Acquiring and Transferring Intellectual Skills  
With Modifiable Software Advisors  
in a Virtual Inquiry Support Environment  

Todd A. Shimoda  
University of California at Berkeley  
tshimoda@socrates.berkeley.edu  

Barbara Y. White  
University of California at Berkeley  
bywhite@socrates.berkeley.edu  

John R. Frederiksen  
Educational Testing Service  
jjfrederiksen@ets.org  

Abstract  

Our previous work with the ThinkerTools Inquiry Curriculum found that students who were prompted to reflect on their work performed better on inquiry projects, and attained a better understanding of the inquiry process. These prompts, however, were in a pencil-and-paper form, which did not allow for individual, on-line needs. We hypothesize that improvement will be further enhanced by introducing an adaptive, user-controlled system of software agents that can advise students on completing specific inquiry tasks and their application of general intellectual and collaborative skills. In addition, students can modify the system to help students meet their own knowledge-building goals. Factors such as an advisor’s goals, the type of help an advisor can offer, when the advisor gives advice, and an advisor’s personality can be modified. For example, an inquiry task advisor, the Hypothesizer, helps students come up with alternative hypotheses that are testable. The students and the advisors work in a virtual inquiry support environment of typical artifacts such as a journal, progress report, meeting rooms, and a dialogue box for communication.

Our paper has two parts: First, we discuss what we mean by intellectual skills and how one acquires them and transfers their use to new situations. We also summarize our previous work in increasing science accessibility to students through inquiry, modeling, and metacognition. We also look at the role of technology in our previous work and how we can exploit new technologies to further successes and eliminate weaknesses. Second, we describe our prototype system that uses software agents (“advisors”) and a virtual inquiry support environment to facilitate acquisition and transfer of intellectual skills. We also describe a research plan for evaluating the system.

Intellectual skills, accessibility, and the role of technology  

The term “knowledge construction” in any field of study implies more than just an accumulation of facts. For example, in science, Simon, Valdés-Pérez, and Sleeman [26] point out the great variety of processes used in discovery such as identifying research problems and representations that describe them, making observations and carrying out experiments, developing instruments, seeking regularities and patterns that describe the data parsimoniously, generalizing laws for use in creating more comprehensive theories, drawing inferences from theories and making empirical predictions that can be tested by new observations and experiments. They also point out that an individual scientist, or even a single laboratory, may be devoted to one or a few of these processes. They have become specialized, and their efforts become coordinated through publications, conferences, collaborations, and less formal gatherings. In addition to the discovery process, scientists often engage in the meta-scientific activity of building general theories of the scientific discovery process itself, particularly when impasses in current methods of constructing knowledge are reached.

Collins and Ferguson [2] and Morrison and Collins [15] describe this multi-stage, multi-participant, meta-process as playing the game of using “epistemic forms.” These target structures guide inquiry, and the strategies or sets of moves and constraints in filling out a form are called “epistemic games.” Examples are the list game, the theory-and-evidence game, and the critical-event-analysis game. To play a game, participants must be able to recognize and practice a culture’s epistemic games, understand their different forms of expression and evaluation, and take the perspective of those operating within different epistemic frameworks. This ability is “epistemic fluency.” Thus, learning a science or any other subject is not only mastering the content of a domain, but
also understanding how a field acquires and constructs its knowledge.

The science education reform movement emphasizes teaching students the authentic activities, methods, and processes of science, as well as its content. Anderson [1] includes activities such as empirical discovery through inquiry; the building of models, hypotheses, and theories that explain and predict phenomena; the application of scientific principles to design or to new problems. Ohlsson [17] states that instruction in science cannot limit itself to teaching students the existing knowledge or content of science, because that content is always changing. Because of the enormous changes in the world’s knowledge that can take place in a lifetime, Simon [25] believes that teaching generalized procedures for problem solving in new situations is necessary.

