ePortfolios in ActiveMath

Homik, Martin, University of Saarland, Germany, homik@ags.uni-sb.de
Melis, Erica, DFKI, Germany, melis@dfki.de

Abstract. Existing ePortfolios record the work of a learner and his assessments. The standalone application “workbook” of the web-based and user-adaptive learning environment ActiveMath tries to go beyond that. It aims at supporting explicit learning by providing interaction, construction and reflection opportunities.

1. Introduction

Existing ePortfolios record the work of a learner and his assessments. Prior to the selection of evidences to be included into a portfolio, a reflection process takes place: "What kind of evidence do I want to include into my portfolio and why? Why do I prefer some evidences over others?" These reflections are usually not stored together with the evidence itself. Instead, the information is stored in the learner's memory. This implicit information if not used regularly is transient and vanishes after a while. A learning log is a mean of storing and transferring implicit information into explicit one.

Implicit memory is closely related to implicit learning which is defined as “learning complex information without complete verbalisable knowledge of what is learned” (Seger, 1994). In other words, learners acquire knowledge, e.g. by reading, or they learn a sequence of steps to perform a task, but most of them are unable to report the principles underlying their performance. Hence, there is a need for cognitive tools supporting explicit learning.

The standalone application “workbook” of the learning environment ActiveMath tries to go beyond pure collection of work and acquisition of knowledge. It aims at supporting explicit learning by providing interaction, construction and reflection opportunities. We show how this is realized currently: the learner can interactively assemble his own learning material, reflect on his learning process and learning results in a (adaptively) structured environment and he can recall knowledge in a situation-related way. When “workbook” is run together with ActiveMath, it can take advantage of the systems various functionalities such as knowledge base and learner model.

2. ActiveMath

ActiveMath (Melis et al., 2001) is a web-based, multi-lingual, user-adaptive, interactive learning system for mathematics. It employs technology for enhancing learning with scaffolding and instruction as well as with constructivist elements, among them cognitive tools such as an interactive concept mapper, a semantic lexicon and those described below.

ActiveMath bases on semantically OMDoc-encoded (Kohlhase 2000) learning objects that are annotated with pedagogical and other metadata. The system provides a modular and open architecture to use various components and external services such as a learner model, a knowledge base, a course generator and several service systems. The course generator automatically assembles individual books according to learner's goals, preferences and knowledge.

Domain knowledge is represented by a collection of learning objects encoded in OMDoc. The metadata annotations define mathematical and educational properties of knowledge items and relate these to each other. This establishes ontologies of mathematical and educational knowledge. The domain knowledge contains *concept* and *satellite* items. Concept items (e.g. definition, theorem) are the main items of content ontology, whereas satellites (e.g. example, exercise) are additional content items grouped around the concepts. Each item is stored in ActiveMath's knowledge base and is accessible via a unique identifier.

3. Workbook

The “workbook” is ActiveMath’s first version of an ePortfolio which enables students to collect and reflect on their work. It includes two tools: learning log and assembly tool.

3.1 Learning log tool

A learning log provides a model for self-managed documentation of learning objectives and processes and supports the awareness and reflection of learning activities. ActiveMath’s learning log offers two types of log entries: one for reflecting on lectures and one for stating hypotheses. In the following, we discuss the pedagogical motivation for learning logs, introduce both types of log entries, and describe how to capture the semantics of learning logs.

3.1.1 Pedagogical motivation

Gallin and Ruf (1990) emphasize the importance of documentation and presentation of core ideas by the learner in his own words. They empirically validated paper learning logs with students in primary school as well as in secondary school (Gallin/Ruf 1993). Further pedagogy researchers (e.g. Mayr 1997) tested paper learning logs with students of different educational levels. Their results show that learning logs help to strengthen the learner's autonomy and support his meta-cognitive reasoning such as planning, reflecting, correcting, looking back, memorizing, and exploring different viewpoints in a situated way. This observation motivates electronic learning logs in any eLearning environment.

