Authoring Learning Environments Using an Embedded Pedagogical Model

Smadar Kedar, David Guralnick, Ann Holum
Ray Bareiss, Richard Beckwith, Alex Kass

Institute for the Learning Sciences
Northwestern University,
1890 Maple Ave.
Evanston, IL, 60201 USA
Email: {kedar,guralnick,holum,bareiss,beckwith,kass}@ils.nwu.edu

Abstract: For computer-based learning environments to make a major impact on education, we need tools to scale up construction of these systems in order to deploy the technology. On the one hand, commercial authoring tools can be used by educators to ease construction of learning environments. However, these tools provide no guidance in terms of pedagogy. On the other hand, construction tools for intelligent tutoring systems (ITSs) can assist educators in authoring tutoring systems by helping them encode instructional content, but many require authors to specify *explicit* models such as teaching strategies and student models. Although the resulting software is pedagogically principled, encoding these models is cumbersome. For educators to create quality tutoring programs with relative ease, our approach combines the commitment to pedagogy of the ITS tools, with the ease of authoring of off-the-shelf tools, by embedding a pedagogical model *implicitly* in a tool. The embedded model ensures that the resulting learning environments will be pedagogically sound, while freeing educators to concentrate on curriculum content. We demonstrate this with a case study of a prototype authoring tool, Investigative Goal-Based Scenario (IGBS) Builder, intended for educators building multimedia learning environments using an embedded pedagogical model of situated instruction.

1. Introduction and Motivation

For computer-based learning environments to make a major impact on education, we need tools to scale up construction of these systems in order to deploy the technology. Educational software today is frequently a commercially available prepackaged unit of instruction, allowing educators no control in changes in the content or in teaching style. Commercially available authoring tools [e.g. Hypercard™ or Macromedia Director™] provide opportunities for creating programs tailored to instructional needs, but these tools provide educators with no guidance in terms of selecting or structuring pedagogical content [Ed-Media, 1994]. As a result, the software that is created may not be educationally sound. Recent research on model-based *automated generation of interfaces*, e.g. [Puerta, *et. al.,* 1994; Szekely, *et. al.,* ,1993], intends to alleviate some of the limitations of commercial authoring tools by generating interfaces automatically from high-level models. However, most of these tools aim for generality, creating interfaces with limited, generic styles (usually form- or menu-based), not specifically tailored to instruction.

One aim of intelligent tutoring systems (ITSs) is to instruct using sound pedagogical theories. In order to make a significant impact in the classroom, the future of these systems lies in scaling up their construction. To address this issue, some researchers have created construction tools for ITSs. [e.g. Murray and Woolf, 1992; Major, 1993, Frasson, *et. al.,* 1992, Russell, *et. al.*, 1988; Macmillan, *et. al.*, 1988; Towne, *et. al.*, 1990]. Most of these tools are generic shells that allow educators, who need not be programmers, to specify the knowledge base of an ITS. However, in order to build a tutoring system, a serious knowledge representation effort is required. An educator needs to encode not only instructional content, but also *explicit* models of teaching strategy, the student, and so on. For many potential authors, this is simply not practical. Although pedagogically principled, encoding these models along with content makes building tutoring systems is cumbersome.

If educators are to create quality learning environments with relative ease, we need to combine the commitment to pedagogy of the ITS construction tools with the ease of authoring of the off-the-shelf tools. Our approach is to design authoring tools whose pedagogical model is *implicit*; that is, embedded in the tool. Limiting the scope of a tool to a specific model gives it its power. This is the common 'generality versus power' tradeoff found in AI, and more specifically in tools such as knowledge acquisition tools based on role-limiting methods [McDermott, 1988]. We claim that embedding the model implicitly in the tool (1) ensures that the resulting learning environments will be pedagogically sound, and (2) frees educators to concentrate solely on curriculum content, thus easing construction of learning environments. We demonstrate these claims with a case study of a prototype authoring tool, Investigative Goal-Based Scenario (IGBS) Builder, intended for educators building multimedia learning environments using an embedded pedagogical model of situated learning.

