Building a Community Memory for Intelligent Tutoring Systems[1]

Beverly Woolf and Pat Cunningham†
Department of Computer and Information Science,
University of Massachusetts, Amherst, Massachusetts 01003
†The Hartford Graduate Center, Hartford, Conn 06101

Abstract

This article discusses the need for multiple experts to work together to develop knowledge representation systems for intelligent tutors. Three case studies are examined in which the need for a pragmatic approach to the problem of knowledge acquisition has become apparent. Example methodologies for building tools for the knowledge acquisition phase are described including specific tasks and criteria that might be used to transfer expertise from several experts to an intelligent tutoring system.

I. A Community Memory

Building intelligent tutoring systems requires community knowledge, i.e., multiple experts working together to encode individual expertise in an intelligent tutor. This knowledge acquisition phase might span months or years. Thus, we need a framework to simplify *changing knowledge* in the tutors as well as a suite of programming tools for browsing and summarizing knowledge, for tracing and explaining the student model, and for tracking reasoning about teaching strategies. In short, tools and methodologies are needed that can be used specifically for knowledge acquisition activities within an intelligent tutor. In this paper we share our experience of building three intelligent tutors and describe the criteria for, and in some cases, the emerging tools used within this acquisition process.

The concept of a community memory for intelligent tutors reflects the fact that knowledge of tutoring is often distributed, incomplete, and acquired incrementally [Bobrow, Mittal and Stefik, 1986] and thus requires contributions from several experts. This is especially true in tutoring systems because the domain expert, cognitive scientist, and teaching expert are typically not the same person. Given multiple experts who contribute to building the system and the need for a large amount of testing and modification to fine tune the tutor, completion of a tutor can not be the "final" step in development of a single system, but rather must be a forcing function between the completion of one system and the beginning of another. A completed knowledge base provides grit for our collective grinder, forcing us to further clarify and amplify teaching and learning knowledge and to improve communication between those experts who contribute to it.

Articulating and incorporating communal knowledge into a tutor reveals a great deal about each area of expertise and about the tools used by the experts to perform problem solving in the domain. For example, building the boiler tutor described in Section 2.1 indicated several weaknesses in the tools available to industrial boiler operators. We therefore developed simulation tools, including abstract meters and trends (Figure 1) that might ultimately be integrated into the equipment used by boiler operators. Similarly, in building a geometry tutor [Anderson, Boyle, and Yost, 1985] provided an environment that would be a valuable aid to motivated learners, even without help from any on-line tutor. Anderson introduced visualization and forward and backward reasoning templates that would facilitate geometry problem-solving independent of teaching media.

In the next section, we briefly describe our three intelligent tutors and in Section III indicate some methodologies for how knowledge can be acquired from multiple experts to build additional tutors.

II. CASE STUDIES

A. RBT for Teaching Complex Industrial Processes

The first tutor to be discussed is fully implemented, tested, and now used for training in nearly 60 industrial sites across America. The Recovery Boiler Tutor, RBT[2], is described elsewhere [Woolf, Blegen, Jansen and Verloop, 1986], and will only be summarized here. It provides multiple explanations and tutoring facilities tempered to the individual user, a control room operator. The tutor is based on a mathematically accurate formulation of the boiler and provides an interactive simulation, (Figure 1) complete with help, hints, explanations, and tutoring.

[1]This work was supported in part by the Air Force Systems Command, Rome Air Development Center, Griffiss AFB, New York, 13441 and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). Partial support also from URI University Research Initiative Contract No. N00014-86-K-0764.

[2]RBT was built by J. H. Jansen Co., Inc., Steam and Power Engineers, Woodinville (Seattle) Washington and sponsored by The American Paper Institute, a non-profit trade institution for the pulp, paper, and paperboard industry in the United States, Energy Materials Department, 260 Madison Ave., New York, NY, 10016.

