Lessons Learned from Authoring for Inquiry Learning: A tale of authoring tool evolution

Tom Murray, Beverly Woolf, David Marshall

University of Massachusetts, Amherst, MA
tmurray@cs.umass.edu

Abstract: We present an argument for ongoing and deep participation by subject matter experts (SMEs, i.e. teachers and domain experts) in advanced learning environment (LE) development, and thus for the need for highly usable authoring tools. We also argue for the "user participatory design" of involving SMEs in creating the authoring tools themselves. We describe our experience building authoring tools for the Rashi LE, and how working with SMEs lead us through three successive authoring tool designs. We summarize lessons learned along they way about authoring tool usability.[1]

1. Introduction

Despite many years of research and development, intelligent tutoring systems and other advanced adaptive learning environments have seen relatively little use in schools and training classrooms. This can be attributed to several factors that most of these systems have in common: high cost of production, lack of widespread convincing evidence of the benefits, limited subject matter coverage, and lack of buy-in from educational and training professionals. Authoring tools are being developed for these learning environments (LEs) because they address all of these areas of concern [1]. Authoring tools can reduce the development time, effort, and cost; they can enable reuse and customization of content; and they can lower the skill barrier and allow more people to participate in development and customization ([2], [3]). And finally, they impact LE evaluation and evolution by allowing alternative versions of a system to be created more easily, and by allowing greater participation by teachers and subject matter experts.

Most papers on LE authoring tools focus on how the features of an authoring tool facilitate building a tutor. Of the many research publications involving authoring tools, extremely few document the use of these tools by subject matter experts (SMEs, which includes teachers in our discussion) not intimately connected with the research group to build tutors that are then used by students in realistic settings (exceptions include work described in [2] and [3]). A look at over 20 authoring systems (see [1]) shows them to be quite complex, and it is hard to imaging SMEs using them

[1] We gratefully acknowledge support for this work from the U.S. Department of Education, FPISE program (#P116B010483) and NSF CCLI (# 0127183).

without significant ongoing support. Indeed, tutoring systems are complex, and designing them is a formidable task even with the burden of writing computer code removed. More research is needed determine how to match the skills of the target SME user to the design of authoring tools so that as a field we can calibrate our expectations about the realistic benefits of these tools. Some might say that the role of SMEs can be kept to a minimum--we disagree. Principles from human-computer interaction and participatory design theory are unequivocal in their advocacy for continuous, iterative design cycles using authentic users ([4], [5]). This leads us to two conclusions. First, LE usability requires the participation of SMEs (with expertise in the domain and with teaching). LE evaluations by non-SMEs may be able to determine that a given feature is not usable, that learners are overwhelmed or not focusing on the right concepts, or that a particular skill is not being learned; but reliable insights about *why* things are not working and how to improve the system can only come from those with experience teaching in the domain. The second conclusion is that, since authoring tools do indeed need to be usable by SMEs, then SMEs need to be highly involved in the formative stages of designing the authoring tools themselves, in order to insure that these systems can in fact be used by an "average" (or even highly skilled) SME.

This paper provides case study and strong anecdotal evidence for the need for SME participation in LE design and in LE authoring tool design. We describe the Rashi inquiry learning environment, and our efforts to build authoring tools for Rashi. In addition to illustrating how the design of the authoring tool evolved as we worked with SMEs (college professors), we argue for the importance of SME involvement and describe some lessons learned about authoring tools design. First we will describe the Rashi LE.

2. The Rashi Inquiry Environment for Human Biology

Learning through sustained inquiry activities requires a significant amount of reflection, planning, and other metacognitive and higher level skills, yet these very skills are lacking in many students ([6],[7]). Thus it is crucial to support, scaffold, and teach these skills. This support includes providing "cognitive tools" [8] that relieve some of the cognitive load through reminding, organizational aides, and visualizations; and providing coaching or direct feedback on the inquiry process. Our project, called Rashi, aims to address these issues by providing a generic framework for supporting inquiry in multiple domains.