The rest of the paper is organized as follows: Section 2 describes the pedagogical model embedded in the tool. Section 3 describes a sample interaction of constructing a tutoring system with the authoring tool, IGBS Builder. Section 4 presents an evaluation of the tool and a discussion of future work. Section 6 concludes.

2. The Pedagogical Model: Goal-Based Scenarios

An authoring tool with an embedded pedagogical model is only as good as the model embedded in it. The model must be pedagogically sound and encompass a significant class of instruction. We discuss these two qualities in terms of the pedagogical model of *Goal -Based Scenarios* [Schank et. al., 1993], which underlies the Investigative Goal-Based Scenario Builder tool. The central tenet of Goal-Based Scenarios is that skills and concepts are best learned when embedded within a problem-solving task. In a problem-solving task, information is more likely to be retained by the student and transferred to similar problems, since its relevance and connection to the problem can be more easily understood [Adams, *et. al.*, 1988]. This position is shared by other approaches to situated instruction, notably *cognitive apprenticeship* [Collins, et. al., 1989] and *anchored instruction* [Bransford, et. al., 1990]. Goal-Based Scenarios are a specialized form of situated instruction, addressing, in particular, the motivational needs of the student. A GBS can be thought of as a play, where the student is the main actor, and some problem or conflict needs to be resolved, e.g., "Broadcast News" [Kass, *et. al.*, 1994], "Creanimate" [Edelson, *et. al.*, 1993]. More specifically, in a GBS, there is a (1) *mission:* the student assumes a motivating simulated role related to a real-world task, and is given a goal to achieve which is intrinsically motivating. For example, students may play the role of a forensic chemist helping to solve a crime, as a way of engaging them to learn aspects of chemistry. In addition, there are (2) *pedagogically-structured activities:* what to do and what is learned are tightly linked. In order to perform the activities to accomplish the goal, the student will *need* to acquire the intended skills and concepts (and therefore is assumed to be highly motivated to do so). To continue in our example, in order to interpret the results of a test that evaluates toxicity in a liver sample, students would need to learn relevant aspects about chemical compounds. Finally, there is (3) *tutorial support:* expert advice and guidance are available 'just-in-time', and are strongly linked to the activities of the GBS, so that the student is given feedback both about mission and what he or she needs to learn. For example, errors in evaluating the evidence may lead to a dangerous criminal's mistaken release. Currently, our tool, called Investigative Goal-Based Scenario (GBS) Builder, embodies a specialized model of Goal-Based Scenarios, where the mission of the student is specialized to investigation tasks (as opposed to design or simulation tasks, say).

3. Investigative Goal-Based Scenario Builder

3.1. Overview

In this section we describe how Investigative GBS Builder aids in authoring computer-based learning environments, and discuss to what extend the tool satisfies our claims that embedding the model implicitly in the tool (1) ensures that the resulting learning environments will be pedagogically sound, and (2) frees educators to concentrate solely on curriculum content, thus easing construction of learning environments.

IGBS Builder's key task is to use the embedded model of an investigative GBS to provide strong guidance to the teacher or curriculum designer about how to structure the content of what to teach around an engaging task, and how to implement it as an interactive multimedia program, so that the resulting program will be a GBS. It does that by a step-by-step elicitation of the details of activities the student is to perform, and connects these to the required teaching content. The resulting interactive program is designed to engage the student in playing a role and performing a task through conducting simulated activities such as asking questions of video experts and performing laboratory tests by running animations and reviewing video-based evidence. Learning the intended teaching content thus occurs in a situated context.

As mentioned above, IGBS Builder embodies a specialized model of Goal-Based Scenarios, where the mission of the student is investigation (see Section [Evaluation and Futur Work] for a discussion of future work to expand the scope by a suite of tools). An *investigative* GBS is structured around a goal the student is asked to achieve which requires performing an investigation. More specifically, the overall mission of an investigative GBS model is divided into five phases: The *Problem* phase provides the student with their role and goal (i.e. a problem to solve); in the *Do* phase, the student performs the investigation; deciding on a solution to the problem occurs in the *Decide* phase; communicating the decision occurs in the *Communicate* phase; and in the *Wrap-Up* phase, the student sees the consequences of their investigation and decision.