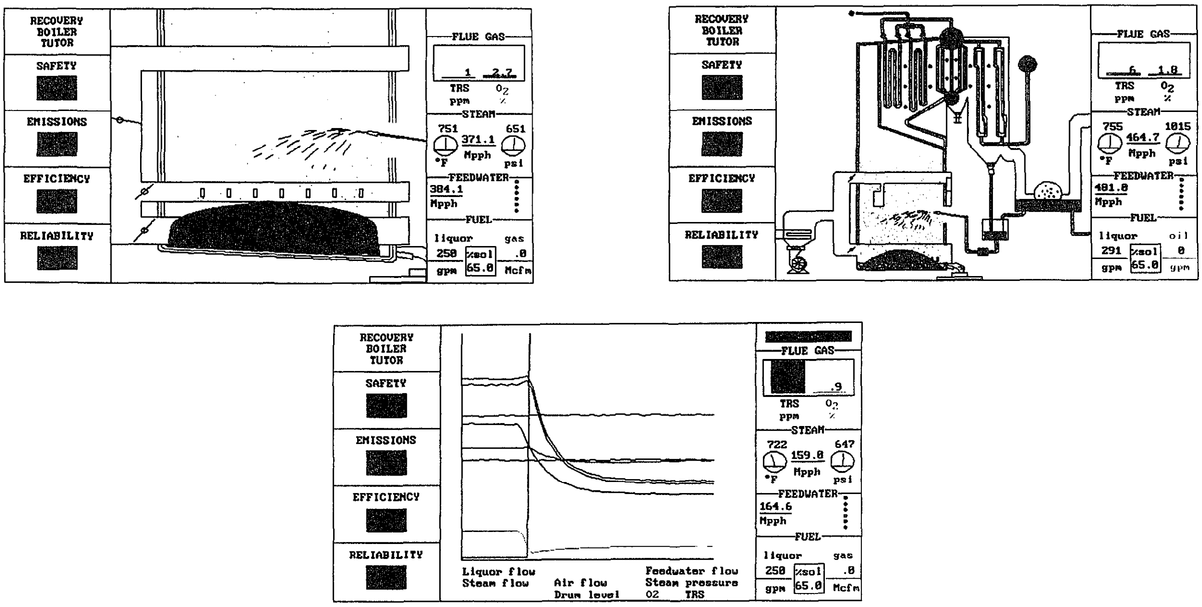

Figure 1: Several Views of the Recovery Boiler Tutor

The tutor challenges operators to solve boiler emergencies while monitoring their actions and advising them about the optimality of their solutions. The tutor recognizes less than optimal and clearly irrelevant actions and modifies its response accordingly. Operators can continue their freewheeling or purposeful problem-solving behavior while the tutor offers help, hints, explanations, and tutoring advice when needed or when requested. Operators gain experience in recognizing the impact of their actions on the simulated boiler and to react before the tutor advises them regarding potential problems.

Meters, as shown on the left side of screens in Figure 1, record the state of the boiler using synthetic measures for *safety*, *emissions*, *efficiency*, and *reliability* of the boiler. The meter readings are calculated from complex mathematical formulas that would rarely (if ever) be used by operators to evaluate the boiler. The meters have already proved effective as training aids in industrial training sites and could possibly be incorporated into actual control panels.

Operators have reported using the system as much as 70 hours in three months to practice solving emergencies. They handle the simulation with extreme care, behaving as they might if they were in actual control of the pulp mill panel, slowly changing parameters, checking each action, and examining several meter readings before moving on to the next action.

B. Caleb for Teaching a Second Language

Our second intelligent tutor teaches languages based on a powerful pedagogy called the "silent way"–a method developed by Caleb Gattegno. The system uses non-verbal communications within a controlled environment to teach Spanish [Cunningham, 1986]. It uses graphical Cuisenaire rods[3], to generate linguistic situations in which the rod plays various roles. For example, it is used as an object to be given or taken by a student, or it is used to brush teeth. As a new rod is presented, the student theorizes about what situation is encountered and types the appropriate phrase below the picture. In the case illustrated at the top of Figure 2 the tutor presents a rod in the center box. The student responds by typing the word for the new piece at the cursor. In the bottom figure, the tutor corrects a student who places an adjective before rather than after a noun. In this exercise, students might have classified the word "blanca" as an adjective referring to the size of the rod before knowing its meaning. The tutor does not clarify student's conjectures. Students can later change a hypothetical definition if in fact the new word turns out to define the color of the rod. Meanwhile, they will have learned to write the word, spell it, and place it correctly in a sentence.

C. ESE for Teaching Physics

A third tutor is now in the early implementation stage. It is part of a program to develop interactive and monitored simulations to teach physics at the high school or college level.[4] One of these tutors teaches the second law

[3]originally developed by Gattegno for teaching arithmetic

[4]These tutors are being built by the Exploring Systems Earth (ESE) consortium, a group of three universities working together to develop intelligent tutors. The schools include the University of Massachusetts, San Francisco State University, and the University of Hawaii.