3.1.2 Reflecting on lectures

Reflection on lectures by creating own learning logs is a simple process of taking several reflection steps: on the content, on the process, and on meta-cognition. Table 1 lists the questions used in ActiveMath’s learning log. A picture of the learning log application is shown in figure 1.

Each learning log entry contains basic information such as label, date, and privacy. The label is the entry’s subject. Storing the date enables to review log entries chronologically and to retrieve log entries for a specific period. By setting the privacy option, the learner indicates whether his entry is accessible for the public.

Reflection on the content comprises recapitulation of learned content in the learner’s own words. He is encouraged to write a summary of the last lesson, to explicitly refer to learning objects he has understood as well to those he has not understood, and to formulate questions he might ask his tutor in order to fill a learning gap.

Reflection on the (production) process includes information on resources in general and persons in particular that were involved in the problem solving task. Storing this information creates links between content and problems on the one side and resources on the other side. In watching his learning logs the learner can deduce which resources are useful in which context, or which resources are most helpful.

Reflection on meta-cognition implies self-assessment (where you are now), checking and setting objectives (where you want to get to), action planning (how you will get there), and charting progress (how you are doing). The intention is to increase the learner’s awareness of his cognitive skills. To do so, the learner has to reflect about his difficulties and about his goals. He has to argue if he has achieved his goals. If not, he should try to identify the problems. Moreover, he is asked to refer to learning objects that are most helpful to understand the learning content. Finally, he can rate his performance which expresses satisfaction on learning progress.

In ActiveMath, the learner introduces a log entry in a dialogue window which contains a tabbed panel. Each panel represents one of the reflection fields described above. The introduced questions expect different types of answers, for example yes/no, text, or single/multiple selections of a given set. To answer these questions, we use a palette of diverse widgets such as text fields, text areas, list boxes, or check and radio buttons. Referring to learning objects is a simple drag and drop gesture: point the mouse cursor to the learning object’s title in the browser ActiveMath is currently running in and drag it to the learning log entry.

General data:	
	Label, privacy, date
Questions related to content:	
	What was the lesson about? Write a summary.
	What did you (not) understand?
	Formulate questions to issues you did not understand.
Questions related to the (production) process:	
	What kind of tools or resources did you use for solving?
	Who helped you?
Questions related to meta-cognition:	
	What difficulties arose during problem solving?
	Which learning objects were most helpful to you?
	How would you rate your performance?
	Did you achieve your goals? If not, why?
	Who can help you?

Table 1: Learning log questions

3.1.3 Writing Hypothesis

A hypothesis is a proposed explanation for a phenomenon. A phenomenon can have several hypotheses and each can be annotated by different beliefs. Formulating hypotheses is a meta-cognitive strategy that supports the acquisition of reasoning and problem solving skills. Further, hypotheses are basic building blocks of argumentations, and thus, knowing how to formulate and justify hypotheses, improves argumentation skills which are in particular important in scientific disciplines.

