  They manage individual courses by starting different learning events, assessments and exams. They monitor the learners' progress, provide tutoring support and consultations to students, and form the final grade for any students in the class. They also analyse all the tracked information from their classes as well from the Virtuoso system, and on the base of the obtained results make decisions for the improvement of the teaching process.
- Administrators:
  They provide all the administrative information into the system: create students and teachers accounts, create classes, enrol teachers and students to the classes, and provide any additional administrative support to the teaching process.

All Virtuoso users can participate in various communities of practice worldwide, created and supported through Virtuoso system. These communities of practice are based on the user profile, on the course subject domain, as well as on the features defined by the users themselves. All members of a given community of practice can communicate through individual emails, discussion boards, help centres and best practice case studies.

All students can use the system from a distance, as well as from the on-line labs at the University of Sofia. Face-to-face sessions are exceptional. As a general rule the students work individually on the learning materials or collaborative sessions from a distance, and participate in practical exercises and assessments in the on-line labs with the presence of the teachers. But the students can work with the learning materials in the labs together with the teachers, as well as they can make practical exercises from a distance.

2.2. Design Model

From the pedagogical point of view, the system is following the standard behavioural instructional design model of learning, based on the performance objectives formulation and the appropriate assessment procedures development. From the technological point of view, the system is built around the concept of Learning Object, and is fully compatible with the IMS learning standards and specifications [2]. The Virtuoso follows the Cisco model for Reusable Learning Object (RLO) and Reusable Information Object (RIO). The RIO contains content, practice, and assessment components. Figure 1. presents Concept

RIO Template . Each RIO is defined as a concept, fact, process, principle or procedure, and tagged appropriately.

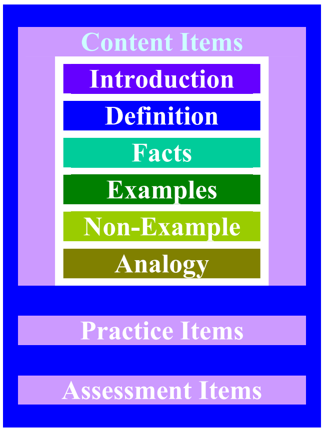


Figure 1. Concept RIO Template

Several RIOs (between five and nine) are combined together to create a RLO. Each RLO, which also includes introduction, summary, and assessment items, is designed to meet a learning objective derived from a specific job task. RLO is the sum of RIOs needed to fulfil that objective.
A RIO can function as an independent performance support aid that can be called up by a learner who needs a specific piece of information or a more in-depth learning experience. RLOs can be sequenced to create a full course on a particular subject. RIOs can be combined together to build custom RLOs that meet the needs of individual learners.
The Cisco strategy aims for creation of personalized learning paths showing each learner which RIOs they've experienced and which are recommended to get them to the desired skills level.
All courses delivered through the Virtuoso system are in the domain of Computer Science – introduction Information technologies courses, basic courses in programming and operating systems, and most courses in computer networks.

3. What Data is Collected for the Analysis and Why

Course designers need the collected and analysed data to improve and make more clear the presentation of the concepts, to improve the assessment, and to improve the practical exercises.
The teachers and tutors need the data to have information about regulation and learning assessment. Knowing where most students face problems from the analysis of the assessments and performance data from previous courses, the teacher/tutor can stress more during the current course delivery on the most difficult for the learners' topics and practices. From the analysis of each particular student's history the tutors can provide the right explanations and give individual advice to each particular student how to proceed further in the course.
Learners can access their history and assessments, and can reflect on the mistakes they have made and on the practice exercises the have failed.
Below we give examples of how data is stored, analysed and used.

3.1. Assessment History

As assessment history we consider: all questions to tests, errors, time and date when assessment was performed, etc. Several examples of these data are:

- Progress of each student (tests passed):
  Data about the progress of each student (tests passed) during the concrete course are collected into the class grade book.
- Errors on every question:
  Errors on every question for each test the teacher can view the detailed picture. Errors on every question are marked with earned by student points (showing that they are not maximum points) in the row of each student. Columns of errors (numbers) formed under some questions show that either topic was not well explained in the course materials, or the question was not clear for the students. Instructors and students can view for each error the correct answer and the wrong answer given by the learner.
- Time and date of assessments:
  The list of all past test activations is available with time and date when assessment was active. For every test's activation the learning system recorded the assessment history data for time and date when assessment was performed.

3.2. Practical Exercises History

As practical exercises history we consider: sequence of steps to achieve a solution, errors made, time consumed, score achieved, help consumed, etc. Several examples of the data from this type are:

- Time reserved and time consumed:
  Time reserved and time consumed for any practical exercise time reserved and time consumed is available through the system. For all students system store information how many times she/he logged to the system, and the time of last login.
- Activities performed
  Activities performed for each student system can show what activities were performed (from the sessions log file) and the final result (configuration files) after activity was completed. When reservation for practical activity was performed the instructor, depending on system configuration, could receive automated e-mail with report on the student activity. This report shows sequence of steps to achieve a solution as well as errors made and is attached as a zip archive with several log files to the automated email.

3.3. Records of Participation in all Community of Practices Events

As community of practices events we consider: discussions, questions, answers, opinions, etc. Example for data from this type is:

- Teachers and learners have access to all the discussion forums for which they are eligible, and also to the full Help desk archive containing all important help advices given to the learners and tutors.

3.4. Additional Data

As additional data we consider: scenarios for course presentations, exercises and course delivery time schedules on the FTP server, personal data for each learner and teacher (name, age, preferences, etc.), metadata (mostly available for teachers, regarding courses and Learning Objects), contextual data (number of students, number of courses, etc.).
Virtuoso system collects lot of information for further analysis and use, but there are still important information not stored from the system– how long the student read the course materials, and what time was needed for getting with each Learning object.

In general, all actors in the learning process have the access to the appropriate data stored from the system. The system not only store a lot of data tracking the users performance, but also provides very rich set of statistical analysis of almost all kind of data stored. So each actor can have different view on the data stored, depending of the needs and the wishes of the user:

- Teachers analyse the huge amount of available statistics on students performance (answers to test questions, solutions to practical exercises, etc.) to produce the individual performance report for each student and to use it for student's grades, identifying strong and weak points of the students performance, and to plan the future learning sessions. They need to know which test questions are most difficult for the most of the students, and to stress during the preparation for the tests on the appropriate examples and explanations. They can use the data from the student's participation in the communities of practice, in order to better understand their wishes, needs and problems, and to plan the individual session in the most appropriate way.
- Learners analyse the results from their performance in order to identify their own weak points and to know what additional learning they need in order to surmount the difficulties they met. On that base they plan their further learning activities.
- Designers, after careful analysis of the statistics available, can identify which test questions seem to be not adequate (either the text is misleading, the question seems to be very difficult, or question seems to be too easy), which learning objects are not well presented, and which practical exercise is either too difficult or too easy for the students. They use the results of this analysis to readjust pedagogical scenarios – to change the objectives, to change the course design, to change learning objects, to change assessment, etc. They can revise the identified not adequate learning objects, test questions or practical exercises.