The tool could, in theory, present the abstract model of investigative GBS to the designer (i.e. educator), and allow the designer free reign in implementing that model as an interactive program. However, this would be difficult since there is a vast design space for how to instantiate the model. The tool therefore guides the designer not only with a model, but also with an exemplar GBS that has implemented that model successfully. The designer uses the exemplar to make the model more concrete (as in case adaptation in case-based reasoning, e.g. [Kolodner, 1993], [Bell, Kedar & Bareiss, 1994]]. The exemplar currently in the tool is an investigative GBS called "Sickle Cell Counselor" [Bell, Bareiss & Beckwith, 1993], built for the Museum of Science and Industry in Chicago. This GBS teaches basic concepts of genetics, and in particular, of Sickle Cell Disease, by allowing the museum visitor to assume the role of a genetic counselor, with the goal of advising clients who are considering having children, about the risk of Sickle Cell Disease for their offspring.

The exemplar helps the designer by illustrating the phases of the investigative GBS using Sickle Cell Counselor, and this makes the investigative GBS model more concrete. For example, in Sickle Cell Counselor, the phases are instantiated as follows: In the *Problem* phase, clients seeking advice about Sickle Cell Disease appear on video, and the student is given his or her role and goal. Investigation occurs in the *Do* phase, for example, the student performs an electrophoresis test to establish the clients' genotypes. In the *Decide* phase, a decision is reached by calculating the probable genotypes of the client's offspring using an animated Punnett Square. The student conveys the decision to the clients in the *Communicate* phase. The *Wrap-Up* phase offers feedback for the student's actions, demonstrating the ramification of students' final decision: the clients return "a year later" to talk about whether or not they had children, and what the ultimate outcome of the disease was for them. Preliminary evaluations show that Sickle Cell Counselor is an effective GBS: the student's role and task is engaging, and performing the activities leads the student to acquire the intended teaching content [Bell, Bareiss & Beckwith, 1993].

The exemplar is not useful for guiding the authoring unless it is clear how it is connected to the model. For that, the tool provides a *map* of how each step in the interactive program fits into a phase and its subphases. The tool also provides *design rationale* to explain how each individual screen component fits into the whole.

Investigative GBS Builder, then, uses the embedded pedagogical model with strong guidance for authoring: it provides both an abstract model of an investigative GBS, an exemplar, and a link between them. In the next two sections we provide a sample interaction with the tool. We illustrate the interaction in terms of an example of a GBS authored with the tool by a pair of middle school teachers (more on the evaluation and preliminary user testing of the tool in Section 4). The teachers wanted to build a GBS to teach chemistry skills and concepts, and decided to create an "Arson Investigator" GBS. In that GBS, the students' role is a forensic chemist, and his or her goal is to investigate the cause of a fire to determine if it was arson. As a byproduct of performing the activities of the GBS and asking questions of experts, the student learns the intended chemistry skills and concepts.

3.2. Design Support

Authoring a GBS is divided broadly into two stages: *design* and *implementation* (often iterative). In the design stage, the designer sketches the broad brush outline of the three key aspects of the GBS, and makes sure they satisfy certain constraints: (1) The designer matches what to teach to a student's *mission* in the GBS, and ensures it is motivating (arson investigation); (2) the designer sketches the top-level outline of the *pedagogically-structured activities,* ensuring that the activities *require* the student to seek out what is being taught; and (3) the designer sketches the outline for *tutorial support* by assigning characters in the GBS to various categories of support.