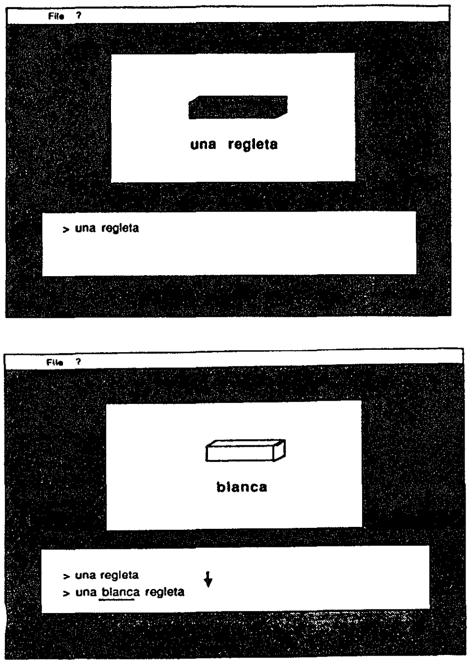

Figure 2: Caleb: A System for Teaching Second Languages

of thermodynamics[5] and provides a rich environment at the atomic level through which the principles of equilibrium, entropy, and thermal diffusion can be observed and tested [Atkins, 1982]. Students are shown (and are able to construct) collections of atoms that transfer heat to other atoms through random collision (see Figure 3). They can create areas of high-energy atoms, indicated by dark squares, along with variously shaped regions within which the high energy atoms can be monitored. Concepts such as temperature, energy density, and thermal equilibrium can be plotted against each other and against time.

The tutor uses all student activities – including questions, responses, and requests – to formulate its next teaching goal and activity. It uses student actions to determine whether to show an extreme or near-miss example, whether to give an analogy or whether to ask a question. To refine the tutor's response, we are now studying student misconceptions and common errors in learning thermodynamics and statistics.

III. Tools for Knowledge Acquisition

Given the complex heterogeneous nature of the knowledge required to build each of these systems, we need methodologies and tools to transfer teaching and learning knowledge from human experts to systems under construction. Few such tools exist.

[5]The second law states that heat cannot be absorbed from a reservoir and completely converted into mechanical work.

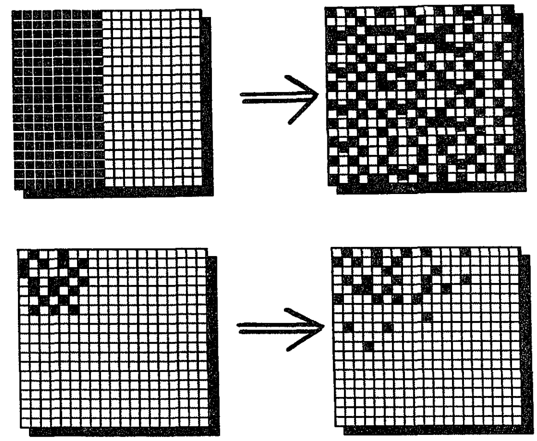

Figure 3: Systems Moving Towards Equilibrium

Expert system shells contain a framework for building knowledge bases about concepts and rules and for making inferences about them. However, they are limited as specific tools for designing and storing tutoring knowledge. They are frequently based on production rules and are limited in representing history and dependency of the tutoring interaction. Also, they inadequately represent tutoring and misconception knowledge such as how to reason about teaching strategies, how to update and assess student models, how to select a path through domain concepts, and how to remediate for misconceptions. In this section, we describe the criteria for developing tools specific to this knowledge acquisition process.

A. Environment Expert

The first expert needed to build an intelligent tutor is the environmental expert. This person often uses a majority of system memory [Bobrow, Mittal and Stefik, 1986] to provide an envelope within which students and system interact. The environment provides specific tools and operators for solving domain problems or for performing domain activities.

Environmental, teaching, cognitive, and domain expert contributions interact strongly with each other–especially those from the environmental expert. For example, a system that asks students to record entrance and exit angles for light in an optics experiment, assumes that the environment supplies such measuring devices.

The following criteria for developing a tutoring environment have begun to emerge:

1) Environments should be intuitive, obvious, and fun. Student energy should be spent learning the material, not learning how to use the environment [Cunningham, 1986].

For example, to indicate errors, express feelings or convey meaning, the second-language tutor's, visual activities mimic the human Silent Way teacher's gestures, facial expressions, and rods.