However, previous research on acquiring and transferring intellectual skills has resulted in mixed reviews. The main sources of contention include [28]: defining general vs. specific skills, accounting for individual differences, and integrating the ideas of the sociocultural and active processing paradigm. Learning of general skills, such as heuristics and metacognitive expertise, and specific domain-related skills, such as applying formal models in physics, are hard to disentangle, particularly the question of which comes first. Some of our previous work [8] has been directed at this issue, and we have had success using an intermediate area of reasoning based on causal models [29]. Voss et al. [28] offer two main questions concerning individual differences: what is the basis of ability-level differences, and to what extent can intellectual skill acquisition for low ability-level individuals be facilitated? Recent evidence indicates that intellectual skill acquisition is facilitated when individuals can generate their own solutions to problems, explain and elaborate upon their solutions, and employ metacognitive skills. Voss et al. also believe evidence shows that the sociocultural influence can act to produce more processing, in terms of elaboration and justification, than may otherwise occur. White and Frederiksen’s Inquiry Curriculum [30] studies have shown that prompting students to perform metacognitive and reflective activities can improve performance. They also showed that pairing of lower-achieving students with higher-achieving students can have benefits for both, especially the lower-achieving student.

Measuring the transfer of intellectual skills (from one domain, say physics, to another, say biology or even social science) has been particularly slippery. Cormier and Hagman [3] identify four issues: (a) how transfer should be measured, (b) how training for transfer differs from rapid acquisition, (c) how direction and magnitude of transfer are determined, and (d) whether different principles of transfer apply to motor, cognitive, and metacognitive elements. Singley and Anderson [27] propose (within the ACT* theory) that skill acquisition occurs by general problem solving methods (e.g., means ends analysis or analogy) working with declarative knowledge to create productions rules (condition-action). Transfer of training is predicted to occur to the extent that two tasks share common production rules. However, this version of skill acquisition applies only to tasks that can be automated. Ng and Bereiter (16) identify three levels of goal orientation of students toward learning during a self-directed sequence of computer programming instruction: (a) task-completion goals, such as write a GOTO statement, (b) instructional goals, or pursuing the manifest objectives of the course such as understanding how GOTO statements work, and (c) knowledge-building goals that are more personally motivated, as in applying GOTO statements in new ways. Their study showed the students’ performance (in a near transfer task) generally improved with higher goal orientation level. In summary, the research on transfer of skills is not clear, and transfer appears to depend on many factors, including the experimental task, the relation of the learned material to the transfer material, the amount of training and the explicitness of the training.

Our previous and continuing work with inquiry curricula (the ThinkerTools Inquiry Curriculum), modeling software, and inquiry support software has begun to address many of these issues [8, 30, 31]. The ThinkerTools Inquiry Curriculum incorporates a number of pedagogical strategies related to teaching inquiry and metacognition. These include:

(a) making inquiry tasks and goals explicit using an Inquiry Cycle; the six-steps are: Question (formulating a focused, testable research question), Hypothesize (generating competing hypotheses), Investigate (testing hypotheses, either with real-world experiments or computer simulations such as our ThinkerTools Force and Motion software), Analyze (looking for significant patterns), Model (explaining findings with a law, causal model, or theory), and Evaluate (applying the model to new situations, and determining limitations that suggest new Questions).

(b) scaffolding the inquiry process through methods for achieving those goals. For example, a teacher might begin the curriculum by tossing a bean bag around the room and asking the students to describe all of the factors affecting its motion.

(c) introducing and defining cognitive and social evaluative criteria, such as “reasoning carefully” or “being collaborative.” For instance, “being systematic” is defined as: “Students are careful, organized, and logical in planning and carrying out their work. When problems come up, they are thoughtful in examining their progress and in deciding whether to alter their approach or strategy.”

(d) having students use the criteria to monitor their performance and reflect on their inquiry processes.
in order to determine how they could be improved.

In our research that compared middle-school students who did the monitoring and reflecting (“d” above, in a paper-and-pencil form) with those who didn’t, we found that students in the reflective group gained more on an Inquiry Skills test. Also, this was particularly true for the lower-achieving students those with lower scores on a comprehensive basic skills test score). Also, we found that students in the reflective group, who worked with the criteria throughout the curriculum, showed significant agreement with the teachers in judging their work, while this was not the case for students in the non-reflective group. Our results in the sociocultural category, specifically with pair-work, showed lower-achieving students working with a higher-achieving partner had higher levels of performance than when they worked with a lower-achieving partner, but this advantage was only seen when they were in the reflective assessment group.

If the reflective assessment concepts and activities are acting as metacognitive tools to help students, then the students' performances in developing their inquiry projects should depend upon how well they have understood the assessment concepts. Our results show that students who had learned to use the interpretive concepts appropriately in judging their work produced higher quality projects than students who had not. And again we found that the benefit of learning to use the assessment concepts was greatest for the lower-achieving students.

