Design Choices for Learning-by-Doing Software: When to Choose Advise

Michael Korcuska, Alex Kass & Menachem Y. Jona
The Institute for the Learning Sciences/Northwestern University
{korcuska,kass,jona}@ils.nwu.edu

Abstract: While there is growing awareness of the value of developing computer-based learning-by-doing environments, there is still a need to analyze the strengths and weaknesses of particular architectures. This paper examines some of the common design issues faced when building learning-by-doing environments and analyzes the relationship between those design issues and the key features of learning architectures, especially an architecture we are currently developing called *Advise*. We also discuss the circumstances under which Advise is appropriate and when other architectures would be a better choice.

1. Introduction

We believe that for learning-environment design and development to make more rapid progress, it is important for educational researchers to stop focusing primarily on such abstract questions as whether to pursue an approach like learning-by-doing, and more on how to pursue the particular approach they advocate. For instance, even if one accepts the basic philosophy – as we do – that some sort of coached learning-by-doing environment is the most effective way to teach, numerous design alternatives for teaching any particular subject area remain open. The real question for advocates of learning-by-doing is "Learning *what* by doing *what*?". Our own experience building computer-based learning-by-doing environments (e.g. Bareiss & Beckwith, 1993; Bell & Bareiss, 1993; Kass, Dooley, & Luksa, 1993; Schank, 1992) as well as the experiences of others doing similar work (e.g. Anderson, 1993; Bransford, 1990; Lesgold, Lajoie, Bunzo, & Eggan, 1988; Soloway, Guzdial, & Hay, 1994) have taught us that many decisions related to the structure of learning-by-doing environments that have a major impact on the effectiveness of the final product. Mapping a set of pedagogical goals and practical considerations into an appropriate design is a complex process. That process, which is relatively unexplored in the literature, is the central topic of this paper.

1.1 Advise: A Sample architecture

In this paper we'll use a particular learning-by-doing architecture, called Advise, with which we're currently experimenting to illustrate some issues that relate to choosing and implementing appropriate learning-by-doing environments. Advise is a computer-based learning architecture which places students in the role of advisor to an important decision-maker, such as the President of the United States. Students are given the mission of evaluating the effectiveness of several courses of action, providing evidence for their evaluations, and recommending one of the options. To help complete the task, students are given a large hypermedia reference library of information and expert opinions about the issue, a skeletal report, and a panel of opinionated experts who represent conflicting perspectives about the issue. Upon submitting the report for review, students receive a critique which focuses on the quality of the evidence he used to support his conclusions. Students can then revise and resubmit the report.

We're currently using the Advise architecture to create applications designed to teach topics in foreign policy, earth science, and statistics. For example, the foreign-policy application is based around a crisis situation in a fictional country we call Krasnovia, which is modeled after events in the former Yugoslavia. The structure of the learning environment is as follows:

- The student gets an emergency phone call from the White House, asking him to come down on the double to help with an emerging crisis.
- On the way to the White House the student watches a new report explaining that the fighting in the Krasnovian capital has intensified and there is new pressure on the US to respond.
- At the White House, the President tells the student that he needs help because his knowledgeable advisors are providing mutually contradictory advice. He asks the student to evaluate the possible options and recommend one.

- The student then must evaluate each of the options—major military action, diplomacy, and economic embargo—and decide how well each one meets a set of objectives the President provides.
- The student uses information in a hypermedia reference library to support his opinions, and the President's advisors critique the student's reasoning and make sure he considers all sides of the issue.
- A plausible argument can be made for several of the options. When the student decides on an option and recommends it to the President, the President will be satisfied only if the student has provided enough evidence for his position.

It is important to note that more than one of the options for solving the problem can be defended using the evidence in the hypermedia reference library. The advisors play an important role in this part of Advise applications. Students can ask them specific questions about their opinions of particular plans, and then each advisor can critique the report and offer advice about how to improve it. This means that students can't succeed at the reporting task simply by finding a set of opinions that no one disagrees with. They must instead concentrate on ensuring that good support is provided for opinions that are included in the report. Another important feature of Advise applications is that since they deal solely with proposal formation and analysis they avoid the complications associated with intricate simulation.