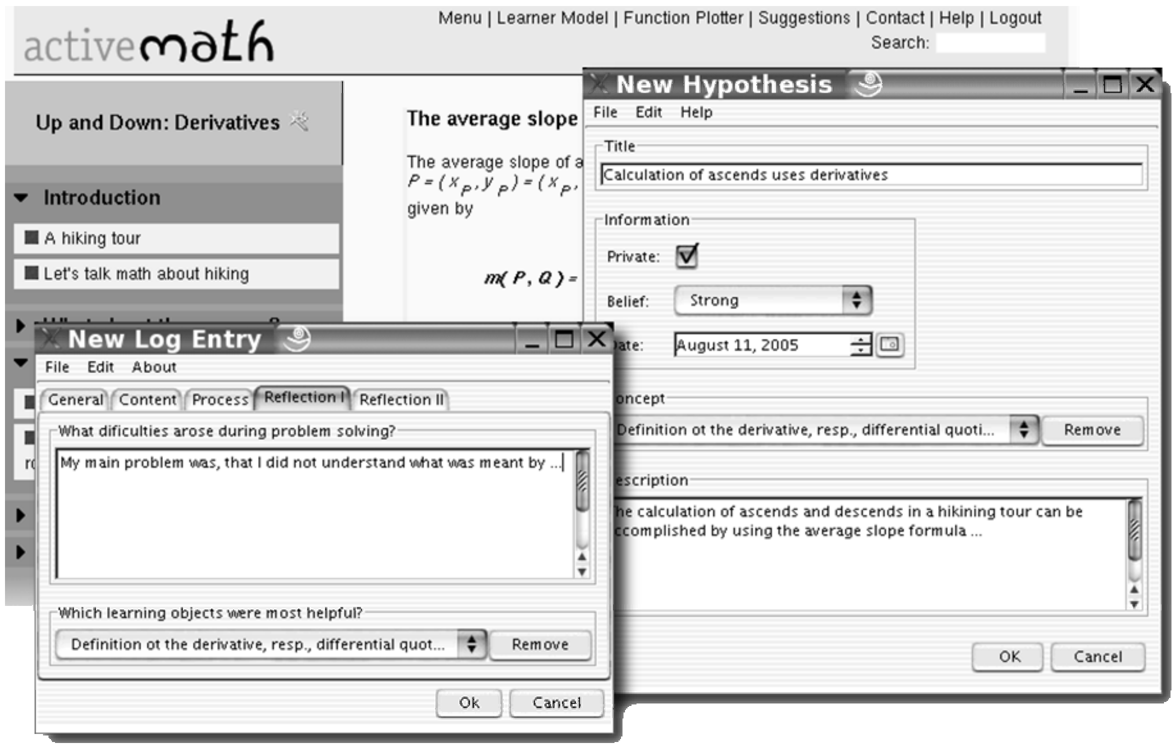
ActiveMath's learning log supports formulation of hypotheses and their justification with evidences which are, in this context, learning objects (see figure 1). The learner introduces a hypothesis in a dialogue window, where he first inputs the label of the hypothesis. Then, in the description field, he can elaborate on his line of argumentation. To justify a hypothesis, the learner can drag any learning object from ActiveMath's learning environment to the dialogue. Finally, he can rate a hypothesis, i.e., he can state how much he believes the hypothesis is true.

3.1.4 Capturing semantics

Learning logs carry structure and semantics. Plain text input without structure is accessible to human mind, but not to machines. Thus, our learning log tool stores and processes information as partially machine-readable entities. This enables semantic evaluation and reuse of learning logs.

Learning logs' structure can address different aspects of learning such as motivation, content, process, and meta-cognition. They depend on the target group and therefore differ in aspects, questions and ordering of questions. Our structure of learning logs is adopted from pedagogical experience for university students.

Learning log structure provides first information on semantics. That is, we know what the questions are about, but we do not understand the learners' input. To capture even more semantics from input our learning log tool applies drag&drop interactions, single/multiple choice questions and template answers.



The average slope of a $P=(x_P, y_P)=(x_P,$ given by

$m(P,Q)=$

Calculation of ascends uses derivatives

Strong

August 11, 2005

Definition ot the derivative, resp., differential quoti...

he calculation of ascends and descends in a hikining tour can be ccomplished by using the average slope formula ...

My main problem was, that I did not understand what was meant by ...

Definition ot the derivative, resp., differential quot...

Figure 1: Learning Logs in ActiveMath

Drag&drop. In ActiveMath, learning objects are identified by unique identifiers. They are presented to the learner within a book or a lexicon. Each presented learning object displays the object's title as a link that encapsulates the object's identifier. This way, the identifier is accessible by various actions: clicking on the title of an object in a book opens the lexicon with a detailed description of the item; dragging the link out of the browser's window and dropping it on learning log's widgets stores the identifier and establishes an association. This way, drag&drop actions allow the user to refer to learning objects. Hence, the system "knows" what learning object the learner describes and what evidences he uses for justifying hypotheses.