The success of the ThinkerTools Inquiry Curriculum, particularly the reflective-assessment component, supports our hypothesis that making students aware of cognitive and social processes related to inquiry will enable them to acquire metacognitive expertise which will then play an important role in enabling them to learn via inquiry. We believe that science (and other domains as well) can be improved. In our research that compared middle-school students who did the monitoring and reflecting (“d” above, in a paper-and-pencil form) with those who didn’t, we found that students in the reflective group gained more on an Inquiry Skills test. Also, this was particularly true for the lower-achieving students those with lower scores on a comprehensive basic skills test score). Also, we found that students in the reflective group, who worked with the criteria throughout the curriculum, showed significant agreement with the teachers in judging their work, while this was not the case for students in the non-reflective group. Our results in the sociocultural category, specifically with pair-work, showed lower-achieving students working with a higher-achieving partner had higher levels of performance than when they worked with a lower-achieving partner, but this advantage was only seen when they were in the reflective assessment group.

If the reflective assessment concepts and activities are acting as metacognitive tools to help students, then the students' performances in developing their inquiry projects should depend upon how well they have understood the assessment concepts. Our results show that students who had learned to use the interpretive concepts appropriately in judging their work produced higher quality projects than students who had not. And again we found that the benefit of learning to use the assessment concepts was greatest for the lower-achieving students.

The success of the ThinkerTools Inquiry Curriculum, particularly the reflective-assessment component, supports our hypothesis that making students aware of cognitive and social processes related to inquiry will enable them to acquire metacognitive expertise which will then play an important role in enabling them to learn via inquiry. We believe that science (and other domains as well) can be more accessible with these techniques.

We also see ways to improve our curriculum. Most importantly, we believe that a more individualized approach will allow students to acquire the skills at the level they need and at the time they need them. This area of improvement addresses the individual differences problem. We also believe that if the students have control over the type and timing of the advice they receive and the criteria they use to judge their progress, then they will better acquire the skill and be able to transfer it to new situations. We are designing a computer-based system that we hope will accomplish these improvements.

Another goal of our new system is to facilitate the acquisition (or improvement) of reflective, meta-scientific ability to build general theories of the inquiry process and learning in general. A measure of the success would be epistemic fluency, which would include not only the ability to recognize and practice a culture’s epistemic games, but also the ability to develop new epistemic forms and games, as Simon [25] would advocate. Another measure of success would be an awareness of and ability to modify one’s own learning process.

There are many different types of systems that might accomplish our goals. However, the level of modifiability and adaptability required to handle individual differences, would mean that the system needs some level of artificial intelligence (AI). McArthur, Lewis, and Bishay [12] surveyed the state of AI and education and found several interactive-learning environments that appeared in various forms as inquiry-based, student-centered, constructionist, constructivist, discovery-based, and interactive-learning environments. Results from their own investigations showed that students working with mathematics-related microworlds acquired rather broad insights into the nature of mathematical knowledge and problem-solving. Students began to understand that they could make their own mathematics, rather than learn about theorems discovered by others and they abandoned the notion that math comprises a series of separate problems to solve. They also found that the pedagogical strategies are often best presented as locally intelligent agents with interventions carefully chosen so that student initiative is not interrupted unless there is very strong evidence that the student is thrashing in unprofitable ways. The student is also free to disregard the advice offered by the local agents. The agents’ roles can also be adaptable, for example, if students are learning inquiry skills themselves, or how to brainstorm in groups, then locally intelligent agents may play very modest roles. Similarly, a novice student may learn best in environments that include agents which can intensively model and coach formative skills. As the student acquires expertise these agents will probably “fade,” permitting much more student initiative. This type of system seems to best fit our needs and the second section further elaborates our prototype system.

Prototype system: SCI-WISE

This section describes the characteristics of the software agents in the prototype system and the architecture and interface design. The programming platform being used for the prototype is Macromedia Director 6 and its Lingo code. While not as sophisticated a language as C++ or others, it allows an object-oriented and an agent-based style of programming, and can handle complex message passing and data tracking. Most importantly, the program allows for relatively quick interface prototyping and evaluation. We are evaluating other languages, interfaces, and reasoning engines, such as KQML (Knowledge Query and Manipulation Language), Java and JESS (Java Expert Shell System), for the next generations of our system.