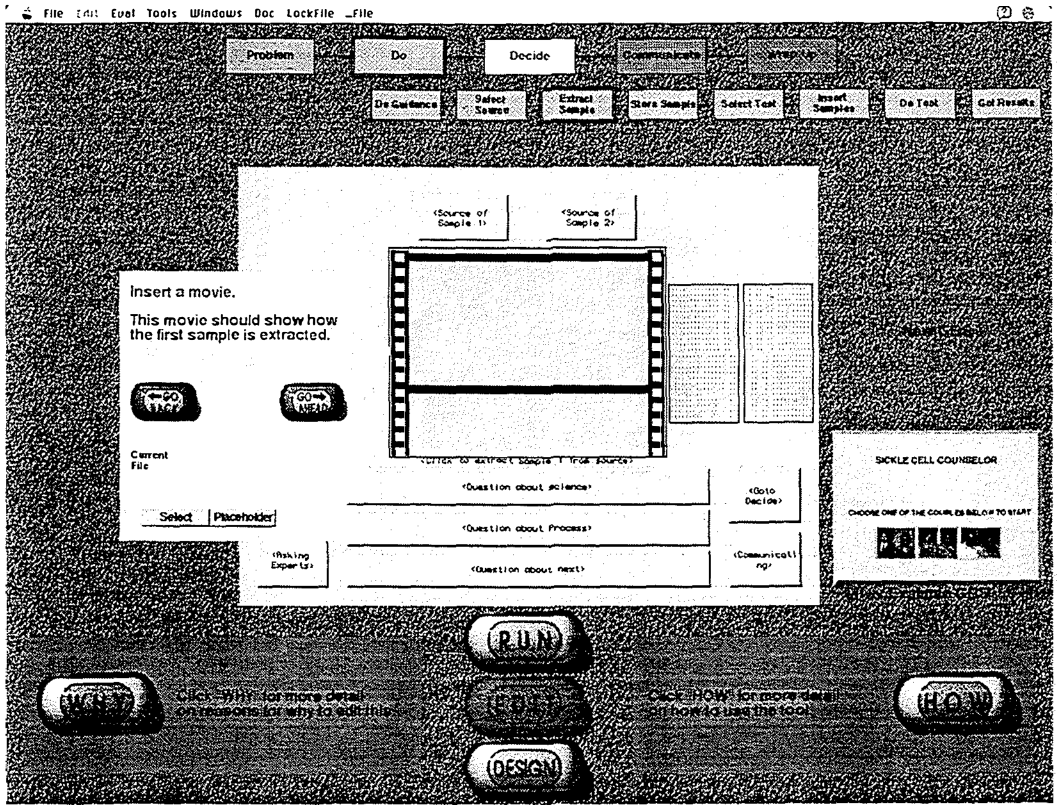

Figure 1: An Empty Template for the Extract Sample Subphase of the Do Phase

Conceptual design is an art. Currently, designers work with people experienced in designing GBS's during the initiation of a project, and iterate through many different designs until one is found where the match of content to mission is appropriate, and the mission is motivating. An on-line *conceptual design document* is used to structure the resulting information and relate it to the model. Currently, the design document is not connected to the rest of the tool. However, we are working on incorporating it, and automatically migrating concepts from the conceptual design stage to the implementation stage (see Section [Design Support]).

3.3. Implementation Support

In the implementation stage, the tool guides the designer to implement the activities of the GBS as interactive actions in a multimedia program. The tool constrains these to be *pedagogically structured activities* by guiding the designer to specify teaching content which is indexed to each activity in the following ways: First, the designer specifies the details of an activity (e.g. collecting a burnt drape sample to determine arson). Then, the designer specifies a set of initial and follow-up questions that are likely to be raised in the student's mind during the activity, and provides video clips of experts answering them. This supplies the intended teaching content to be delivered 'just in time'. In addition, the tool guides the designer in implementing the *tutorial support* by specifying, for each of the characters in the GBS, movie clips that deliver guidance, critique, and feedback at key points in the program. We illustrate these features of implementing the GBS in the process of defining the "Extract Sample" subphase of the Do Phase. The activity of "Extract Sample" is to obtain a physical sample is obtained in order to analyze it as part of the investigation. In Arson Investigator, a sample of a burnt drape from an apartment fire is collected in order to analyze whether there was an accelerant (e.g. fuel) in it, which is possible evidence that the fire was intentional.