A number of educational software projects have addressed the support of inquiry learning in computer based learning environments and collaborative environments (for example: Inquiry Island [9], SimQuest [10], Bio-World [11], Belvedere [12], CISLE [13]). These projects have focused on various aspects of inquiry, including: providing rich simulation-based learning environments for inquiry; providing tools for the gathering, organization, visualization, and analysis of information during inquiry, and---the main focus of our work---directly supporting and scaffolding the various stages of inquiry. Our work advances the state of the art by providing a generic framework for

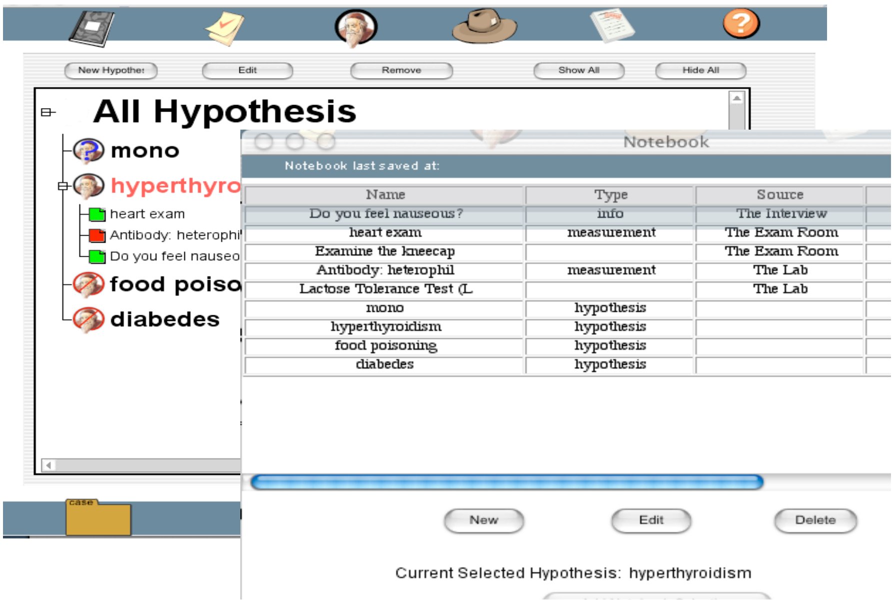

Figure 1 A&B: Rashi Hypothesis Editor and Inquiry Notebook

LE tools for: searching textual and multimedia recourses, using case-based visualization and measurement, supporting organization and metacognition within opportunistic inquiry data gathering and hypothesis generation. The project also breaks new ground in its development of authoring tools for such systems--SimQuest is the only inquiry-based system that includes authoring tools, and its focus is more on authoring equation-centric models than on case-based inquiry.

Students use Rashi to accomplish the following tasks in a flexible opportunistic order ([14] [15]):

- Make observations and measurements using a variety of tools
- Collect and organize data in an "Inquiry Notebook"
- Pose hypotheses and create evidential relationships between hypothesis and data using a "Hypothesis Editor"
- Generate a summary of their final arguments with the Report Generator.

Figure 1 show the Rashi Hypothesis Editor (A) and Inquiry Notebook (B). Students use a variety of tools (not shown) to gather data which they store and organize in the Inquiry Notebook. They use the Hypothesis editor to create argument trees connecting data to hypotheses. Rashi also includes an intelligent coach [14], requiring the SMEs to enter not only the case data that the student accesses, but the evidential relationships leading to an acceptable hypothesis. Domain knowledge which must be authored in Rashi consists of cases (e.g. the patient Janet Stone), data (e.g. "temperature is 99.1"), inferences (e.g. "patient has a fever"), hypotheses (e.g. patient has hyperthyroidism), evidential relationships (e.g. fever supports hyperthyroidism), and principles (references to general knowledge or rules, as in text books).