In this paper we'll examine some of the important design issues faced when building learning-by-doing environments; analyze the relationship between those design issues and the key features of the Advise architecture; and discuss how to decide when that architecture is appropriate.

2. Learn *what* by doing *what*?

Designing a learning-by-doing environment is not simply a matter of straightforwardly inspecting the teaching objectives and then creating an environment which employs the target skills. For one thing, at the outset of the design process, it is frequently unclear what the teaching objectives related to a particular subject matter should be. The first question that must be answered is: Learn *what* by doing *what*? For instance, if one starts with the goal of teaching students about foreign policy, the first question one has to ask is, teach them *what* about foreign policy? The mechanisms by which foreign policy is formulated? The mechanisms by which it's implemented? The trends in its recent evolution? Or, perhaps, the rationale behind recent foreign policy decisions?

Only after specifically defining what one wants students to learn does it make sense to start thinking about what kind of activity they should engage in. Perhaps the clearest way to think about the structure of different learn-by-doing architectures is in terms of different possible roles that students could take on. Here are some possible roles that might be relevant to teaching foreign policy:

- **Advisor** — Students advise a decision maker about what course of action to take. In the US foreign policy example, they might be assistants to the President and be asked to recommend policy options.
- **Reporter** — Students create an explanation of some event or situation. As a reporter, students might create a story about the results of US foreign policy in Vietnam.
- **High-level Decision Maker** — Students make important decisions and control action in the learning environment—they primarily focus on what to do and why. Secretary of State is an example of this role that would be used to teach about foreign policy.
- **Low-level Functionary** — Students are given tasks to execute and are primarily concerned with executing those tasks in the current environment. The role of a diplomat stationed in Saigon would be a foreign-policy-related example of this.

This list is intended to give a sense of *some* of the possibilities for structuring a learn-by-doing environment that would teach about a topic like foreign policy. There are many possible roles, some of which don't easily map onto foreign policy.

Deciding which role is best suited to a particular learning situation requires an analysis of the pedagogical opportunities and limitations of each role. Furthermore, the decision about what role students will play is only one part of designing the activity that students will be given – it's just the beginning of deciding what students will do in the learning environment (Schank, Fano, Bell, & Jona, 1994). For example, once one has decided that the best idea for a particular subject area is to put students in the role of a reporter, there's a whole range of questions that follow:

- What should students be assigned to report about? Should they be asked to report on the causes of a particular situation? The likely effects? The opinions of various people involved?

- What type of scaffolding should students receive? Should they be given a partially completed task and asked to finish it? Should they be given an incorrect solution and asked to correct it? Or should they have to start from scratch?
- What kind of coaching should students receive? Should the coaching occur whenever students make an error? Only when they ask for it? Or only after the effects of an error become apparent?

One of our highest priority research goals is to begin to form a framework for analyzing some of the factors that should influence the choice learning-by-doing architectures.

3. Factors that Influence the Choice of Learning Architecture

One way to approach the issue of how to make design choices is by examining the question: Why isn't the most straightforward design always the best? The most straightforward learning-by-doing approach would seem to be to create a simulated environment that allows students to execute the skills one wants to teach. In foreign policy, for example, the most straightforward learning-by-doing design might be to have students execute foreign policy by, say, playing the role of the Secretary of State. Why might this not always be the best choice?

There are both pedagogical and practical reasons why this might not be the best choice. Five common factors that influence the choice of the particular learning-by-doing architecture are:

1. The real-world domain might be too hard for students to understand.
2. The domain might be too difficult to simulate.
3. The most obvious choices might focus attention on irrelevant issues.
4. The level of abstraction which is required might not match the pedagogical goals.
5. There might be a need to simplify the task so that coaching becomes tractable.
6. The obvious role might not be motivating to the majority of students

Next we examine these issues and discuss some strategies for dealing with them, using examples drawn from teaching foreign policy.