The tool interacts with the designer to implement the GBS screen by screen as follows [see Figure 1]: At the top of the screen, the tool provides a *map* of the investigative GBS model in terms of its phases and subphases, highlighting the current phase and subphase being implemented. In the center is the *work area*, displaying the current screen of the GBS being created. On the left, a *dialog window* guides the designer to instantiate each component of the screen, describing how it implements the current phase and subphase. To the

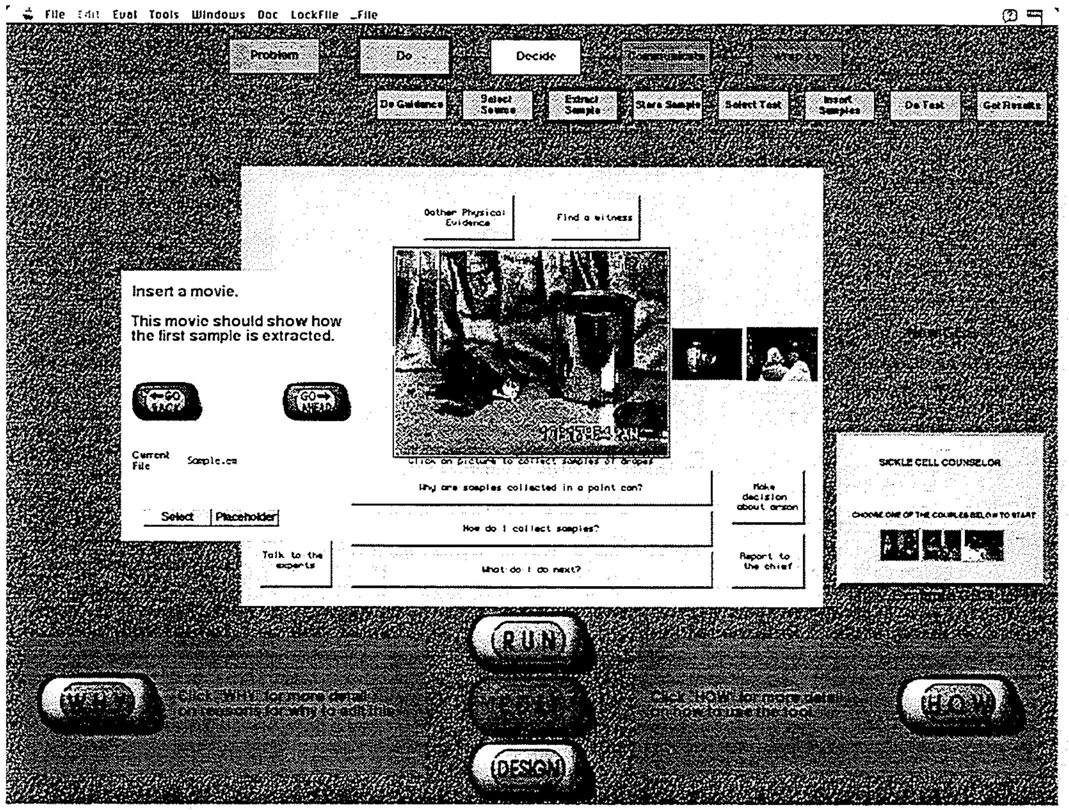

Figure 2: Implementing Extract Sample subphase of Do Phase for Arson Investigator GBS

left, clicking on the small GBS icon lets the designer view the corresponding phase and subphase of the exemplar GBS (here, Sickle Cell Counselor). The buttons below provide help on *design rationale* (WHY) and *guidance* in using the tool (HOW). The center bottom *mode* buttons allow the designer to switch to RUN mode in order to 'play' and test the portion of the GBS built so far, or to the DESIGN mode in order to review the design. EDIT mode is the name of the implementation mode the designer is in right now.