Each actor, depending of his goals, can use the data stored at different time and on different regularity:

- Usually designers analyse the data at regular time intervals in order to identify if they have to change the course design and delivery.
- Teachers and tutors generally analyse the results before the sessions in order to prepare better for each individual session.
- Learner usually analyse the results after the sessions in order to adjust their learning according to their own performance.

4. Generalization of our Experience

In order to be able to generalize, we have to describe our context in more abstract way. We can identify the following important features:

- learning content management system in which all learning activities are represented as Learning Objects (LOs)
- assessment system, in which it is possible to assign several different assignments to every single LO
- statistical system which can calculate the mean scores for every single student, group of students or for all students regarding given course, every single LO, every single assessment, etc. This system should make it possible to calculate automatically some basic statistical variables used to make conclusion in statistical analysis.

This context is typical for any full featured based on the standards Learning Content Management System. Tests have to be based on the IMS QTI specification. Practical exercises and user data tracking are also typical components for a standard learning environment based on the constructivist approach. The other contextual data (learner details, metadata, etc.) can also be easy found in a standard LCMS.

4.1. Tracking Problems

At a general level we faced and solved the following most significant tracking problems:

- Monitoring participation and the level of discussion in CoP events
- Analyzing learners' performance to identify:
  - Very difficult topic
  If for a given Learning Object (LO) all students score significantly lower than for other LOs, this indicated that the LO is very difficult for them (could be explained badly, or not appropriate, or not well supported by other needed input LO).
  - Very easy topic
  If for a given LO all students score significantly higher than for other LOs, this indicated that the LO is very easy for them (could be not appropriate, or not well linked with other LOs).
  - Wrong question / assessment
  If for a given LO one of the assessments (test question, exercise, etc.) scores for almost all students differently than other assessments for the same LO.
  - Student weak point
  If for a given LO student scores significantly lower than for other Los.
  - Student strong point
  If for a given LO student scores significantly higher than for other Los.

4.2. General Solutions

On the base of our experience we can suggest the following general solution/recommendation to people who could face the same problems:

- Use powerful and user-friendly statistical package for the analysis of tests and exams results in order to identify easy/difficult course topics, details regarding the learner's model, and what tests/content/exercises seem to be not adequate.
- Use tracking of learner's solutions to practical tasks to identify common performance gaps and to individualize the future learner's path.

4.3. Positive and Negative Effects

In general, we can point the following positive and negative effects:

- When learners have their own view on the tracking data together with recommendations for improving their knowledge and performance, they feel more confident and achieve better results.
- When tutors and teachers can analyse the statistical data from huge number of learners, they are sure what decision to take. From the other side, the use of statistical data coming from different contexts may lead the teachers to wrong predictions and conclusions regarding their own students.

5. Example Tracking Design Pattern

On the base of the analysis and generalization presented above, as well as on the analysis of the state-of-the-art in the field of design and usage learning patterns ([4], [5], [6], [7], [8]), we are proposing one example structure and description of one learning usage tracking pattern:

Problem domain: Course usage

Problem name: Very difficult course topic

Problem symptoms: Learners have problems to achieve the needed knowledge or skills regarding some particular topic of learning. These symptoms can be seen from discussions, practical exercises, questions raised to the help desks, assessment scores.

Contextual information: We have a learning system in which all learning activities are represented as Learning Objects (LOs). We also have assessment system, in which we can assign several different assignments to every single LO. We also have some statistical system which can calculate the mean score for every single student, group of students or for all students regarding given course, every single LO, every single assessment. The system is able to calculate automatically some basic statistical variables used to make conclusion in statistical analysis.

Problem statement: There is a LO which is causing problems to the learners.

Problem identification: If for a given LO (for all assessments) all students score significantly lower when compared to other LOs, this indicated that the LO is very difficult for them.

Used indicators:

In order to identify the problem, we need to measure, calculate and compare the students' results and performance. The Virtuoso system can give to use two sets of indicators to use – quantitative (direct scores from assessments) and qualitative (in the form of metadata descriptors of the learning objects and the learning process in general). The quantitative indicators are direct scores and calculated scores:

- direct score: an integer in the range 0-100 (higher the number, better the score) assigned to each individual assessment of a single student regarding one particular LO
- calculated scores: integer values calculated on the base of the basic scores using standard statistical techniques. Example for such a score is the mean score for all students for all available assessments regarding particular LO. The system Virtuoso is offering very rich set of calculated scores, and even the ability for the user to define its own calculated score.
- meta-data for the LO (difficulty, prerequisites, technical prerequisites, etc.). They are used for the human analysis of the calculated scores.
- meta-data for the course or set of learning activities (list and sequence of LOs, their possible correct order, etc.) They are also used for the human analysis of the calculated scores.

Possible reasons:

- Reason (1) - this LO could have very low quality (bad explanation caused by wrong design, wrong examples caused again by the wrong course design, very low technical quality caused by the wrong design or the error in the learning system, etc.)
- Reason (2) - this LO could be not appropriate for the current course or learning (designed for MSc level but delivered at BSc level for example) – if the needed LO was not available and was substituted with what is available, or if there is an error in the learning system causint the delivery of not correct LO.
- Reason (3) - this LO could have a list of prerequisite LOs and some of them is not available in the current course (wrong design or error in the learning system)

Solution:

The solution depends on the reason. First we have to identify the reason. (2) and (3) can be checked by examining the appropriate meta-data. If (2) and (3) are not the case, than we have (1).

- Solution for (1):
  The course designer has to re-design this particular LO. In the mean time the teacher/tutor have to plan some support activities in order to help the students in understanding the concepts/skills covered by this LO (additional explanations, special group session, additional practical sessions, etc.)
- Solution for (2):
  Change the LO with the right one, or if not available, apply the solution from (1).
- Solution for (3):
  Find the missing LO and re-arrange the learning. If the missing LO is not available, we have to plan again supporting activities.