Single/Multiple choice questions. Some questions provide a restricted set of answers. These are modeled either as single or as multiple choice questions. Single choice questions accept only one answer (e.g. "How would you rate your performance?"), multiple choice questions accept several answers (e.g. "What kind of tools and resources did you use for solving?"). Here, the system is able to "distinguish" between different answers.

Depending on the question, it can happen that none of the provided answers applies. Then the learner can input an additional answer, which will be stored and which will be available in further learning logs. For instance, he might have used a tool that has not been listed beforehand. Then he inputs the name of the tool. If he uses the tool more often the learning log offers the tool's name as an answer in this question again.

Template answers. A combination of single choice questions and a user's plain text input are template answers. Each template encodes a specific beginning of a sentence. Examples are: "I am confident that...", "I doubt that...", "I know that..." etc. On the one hand this approach supports the learner in articulation and on the other hand the learning log is able to "classify" answers.

3.1.3 Views and Queries

ActiveMath learning logs are stored in a database and can be retrieved at any time. Database queries profit from the learning logs' structure and semantics, as this data is stored explicitly. That is, we can provide views or search for entries according to a specific criteria. Views are predefined queries to the database, while search statements are custom queries defined by the learner. Some interesting example queries are: the learner can ask the database for a list of all entries; for entries of a specific type; for entries of a specified period; for entries related to a specified learning object; for entries related to learning objects he does not understand; or for entries related to learning objects he did not understand in first place, but has learned meanwhile.

ActiveMath's learning log lists the result of a query in a table ordered by date. The table ordering can be changed by clicking a table header. You can order by date, label, or privacy in respect of ascending or descending order. Clicking on a row opens a window, displays the complete entry, and allows for editing.

3.2 Assembly tool

3.2.1 Pedagogical motivation

Bredo (1993) points out that learners are never integrated into the constructive process of assembling learning objects. Instead, they are given learning material and a task that puts them in a passive role with respect to finding their own problems and developing their own expertise. Learners become “book smart and practically stupid”. This observation motivates the assembly tool.

The goals of an assembly tool are to support the learner's meta-cognitive reasoning such as planning, reflecting, structuring and presentation as well as to help memorizing knowledge and to work with content actively.

3.2.2 Interactive actions

On the one hand, ActiveMath provides students with authored books or adaptively generated books by the course generator. On the other hand, ActiveMath offers a learner to manually create or edit his own book with the assembly tool and to upload it into ActiveMath. If the learner sets access rights for the public this book becomes publicly available to other students who are free to browse it, to work it through and to comment on it.

Currently, the assembly tool offers learners to newly assemble content (learning objects) in the sense of creating and ordering new chapters and sections as well as dragging learning objects from ActiveMath learning environment into the book the learner is currently assembling.

4. Architecture

The learning log as well as the assembly tool runs standalone. Both can be invoked either locally or as modules within the Java Web Start framework (jnlp) from ActiveMath’s learning environment. Data is stored locally, and since they are connected to ActiveMath's services, data exchange is possible. Figure 1 illustrates ActiveMath’s architecture and embedding of jnlp-based applications. The knowledge base and the user model operate as services and can be accessed by secured XML-RPC interfaces which restrict read and write actions to data. The session manager is the key control instance that manages all system properties and threads. It is connected to the knowledge base and user model services. It is in charge of assembling content according to the user’s model and of passing it to the presentation generator which in turn transforms the content into a web-presentable format. Finally, the web server presents the learning objects in a web browser to the user. Moreover, the web server watches user actions which are passed to and interpreted by the session manager again.