Rashi is being used in several domains (including Human Biology, environmental engineering (water quality), geology (interpreting seismic activity), and forest ecology (interpreting a forest's history), and in this paper we focus on our most fully developed project, in the Human Biology domain, which is based on a highly successful college course. "Human Biology: Selected Topics in Medicine" is a case-based and inquiry-based science course designed to help freshmen develop skills to complete the science requirement at Hampshire College. Students are given a short case description and then scour through professional medical texts (and on-line sources) looking for possible diagnoses. They request physical examination and laboratory tests from the instructor, who gives them this data piece-meal, provided they have good reasons for requesting it. The problem solving process, called "differential diagnosis" can last from two days to two weeks, with students usually working in groups, depending on the difficulty of the case. Classroom-base evaluations of students over seven years of developing this course show increased motivation to pursue work in depth, more effective participation on case teams, increased critical examination of evidence, and more fully developed arguments in final written reports ([16]). Rashi-Human Biology is our attempt to instantiate this learning/teaching method in a computer-based learning environment.

3. The Complexity of the Authoring Process

In this section we will describe some of what is involved in developing a case-based tutorial for Rashi-Human-Biology, and in so doing we will illustrate both the need for SME participation and the task complexity that the authoring tool needs to support. The complexity of the Rashi LE and of authoring content in Rashi is comparable to that of most other LEs and LE authoring systems. For Rashi-Human-Biology our experts are two college biology professors skilled in using case-based learning and problem-based learning (CBL/PBL, see [17]) methods in the classroom (one of them does the bulk of the work with us, and we will usually refer to her as "the" expert). Given the relative complexity of the data objects involved in designing a case, the expert assists with the following tasks: develop medical diagnosis rules (inferential argument links), create descriptive scenarios and patient signs/symptoms for cases, articulate the details of a problem-based inquiry learning pedagogy, identify primary and secondary sources that students may go to for medical information, and inform us about the expected level of knowledge of the target audience. Our expert also helped us set up formative (clinical and in-class) evaluative trials of the system, and was critical in the analysis of trial results to determine whether students understood the system, whether they were using the system as expected, and whether they were engaged and learning in ways consistent with her goals for classroom CBL. The creation and sequencing of cases that introduce new concepts and levels of difficulty requires significant expertise. This involves setting values for the results of dozens of patient exams and laboratory tests, some of which are normal (for the age, gender, etc. of the patient) and some abnormal. Data must be authored not only for the acceptable hypothesis, but also to anticipate other alternative hypotheses and tests that the students may propose. Student behavior in complex LEs can never be anticipated, and a number of iterative trials are needed to create a satisfactory knowledge base for a give case.

The author uses the Rashi authoring tools to enter the following into the knowledge base:

- Propositions and hypotheses such as "has a fever", "has diabetes"
- Inferential relationships between the propositions such as "high fever supports diabetes"
- Cases with case specific values: Ex: the "Janet Stone Case" has values including "temperature is 99.1" "White blood cell count is 5.0 x 10^3 "

For the several cases we have authored so far there are many hundreds of propositions, relationships, and case values. Each of these content objects has several attributes to author. The authoring complexity comes in large part from the sheer volume of information and interrelationships to maintain and proof-check. The authoring tools assist with this task but can not automate it, as too much heuristic judgment is involved.

The above gives evidence for the amount of participation that can be required of a domain expert when building novel LEs. Also, it should be clear that deep and ongoing participation is needed by the SME. We believe this to be the case for all almost all adaptive LE design. Since our goal is not to produce one tutor for one domain, but tutors for multiple domains and multiple cases, and to enable experts to continue to create new cases and customize existing cases in the future, we see the issues of authoring tool usability as critical and perennial. The greater the complexity of the LE, the greater the need for authoring tools. In designing an authoring tool there are tradeoffs involved in how much of the complexity can be exposed to the author and made a) inspectable, and b) authorable or customizable [4].

The original funding for Rashi did not include funds for authoring tool construction, and the importance of authoring tools was only gradually appreciated. Because of this, initial attempts to support SMEs were focused on developing tools of low complexity and cost. In the next section we describe a succession of three systems built to support authors in managing the propositions and evidential relationships in Rashi. Each tool is very different as we learned more in each iteration about how to schematically and visually represent the content. In one respect, the three tools illustrate the pros and cons of three representational formalisms for authoring the network of evidential relationships comprising the domain expertise (network, table-based, and form-based). In addition, each successive version added new functionality as the need for it was realized.