Sketch or Diagram: Not available
Links with other to consider:

- It could be possible that the LO is OK, but all assessments for this LO are wrong.
- It could be checked if the learning system has some error which is causing the bad quality of the delivery of this particular LO.
- It could be checked if the learning system has some error causing one of the prerequisite LO not to be delivered to learners.

6. Conclusion

This case is one among several different cases developed and studied during the DPULS workshops. They mark the beginning of the joint research targeting the design and development of the set of Usage Patterns for collecting the experiences of using Learning systems and identifying common generic tracking problems and solutions.
The next steps are linked with experiments for transferring the methods described for other LCMS and collecting the results. This will help us to show how general is our approach and how it could be applied in different learning settings.
Another important step forward is linked with more systematic analysis of the situations described, in order to identify as rich set of possible design patterns as possible. On the base of this analysis, we can combine our research outcomes with the findings of our DPULS project partners from other learning situations, thus going further to the creation of a structured set of Design Patterns for recording and analysing the usage of e-learning systems.

7. References

[1] DPULS: Design patterns for recording and analysing usage of learning systems - description of work document, NoE Kaleidoscope, 2004.
[2] IMS Global Learning Consortium, http://www.imsglobal.org/.
[3] Virtuoso system, Cisco Learning Institute, http://www.ciscolearning.org/.
[4] Joseph Bergin. A Pattern Language for Initial Course Design. ACM SIGCSE 2001 2/01 Charlotte, NC USA. http://csis.pace.edu/~bergin
[5] Jutta Eckstein. Learning to Teach and Learning to Learn. Workshop at EuroPlop 2000 in Kloster Irsee, Germany. http://www.pedagogicalpatterns.org/examples/LearningAndTeaching.pdf
[6] Frans Ronteltap, Peter Goodyear, Sonia Bartoluzzi. A Pattern Language as an Instrument in Designing for Productive Learning Conversations. EDMEDIA'04 Symposium: Design Patterns For Elearning. http://www2.tisip.no/E-LEN/papers/EDMEDIA04/Paper_EM2004_Ronteltap_Goodyear_Bartoluzzi.pdf
[7] P. Avgeriou, A. Papasalouros, S. Retalis, E. Skordalakis. Towards a Pattern Language for Learning Management Systems. Educational Technology & Society, ISSN 1436-4522, Volume 6, Issue 2, pp. 11-24, 2003. http://ifets.ieee.org/periodical/6-2/2.html
[8] Sherri S. Frizell and Roland Hubscher. Aligning Theory and Web-based Instructional Design Practice with Design Patterns. E-Learn 2002. http://www.auburn.edu/~frizess/frizell-elearn.pdf

Recording and Analyzing usage of Learning Systems: experience and reflection with an on-line course on Java Programming

Patrick Duval, Agathe Merceron, Michel Scholl
Department of Computer science,
ESILV, Pôle Universitaire Léonard de Vinci, F-92916 Paris La Defense Cedex
Patrick.Duval, Agathe.Merceron, Michel.Scholl@devinci.fr

Abstract. In this poster we summarize an experience of data collecting and analysis with an on-line course on the programming language Java following the format proposed in the DPULS project [3].

Introduction

There is a whole body of experiences with Technology Enhanced Learning. There is also a need to learn from these experiences for all actors involved in this field. A way to answer this need is to gather experiences, to structure them and make them available with appropriate tools. Various projects cope with this issue, among others the DPULS project [3] inside the European Network of Excellence Kaleidoscope [6]. The DPULS project is concerned with usage of learning systems. The approach that it has adopted is to gather, structure and make available experiences on usage of learning systems based on a Design Pattern approach.
The aim of this poster paper is to summarize about our experience in this area. We finally identify and make suggestions for two general problems not already tackled in this experience: diagnosis of individual students and grouping students by abilities.

Experience

Our learning system consists of a distance course "Introductory programming with Java" accessible through the Ganesha platform [5]. More details on this system can be found in [2]. Though still experimental, this system has been used in our Engineering school by students in their second year of studies. The Java course is offered as an extra resource to a face to face, and as a distance course to catch up during summer. It is based on real programming exercises, where students have to complete real code. Each chapter, covering a specific concept, starts with an exercise serving the purpose of understanding, following exercises asking students to use the newly acquired concept. Students receive immediate feedback to their code submissions, their code being automatically evaluated by configurable components, applying vector tests specified by authors. Compilation and execution are transparent for learners. This course has been designed for students to acquire programming skills and knowledge on Java, not for marked assessment.

The design of the course and of the exercises uses ideas of cognitive load theory: the "learning by doing" approach encourages the construction of schemas to help students for solving problems; completion exercises reinforce this construction, see [7].

The Java course records all students' answers to exercises, including mistakes, in a database. We mine this data to extract pedagogical information and produce reports of interest for learners, teachers, and tutors. Tutors and teachers run queries and diagrams as students do exercises. This allows tutors to know exactly where students are in the completion of their work, what kind of difficulties are encountered, and how students try to get to the solutions. Then tutors can be pro active towards students. Another benefit for teachers is to be aware of the adequate difficulty level of exercises. Additional primitive data that we intend to collect are:

Vector of tests. Which test vectors failed when evaluating an answer.

Programming style. Some of the test vectors could evaluate programming styles as in [4].

Code execution time. Might give hints about the efficiency of the code.

Course material. Which course items have been consulted before answering.

Defining skills and concepts

Mining this raw data does not necessarily provide a good picture of the skills or concepts mastered by each learner. Following the stereotype idea explored in the Pepite software [1], we think that it would be interesting to define a set of concepts and skills that students have to master when completing the course and to be able to produce a picture of students' progress according to this set. At this end it is necessary (i) to map each exercise, or even each test of each exercise, to an element of this set, (ii) to propose a method that calculates an indicator describing skills or concepts acquired by students from his/her answers to exercises and tests.

We see several benefits in such an approach. The first one concerns *remedial measures* in cases where indicators show that some skills are not mastered. The second benefit concerns the general problem of *how to create an individual diagnosis for students?* Diagrams focusing on acquired skills and concepts could be drawn for each student. A third benefit concerns the general problem *of how to be aware of the adequate difficulty level of a concept or skill?* Finally, a last benefit concerns *the general problem of grouping students by abilities*. This can be done using data mining techniques such as hierarchical or k-means clustering. A natural way to use these techniques would to consider all the results to all exercises as the variables or attributes for the clustering. However, when this number gets large relative to the number of students, there is a curse of dimensionality effect and no reasonable groups can be found with these techniques. For these techniques to work, it is necessary to reduce a priori the number of attributes. Defining a set of skills and concepts appears as a promising approach.