Below is a sample interaction (corresponding to two steps in the dialog) which implements part of the Extract Sample subphase of the Do phase, demonstrating the creation of pedagogically-structured activities and tutorial support:

Tool: `In the Extract Sample subphase, the student selects the source of the sample, extracts the sample, and stores the sample for analysis.`

«Designer switches to viewing SCC»

Tool: `In SCC, the student selects blood samples from each client, extracts the sample by drawing blood using a syringe, and places each blood sample in a test tube. Now you will define the screens and their components for the Extract Sample subphase of Arson Investigator.`

Next, the dialog window points to each empty screen component in turn, guiding the designer to fill it with appropriate graphical items (text, graphics, video clips, etc.). In the case of graphics or video, the designer either can fill in the appropriate item if it already exists in a library, or if unavailable on-line, the designer can specify a "placeholder", a to-do note which holds the place of the item until he or she shoots the appropriate video or creates the needed graphic and inserts it. The designer of Arson Investigator has already created a short Quicktime™ video clip showing a burnt drape sample being collected from the apartment fire. In the following interaction, the designer selects that movie to insert for the Extract Sample activity in the GBS. [See Figure 2] for the implementation of one screen of this subphase of the Do phase.

Tool: Insert a movie. This movie should show how the first sample is extracted.

Designer: (chooses a movie of a person collecting a sample of a burnt drape and storing it in a container).

We now demonstrate how the designer is guided to specify the teaching content related to this activity. Three questions, follow-up questions, and their answers (in video clips by experts) are elicited from the designer. The dialog window specifies three categories of questions related to the current activity: a question on basic concepts, a question about skills, and a questions about how to do the activity in the GBS. The three questions (along with their answers and follow-up questions) in Arson Investigator, indexed to the "extract sample" activity, are as follows:

Why are samples collected in a paint can?

How do I collect samples?

What do I do next?

Finally, we illustrate how the tool guides the designer in specifying one type of tutorial support: feedback about the ramifications of student actions. The designer is asked to specify three possible conclusions for the student to communicate (one correct, one incomplete, and one incorrect):

This was clearly a case of arson.

There was not enough evidence to conclude.

This was an accidental apartment fire.

In the Communicate Phase, the student is presented with a choice of these three conclusions. Each conclusion leads to a different ramification, and therefore different feedback to the student. In one of the problems presented in Arson Investigator, the investigation clearly demonstrates that there has been arson (e.g. the drape sample contained an accelerant, and witnesses reported valuable objects stolen from the apartment). If the student chooses the incorrect communication "This was an accidental apartment fire," the tool leads the designer through specifying the feedback. Possible feedback in Arson Investigator is a movie clip of a fire chief speaking to the student a 'year later', telling the student that based on the conclusion, an arson suspect was let go, and there has been a string of fires of a similar nature in the neighborhood since then.

4. Evaluation and Future Work

We first performed a formative evaluation [Nielsen, 1993] of the tool in order to improve it as part of iterative design. The tool was used by 20 graduate students in computer science and education, building 10 prototype GBS's. One successful example is Aztec P.I., in which the student, as a museum curator, is assigned to verify the authenticity of an Aztec artifact. The students reported feeling restricted in copying and editing an exemplar, so during the course of the evaluation we provided a facility to let them alter their GBS by adding or deleting any screen or component they wished. As a result, only 5 of the 10 GBS's kept the gist of an investigative GBS. One explanation is that it was more difficult to be self-disciplined in staying to the gist of the model than to have the tool enforce that discipline automatically. Among the lessons learned from the first evaluation was that although copying and editing an exemplar is too restrictive, allowing unconstrained changes to the interface can turn a GBS into an arbitrary (and possibly educationally unprincipled) program.

After nine months of development, we performed a second iteration of user testing with a group of eight elementary and middle school teaches from a school district in suburban Chicago. The teachers, who had little computer experience, used the tool in pairs to build four prototype GBSs over the summer. The four GBSs were: Arson Investigator for teaching chemistry concepts and investigative skills, Debt Buster for teaching mathematics by helping a teenager manage his debt; Kid Counselor for teaching social skills by helping youngsters reason through social problems, and Lend a Hand, for teaching accessibility issues for the handicapped by helping a person navigate in a wheelchair to a baseball game. It took the teachers, with significant ILS staff support, one and a half months, in pairs to build the prototype GBS (about a third of the content of a full GBS). We extrapolate that building a full GBS could take them nine person/months. This is still less than the one and a half person/years it tool to build Sickle Cell Counselor from scratch, using a programmer, a graduate student, and a knowledge engineer. During this evaluation the teachers could not make any unconstrained changes to the GBSs built with the tool. The quality of the resulting GBSs was good overall: all four GBSs kept to the model of the investigative GBS.