Software agent definitions

Franklin and Graesser [7] attempt to differentiate a computer “program” from a software “agent.” As they note, there is no definitive or agreed upon definition of
software agent, rather there is an emerging taxonomy, much as there is to the question: “What is life?” Still, they propose this definition: “an autonomous agent is a system situated within and part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future” (p. 25). Key to their definition is “autonomous” and to describe an autonomous agent, one must describe its environment, sensing capabilities, actions, drives, and action selection architecture. They claim most non-agent programs lack this sense of autonomy, for example, a payroll program’s output won’t affect what it senses later, nor does it continue acting after its immediate function is complete. Autonomy need not be the only principal component of agenthood. Others may include consciousness, intentionality, mobility, and for Russell and Norvig [21] rationality:

For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has. … Doing actions in order to obtain useful information is an important part of rationality … p. 33.

The argument will likely continue for a long time as to what defines an agent, if it ever comes to resolution. What can be applied from this programming paradigm to an inquiry support environment are the techniques used to create different classes of agents. Russell and Norvig define four levels of agent complexity: (a) simple reflex agents that use condition-action rules, (b) agents that keep track of the world with an internal state, (c) goal-based agents that seek a goal-state, and (d) utility-based agents, those that want to maximize their state of utility or “satisfaction.” Franklin and Graesser [7] classify agents according to the subset of properties that they have, e.g. reactive (senses and acts, responds in a timely fashion to changes in the environment), autonomous (exercises control over its own actions), goal-oriented (is proactive, purposeful, does not simply act in response to the environment), temporally continuous (is a continuously running process), communicative (is socially able, communicates with other agents, including people), learning (is adaptive, changes its behavior based on its previous experience), mobile (is able to transport itself from one machine to another, flexible (actions are not scripted), character (has believable personality and emotional state). The agents then act in a social manner, with the user and other agents. It is useful to think of a society of agents, or agency, [cf. 14] as a metaphor for an educational system, each with some specialized expertise.

**Characteristics of software agents**

In narrowing the focus for design purposes, four main characteristics of agents will be discussed: (a) goals and actions, (b) knowledge and beliefs, (c) communication, and (d) adaptability. As with most system designs, the characteristics are not easily separated, nor is this the only way to categorize the characteristics. For instance Dignum and van Linder [5] propose that social agents consist of these four components: information (containing knowledge and belief), action, motivation (where goals and intentions play a role), and social (containing aspects of speech acts and relations between agents).

(a) Goals and actions. Selker [24] defines an advisory-style agent that builds a user relationship with the explicit goal of educating the individual. To achieve this goal, the advisor-agent must assume (or be told) what the student already knows. Selker identifies four levels of user experience as (1) the novice who needs basic information, (2) the level at which users need to know how and when the learnable thing can be used, (3) a higher level that shows the rich and variable uses of the learnable thing, and (4) the highest level at which expert interactions of things they are learning can be explored. Selker’s levels have some correspondence to Ng and Bereiter’s [16] levels of goal orientation (1) corresponds to task completion, (2) and (3) seem to fit with instructional goals, and (4) with knowledge building.

Three broad classes of agents for SCI-WISE have been developed: *task advisors* that help students complete the steps of the inquiry process (task completion level), *general purpose advisors* for help in understanding and using general cognitive and social skills embodied in the inquiry tasks, and *system-development* advisors that help students construct a model of inquiry and theory building through modification of the advisors. These are not the only classes of agents that can be useful; but they satisfy a model of reflective, goal-driven inquiry as proposed by White and Frederiksen [30].

Task advisors, such as an Inquiry Advisor, provide strategies, examples, and other advice for accomplishing tasks, in this case, doing an inquiry research project. Tasks can be composed of subtasks, as in the case of the inquiry cycle, and each subtask might have an advisor. In theory, a single rule could be designed as an agent, the trade off becomes one of the number of agents vs. their complexity. For example, the Questioner has these goals: Goal 1—To advise the student how to formulate a good research question. Goal 2—To advise the student during the other steps of the inquiry process (for example, during the hypothesize subtask, the Hypothesizer can refer the student back to the Questioner to get advice on focusing the question.) Goal 3—To give students advice during other tasks such as preparing a presentation of their work or reflecting on their work.