References

[1] Delozanne El, Vincent C. “Stereotypes in pepite”, private communication, 2004
[2] Duval, P., Merceron, A., Scholl, M., Wargon, L., “Empowering learning Objects: an experiment with the Ganesha Platform”. ED-MEDIA 2005, Montreal, Quebec, June 27-July1, 2005.
[3] DPULS project http://www.noe-kaleidoscope.org/pub/patterns/index.html
[4] S. Foubister, G. Michaelson, and N. Tomes, “Automatic assessment of elementary standard ML programs using Ceilidh”, Journal of Computer assisted Learning, 1996
[5] Ganesha www.anemalab.org, October 31, 2004.
[6] Kaleidoscope http://www.noe-kaleidoscope.org/pub/
[7] Sweller J., van Merrienboer J.G. and Paas F.G.W.C. “Cognitive architecture and Instructional Design”. In: Educational Psychology Review, Vol. 10, No. 3, pp. 251-295, 1998.

Usage Analysis for Studying Scientific Discovery Learning Processes: Results from Three Different Learning Environments

Casper D. HULSHOF
Faculty of Behavioural Sciences, Department of Instructional Technology
University of Twente
The Netherlands

Abstract. Scientific discovery learning is an action-based learning method that stresses experimentation and hypothesis testing. Usage analysis can be an effective tool for understanding the particular learning processes involved, which may lead to ways of actively regulating it. Here, usage analysis has been applied to three different learning environments that we have studied before: FILE, Bubbles, and Optics. We show how a generic measure of learning efficiency improves the ability to compare different learning environments, to predict and model learner behavior, and to create guidelines for supporting scientific discovery learning.

1. General background

Interactive learning fosters learning processes that are hard to obtain with traditional methods. A type of instruction that makes good use of computer-regulated interactive learning is *scientific discovery learning* [1]. In scientific discovery the content of a conceptual domain is not explicitly stated to students. Instead, they 'discover' the material as a scientist would. Scientific discovery learning can be elicited using computer simulations [2]: virtual environments in which students can design and perform different types of experiments, observing the effect of manipulating variables. In a simulation, students' task is to induce the relations between variables in the simulated domain through exploration, experimenting, and discovery. Discovery learning has been criticized because it is a difficult way to learn. To examine the difficulties students experience during learning, usage analysis (in this case on logfiles) can be a fruitful approach [3].

2. Aim

Our aim was to devise a generic measure for comparing learning with three different simulations: FILE, Bubbles, and Optics. The advantage of this is that it allows for a better comparison of learning styles between different learning environments and that we can use results from one learning environment to predict behavior in another environment. Also, generic usage measures foster modeling of discovery learning processes and creating guidelines to support learning. All measures were extracted from students' logfiles through an automated analysis, driven by a set of Perl scripts.

3. Usage analysis of FILE, Bubbles, and Optics

FILE ('Flexible Inquiry Learning Environment') is used to study inquiry learning in a relatively simple environment [4]. There are only a limited number of 'experiment states'

(48 in total). Exploring all experiment states should be sufficient to induce all rules that govern the outcome of experiments. However, students usually do not explore all states. So what *do* they do? To answer that, students' actions were registered while they worked with the task. The results show that students with lower general abilities explore fewer experiment states. From the relation between the total number of experiments students perform in the task and the number of different experiment states they cover a generic measure of *learning efficiency* can be derived. Learning efficiency is the number of experiment states explored, divided by the total number of experiments performed. In FILE, mean efficiency for low ability students is similar to that of high ability students.

Bubbles simulates a fictitious chemical reaction that takes place when certain conditions are met [5]. Bubbles uses continuous variables, but analysis of the environment results in 38 different experiment states. Learning efficiency for students with relevant prior domain knowledge appears to be higher than that for students with no prior knowledge.

Optics simulates the principles that govern image formation properties of lenses [6]. It is dynamic in that there is no demarcation between individual experiments. Work is now underway to make a distinction between experiment states in the Optics simulation and to apply the measure of learning efficiency to it. It is predicted that a problem in learning with Optics is that students explore only a minimum number of experiment states.

4. Conclusion

The analysis of three learning environments of varying complexity led to the derivation of a common measure that has a number of advantages over previous analyses. It allows for a direct comparison between learning environments, which helps prediction of learning behavior in one environment based on results from another. We found it useful to do a thorough analysis of the learning environments themselves, even before a usage analysis is undertaken. It is hoped that the efforts described here, and those recently undertaken by the 'Interaction and Collaboration Analysis Supporting Teachers and Students' Self-regulation' JEIRP (part of the Kaleidoscope NoE) and its successor 'Interaction Analysis', contribute to finding and utilizing commonalities in different approaches to learning and teaching, in order to further improve usage analysis. With respect to scientific discovery learning, the result should be a set of guidelines on how to support interactive learning processes while maintaining the freedom of action associated with them.

5. References

1. De Jong, T., & Van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68,* 179-201.
2. De Jong, T. (1991). Learning and instruction with computer simulations. *Education & Computing, 6,* 217-229.
3. Hulshof, C.D. (2004). Log File Analysis. In K. Kempf-Leonard (Ed.), *Encyclopedia of Social Measurement* (pp. 577-583). Elsevier.
4. Hulshof, C. D., Wilhelm, P., Beishuizen, J. J., & Van Rijn, H. (2005). FILE: A Tool for the Study of Inquiry Learning. *Computers in Human Behavior, 21,* 945-956.
5. Hulshof, C. D. (2001). Discovery of ideas and ideas about discovery: The influence of prior knowledge on scientific discovery learning in computer-based simulations. Unpublished Doctoral dissertation, University of Twente, The Netherlands.
6. Hulshof, C.D., & De Jong, T. (Submitted). *Using Just-in-Time Information to Support Scientific Discovery Learning in a Computer-based Simulation*. Submitted to Journal of Experimental Education.

Acknowledgement

The work described in this paper is related to the Jointly Executed Integrated Research Project 'Interaction Analysis', part of the Kaleidoscope NoE.