3.1 Strategies for simplifying domains that are difficult to understand

Many domains are difficult to understand, and many students will find it impossible to successfully complete realistic tasks in learning-by-doing environments which preserve significant real-world complexity. The issues that the Secretary of State must face when formulating foreign policy are quite complex and contain an overwhelming amount of detail. Making foreign policy decisions might be daunting or discouraging to many students. There are several choices open to designers in this situation, including the following:

1. **Find a "toy" example.** This strategy involves finding a situation that is simple and familiar enough that students will not find it overwhelming. One way to do this is to find a "close to home" situation that is similar to real-world situations. Instead of using a real diplomatic situation involving the US, Israel and Syria, for example, a designer might choose an example from a context more familiar to students like a dispute between the basketball team and the volleyball team about who gets to use the gym.
2. **Break the real domain into small chunks.** This strategy involves breaking down the domain into small chunks and addressing each piece with a separate learning environment. Instead of dealing with all the issues that impact US foreign policy students might be asked to work in a scenario which only involved economic or trade policy, and ignored political and security issues.
3. **Limit the scope of the job.** This strategy involves giving students a constrained job in the learning-by-doing environment. Unlike the previous strategy students encounter the full complexity of the domain but do so in a limited and well-defined manner. Instead of being responsible for all aspects of US foreign policy (including collecting information, setting goals and formulating and executing plans) they might only be responsible for one aspect, say formulating plans.
4. **Provide scaffolding.** If the task is too complicated, you can provide assistance for students. You might, for example, provide a draft of a report to modify instead of having students create one from scratch. The priorities and courses of action are pre-determined so that students does not need to reason at the very highest levels of abstraction.

What Advise Does: Advise applications deal with difficult to understand domains by limiting the scope of the job—students are not asked to invent or execute options, but to evaluate how well the options achieve a predefined set of priorities. The options and priorities are chosen so students will encounter the complex interactions and tradeoffs that may be the defining characteristic of the domain.

Each of these strategies has various attributes which make it more or less appropriate depending on the goals of the learning environment. The first two strategies, finding a toy example, and breaking down the domain, are less likely to engage students in complex interactions and tradeoffs between important factors of full-fledged problems in the domain. Breaking the domain into chunks isolates students from such interactions—that is how it makes the situation easier to understand. While the "toy example" strategy allows students to encounter tradeoffs (between the needs of each team), these tradeoffs may not be similar to those faced in the real-world domain (between economic and political concerns). It also creates a need for students to transfer from the toy problem to the intended real-world domain of foreign policy. The third strategy, limiting the job, better preserves the actual features of a domain—but at a cost. One of these costs if flexibility; restricting the job gives students fewer options. Thus, if the task is to formulate plans, students might not participate in the execution of those plans; if they execute the plans then they miss out on their development. This can reduce the level of excitement or engagement with the task.

3.2 Strategies for simplifying domains that are hard to simulate

Another reason for simplifying a domain is because it is difficult or expensive to accurately simulate. There are several strategies for dealing with this situation:

1. **Build a state-based pseudo–simulation.** Instead of trying to construct an actual simulation of the effects of foreign policy on the world, a set of linked, pre-defined states can be constructed. The system knows that when a user performs an action at a certain state that it should present him with a pre-determined new situation. For example when a user decides to ignore Yugoslavia and focus most of his attention on Iraq, he is soon confronted with a crisis in Yugoslavia.

2. **Avoid simulating the most complex activities.** Some aspect of the domain may be easier to simulate than others. The difficult aspects can either be treated separately or can be partially connected to the simulation. It might be possible to simulate the economic effects of foreign policy, for example, without being able to capture the political effects. Furthermore, when designing a learning environment, one often has a choice about which activities to include. An activity that is less directly related to the topic of the learning environment may nevertheless be a good choice if it is relatively easy to simulate.