2) Environments should record not only what students do, but what they did, intended to do, might have forgotten to do, or were unable to do [Burton, in press]. Environments should provide a "wide bandwidth" within which multiple student activities can be entered and analyzed. For example, the Pascal tutor developed by Johnson and Soloway [1984] processed and analyzed an entire student program before offering advice.

3) Environments should be motivated by teaching and cognitive knowledge about how experts perform tasks and the nature of those tasks. For example, Anderson [1981] performed extensive research with geometry students before developing his geometry tutor interface, and Woolf et al. [1986] incorporated knowledge from experts with more than 30 years experience working with boiler operations before building the RBT interface.

4) Environments must maintain physical fidelity[6] [Hollan, Hutchins and Weitzman, 1984]. The RBT tutor presents a mathematically exact duplicate of the industrial process. It models and updates over 100 parameters every two seconds. Visual components of the industrial process such as alarm boards, control panels, dials, and reports are duplicated from the actual control room.

5) Environments should be *responsive, permissive,* and *consistent* [Apple, 1985]. They should target applications based on skills that people already have, such as moving icons, rather than forcing people to learn new skills. By responsive, we mean that student actions should have direct results–that students need not perform rigid sets of actions in rigid and unspecific order to achieve goals. By permissive, we mean that students may do anything reasonable and that multiple ways should exist for taking action. By consistent, we mean that moving from one application to another, (for example, from editing text to developing graphics), should not require learning new interfaces. All tools should be based on similar interface devices, such as pull-down menus or single and double mouse clicks.

No one environment is appropriate for every domain. We must study each domain to determine how experts function in that domain, how novices might behave differently, and how novices can be helped to attain expert behavior.

B. Teaching Expert

Acquiring sufficient and correct teaching expertise is a long term problem for builders of tutoring systems–in part, because sophisticated knowledge about learning, teaching, and domain knowledge remains an active area of research in most domains. Teaching expertise includes decision logic and rules that guide the tutor's intervention with the student. Tools to facilitate teasing apart and encoding teaching knowledge are just beginning to emerge. For example, we have developed a framework for managing discourse in an intelligent tutor [Woolf and Murray, 1987] that reasons dynamically about discourse, student response, and tutor moves.

The framework (Figure 4) reasons about which pedagogical response to produce and which alternative discourse move to make. It custom-tailors the tutor's response in the form of examples, analogies, and simulations. Discourse *schemas*, or collections of activities and response profiles, are responsible for actually generating system actions and for interpreting student behavior. The number and type of schemas used is dependent on context.

We used empirical criteria to define discourse schemas: tutoring responses were analyzed from empirical studies of teaching and learning and from general rules of discourse structure[Grosz and Sidner].

The framework is flexible and domain-independent; it is designed to be rebuilt – decision points and machine actions are modifiable for fine-tuning system response.

We are now using this framework to improve the physics tutor's response to idiosyncratic student behavior. Response decisions and machine actions, explicitly represented in the system, can be modified through a editor. Appropriate machine response can be assessed continuously and improved. In the long term, we intend to make this reasoning process available to human teachers, who can then modify the tutor for use in a classroom.

No single teaching strategy is appropriate for every domain. For example, Anderson et al. [1985] built geometry and Lisp tutors that responded immediately to incorrect student answers. These authors argued that immediate computer feedback was needed to avoid fruitless student effort.

This pedagogy was opposite to that used by Cunningham [1986] and Woolf et al. [1986]. These latter tutor's advice was passive, not intrusive. The strategy was to subordinate teaching to learning, and to allow students to experiment while developing hypotheses about the domain. The tutors guided their students toward developing their own intuitions, but did not correct them so long as their performance appeared to be attaining a precise goal.

In industrial settings, particularly, trainees must learn to generate multiple hypotheses and to evaluate their own performance based on how their actions affect the industrial process. For example, no human tutor is available during normal boiler operation.

C. Cognitive Expert

At present, the role of the cognitive scientist is incompletely understood; in part, this expert seeks to discover how people learn and teach in a given domain. For example, cognitive science research in thermodynamics will enable systems to recognize common errors, tease apart probable misconceptions, and provide effective remedia-

[6] Fidelity measures how closely simulated environments match the real world. High fidelity identifies a system as almost indistinguishable from the real world.