Analysis of tracks from student's use of a system to teach a problem solving method

Françoise LE CALVEZ[1], Hélène GIROIRE[2], Gérard TISSEAU[2], Jacques DUMA[3]

[1]CRIP5, Université René Descartes, 45 rue des Saints Pères, 75270 Paris cedex 6

[2]Equipe SysDeF - LIP6, Université Paris6, 8 rue du Capitaine Scott, 75015 Paris

[3]Lycée technique Jacquard, 2 rue Bouret, 75019 Paris

Francoise.Le-Calvez@math-info.univ-paris5.fr, Helene.Giroire@lip6.fr, Gerard.Tisseau@lip6.fr, dumajd@club-internet.fr,

Abstract. In this paper, we present the results of the analysis we made about the tracks collected when students use a specific computer supported educational system. This system aims to teach a method to solve combinatorics exercises. The analysis of the tracks enables to make visible some students' abilities or difficulties in order to individualize the teaching.

We present an ongoing study about the analysis of data collected from the usage of Combien? (How Many?) Software [2] [7]. Pedagogical interfaces are proposed to students to help them to learn combinatorics using mathematical language. These interfaces are based on our solving method: the "Constructive Method": to obtain the numeric solution of an exercise, the constructive method consists in building one element of the set of configuration-solutions; this element is defined by a set of exclusive constraints; for each constraint the number of various possibilities can be calculated; then, the solution is the product of all these possibilities (cf multiplicative principle). Each interface corresponds to a class of problems according to their solving schemata. It trains the student to build a solution using the constructive method and it detects the errors incrementally. The first aim of the software is not so much to turn students into counting experts, able to determine the number of elements of a set, as to train them for a modeling task and to make them able to represent a situation by a complex structure. These interfaces, which we call "machines", have been used in different contexts in classrooms for senior high school students and first-year university students.

1. What are the tracks?

When a student uses the software, all the events and inputs are recorded so that the session can be re-played. The tracks are the results of this recording. The tracks are structured according to the model of the solution. When the input is a validation-click the machine tests the validity of the sub-solution, and records the action and possible errors. These data are organized using the "descript" language [4] and are transformed into XML-files. Track analysis is performed from the XML files.

2. The aims and use of data analysis

The exercises assigned to a student who uses the machine for the first time are chosen for their capacity to highlight the student's potential difficulties. The tracks contain relevant information about the solving method comprehension and the underlying mathematical concepts (constructive method, sets, properties, constraints, functions...). Then, the tracks are analyzed according to two directions. On the one hand, general information (here called generalLevel) can be extracted about the session: the name of the used "machine", the number of attempted exercices, the number of completed exercises, the total duration of the session, the order in which the exercises have been processed and for each exercise, its name and the duration of the solving process. On the other hand, specific information (here called domainLevel) can be extracted from the inputs from the learner and the reactions from the system about these inputs: errors and hesitations. The use of track analysis is to give first the students a summary of their work so that they become aware of their abilities or difficulties, and second to improve teaching and enhance learning by the means we present below .

2.1. Personalized courses and exercises

During the solving of the exercise, when the student makes an error, the system displays a message. This message is neither an explanation, nor the solution of the part of the exercise the student is trying to solve. It is composed of hints about the misused concept, to help the student to overcome cognitive conflicts and continue the process. In the track, the name of the error and the error message are recorded. According to their work, students need different explanations and a more personalized help (course, exercises...). At present, we are studying how we can build some explanations based on students' errors, may be by grouping them in more general concepts. For example, in the "SetConstruction Machine", thirteen types of errors are possible, and the underlying concepts are "Universe", "constructive method" and "multiplicative principle".

According to the errors made and to the categorisation of exercises, we can propose some new series of exercises which allow the student to work on his/her difficulties. We have begun such a work in collaboration with LeActiveMath group [1].

2.2. Automatic progression in various machines

Progression from a machine to another is linked to the results of the analysis. Each machine corresponds to a class of problems and the complexity of the concepts increases. For example the Universe in the ListConstruction is a set of functions and the Universe in the SetConstruction is a set of basic elements. The upper machine contains exercises from all classes and the challenge is to choose the right machine to solve the problem.

2.3. Categorization of exercises

For a group of students and each exercise, statistics are performed about the number and types of errors made, its solving duration etc. They allow to assign a value to each attribute of the exercise (difficulty, underlying basic concepts which induce errors, duration if used in limited session...) This allows to categorize exercises and then to personalize the courses.

3. Conclusion

The special feature of "Combien?" is to teach an explicit problem solving method. During its use, it records all the students' inputs, events and error diagnosis in a track structured according to a solution model. These tracks used to define a student's profile or exercise features are very complete. Calculations are made to give the various actors (students, teachers, tutors and designers) an overview of the students' activities and abilities so that the teaching can be modified to become more appropriate. According to the result of the track analysis, some personalized courses and exercises can be proposed, an automatic progression in the use of the various machines can be defined, and types of exercises can be determined.

References

[1] Activemath http://www.activemath.org/

[2] Combien? http://www.math-info.univ-paris5.fr/combien/

[3] Descript http://www.math-info.univ-paris5.fr/%7Elecalvez/combien/descript.pdf

[4] Le Calvez F.,Giroire H., Duma J., Tisseau G., Urtasun M., Combien? a Software to Teach Students How to Solve Combinatorics Exercises ; Workshop "Advanced Technologies for Mathematics Education" AIED 2003, In Supplementary proceedings of the 11th International Conference on Artificial in Education, pp. 447-454, Sydney, July 2003

[5] Surawera P., Mitrovic A., KERMIT: A constraint-based tutor for database modeling. ITS 2002, Biarritz, pp377-387, Lecture Notes in Computer Science, Springer Verlag.

Supporting Distributed Collaborative Learning with Usage Analysis Based Systems

Aloys Mbala

aloys@cs.rmit.edu.au

School of CS &IT
RMIT University
Melbourne, Australia

Christophe Reffay

Christophe.Reffay@univ-fcomte.fr

Laboratoire d'Informatique de Franche-Comté, Besançon, France

Alain G. N. Anyouzoa

aanyouzoa@cougaarsoftware.com

Cougaar Software Inc., McLean, USA

1 – Introduction and Motivations

We showed in [4] that a Usage Analysis can be used to master the progression of DL training and to identify at-risk users. That paper presents weekly cumulated intra-group interaction data and demonstrates how the failure of one of the groups involved in a DL training experimentation could be foreseen from the low level of interactions of and amongst group's members. Additionally, we built a Conceptual Model of Activity [4, page 844] that links the usage of a given tool in the Learning Management System (LMS) to the achievement of a pedagogical task or activity (we view an activity as a set of tasks, each linked to a tool). This offered us an insight in the progression of the DL training; we could at any time, know the achievement state of an activity and the appreciation of the involvement of a user in the course. These functionalities were built in relatively low-sophisticated indicators based on the volume and the type of interaction data of a targeted group or user. Our Conceptual Model of Activity allows the transformation of users' actions into interactions captured in the LMS.