3. **Place students in an advisory role.** One reason for creating a simulation is to give students control over what happens. When you can't accurately simulate the domain you can, instead, reduce the amount of control students have. Placing them in an advisory role instead of a decision-making role is one way to do this because the decision-maker will ultimately take actions that effect the simulated world. If the decision-maker's actions are to those you know how to simulate, then students are free to explore a range of actions—even many that can't be simulated. Taken to an extreme there may be no need for a simulation at all and the job is only to give advice.

4. **Avoid simulation entirely**. Sometimes simulation will be so difficult that it is better not to even try. Our models of human psychology are currently rather primitive, for example, and creating simulations of diplomats and politicians is unlikely to lead to an effective learning environment

What Advise does: Advise applications deal with domains that are difficult to simulate by placing students in an advisory role and doing no simulation. Students' advice is accepted by the decision-maker who then puts the plan into action. The results of the plan can be presented very simply by having one ending scenario, usually in video, for each course of action. Alternatively the final results can be omitted and the decision maker can compliment students on a job well done and assure them that even if the situation turns out badly he can at least feel good about having made a reasoned decision.

Again, each of these strategies has strengths and weaknesses. The second strategy, simulating a subset of the domain, is effective because it maintains the strengths of simulations, namely, students have a great deal of control, can experiment with a wide range of decisions and can see the effects of their actions. On the other hand it is often the case that the most interesting aspects of the domain, the ones you wanted to teach in the first place, are precisely those that are hardest to simulate. A state-based simulation, on the other hand, can allow students to engage in those aspects of a domain that can't be mathematically modeled. The difficulty with this strategy is that the range of options given to students are severely restricted—usually only a small list of

alternatives can be supported without creating unmanageable branching. Finally, placing students in an advisory role can allow you to build a simpler situation. The drawback of this strategy is that the decision-maker must reject the advice when the system can't simulate the effects of that actions, which can be frustrating.

3.3 Strategies to control the focus of attention

One of the major advantages of learning-by-doing is the motivational effect that comes with giving students real-world problems to solve. This can paradoxically create a problem if there are learning goals that, from a student's perspective, don't seem to be directly related to the learning-by-doing situation. Students might be so focused on performing the real-world task that they ignore important aspects of the learning environment. Students given the role of President, for example, would focus on trying to make the correct foreign policy decision, that is, the foreign policy decision that causes the simulation to behave in the best way. This is especially dangerous when the simulation is not perfectly accurate—it can lead to students becoming experts on running the simulation instead of learning about the subject area. But even if we have a high-fidelity simulation we could simply more interested in having students learn about the range of possible options than in helping them make "correct" foreign policy decisions (if there are such a thing). Again, there are several strategies to focus students' attention of different aspects of a task:

1. **Manipulate the role**. Placing students in a high-level decision making role will focus students on making correct decisions. An advising role, on the other hand, can be used to focus attention on the analysis that leads up to making a decision instead of the decision itself.
2. **Manipulate the simulation**. If you want to focus students on analysis skills rather than on decision-making procedures, you can cause the simulation to behave badly whenever students do a poor job of analysis—regardless of the decisions they make.
3. **Modify the task**. Instead of simply allowing students to execute the actions that the simulation allows, they can also be asked to justify their actions with some analysis.

What Advise does: Advise applications control the focus of attention by using the advising role. The advising role focuses more attention on the *reasons* for making a specific decision and less on the decision itself. The goal of Advise applications is to help students learn to consider many possible alternatives and to bring appropriate evidence to bear on the situation. Because advice is effective only to the extent that good reasons are provided for following that advice, the role of the advisor focuses attention on providing quality evidence supporting their evaluations.