Through focus groups, surveys, and informal interaction during our evaluation, we learned a number of lessons regarding the overall effectiveness of the tool. First, teachers are expert at content, not necessarily software design. The teachers found the guidance in structuring the content useful (e.g. associating types of experts and questions with each activity in the tool) but needed even more guidance (e.g. software design considerations such as level of interactivity in designing a test). Second, although we provided some design support, the teachers found that in implementing a GBS they lost sight of the conceptual design, and could not 'see the forest for the trees'. The problem is that while strong guidance is being provided, it is often at an interface rather than a conceptual level. We provided the teacher hard copy design support documents, and the teachers themselves produced detailed outlines and spreadsheets to aid their design. In addition, we are working to make the on-line conceptual elicitation stage of the tool richer, and to migrate the information elicited automatically into the implementation [Bell & Kedar, 1995], [Bell, 1995]. Third, while the teachers found the guidance from embedded model useful, they found the Sickle Cell Counselor exemplar especially useful. However, as much as they found these helpful, they also felt constrained by them. The current model and exemplar are only one instance of what an investigative GBS could be. The tool needs to incorporate a broader range of investigation models (e.g. a factual investigation versus a laboratory investigation), and a library of prototypes of these models.

5. Conclusions

Our evaluation to date does not fully substantiate our claims that the tool (1) ensures that the resulting learning environments are pedagogically sound, and (2) frees educators to concentrate on curriculum content, but the results are in the direction we expect. In future evaluations, we intend to investigate students' use of GBSs resulting from the tool, and study more closely how educators use the tool to build complete GBSs.

It is clear that educators will want to build GBSs that support tasks broader than what this tool can support. The IGBS Builder is the first of several GBS Builder tools currently being developed in our lab, each supporting the construction of a different class of GBSs. The approach currently being pursued at the Institute for the Learning Sciences is to build a suite of tools, each supporting the construction of a different class of GBSs. The other GBS Builders are intended for tasks such as everyday procedures, interpersonal behavior, running an organization, or evidence-based reporting [Schank, forthcoming]. The IGBS Builder is thus the first of several GBS Builder tools, and provides an important first step towards deploying computer-based learning environments in the classroom.

References:

Adams, L., Kasserman, J., Yearwood, A., Perfetto, G., Bransford, J., and Franks, J. (1988). The effects of facts versus problem-oriented acquisition. In Memory and Cognition, 16, 167-175.

Ed-Media 1994 Proceedings. Vancouver, B.C., Canada. June 25-30, 1994. The Association for the advancement of Computing in Education.

Bell, B.L. (1995). Toward Modularity in Specializing Authoring Tools. In *Proceedings of the Workshop on Authoring Shells for Intelligent Tutoring Systems, 7th World Conference on Artificial Intelligence and Education*, Washington, DC., pp. 4-9.

Bell, B.L., Kedar, S.T. (1995). When Less is More: Supporting Authoring and Interface Building via Special-Purpose Task Models. In *Proceedings of the 7th World Conference on Artificial Intelligence and Education*, Washington, DC., pp. 533-540.

Bell, B.L., Bareiss, R., and Beckwith, R. (1993/1994). Sickle Cell Counselor: A Prototype Goal-Based Scenario for Instruction in a Museum Environment. *Journal of the Learning Sciences, 3*(4).

Bell, B.L., Kedar, S., and Bareiss, R. (1994). "Interactive Model-Driven Case Adaptation for Instructional Software Design." A poster in *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society,* Atlanta, GA, 1994.

Bransford, J.D., Sherwood, R.D., Hasselbring, T.S., Kinzer, C.K., and Williams, S.M. (1990). Anchored instruction: why we need it and how technology can help. In R. Spiro and D. Nix (Eds.), *Cognition, Education, and Multimedia: Exploring Ideas in High Technology.* Lawrence Erlbaum: Hillsdale, N.J.