We devised SIGFAD, a multi-agent system (MAS) featuring functionality that should be coupled to LMS in order to support DCL users in their tasks. The design, specifications and development of SIGFAD are detailed in [1][2][3]. The need to make SIGFAD generic has led us to the specification of interaction objects reflecting the actions and the interactions of the users. These interaction objects should be accessed straightforwardly in LMS rather than being obtained through intensive computerised mechanisms. They will interface the LMS and systems like SIGFAD for the advent of useful Usage Analysis Learning System.

2 – Towards a specification of Interaction Data in Learning Management Systems

By defining certain concepts and making them easy to find in the LMS, we can overcome the drawback mentioned in the last sentence of the previous section. We specify hereafter an initial set of concepts representing users' usage and interactions that should be present in LMS. Such concepts will ease the plug-in of systems like SIGFAD to LMS.

User: A user is defined by his identification, his group and the status he has in the group. A user can belong to more than one group and many statuses. However, a user has one and only status in one given group.
User: <id_user, group, status>

Task: A task is an atomic pedagogical assignment attached to one single tool. It has a starting date and an end date and is carried by a group or a sub-set of the group.
Task: <id_task, user list, start_date, end_date, tool>

Activity: An activity is a set of different tasks and is filled by the group or a sub-set of the group. It has a commencement date and an end date.
Activity: <id_activity, task_list, user_list, start_date, end_date>

Session: A session is defined by (a) the time interval constituted by the login and the logout of a user in the LMS (b) the group in which he logs on and (c) the list of tools he used.
Session: <id_session, user, group, start_date, end_date, tool_list >

Interaction Database: The most important and ultimate entity would be an interaction database. The concepts defined above are meant to feed this database. The database should keep up to date the list of tasks achieved by the users and the dates of achievement.
DataBase: <user, group, status, task, achieve_date >

3 - Conclusion

In [4] we showed that the closure of a group during a DL training experimentation we conducted could be predicted from a usage analysis. This analysis has the advantage to avoid getting into the contents and the semantics of interactions and is only based on the number and the types of interactions. Thereafter, we built a multi-agent system based on a usage analysis and providing support to users and especially to tutors responsible for learning groups and to the manager in charge of the good progression of the DL training. One major concern is the plug-in of such a multi-agent system to LMS in a broad sense. This brought us to suggest some objets that could represent the users' actions and interactions; we call them interaction data. We outlined initial and basic specifications for those interaction data which should facilitate their implementation in LMS. The overall goal of the entities illustrating users' actions and interactions is to build a database reflecting in real time the tasks performed by users, their presence in the platform, their group and their sociability: the propensity to interact with others via synchronous and asynchronous communication tools.

References

[1] Aloys Mbala, Alain G. N. Anyouzoa. A multi-agent system to support users in Online Distance Learning. *Accepted in the Agent Based Systems for Human Learning Workshop at AAMAS'2005*, 6 pages.

[2] Aloys Mbala. *Analyse, Spécification, Conception et développement d'un système multi-agent pour le soutien des utilisateurs en formation à distance*. Thèse de l'Université de Franche-Comté Numéro 977. Octobre 2003, 298 pages.

[3] Aloys Mbala, Christophe Reffay, and Thierry Chanier. SIGFAD: Un système multi-agents pour le soutien des utilisateurs en formation a distance. In *Actes de la Conférence sur les Environnements Informatiques pour l'Apprentissage Humain 2003*. Pages 319–330, 2003.

[4] Aloys Mbala, Christophe Reffay, and Thierry Chanier. Integration of automatic tools for displaying interaction data in computer environments for distance learning. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 841–850, 2002.

Use analysis for the design of the ITS AMBRE-add

Sandra NOGRY
LIRIS, Université Lyon 1 - CNRS
8, bd Niels Bohr - la Doua
69622 Villeurbanne Cedex
sandra.nogry@liris.cnrs.fr

1. Introduction

This poster describes the analysis of learning activities used as support to the design of the ITS (Intelligent Tutoring System) AMBRE-add [5]. The aim of this analysis is to evaluate if our ITS is adapted to nine years old pupild and if there is a divergence between the expected use and the effective use of the system [2][6].

The purpose of the ITS AMBRE-add is to help learners to acquire a classification of arithmetic word problems (and solving tools) [7][8][9] by using the Case-Based Reasoning paradigm [3] as a learning strategy. Thus, in order to help the learner to acquire a classification, we propose to present him a few worked-out examples representative of each class (typical problems). Then, the system guides the learner's solving of the problem by following each step of the CBR cycle:

- the learner reformulates the problem in order to identify problem structure features.
- Then, (s)he compares the problem to be solved with the typical problems in order to choose the closest one.
- Next, (s)he adapts the typical problem solution to the problem to solve.
- Finally, (s)he classifies the new problem. The steps are guided by the system, but done by the learner. The responses are diagnosed in each step of the cycle.

2. How to record and analyse the effective use of the system?

We analysed the effective use of the system by a class of 21 pupils during the class. We combined several methods. We recorded interaction traces; made observations, interviews, and proposed a questionnaire. During the individual observation, we noticed the learner's activity in each step, difficulties encountered, his (her) reactions when (s)he made errors and how (s)he took explanations into account. To complete these observations, we asked them at the end of the last session in order to know what strategy each pupil used to redact the solution during the adaptation step. After the sessions, the learners filled up a

questionnaire which dealt with learner satisfaction, difficulties encountered and vocabulary understanding.
Then, we analysed the effective use: interaction traces allowed us to know how many problems they solved and how many time they passed in each step.

3. Usage analysis and design of AMBRE-add

The experiment shows that AMBRE-add was not used as accurately as we expected. For example, we expected that pupils compare the problem to solve with typical problems in order to detect similarities; and that they adapt the solution of the typical problem. The observations showed that they did not really compare the problems, but often mapped the reformulation using perceptive similarities. However, in the adaptation step, half of pupils adapted the typical problem, as we expected. 37% of pupils redacted the solution using the reformulation. Both these results are encouraging.
Besides, this analysis shows that pupils are very autonomous, and understand the AMBRE principle well. However, several difficulties persist and were identified, learners had difficulties to understand the diagram used in reformulation. We proposed several recommendations to solve these difficulties: we proposed a new demonstration of the system in order to better introduce the reformulation of the problem.
Finally, we noticed that 4 pupils used a trial-and-error strategy and were "gaming the system" [eg. 4] This behaviour is strongly negatively correlated with learning [1]. So we want to detect trial-and-error strategy in order to remedy to it.
In this aim, we are realising an experiment in order to quantify this behaviour by the quantitative observational method by Baker, Corbett et Koedinger (2004) [1]. In a second time, we would like to compare observations and interaction traces in order to identify the "signature" of this behaviour to detect it automatically.