All of these strategies are designed to move the focus of attention from the final goal of the learn-by-doing environment and onto the (possibly more mundane) activities that are more directly related to the pedagogical goals. The strategy of using the advising role suffers from the same disadvantages as in the previous instance, namely, it potentially reduces the feeling of engagement with the learning (see (Malone, 1981) for example). Manipulating the simulation is problematic for two reasons. First, it might be difficult to control the simulation so precisely. Second, students who use the simulation over time may feel that the learning environment is being unfair. Modifying the task works well in many cases but potentially faces practical problems—it can be quite difficult to analyze justifications.

3.4 Strategies for controlling the level of abstraction

Sometimes the obvious learn-by-doing scenario will be at the wrong level of abstraction given the pedagogical goals. There are a wide variety of aspects to foreign policy ranging from the formulation of overall goals, to deciding how to achieve those goals in different regions, to crafting the correct diplomatic language for communicating in particular countries. The design choices appropriate for teaching about policy formation are unlikely to be the same as those for teaching about writing letters to foreign dignitaries from Japan.

1. **Manipulate the role**. Adjusting the role is a simple way to control what level of abstraction students will be dealing with. The roles of high-level decision and low-level functionary clearly emphasize different levels of abstraction.
2. **Provide insulation.** If the role you'd like to give students naturally includes some activities at inappropriate levels of abstraction then you can provide assistance which will help students with those activities. Creating reports involves deciding what to say, and also how to say it. If you don't want to worry about teaching writing, you might insulate them from those details by ask them to tell an assistant what to write, rather than writing it themselves.

What Advise does: Advise applications use the "assistant" technique to provide insulation from the writing task, and keep students at a relatively high level of abstraction. The advising task itself helps focus students on the analysis issues by avoiding execution-related issues, and by providing the plans and goals

3.5 Strategies to make coaching tractable

The wider the range of actions and the more free-form the task, the more difficult it is to provide automated coaching and critiquing. Simulations can often be difficult to coach students through, because it is difficult to know why they performed a particular action in a complex simulated world. It is also the case that certain simulation architectures, such as purely numerical simulations, make it difficult to trace the path from causes to effects (Forbus, 1994). The following strategies can help provide reliable coaching:

1. **Add structure to the task**. Restricting the flexibility students have can facilitate coaching. Reducing the number of actions students can take makes it easier to decide which action caused the problem they are facing. Furthermore, structured tasks make it much easier to know what students are doing and why—both crucial ingredients in delivering assistance.
2. **Build a self-explanatory simulator**. Qualitative reasoning has demonstrated the ability of self-explanatory simulations, which are combinations of quantitative and qualitative simulations, to provide the kind of explanations that are useful for coaching (Forbus, 1994).
3. **Require students to explain actions**. Instead of simply allowing students to execute actions in the simulation, they can also be asked to explain why they are doing what they are doing. Coaching can be based on the relationship between their explanations and actions.
4. **Avoid simulation**. Sometimes providing coaching in the context of a simulation will be so difficult that it is better not to even try. Figuring out what students are doing and why can be so difficult in some circumstances that coaching will be impossible.

What Advise does: Advise applications provide a highly structured task and do no simulation. The task is structured around priorities and options that the system understands. The system therefore knows what students are trying to accomplish and why. The lack of simulation also makes identifying appropriate coaching opportunities that much easier.

The advantages of self-explanatory simulators are compelling—they provide qualitative explanations that are useful for coaching and tutoring without sacraficing the dynamic interaction of a mathematical simulation. The main disadvantage is that they are very difficult to build. Finally, the last strategy, avoiding simulation, doesn't say much about what the learning environment should be like instead. Another drawback of not building a simulation is that students don't get to see the results of their actions in as direct a way as a simulation affords.