ActiveMath provides access to the learning log and assembly tool via a web link which upon activation triggers a request to ActiveMath’s servlets. The latter generates a jnlp file and includes the user’s id, name, language, as well as urls to ActiveMath’s host, to the user model and to the knowledge base services. Further, it contains the application’s arguments and resource path. The manager sends the created jnlp file back to the client browser which recognizes it as such and processes it, i.e., it downloads the application’s resources if not already present on the client side and starts it.

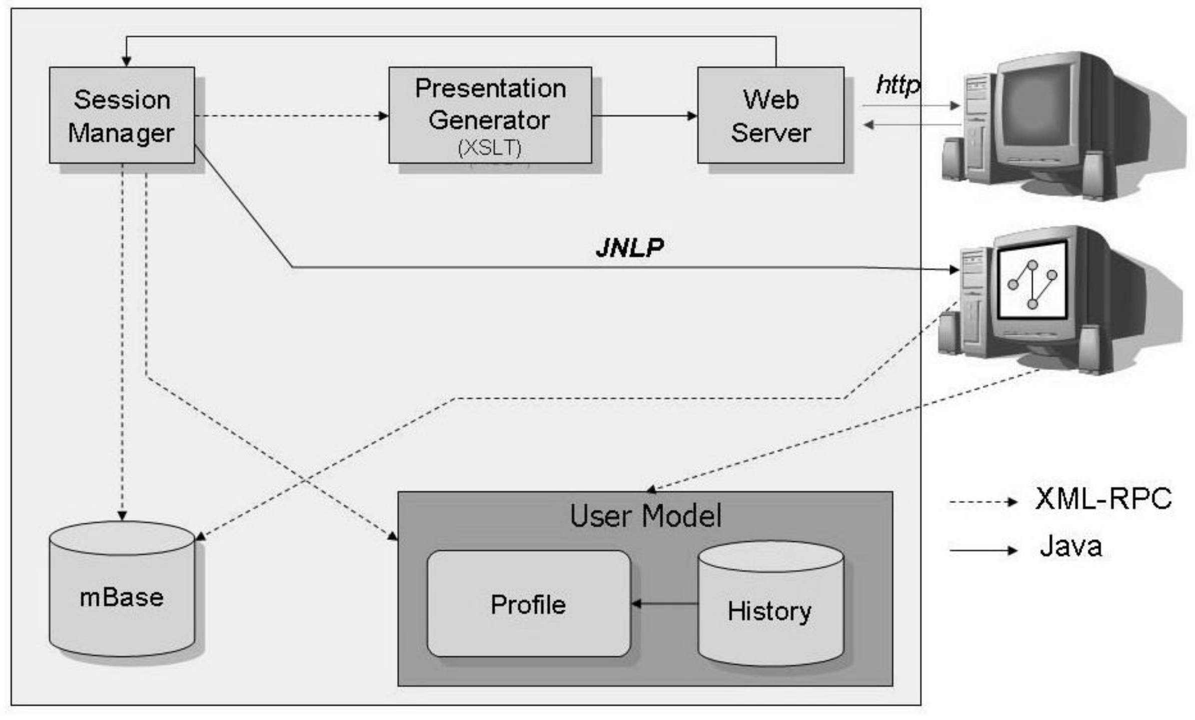


Figure 2: ActiveMath architecture

Communication between the learning log as well as the assembly tool and ActiveMath is two-folded: via drag&drop and via services. Drag&drop is a one-way communication. Links (w.r.t. unique identifiers) can be dragged from ActiveMath learning environment and dropped onto the learning log or assembly tool. Both tools exchange data with the knowledge base service upon request with respect to the unique identifier.

The learning log contains a backend for storing entries in a relational database. The database is organized in tables for the learning log structure and entries. Figure 3 illustrates briefly the database scheme. In particular, it highlights how learning log objects are linked. Log entries and hypotheses share a common structure that contains basic information. Identifiers of learning objects are stored as strings in the “MbaseID” table. The tables “Understood”, “NotUnderstood”, “MostHelpful”, and “Evidence” link MBase identifiers to log entries as well as to hypotheses.