General purpose advisors can provide advice during almost any task. These advisors correspond to some of White and Frederiksen’s [30] self-assessment criteria and are categorized as either cognitive or social. The cognitive advisors and the criteria they are derived from are the Planner (being systematic), the Inventor (being inventive),...
the Reasoner (reasoning carefully), and the Representer (using tools and representations of science). For example, the goal of the Reasoner is to advise students how to reason carefully with such strategies as supporting their claims with evidence. The social skill advisors that were derived from teamwork and communication criteria are the Collaborator, the Debater, the Mediator, and the Communicator. For example, the Collaborator might suggest ways to divide the work fairly.

The system development advisor is the Modifier. The goal of this advisor is to help students customize the agents and create new ones. Actions that can be performed include adding to or modifying an agent’s knowledge base or setting when an agent should give advice and what kind of advice it should give. Other advisors are being considered for this category, for example, a Learning Theory advisor, whose expertise is to help students identify and possibly change the theory of learning they have. We hypothesize that these modification or learning theory advisors will help students construct a model of inquiry which can be useful when working in new domains or confronting novel problems.

The actions an advisor takes largely consist of giving advice appropriate for a context. Other actions, besides giving advice, might include calls to other advisors or requests for information. An advisor determines the context by information it receives from the environment (such as the task the student is working on), and the advisor’s student model (in this case, goal orientation). The reasoning is accomplished by advice drivers using rule-driven reasoning engines for establishing beliefs (disposition toward some action) about which advice or message is appropriate to display. Once the beliefs are established, the appropriate advice is selected from the advisor’s knowledge base.

(b) Knowledge and beliefs. Russell and Norvig [22] argue that the standard approach to designing agents is to create agents that have some knowledge and that they try to learn more. One problem is to figure out what knowledge base the agents should start with. Self [23] approaches background knowledge as “viewpoints.” “A viewpoint is a set of beliefs, a belief is a dispositional state held by some agent” (p. 22) Essentially, agents need to adopt a viewpoint from which to reason. A viewpoint may be represented as a set of beliefs, and at any time, an agent or a community of agents may possess any number of viewpoints. To permit cooperative problem solving, an agent needs to be able to represent and reason about other agents’ viewpoints (either software or human). If two or more agents have conflicting views which have to be reconciled then, with or without a mediator, some negotiation is necessary.

Goodyear and Stone [9] note that an intelligent system will typically contain, and be able reason with, explicit representations of knowledge of four kinds: subject-matter (domain) knowledge, knowledge about teaching (tutoring or pedagogic knowledge), knowledge about the learner currently interacting with the system (encoded as a learner model), and knowledge that facilitates communication between the system and the user (the user-computer interface). Problems in knowledge representation include uncertainty and incompleteness, formation and use of concepts, explanation and causal modeling, and learners having knowledge of the real world which is substantially richer than the real world knowledge encoded in the machine. A knowledge based system in a domain with more open-end questions, such as social science, will need to have some basis for making claims about the status of knowledge, whether its own, or the student’s. It will need to be able to establish and challenge warrants for that knowledge. In so doing, it must be able to draw on rules of evidence that are founded in explicit epistemological positions—otherwise it will be unable to emulate the reasoning of a social scientist adopting that position.

The advisors in the prototype SCI-WISE system are knowledge based, and are intended to provide general guidance in many content areas including those in the physical and biological sciences, as well as in the social sciences. As these general advisors evolve, specialist advisors will likely emerge, such as a “genetics-smart” Questioner. The knowledge base of SCI-WISE advisors consists of the broad areas of domain knowledge (for example, the Questioner’s domain is coming up with research questions), pedagogical knowledge, student user knowledge (student models), knowledge about other advisors, “self” knowledge, and computer interface-environment knowledge. In general, the types of knowledge in these areas consist of some or all of the following: states (a state might be a student user just starting a task), goals of the advisor and student, criteria (for example, what makes a good research question), strategies to reach a goal state, examples, and concepts.

Beliefs in the SCI-WISE system establish some disposition toward a specific action. An example of a belief for the Questioner is: the student is working on the Question task (coming up with a good research question) and has just begun, and he or she hasn’t worked with the Questioner before. These beliefs, taken together, imply the action of giving the student the criteria for good research questions.