References

[1] Baker, R.S., Corbett, A.T., Koedinger, K.R.: Detecting Student Misuse of Intelligent Tutoring Systems. International Conference on Intelligent Tutoring Systems (ITS'2004), Springer, (2004), 531-540.
[2] Béguin, P., Rabardel P.: Concevoir des activités instrumentées. Revue d'Intelligence Artificielle, Vol. 14 (2000), 35-54.
[3] Kolodner, J.: Case Based Reasoning. San Mateo, CA: Morgan Kaufmann Publishers, (1993).
[4] Mostow, J., Aist, G., Beck, J.E., Chalasani, R., Cuneo, A., Jia, P., & Kadaru K.: A La Recherche du Temps Perdu, or As Time Goes By: Where does the time go in a Reading Tutor that listens? Sixth International Conference on Intelligent Tutoring Systems (ITS'2002), Springer, (2002).
[5] Nogry S., Jean-Daubias S., & Duclosson N.: ITS evaluation in classroom: the case of the AMBRE ITS. Conference on Intelligent Tutoring Systems (ITS'2004), Springer, (2004), 511-520.
[6] Perriault, J.: La logique de l'usage, essai sur les machines à communiquer, Flammarion, (1989).
[7] Rogalski, M.: Les concepts de l'EIAO sont-ils indépendants du domaine? L'exemple d'enseignement de méthodes en analyse. Recherches en Didactiques des Mathématiques, vol 14 n° 1.2 (1994), 43-66.
[8] Schoenfeld, A.: Mathematical Problem Solving. New York: Academic Press, (1985).
[9] Vergnaud, G.: A classification of cognitive tasks and operations of the thought involved in addition and substraction problems. Addition and substraction: A cognitive perspective, Hillsdale: Erlbaum, (1982), 39-58.

Interaction Analysis of Learning and Teaching Processes of e-learning Systems

Alexandra Totter, Gudela Grote
Organisation, Work and Technology Group, Swiss Federal Institute of Technology Zurich (ETH), Kreuzplatz 5, 8032 Zürich, Switzerland

Abstract.
We introduce a method for processes-oriented analysis of collaborative interactions as part of the formative evaluation of the Lab@Future platform. We examine the role of this e-learning platform as a tool for mediating interaction and task accomplishment. Furthermore we investigated the extent to which our realized pedagogical framework evokes different types of interactions.

1. Introduction

Modern computer supported cooperative learning environments (CSCL) enable verbal interactions in distributed collaborative settings. These collaborative interactions involve explanation, reflection, verification, critical assessment, argumentation, co-construction of knowledge and meanings [1]. To be able to analyse this activities as part of the evaluation of CSCL environments the collaborative interactions have to be investigated along two dimensions. First with respect to system design – the aim is to understand the role of the e-learning platform as a tool for mediating interaction, enabling features for collaboration (e.g. setting rules for turn taking, etc.). Second pedagogical design aspects have to be explored defining and supporting types of interactions that are expected to promote learning.

A generic "mixed and augmented reality" – learning platform was developed enabling distributed CSCL. The Lab@Future platform (see Figure 1) supports a pedagogical framework derived from constructivism, activity theory and the theory of expansive learning [2]. Based on this pedagogical framework different types of interaction were expected to occur such as students collaboratively perform practical tasks; structure and control their learning process; choose methods for solving the task etc. The teacher was expected to act as a coach; diagnoses mistakes and supports students. Research questions in the course of the formative evaluation were (i) the e-learning platform as a tool mediating interaction with respect to interceptions of tasks-related activities which according to [3] are the only one related to achievement and (ii) providing empirical evidence to which extent the learning and teaching approach (pedagogical framework) evoked expected types of interaction.

2. Formative evaluation: Processes-oriented analysis of collaborative interactions

During this formative evaluation students had the tasks to collaboratively build an industrial safety circuit for a real machine in a distributed setting. Within five sessions four students supported by one teacher tried to solve this task. The sessions varied between 50 and 120 minutes. Students could freely choose which communication and collaboration tools to use. An observer manually recorded every sequence of utterances that occurred. This record was transcribed; single sequences were identified. A sequence could contain several utterances. In total 736 utterances were recorded. Out of this corpus a content driven category system was developed in order to categorize each utterance. In total 31 mutually exclusive categories of utterances were identified covering all different themes/contents. These categories were further grouped into nine meta-categories of utterances (Table 1).

Our content driven category system supports categories identified in prior research [1, 4] distinguishing between *task-oriented, dialogue-oriented categories* and *social-emotional activities*. With respect to research question (i) the e-learning platform as a tool enabling computer-based tasks accomplishment and mediating interaction in addition to the above mentioned categories we identified three categories, so-called *tool-oriented activities*. These categories are “(3) system use” (i.e. questions, answers and comments about the use of the Lab@Future platform), the need for “(4) identifying people and systems” (i.e. people hearing and seeing each other, finding out who says or does what) and “(6) media choice” (i.e. making decisions which tools to use). In particular categories “(4) identifying people and systems” and “(3) system use” indicate interceptions of tasks-related activities which should be minimized. These results initiated the implementation of better features for identification and the necessity to calculating enough time for the participants to not only get a first idea of a system, but really learn to use the system

Already in this early developmental stage of the Lab@Future platform according to (ii) expected types of interaction have been found. Our proposed pedagogical framework clearly supports collaborative interaction between students. 50% of the utterances were related to the meta-category “(1) students’ task-related activities”; further 10-15% of the utterances deal with “(2) task coordination” throughout the sessions indicating that students do structure and control their learning process. These first evaluation results denote that the Lab@Future platform offers teachers and students a tool to collaboratively accomplish tasks – even in a distributed setting.