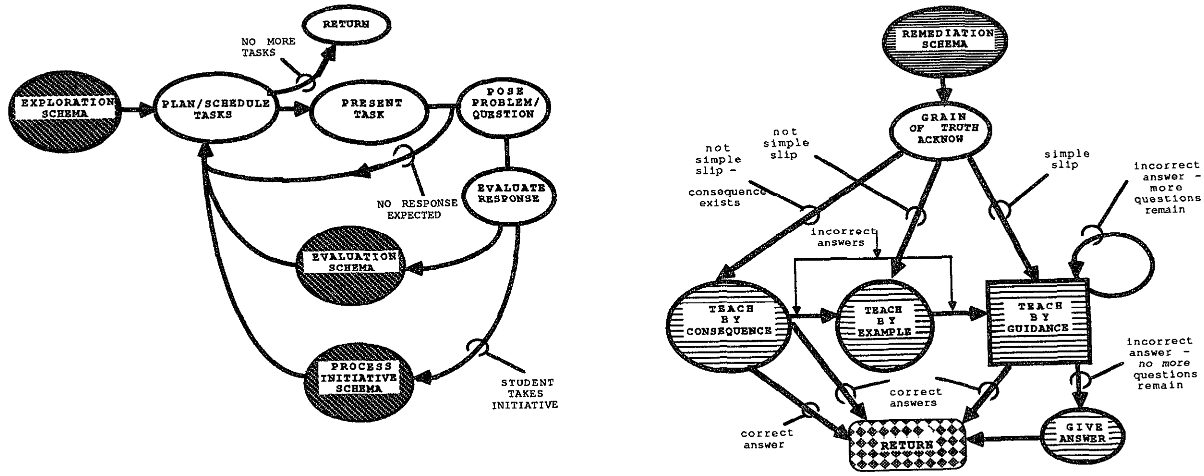

Figure 4: A Framework for Managing Tutoring Discourse

tion. Cognitive science research provides the tutor with a basis for selecting instructional strategies. The importance of addressing common errors and misconceptions in physics is well documented, and the tutor's intelligence hinges on making that knowledge explicit.

We want a tutoring system to help students generate those hypotheses that are necessary precursors to expanding their intuition, and developing their own models of the physical world discover and "listen to" their own scientific intuitions. To do this, we rely on work done by cognitive scientists who study how students reason about qualitative processes, how teachers impart propaedeutic principles (or the knowledge needed for learning some art or science) [Halff, in press], and what tools are being used by experts working in the field.

For example, the cognitive science experiments that must be performed to build our thermodynamics tutor include (1) investigation of real-world tools currently used by physicists, (2) examination of studies that focus on cognitive processes used by novices and experts, and (3) comparison of novice with expert understanding of thermodynamics.

RBT articulates cognitive knowledge by explicitly recording student attempts to solve emergencies. It shows students their false paths and gives reasons behind particular rule-of-thumb knowledge used to solve problems. RBT also provides students with various examples from which they can explore problem-solving activities–perhaps in time showing students their own underlying cognitive processes. By using such knowledge, a tutor can begin to help students *learn how to learn.*

D. Domain Expert

An in-house domain expert is critical to building an intelligent tutoring system. By "in-house", we mean that the domain expert must join the project team for anywhere from six months to several years while domain knowledge is being acquired. Any less commitment than that of full-fledged team member suggests a less than adequate transfer of domain knowledge.

In the tutors described above, the domain experts were (and are) integral to the programming effort. The programmer, project manager, and director of RBT were themselves chemical engineers. More than 30 years of theoretical and practical knowledge about boiler design and teaching strategies were incorporated into the system. Development time for this project would have been much longer than 18 months if these experts had not previously identified the boiler's chemical, physical, and thermodynamic characteristics and collected examples of successful teaching activities.

The second language tutor was developed by a person who holds a graduate degree in teaching English as a second language and has spent more than 7 years using the Silent Way to teach intensive English courses to foreigners living in America and to teach Nepali to American Peace Corps volunteers living in Nepal.

Based on the numerous expert systems projects, the following criteria for acquiring domain knowledge are well understood:

1) Domain experts should be true experts–if possible, the best in the field [Bobrow, Mittal and Stefik, 1986].

2) Domain experts are expensive. Gaining the attention of knowledgeable people is expensive and time consuming. However, the willingness and availability of such experts to participate is critical to the knowledge-engineering process. Assigning the task to a person of lesser ability (or worse, to persons with "time on their hands") might doom a project to failure.

3) Individual domain experts may have incomplete knowledge or conceptual vacuums; therefore multiple experts are needed for testing and modifying domain knowledge throughout the tutor's life.