4. Lessons Learned from Three Authoring Tools

A Network-based representation. At first, the most obvious solution to the authoring challenge seemed to be to create a semantic network tool for linking propositions. The domain knowledge can be conceptualized as a semantic network of evidential relationships (supports, strongly supports, refutes, is consistent with, etc.). We built such a tool, shown in Figure 2 that allowed us to create, delete, and move nodes in the network("propositions"). Nodes could be "opened" and their attributes edited. Nodes of different types (e.g. data, hypotheses, principle) are color-coded. Such a network-style model seemed to fit well with the mental model of the argument structure that we wanted the expert to have. However, in working with both the biology professor and the environmental engineering professor (for a Rashi tutor in another domain), as the size of the networks began to grow, the network became spaghetti-like and the interface became unwieldy. The auto-layout feature was not sufficient and the author needed to constantly reposition nodes manually to make way for new nodes and links. The benefits of the visualization were overcome by the cognitive load of having to deal with a huge network, and more and more the tool was used exclusively by the programming and knowledge engineering team, and not by the domain experts/teachers. We realized that the expert only needed to focus on the local area of nodes connected to the node being focused on, and that in this situation the expert did not benefit much from the big picture view of the entire network (or a region of it) that the tool provided. We concluded that it would require less cognitive load if the authors just focused on each individual relationship: X support/refutes Y, and we moved to an authoring tools which portrayed this in a tabular format.

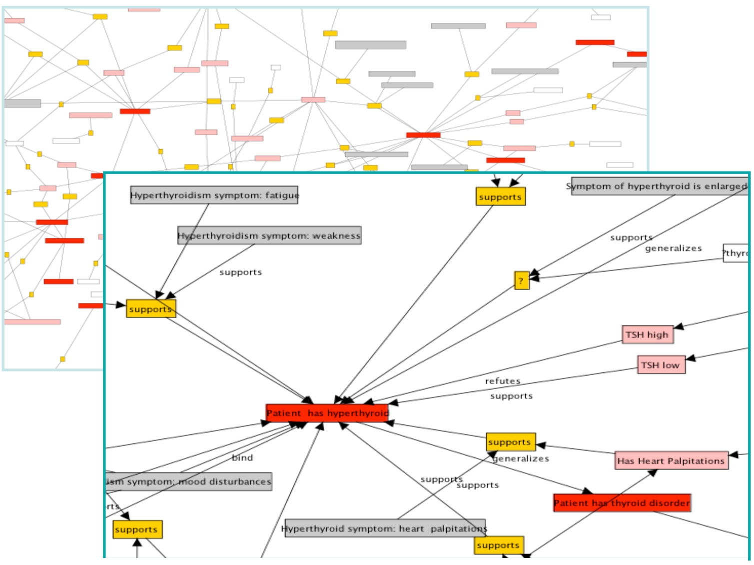

Figure 2: Network-based authoring tool

A table-based representation. The second tool was build using macros and other features available in Microsoft Excel (see Figure 3). The central piece of the tool was a table allowing the author to create Data->RelationshipType->Inference triplets (e.g. "high-temperature supports mono") (Figure 3A). For ease of authoring it was essential that the author choose from pop-up menus in creating relationships (which can be easily accomplished in Excel). In order to flexibly support the pop-ups, data tables were created with all of the options for each item in the triplet (Figure 3B). The same item of data (proposition) or inference (hypothesis) can be used many times, i.e. relationship is a many-to-many mapping. Authors could add new items to the tables in Figure 3B and to the list of relationships in Figure 3A (A and B are different worksheets in the Excel data file). Using the Excel features the author can sort by any of the columns to see, for example, all of the hypotheses connected to an observation; or all of the observations connected to a hypothesis; or all of the "refutes" relationships together. This method worked well for a while. But as the list of items grew in length the pop-up-menus became unwieldy. Our solution to this was to segment them into parts where the author chooses *one* from list A, B, C, or D and *one* from list X, Y, or Z (this modified interface is not shown). The complexity increased as we began to deal with intermediate inferences which can participate in both the antecedent *and* the consequent of relationships, so these items needed to show up in both right hand and left hand pop up menus. As we began to add authoring of case-values to the tool, the need to maintain unique identifiers for all domain "objects" was apparent, and the system became even more unwieldy.