3.6 Strategies for increasing motivation

Motivation is a huge problem in education. Using goal-based scenarios instead of drill-and-practice or lecture courses goes a long way to solving the problems of motivation. It is crucial, though, that students adopt the goal that is the central focus of a many learning-by-doing environments. The following strategies can be used to increase motivation:

1. **Change the role.** Some roles are inherently more interesting than others. Choose these roles when they are compatible with the learning goals, even if it requires stretching the realism of the situation. It is more interesting to play the role of President than it is to be an ambassador.
2. **Magnify importance of the problem.** Often the importance of the problem can be magnified. This is usually done by increasing the cost of failure. In foreign policy this means transforming a scenario in which the problem is a minor land dispute into one which puts civilian lives at stake.
3. **Introduce a "meta-problem".** Introducing a "meta-problem" is just a particular way of magnify the importance of the problem for domains. This strategy involves making the learner's task an important step in solving some larger problem.
4. **Add style/humor.** Impressive graphics, stylish video and a good sense of humor are useful for grabbing a learner's interest. They will not be sufficient to sustain interest, however, and this strategy is useful primarily in conjunction with one of the others.
5. **Use collaboration/competition.** Working with or against others can be very motivating for certain learners. Creating collaborative or competitive learning environments creates a whole new class

of design problems which have received even less attention in the research community than individualized learning-by-doing environments.

What Advise does: Advise applications place learners in the role of advisor instead of decision-maker. For many, if not most, learners an advisory role will be less enticing than the role of decision maker, even if the analysis task itself is interesting. Advise applications address this by drawing students into the role with an exciting introdction which emphasises the importance of the problem, and the importance of their role.

3.7 Review: Key features of the Advise Architecture

To summarize, here are the important features of the Advise architecture:

- In the role of advisor students do not need to make or execute the decision themselves.
- The scope of the job is limited to evaluating a predefined set of potential courses of action. Students do not need to generate options.
- Students are given the priorities the plans should satisfy. They do not need to define the important goals.
- Students do not need to write the text of the report themselves.
- More than one plan can be justified with the evidence from the database. Students are not coached on their choice of plans, only on the effectiveness of their support.
- A panel of advisors offers advice and critiques which reflect a variety of opinions on the issue.
- No complex simulation required.

4. When Advise Is an Appropriate Architecture

We believe this set of features is useful in certain circumstances, as follows:

- You need to simplify a hard to understand domains but don't want to break the domain into small, independent chunks because the important issues relate to the interactions and tradeoffs between those chunks. It was clear to us when set out to teach about foreign policy that one of the central points we wanted to make was about the tradeoffs between domestic and international political concerns, so that separating the two would not be appropriate.
- When watching a simulation unfold is not crucial to learning because real-time dynamics and long-term effect are not the highest priority teaching goals.
- When decisions you want to teach about are not appropriate for simulation because there is not agreement about the effects of certain actions. You might want to use a process other than simulation to teach about Kennedy's decision process in the Cuban Missile Crisis Historical since you don't know what would have happened if Kennedy had allowed the missiles to remain in Cuba.
- When a central teaching point is that there is more than one side to an issue. The panel of advisors focuses attention sharply on the opposing points of view. Because students know their conclusions are being reviewed, Advise emphasizes the evidence students have for their conclusions, rather than the conclusions themselves. The advisors are designed with differing viewpoints and can intervene if students ignore an important point on one side of the issue.
- You want students to focus on analysis rather than execution. In subject areas like foreign policy, the most important thing for students to learn the justification and criticism typically associated with different policy options. The role of advisor facilitates this by focusing more attention on the *reasons* for making a specific decision and less on the mechanism for carrying the decision out.
- You want students to operate at an intermediate level of abstraction. The predesigned goals and plans insulate students from the highest levels of abstraction. The virtual assistant and the advising role help avoid details like writing and execution.

Just as important as understanding when to use Advise is understanding when not to use it. Advise is unlikely to be appropriate if your teaching goals include the following:

- When the dynamic aspects of the domain are crucial. Simulations are really the only way to help students learn about domains where cause-and-effect are well understood and immediate feedback from the learning environment will help students understand the important relationships. Much of engineering and physics have these attributes.
- When there are procedural skills to be taught. Advise is not about the execution of already formulated plans. Some example procedural skill are learning how to be a 411 information operator for the phone company or solving algAdvisea problems. Architectures which are appropriate for these kinds of skills including John Anderson's tutors (Anderson, 1993).