The MBase identifier does not say anything about the ActiveMath session or MBase service it has been used in. Instead, it is an information relative to any MBase instance. This enables the learner to maintain learning logs in different ActiveMath environments. For instance, the learner can work with several ActiveMath environments in parallel. Information on the learning object referred by the identifier will be accessed by the MBase service that belongs to the same environment as the learning log. Moreover, the learning log is still able to retrieve information related to any identifier if the user aborts a session and restarts it later on as well as if he changes to a different ActiveMath environment.

The assembly tool stores books in OMDoc format locally in a file. The assembly tool applies the same drag&drop mechanism as learning logs. That is, you drag an identifier and you store an identifier. You never store the information itself. The information is retrieved from an MBase service by need.

If an assembled book is intended to become publicly accessible, the assembly tool contacts ActiveMath via the shipped url and transmits the book in OMDoc format. The session manager integrates the book and creates a new link to the book on the main page and another link for writing public comments by other users.

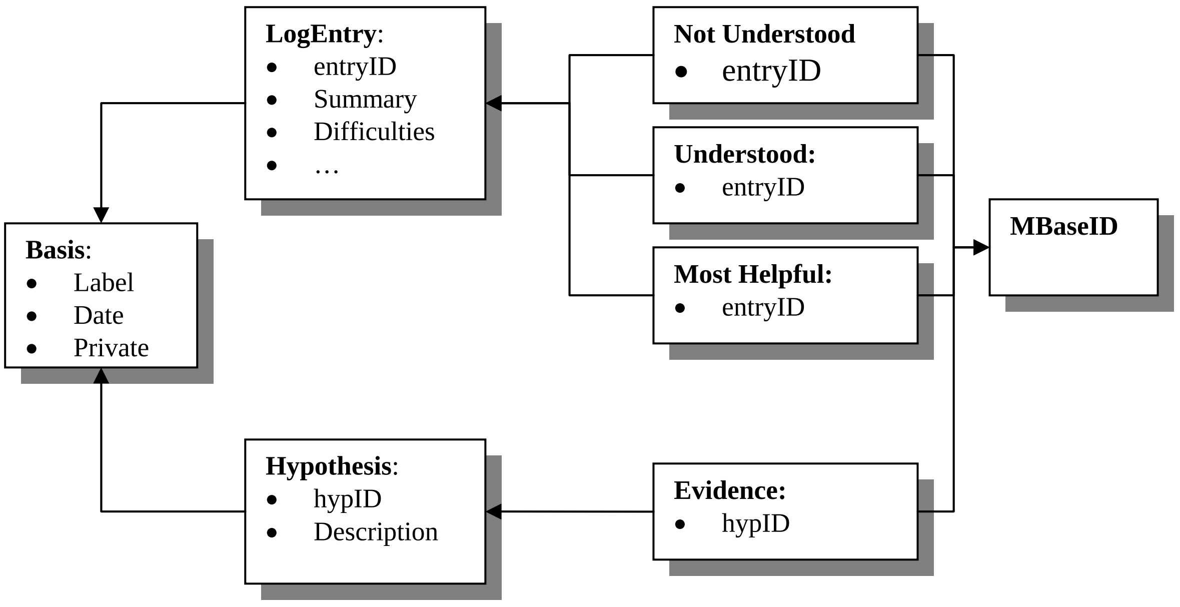


Figure 3: Learning log database scheme

5. Related work

Weblogs (bloggs) are text-based publishing systems that can be used among others for learning logs. There is a huge amount of weblog products and services (free and commercial). Usually, they offer a convenient web front-end for text input and a one-click upload to a web-server. They also offer privacy and collaboration facilities (e.g. comments). More complex systems, such as the learning environment *Moodle* (www.moodle.org), provide a journaling application that is tightly integrated into the environment. eHelp (Nückles et al. 2004) is a tutored system whose learning log structure bases on pedagogical research. It supports the writing of learning diaries through a modelling and scaffolding of the phases of planning, production, and revision.