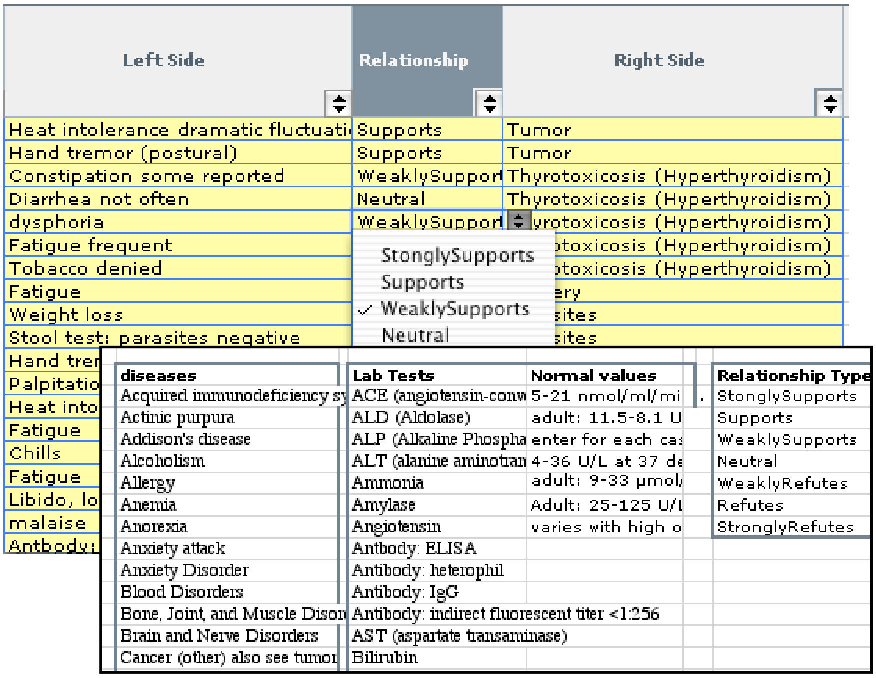

Figure 3 A&B: Table-base authoring tool

A form-based representation. Eventually we conceded that we needed to invest in building a "real" full fledged authoring tool. Our data model of objects, attributes, and relationships is best conceptualized in terms of relational database tables. Because of its abilities in rapid prototyping of user interfaces we used FileMaker Pro. Figure 4 shows some of the screens from the resulting authoring tool, which we have been successfully using over the last year with SMEs. The figure shows the form view and the list view for the propositions database. We have similar screens for the other objects: cases, relationships, and case values. We are able to add "portal" views so that while inspecting one type of object you can see and edit objects of other types that are related to the focal object. Figure 4 shows that while editing propositions the author can edit and manage relationships and case values also. Thus the author can get by using only the propositions screens in figure 4 and a similar but much simpler