- When there is a correct answer. Advise is not appropriate if there is a particular decision making procedure you want to teach which points to a relatively narrow range of correct solutions. More than one solution in an Advise application should be supportable and the advisors are designed to emphasize that there are multiple valid points of view.
- When it is important for students to see the effects of their decisions played out over time. Very little time passes in an Advise application, in part because there is no simulation. If the long-term effects of decisions are important then a simulation-based learning architecture is more appropriate.

5. Conclusion

Designing a compelling computer-based learning-by-doing architecture for a given set of teaching goals is a very difficult, and poorly understood undertaking. In this paper we have explored some of the design issues that must be confronted when mapping a set of teaching goals and practical considerations to a learning-by-doing architecture. To illustrate how these issues manifest themselves in the design of an actual system, we described our development of the Advise architecture and its use in teaching about foreign policy.

To build more creative and engaging computer-based learning-by-doing systems, designers must often go beyond taking the most straightforward approach. Through our work in designing the Advise architecture we have identified five factors that affect the choice and design of a learning-by-doing architecture and a set of design strategies that can be used to address each of these five factors. An awareness of these factors can help practitioners designing their own learning-by-doing architectures by alerting them early on to the sorts of issues that they are likely to face. Moreover, we hope that the specific design strategies we have identified will provide system designers with feel for the range of potential solutions that are available to them.

References

[Anderson, 1993] Anderson, J. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

[Bareiss & Beckwith, 1993] Bareiss, R., & Beckwith, R. (1993). Advise the president: A hypermedia system for teaching contemporary American history. Presented at *American Educational Research Association*, . Atlanta, GA.

[Bell & Bareiss, 1993] Bell, B. L., & Bareiss, R. (1993). Sickle Cell Counselor: Using Goal-Based Scenarios to Motivate the Exploration of Knowledge in a Museum Context. In *Proceeedings of the World Conference on AI in Education*, . Edinburgh: Association for the Advancement of Computers in Education.

[Bransford 1990] Bransford, J. D., et al (1990). Anchored instruction: why we need it and how technology can help. In R. Spiro & D. Nix (Eds.), *Cognition, Education, and Multimedia: Exploring Ideas in High Technology Hillsdale, N.J: Lawrence Erlbaum.*

[Forbus, 1994] Forbus, K. D. (1994). Self-explanatory simulators: making computers partners in the modeling process. *Mathematics and computers in simulation*(36), 91-101.

[Kass, Dooley & Luksa, 1993] Kass, A., Dooley, S., & Luksa, F. (1993). *The Broadcast News Project* (Technical Report No. 40). Institute for the Learning Sciences.

[Lesgold, Lajoie, Bunzo & Eggan, 1988] Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1988). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. Larkin, R. Chabay, & C. Scheftic (Eds.), *Computer assisted instruction and intelligent tutoring systems: Establishing communication* NJ: Lawrence Erlbaum Associates.

[Malone, 1981] Malone, T. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science*, 4, 333-369.

[Schank, 1992] Schank, R. (1992). Goal-Based scenarios (Technical Report No. 36). Institute for the Learning Sciences.

[Schank, Fano, Bell & Jona, 1994] Schank, R., Fano, A., Bell, B., & Jona, M. (1994). The Design of Goal-Based Scenarios. *The Journal of the Learning Sciences*, 3(4), 305-345.

[Soloway, Guzdial & Hay, 1994] Soloway, E., Guzdial, M., & Hay, K. E. (1994). Learner-Centered Design: The Challenge for HCI in the 21st Century. *Interaction*, 1(2), 36-48.

Acknolwedgments

The Advise project is sponsored by the ARPA CAETI program, contract #N-66001-95-C-8617, monitored by NRAD. ILS was established in 1989 with the support of Andersen Consulting and recieves additional support from Ameritech, an Institute partner. Thanks to the many members of the Advise team as well as our reviewers, especially for comments on issues of motivation.