Related systems that support argumentation and hypothesis formulation are *BioWorld* (Lajoie et al. 1995), *Propa* (Linton 1995) and *Belvedere* (Suthers et al., 1995). *BioWorld* provides cognitive tools for the acquisition of scientific reasoning skills in biology for high school students. In an argumentation environment the learner is engaged explicitly in justifying hypotheses with evidences. *Propa* provides an environment for training satellite tracking analysts in which learners explicitly link evidence to explanations (hypotheses) on an argument palette. *Belvedere* provides a graphical environment in which students construct diagrammatic representations of arguments in the areas of science and public policy. Their arguments are analysed for consistency, comprehensiveness and coherence and online advice is available to assist in argument completion and reconstruction.

6. Conclusion

This paper presents the “workbook”, the first ActiveMath version of an ePortfolio. It reviews the basis and benefits of the workbook’s integrated applications, learning log and assembly tool, and describes their models and architecture. Both tools strengthen the learner’s autonomy and support his meta-cognitive reasoning, in particular explicit learning and reflection in a situation-related way.

Further work will examine the integration of more reflection-supporting tools (e.g. notes), the enhancement of collaboration, and the interaction of the workbook with ActiveMath’s services. We will test the workbook tool and conduct an evaluation in our next lecture “Hands-on mathematics for computer scientists” which is scheduled for the period from October 2005 till February 2006.

6. References

Bredo, E. (1993) *Reflections on the ITSs: A Response to Clancey's Guidon-Revisited.* In Journal of Artificial Intelligence in Education, 14, p.35-40

Gallin, P., Ruf, U. (1990) *Sprache und Mathematik in der Schule. Auf eigenen Wegen zur Fachkomepetenz* – Verlag Lehrerinnen und Lehrer Schweiz, Zürich

Gallin, P., Ruf, U. (1993) *Sprache und Mathematik in der Schule – ein Bericht aus der Praxis.* In Journal für Mathematikdidaktik, 14, 1993, p.3-33

Kohlhase, M. (2000) *OMDoc: Towards an internet standard for the administration, distribution and teaching of mathematical knowledge.* In Proceedings Artificial Intelligence and Symbolic Computations AISC 2000

Lajoie, S. P., Greer, J. E., Munsie, S. D., Wilkie, T. V., Guerrera. C., & Aleong, P. (1995). Establishing an argumentation environment to foster scientific reasoning with BioWorld. In D. Jonassen & G. McCalla (Eds.), *Proceedings of the International Conference on Computers in Education* (pp. 89-96). Charlottesville, VA: Association for the Advancement of Computing in Education.

Linton, F. (1995). *Intellectual skills and cognitive strategies: Can one method tutor both?*, In The Proceedings of the World Conference on Artificial Intelligence and Education*,* Washington, DC, August.

Mayr, J. (1997) *Evaluieren.* In F.Buchberger, H.Eichelberger, K.Klement, J.Mayr, A.Seel and H.Heml: Seminardidaktik, p.224-256, Studienverlag, Innsbruck

Melis, E., Andrès, E., Frischauf, A., Goguadze, G., Libbrecht, L., Pollet, M. and Ullrich, C. (2001) *ActiveMath: A generic and Adaptive Web-Based Learning Environment.* In International Journal of Artificial Intelligence in Education, 12(4): p.385-407

Nückles, M., Schwonke, R., Berthold, K., & Renkl, A. (2004). The use of public learning diaries in blended learning. *Journal of Educational Media*, 29 (1), 49-66.

Seger, C.A. (1994). *Implicit Learning*, In Psychological Bulletin, 115, 163-196.

Suthers, D., Weiner, A., Connolly, J., & Paolucci, M. (1995). Belvedere: *Engaging students in critical discussion of science and public policy issues.* In The Proceedings of the World Conference on Artificial Intelligence and Education*,* Washington, DC, August