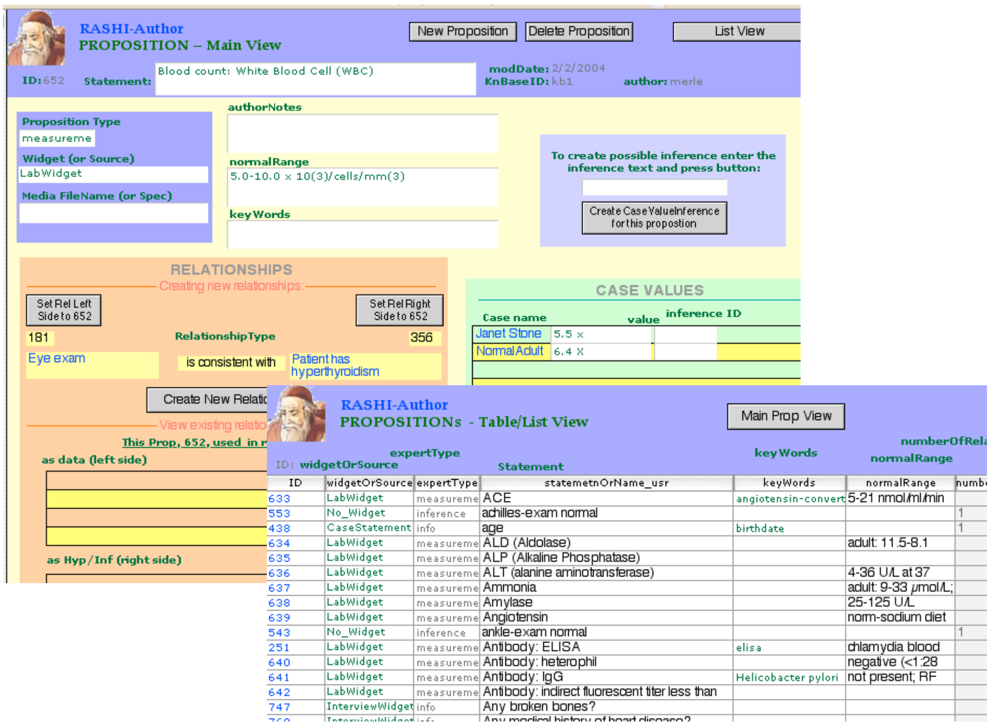

Figure 4 A&B: Final stage authoring tools

screen for cases. Creating fully functioning tools has allowed the expert to creatively author and analytically correct almost all aspects of the Human Biology cases, and participate with much more autonomy and depth (we are using the tool for the other domains as well). It has freed up the software design team from having to understand and keep a close eye on every aspect of the domain knowledge, and alleviates much of the time it took to maintain constant communication between the design team and the domain expert on the details of the content.

5. Discussion

Why did we bother to describe three versions of authoring tools when it was only the final one that was satisfactory? Stories of lessons learned from software development are rare, but the trial and error process can illustrate important issues. In our case this process has illustrated the importance of having SMEs involved in authoring tool design, and the importance of finding the right external representation for the subject matter content.

Comparison with other authoring tool projects. The Rashi authoring tools are relatively unique in that there is only one other project that deals with authoring tools for adaptive inquiry learning environments, the SimQuest/SMILSE project [10]. SimQuest takes a very different approach to authoring inquiry learning environments than Rashi. SimQuest focuses on runnable mathematical models, and supports students in learning science principles through experimentation. The SimQuest authoring environment supports the authoring of equations, graphical portrayals of situations, and the tasks and feedback messages needed in instruction. Rashi focuses on teaching inquiry skills and non-mathematical (symbolic) knowledge (as in biology and geology), and on case-based and rule-based expertise (the evidential relationships are simple rules). Thus the Rashi authoring tools show the application of authoring tools to a new type of domain. However, the elemental methods and interface features used by the Rashi authoring tools does not advance the state of the art beyond other systems (see [18]). However, as mentioned above, the vast majority of authoring tool projects do not focus on what it takes to create tools that can be used generally by SMEs, as we do. Other than this work, only in the Redeem project ([2] and other papers by Ainsworth) includes analyses of not only the successes, but also the ubiquitous problems encountered when employing SMEs to help build adaptive LEs. Redeem studies deal mostly with authoring instructional strategies, vs. our focus on complex subject matter content.

External Representations. We have also seen evidence that the representational formalism used in the authoring tool can affect its usability. The visual representations must match the deep structure of the knowledge in the tutor, must match the cognitive demands of authoring for the intended author characteristics, and must scale up to large content knowledge bases. Studies by Suthers et al. and Ainsworth et al. ([19] [20]) have shown that different external representations facilitate different tasks and internal representations for students using LEs. Similarly, our work has illustrated, albeit anecdotally, the differential effects of three external representations (network, table, and from-based) in knowledge acquisition tools.