Figure 1: Screenshot of the Lab@Future platform

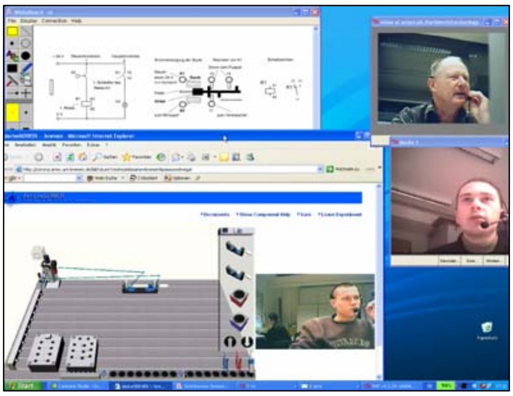


Table 1: Category system

Meta-categories
1. Students’ task-related activities
2. Task-coordination
3. System-use
4. Identification of people & system state
5. Communication rules / behaviour
6. Media / System choice
7. Intervention of the teacher
8. Confirmation
9. Other comments

Acknowledgements

The work presented in this paper has been carried out in the context of the EC-funded project lab@future “School LABoratory anticipating FUTURE needs of European Youth”, IST-2001-34204, 7th IST CALL, 1.5.2002-30.4.2005.

References

1. Baker, M.J. and K. Lund. *Flexibly structuring the interaction in a CSCL environment*. in *European Conference on Artificial Intelligence in Education*. 1996. Lisbon, Portugal: Edições Colibri.
2. Engeström, Y. *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research*. 1987, Helsinki: Orienta-Konsultit Oy, Finland.
3. Webb, N.M., *Task-Related Verbal Interaction and Mathematics Learning in Small Groups*. Journal for Research in Mathematics Education, 1991. **22**(5): p. 366-389.
4. Fisch, R., *Eine Methode zur Analyse von Interaktionsprozessen beim Problemlösen in Gruppen*. Gruppendynamik, 1994. **25**(2): p. 149-168.

Towards Automatic Discovery of Peer Helpers from a Large Message Board System

Mike WINTER, Ben DANIEL and Christopher BROOKS[1]
{mfw127, bkd708, cab938 }@mail.usask.ca[1]
ARIES Laboratory, Department of Computer Science,
Saskatoon, Saskatchewan, Canada S7N 5A9

Abstract. In this paper, we discuss several issues relating to automatic discovery of peer helpers based on the contents of a large educational message board system. In particular, we focus on the discovery of topics within the discussion board.

Introduction

The University of Saskatchewan Computer Science Department supports a popular and varied set of e-learning tools for its students. The iHelp Discussion[1] system is an asynchronous forum and synchronous chat system that has wide use throughout the Computer Science curriculum and the iHelp Learner Content Management System (LCMS) [2] provides complete first year courses which are used in some manner by hundreds of students annually.

One ongoing research project has been to provide peer help to users of the iHelp Discussion system. The PHelps project, which used detailed task maps and a list of pre-determined domain experts to provide appropriate tutors, was the first system developed towards this end [3]. The success of the PHelps project led to the development of a similar peer help system (described in [1]) that was integrated into the learning environment as the predecessor of the iHelp Discussion and LCMS tools. This was a much more difficult domain to work in than PHelps because not only was there not a pre-defined list of domain experts, there was no static list of topics that users would be seeking help on. The topics of discussion in the iHelp Discussion environment vary depending on the interests of the instructors and students and are therefore unpredictable and constantly changing.

To remove the need for explicit metadata from learners in the iHelp environment, we are currently exploring ways to generate a mapping of topics to appropriate peer helpers automatically by examining the data and relationships contained within the iHelp Discussion environment. This problem naturally decomposes into two separate projects: determining the topics being discussed in the forums and developing the mapping between those topics and the experts in the topics. In this paper, we briefly discuss automatic topic discovery and conclude with our future plans for the project.

1. Determining Message Board Topics Automatically

The first step in finding good peer helpers from message board text is to determine the topics of discussion in the message board. Looking at the title of threads is a quick and easy way to

[1] http://ihelp.usask.ca

estimate the topic being discussed, but it is often the case that titles are not informative, only informative in context, or that the discussion within the threads contains many issues that are not referenced in the title. Adding the text of the forum is a useful step in disambiguating: 'Something wrong in compile' becomes 'Something wrong in compile Assignment One CMPT 350', but still does not capture all the facets of the conversation, which is about using a SAX XML parser in Java. Classifying message board threads by title and category alone appears to not precise enough to determine the topics being discussed.

To more accurately classify message board threads, the content of the threads must be examined. This is a harder problem than tagging threads with their titles due to the large increase in the amount of text that has to be processed. There are two general approaches that we are exploring: intra-thread classification and inter-thread classification. Intra-thread classification involves iterating through each thread in a forum and determining keywords for the thread based on the thread's content. Inter-thread classification, which we are currently exploring, involves treating each thread as a document and using a document clustering algorithm to group the threads together. The topics of the clusters (and therefore each thread) can then be identified by the relevant keywords of the cluster. Inter-thread classification is preferred for our current project because it allows us to discover topics that are being discussed in multiple forums and generates a wider candidate list of helpers for those topics. To cluster threads together, we are first adding the thread title and forum category to each thread for greater precision, stripping out the stop words, and then using a spherical k-means algorithm with 10-fold cross-validation from the Mallet toolkit [4] to determine the number of clusters in the data and the clusters' contents. We are currently examining techniques to summarize the contents of each cluster to determine what topics are being discussed in each cluster.

2. Summary and Future Work

In Section One, we discussed the text clustering algorithm that we are currently using to group together forum threads and determine their topics. The next step is to examine the clustered threads and determine who the effective domain experts are in each thread. We are currently exploring different techniques to accomplish this goal.

In future work, we will test the complete peer help finding process, with the peer helpers that the system recommends being tested against the peer helpers humans recommend. We will then put the system through an iterative improvement process, examining different topic finding and thread analysis procedures to try and improve the utility of the system.

References

[1] Greer, J.E., McCalla, G.I., Cooke, J.E., Collins, J., Kumar, V.S., Bishop, A. and Vassileva, J.I. *The Intelligent Helpdesk: Supporting Peer-Help in a University Course*. International Intelligent Tutoring Systems Conference (ITS'98), San Antonio, August, 1998, 494 -503.

[2] Brooks, L. Kettel, and C. Hansen. Building *a Learning Object Content Management System (In Submission)*. World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2005), Association for the Advancement of Computing in Education (AACE), 2005.

[3] McCalla, G.I., Greer, J.E., Kumar, V.S., Collins, J.A., Meagher, P. Tkatch, R. & Parkinson, B. *A Peer Help System for Workplace Training.* Proceedings of the Eighth World Conference on Artificial Intelligence in Education (AI-Ed'97), Kobe, Japan, August, 1997, 183-190.

[4] McCallum, Andrew Kachites. MALLET: A Machine Learning for Language Toolkit. Unpublished. http://mallet.cs.umass.edu, 2002.