Collins, A., Brown, J.S. and Newman, S.E. (1989) Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing and Mathematics. In L. Resnick, Ed. *Knowing, Learning, and Instruction: Essays in honor of Robert Glaser.* Erlbaum: Hillsdale, N.J.

Edelson, D.C. (1993). Learning From Stories: Indexing and Reminding in a Socratic Case-Based Teaching System for Elementary School Biology. Unpublished Doctoral Dissertation. The Institute for the Learning Sciences, Northwestern University. Evanston, Illinois.

Frasson, Gauthier, McCalla, Eds. (1992). *Proceedings of Intelligent Tutoring Systems Conference '92*, Montreal, Quebec, Canada, June, 1992.

Kass, A., Dooley, S., Luksa, F, and Conroy, C. (1994). Using Broadcast Journalism to Motivate Hypermedia Exploration. Educational Multimedia and Hypermedia Annual, 1994. Association for the Advancement of Computing in Education, Charlottesville, VA, 1994.

Kedar, S. T., and Bell, B. L. (1995). Teachers in Charge: Model-Based Authoring of Educational Software. A poster in *Proceedings INTERCHI '95* , Denver, CO.

Kolodner, J.L. (1993). *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

Bell, B., & Korcuska, M. (1995). The Goal-Based Scenario Builder: Experiences with Novice Instructional Designers. Presented at the *American Educational Research Association.* San Francisco, CA, April, 1995.

Macmillan, S., Emme, D., and Berkowitz, M. (1988). Instructional planners: Lessons learned. In Psotka, Massey, and Mutter (Eds.), *Intelligent Tutoring Systems, Lessons Learned.* Hillsdale, NJ: Lawrence Erlbaum.

Major, N. (1993). Reconstructing Teaching Strategies with COCA. In *Proceedings of Artificial Intelligence in Education.* Edinburgh, Scotland, August 23-27, 1993.

McDermott, J. (1988). Preliminary steps toward a taxonomy of problem-solving methods. In S. Marcus (Ed.), *Automating Knowledge Acquisition for Knowledge-based Systems.* pp. 120-146. Boston: Kluwer Academic Publishers.

Murray, T., and B. P. Woolf (1992). Results of Encoding Knowledge with Tutor Construction Tools. In *Proceedings of the Tenth National Conference on Artificial Intelligence,* San Jose, CA, July, 1992.

Nielsen, J. 1993). *Usability Engineering.* AP Professional, 1993.

Puerta, A.R., Eriksson, H., Gennari, J.H., and Musen, M.A. (1994). Model-Based Automated Generation of User Interfaces. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, August, 1994.

Russell, D., Moran, T.P., and Jordan, D.S. (1988). The instructional design environment. In Psotka, Massey, and Mutter (Eds.), *Intelligent Tutoring Systems, Lessons Learned.* Hillsdale, NJ: Lawrence Erlbaum.

Schank, R.C. (1993). Goal Based Scenarios. *The Journal of the Learning Sciences*, *3*(4), 1993.

Schank, R.C. (Forthcoming). Eight GBS Tools. Forthcoming technical report of the Institute for the Learning Sciences.

Schank, R.C., Fano, A., Bell, B.L. , and Jona, M.Y. (1993). The Design of Goal Based Scenarios. *The Journal of the Learning Sciences*, *3*(4), 1993.

Szekely, P., Luo, P., and Neches, R. (1993). Beyond Interface Builders: Model-Based Interface Tools. In *INTERCHI '93*, (Amsterdam, April, 1993), ACM Press, pp.383-390.

Towne, D.M., Munro, A., Pizzini, Q.A. Surman, D.S., Coller, L.D., Woguliz, J.L. (1990). Model-Based Tools for Simulation-Based Training. In *Interactive Learning Environments.* 1:1, pp. 33-50.

Acknowledgments:

Thanks to the IGBSB team for building the tool, and the ILS graduate students and Wilmette School District #39 teachers and administrators for helping us evaluate it.