(c) Communication. Russell and Norvig [20] describe the advantage that a group of agents gains (collectively and individually) by being able to do the following: inform each other about the world (by making statements), query other agents about particular aspects of the world (by asking questions), answer questions, make requests or command other agents, promise to do things or offer deals, acknowledge requests or offers, share feelings and experiences. (p. 652)

To Rimmershaw [19] communication is about shared or negotiated meanings in which “teachers and learners need to coordinate how they represent the topic to themselves, and the language they use for communicating that
selecting appropriate strategies for achieving the desired effectiveness of goal-driven learning depends on being able the learning process should be guided by reasoning about the learning contributes to achieving the learner's goals, systems based on the idea of goal-driven learning, the requires some degree of reflection and self-regulation.

In programming, natural-language understanding and production is a difficult problem, especially doing any of the types of things discussed in Rimmershaw [19]. The prototype SCI-WISE system does not have a sophisticated natural language processor, future generations of the system will likely include such abilities. In the prototype system, the student users can use a Dialogue Box to send a message directly to an agent. The agent checks the message words against its lexicon of key words. If a match is found, the agent displays a message.

The student users can also go directly to an agent for advice either through the Meeting Room which groups agents by function, or the Progress Report which groups agents by tasks. Agents may also respond to other agents' requests or referrals. The information available in a “message” includes text, the sender (where the message originated from), and the receiver. Context information is available as needed from the environment. All advisor messages displayed to the student users are coded in the knowledge base.

(d) Adaptability. Maes [11] gives four different sources from which a learning agent acquires its competence: (a) “looking over the shoulder” of the user, (b) direct and indirect user feedback, (c) examples given explicitly by the user, (d) getting advice from agents that assist others with the same task. McCalla and Wesson [13] claim that adaptability lies in the interplay of such factors as the knowledge states, goals and experience of the student, and the reasoning capabilities and knowledge representations of the system. The partners to this knowledge negotiation do not have equal knowledge of the domain nor equal knowledge of how best to learn the domain. To arrive at a level of understanding that both partners can work within requires some degree of reflection and self-regulation.

Ram, Cox, and Narayanan [18] developed their learning systems based on the idea of goal-driven learning, the central idea underlying their theory of goal-driven learning is that, because the value of learning depends on how well the learning contributes to achieving the learner’s goals, the learning process should be guided by reasoning about the information that is needed to serve those goals. The effectiveness of goal-driven learning depends on being able to make good decisions about when and what to learn, selecting appropriate strategies for achieving the desired learning, and guiding the application of the chosen strategies. Adaptation in such a system means keeping track of changes in the environment and modifying strategies and knowledge to accommodate the changes while still reaching a goal.

The SCI-WISE prototype advisors adapt their advice based on user actions within the system (such as which agents the students are going to for advice) and the advisor’s model of the student’s goal orientation and the advisor’s own goal. The users can establish how the advisors respond by overriding any belief setting processes. The advisors are not able to update their own knowledge base in the prototype (that is, establish new rules), although the advisors can prompt the users to do so through the modification process.

Architecture and interface

Kautz, Selman, and Coen [10] claim that one of the most difficult aspects of agent design is to define specific tasks that are both feasible using current technology, and are truly useful to the everyday user. In their initial testing of their prototype agent system, users had little patience interacting with software agents. It was clear that agents must provide solutions to real problems that are important to real users. deJong and Rip [4] propose guidelines for embedding computer tools in scientific discovery environments including: accessibility; modular design with components that can be developed, used, and maintained separately from other components; interactivity, flexibility, and transparency. To Elsom-Cook [6], the problem is similar to one that arises in interface design generally: the user’s model of the system may not agree with the designer’s model of the system interface.