References

[1] Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. Chapter 17 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publishers, Dordrecht.

[2] Ainsworth, S., Major, N., Grimshaw, S., Hayes, M., Underwood, J., Williams, B. & Wood, D. (2003). REDEEM: Simple Intelligent Tutoring Systems from Usable Tools. Chapter 8 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publishers, Dordrecht.

[3] Halff, H, Hsieh, P., Wenzel, B., Chudanov, T., Dirnberger, M., Gibson, E. & Redfield, C. (2003). Requiem for a Development System: Reflections on Knowledge-Based, Gen-

erative Instruction, Chapter 2 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publishers, Dordrecht.
[4] Shneiderman, B. (1998). *Designing the User Interface* (Third Edition). Addison-Wesley, Reading, MA, USA.
[5] Norman, D. (1988). *The Design of Everyday Things*. Doubleday, NY.
[6] Mayer, R. (1998). Cognitive, metacognitive, and motivational aspects of problems solving. Instructional Science vol. 26, p. 49-63.
[7] Duell, O.K. & Schommer-Atkins, M. (2001). Measures of people's belief about knowledge and learning. Educational psychology review 13(4) 419-449.
[8] Lajoie, S. (Ed), 2000. *Computers as Cognitive Tools Volume II*. Lawrence Erlbaum Inc.: New Jersey
[9] White, B., Shimoda, T., Frederiksen, J. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: computer support for metacognitive development. *International J. of Artificial Intelligence in Education* Vol. 10, 151-1182.
[10] van Joolingen, W., & de Jong, T. (1996). Design and Implementation of Simulation Based Discovery Environments: The SMILSE Solution. *Jl. of Artificial Intelligence in Education* 7(3/4) p 253-276.
[11] Lajoie, S., Greer, J., Munsie, S., Wikkie, T., Guerrera, C., Aleong, P. (1995). Establishing an argumentation environment to foster scientific reasoning with Bio-World. Proceedings of the International Conference on Computers in Education, pp. 89-96. Charlottesville, VA: AACE.
[12] Suthers, D. & Weiner, A. (1995). Groupware for developing critical discussion skills. *Proceedings of CSCL '95, Computer Supported Collaborative Learning*, Bloomington, Indiana, October 1995.
[13] Scardamalia, Marlene, and Bereiter, Carl (1994). Computer Support for Knowledge-Building Communities. The Journal of the Learning Sciences, 3(3), 265-284.
[14] Woolf, B.P., Marshall, D., Mattingly, M., Lewis, J. Wright, S. , Jellison. M., Murray, T. (2003). Tracking Student Propositions in an Inquiry System. Proceedings of Artificial Intelligence in Education, July, 2003, Sydney, pp. 21-28.
[15] Murray, T., Bruno, M., Woolf, B., Marshall, D., Mattingly, M., Wright, S. & Jellison, M. (2003). A Coached Learning Environment for Case-Based Inquiry Learning in Human Biology. Proceedings of E-Learn 2003. Phoenix, Arizona, November 2003, pp. 654-657. AACE Digital Library, www.AACE.org.
[16] Bruno, M.S. & Jarvis, C.D. (2001). It's Fun, But is it Science? Goals and Strategies in a Problem-Based Learning Course. J. of Mathematics and Science: Collaborative Explorations.
[17] Kolodner, J.L, Camp, P.J., D., Fasse, B. Gray, J., Holbrook, J., Puntambekar, S., Ryan, M. (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by Design(tm) Into Practice. *Journal of the Learning Sciences*, October 2003, Vol. 12: 495-547.
[18] Murray, T., Blessing, S. & Ainsworth, S. (Eds) (2003). *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*. Kluwer Academic Publishers, Dordrecht
[19] Suthers, D. & Hundhausen, C. (2003). An empirical study of the effects of representational guidance on collaborative learning. *J. of the Learning Sciences* 12(2), 183-219.
[20] Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education* vol. 33 pp. 131-152.