4) Similarly, domain knowledge can be overly distributed and spread so diffusely among different experts as

to leave severely restricted any system that uses only a single expert [Bobrow, Mittal and Stefik, 1986]. Thus domain knowledge must be acquired incrementally and must be prototyped, refined, augmented and reimplemented. The time needed to build a tutoring system "should be measured in years, not months, and in tens of worker-years, not worker-months" [Bobrow, Mittal and Stefik, 1986].

5) Domain knowledge as found in textbooks is incomplete and idealized[Bobrow, Mittal and Stefik, 1986]. Textbooks rarely contain the commonsense knowledge–the know-how used by expert tutors or professionals in the field–to help choose another teaching strategy or solve difficult problems. Books tend to present clean, uncomplicated concepts and results. To teach or solve real-world problems, tutors must know messy but necessary details of real or perceived links between concepts and unpublished rules of teaching and learning.

IV. Conclusion

Communities of experts are needed to provide a focus for articulating distributed knowledge in an intelligent tutor. The resultant machine tutor should include recent as well as historical research about thinking, teaching, and learning in the domain. Evaluating such an articulation would, in itself, contribute to education–and ultimately to communication between experts.

Compiling diverse research results from environmental, teaching, cognitive, and domain experts is currently hampered by lack of explicit tools to help authors transfer their knowledge to a system. Based on criteria set out above, we intend to continue to develop and integrate knowledge acquisition tools to facilitate assimilation of teaching and learning knowledge into intelligent tutors.

References

[Anderson, 1981] Anderson, J., Tuning of Search of the Problem Space for Geometry Proofs, *International Joint Conference on Artificial Intelligence*, British Colombia, 1981.

[Anderson, Boyle, and Yost, 1985] Anderson, J., Boyle, C., and Yost, G., The Geometry Tutor, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Los Angeles. 1985.

[Apple, 1985] Apple Corp. *Inside MacIntosh*, Vol. 1, Addison-Wesley, Reading, Mass., 1985.

[Atkins, 1982] Atkins, T. *The Second Law*, Freedman Publishers, San Francisco, CA, Scientific American Series, 1982.

[Bobrow, Mittal and Stefik, 1986] Bobrow, D., Mittal, S., and Stefik, M., Expert Systems: Perils and Promise, in *Communications of the ACM*, Vol 29, #9, 1986.

[Burton, in press] Burton, R., Instructional Environments, in Polson, M. and Richardson, J. (Eds.) *Foundations of Instructional Tutoring Systems*, Lawrence Earlbaum Associates, Hillsdale, NJ, in press.

[Cunningham, 1986] Cunningham, P., *Caleb: A Silent Second Language Tutor: The Knowledge Acquisition Phase*, Master's thesis, Tech. Report #87-6. Rensselaer Polytechnic Institute, Troy, NY, 1986.

[Grosz and Sidner] Grosz, B., and Sidner, C., The Structures of Discourse Structure, *Proceedings of the American Association of Artificial Intelligence*, 1985.

[Johnson and Soloway, 1984] Johnson L. and Soloway E., Intention-Based Diagnosis of Programming Errors, in *Proceedings of the American Association of Artificial Intelligence, AAAI-84*, Austin, TX, 1984.

[Halff, in press] Halff, H., Curriculum and Instruction in Automated Tutors, in in Polson, M. and Richardson, J. (Eds.) *Foundations of Instructional Tutoring Systems*, Lawrence Earlbaum Associates, Hillsdale, NJ, in press.

[Hollan, Hutchins and Weitzman, 1984] Hollan, J., Hutchins, E., and Weitzman, L., STEAMER: An Interactive Inspectable Simulation-based Training System, in *The A.I. Magazine*, 1984.

[Mittal and Dym, 1985] Mittal, S. & Dym, C., Knowledge Acquisition from Multiple Experts, in *AI Magazine*, 6(2), 1985.

[Woolf and Murray, 1987] Woolf, B., and Murray, T., A Framework for Representing Tutorial Discourse, *International Joint Conference on Artificial Intelligence*, Milan, Italy, 1987.

[Woolf, Blegen, Jansen and Verloop, 1986] Woolf, B., Blegen, D., Jansen, J., and Verloop, A., Teaching a Complex Industrial Process, *Proceedings of National Association of Artificial Intelligence*, Philadelphia, PA, 1986.

[Woolf and McDonald, 1984] Woolf, B., and McDonald, D., Design Issues in Building a Computer Tutor, in *IEEE Computer*, 1985.