Figure 1 provides a schematic view of the architecture of SCI-WISE. The advisors are constructed for the three classes by the use of template-like structures that include their knowledge base and dialogue subsystem. Agent types can thus be generated from the class and type templates (for example, a Questioner advisor from the Inquiry Task advisor template from the Task Advisor template), and in turn, a specific instance of an agent can be created, as in Quentin Questioner. The students and the advisors operate within an inquiry support environment consisting of four basic artifacts: (a) the Meeting Room, where students can access the advisors by their classification, (b) the Dialogue Box, in which the students can send messages to specific advisors, and also serves as a message builder allowing advisor-to-advisor communication, (c) a Project Journal in which students do and store their task-related work, and (d) a Progress Report that keeps track of progress, context, and student actions, and which provides access to the task advisor associated with a task. All artifacts are in a graphical form (Figure 2).
Figure 1. Prototype SCI-WISE architecture

Figure 2. Graphical interface elements
The advisors’ responses or other messages sent (displayed) for the student user are determined by the message driver subsystem. This subsystem itself is composed of the advice drivers that take some input from the environment, student user, or other agent, and establish a belief. In the prototype testing version, there are five drivers: (a) current status, (b) advisor goals, (c) student goals, (d) advice type, and (e) advice regulator. All the drivers have four possible settings: three that will lock the advisor into some action or belief, and one that allows the advisor to set a belief (a disposition toward an action). Figure 3 shows possible settings for one of the drivers.

For any request or context actuation, the sum of advisor’s beliefs from the drivers will determine the appropriate response via a look-up table. For example, assume all the drivers are set to “let advisor decide”:

- If the current status is “haven’t started,” then
- if the current task is Question, set Questioner goal to Goal 1—give advice related to question, and
- if the student has no recorded experience, set the student goal to “task completion,” and
- if the student has expressed no preference and has no recorded experience, set the advice type to “specific,” and
- if the student has expressed no preference and has no recorded experience, set the advice regulator to “unsolicited,”
- then respond “A research question is your guide for doing your inquiry project, a mystery to be solved” and display two buttons for more information: [Specific example--->] [Advisor referral, Inventor -->]

As the student progresses through tasks, the agents can take student information, such as which classes of agents are solicited for advice, and refine the type of advice given. If a driver is set by a student, for example, the current status driver to “haven’t started,” then the advisor will only give advice when the student is in that status.

The student users can also go to the advisor’s knowledge base (by clicking on “Everything”) to see any advice, and modify it or add to it. This procedure is facilitated by the Modifier advisor.

Testing SCI-WISE

The goal of our system is to help students perform inquiry research (including assessing and presenting their work), and perform meta-scientific activities of building general theories of the scientific discovery process itself. We also hope that they will discover something about their own learning, as well as to see that it can be modified and improved. We will also be testing hypotheses concerning student goals and advisor use.

The overall program of testing our prototype system will be in three phases. In the first two phases, students, working alone or in pairs, will be given tasks to complete using the software. Examples of the tasks in the first phase:

- Find out what you can about the Reasoner advisor (or other advisor). What does it know? How can it help you? When might it be helpful?
- Check out the other things on the screen (e.g., journal, progress report, meeting room), do you understand what they do?
- Try using the system to start an inquiry project on memory. Your research question is: Does repeating words three times help you remember them more than using the words in sentences? (One student acts as the experimenter, and one student acts as the subject. The experimenter reads a list of words and tells the subject to either repeat or use the words in a sentence. When the list is finished, the subject is asked to recall as many of the words as he or she can. The inquiry
task includes coming up with predictions and assessing your predictions.

- How might you change the advisors and system to be more helpful to you? How might you change the system to be useful in another situation (e.g., math class)? Try to make some improvements to the advisors.

For the second phase, the tasks will be similar, except the inquiry project will be more involved (incorporating more phases of the inquiry cycle) and will incorporate portions of the curriculum to be used in the classroom studies. One of our interests is student goal orientation while they are doing inquiry projects. As Ng and Bereiter [16] note, it would be very worthwhile to find out which goals are actually at work as students engage in various school activities.

In the third phase, the teachers will be teaching their classes as usual, with the exception that in the participating middle-school classrooms, the software will be incorporated into the curricula. We will be testing our hypotheses concerning student goals and use of the system, as well as student progress in moving toward higher level goals. We will be able to follow students throughout the year to see what kind of improvement is made in inquiry and collaboration skills, as well as how they transfer to other subject areas. We will also be able to compare classrooms in which the software was not used as a gross comparison of performance.

Acknowledgments

Special thanks to Allan Collins, and the ThinkerTools research group and the Reasoning research group at UC-Berkeley for many helpful comments and suggestions. The graphics for the prototype were designed by L.J.C. Shimoda. For more information on ThinkerTools and SCI-WISE visit http://thinkertools.berkeley.edu/tt.

References


