Representational and Advisory Guidance for Students Learning Scientific Inquiry[1]

Dan Suthers,[2] John Connelly,[3] Alan Lesgold,[3] Massimo Paolucci,[4] Eva Erdosne Toth,[5] Joe Toth,[6] Arlene Weiner[7]

Draft submitted for AAAI book – do not distribute.

Scientific knowledge is dynamic in two senses: it changes and increases extremely rapidly, and it is thrust from the lab into the wider world and the public forum almost as rapidly. These trends place increasing demands on secondary school science education. Besides knowing key facts, concepts, and procedures, it is important for today's students to understand the process by which the claims of science are generated, evaluated, and revised – an interplay between theoretical and empirical work (Dunbar & Klahr, 1989). The educational goals behind the work reported in this chapter are to improve students' understanding of this process, facilitating students' acquisition of critical inquiry skills while also meeting conventional subject matter learning objectives.

In addition to the need to change what is taught, there are grounds to change how it is taught. Research shows that students learn better when they actively pursue understanding rather than passively receiving knowledge (Brown & Campione 1994; Chi *et al.* 1989; Craik & Lockhart, 1972; Greeno *et al.* 1996; Resnick & Chi, 1988; Perkins *et al.* 1985; Webb & Palincsar, 1996). Accordingly, the classroom teacher is now being urged to become a "guide on the side" rather than the "sage on the stage." In parallel, new roles have been recommended for artificial intelligence applications to education, replacing computer-directed learning with software that augments the learning processes of students engaged in collaborative critical inquiry (Chan & Baskin, 1988; O'Neill & Gomez, 1994; Roschelle, 1994; Scardamalia & Bereiter, 1994).

The present chapter describes an educational software package, known as "Belvedere," that supports students collaboratively solving ill-structured problems in science and other areas (such as public policy) as they develop critical inquiry skills. Belvedere exemplifies two ways in which artificial intelligence can contribute to student-centered approaches to learning: by informing the design of representational systems that constrain and guide learner's activities, and by responding dynamically to representations that learners construct in these representational systems.

[1] Work done while all authors were at the Learning Research and Development Center of the University of Pittsburgh. Current affiliations follow.

[2] Dept. of Information and Computer Sciences, University of Hawai'i, 1680 East West Road, POST 317, Honolulu, HI 96822. Email: suthers@hawaii.edu; voice: (808) 956-7420; fax: (808) 956-3548; www: http://lilt.ics.hawaii.edu/
[3] Learning Research and Development Center, University of Pittsburgh; connelly+@pitt.edu, al+@pitt.edu
[4] Robotic Institute, Carnegie Mellon University, Massimo_Paolucci@amalthea.cimds.ri.cmu.edu
[5] Department of Psychology, Carnegie Mellon University, etoth+@andrew.cmu.edu
[6] Pacific-Sierra Research Corporation, Arlington VA, jtoth@psrw.com
[7] KW Consulting, 1433 Denniston Ave., Pittsburgh PA 15217

The chapter begins with an overview of the Belvedere software environment and its use, followed by a discussion of the design history of Belvedere's diagrammatic interface, leading to conclusions concerning the role of external representations in learning applications. Then, the design of Belvedere's automated advice on-demand facility is detailed. Two "advisors" illustrate how useful functionality can be obtained with minimal knowledge engineering, and incrementally extended as the tradeoffs and limitations are better understood. The chapter concludes with a discussion of several approaches to machine intelligence in educational applications, including the approaches exemplified by Belvedere.

1.1 Belvedere: Software for Collaborative Inquiry

The "Belvedere" software is a networked software system that provides learners with shared workspaces for coordinating and recording their collaboration in scientific inquiry. The versions described in this chapter, Belvedere 2.0 and 2.1, are a complete redesign and reimplementation of Belvedere 1.0, previously reported in Suthers & Weiner (1995) and Suthers *et al.* (1995).

1.1.1 Software Interface

Belvedere's core functionality is a diagramming window—a shared workspace in which students construct "evidence maps." Evidence maps are graphs, similar to concept maps, in which *nodes* represent component statements (primarily empirical observations or hypotheses) of a scientific debate or investigation; and *links* represent the relations between the elements, i.e., consistency or inconsistency. The software also includes artificial intelligence advisors, a "chat" facility for unstructured discussions, and facilities for integrated use with Web browsers. The diagramming window is shown in Figure 1. The default "palette" (the horizontal row of icons) makes salient the most crucial distinctions we want learners to acquire in order to conduct scientific inquiry. Left to right, the icons are "data" for empirical statements, "hypothesis" for theoretical statements, and "unspecified" for others statements about which learners disagree or are uncertain; then links representing "for" and "against" evidential relations. The rightmost icon invokes the automated advisors. Learners use the palette by clicking on an icon, typing some text (in the case of statements) and optionally

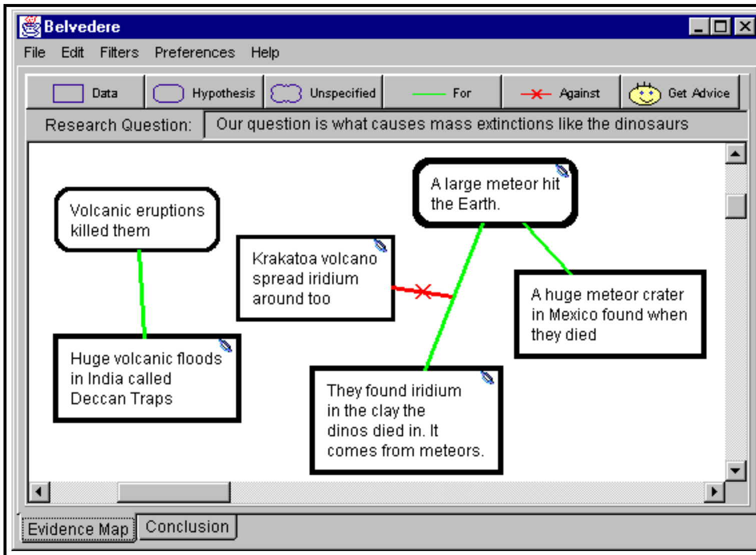


Figure 1. Belvedere Evidence Mapping Software

setting other attributes, and then clicking in the diagram to place the statement or create the link. The palette is configurable: other categories and relations can be added, such as "principle" for law-like statements, and a link for conjunction, enabling expression of N-to-M evidential relations. Extensions underway include alternate views on the workspace (e.g., evidence *tables*), as well as alternate workspace types (e.g., concept maps and causal loop diagrams).

Other features, briefly noted, include the following. Users can set different "belief levels" for the statements and relations, and display these as line thickness with a "filter." Java applets have been embedded in the Web-based curricular materials, enabling learners to send references to these pages into the workspace with a click of a button. (The small "link" icons in the upper right corners of objects in Figure 1 indicate the presence of URLs linking back to these pages.) References to external objects can also be sent from other applications directly into the Belvedere workspace. For example, Koedinger, Suthers, & Forbus (1998) enabled one of Forbus' Active Illustration simulations (Forbus, 1997), to send summaries of simulation runs as "data" objects into Belvedere. The feasibility of embedding other kinds of documents in Belvedere, such as MS Word and Excel documents, has been demonstrated. It is possible to reinvoke these applications in a platform independent manner. Thus Belvedere can be used as a conceptual organizer for use of various tools during an inquiry.

1.1.2 Software Implementation

The Belvedere client application is written in Java, and is available for MacOS, Windows '95, NT, and Solaris. It is deployed within a networked architecture that is designed to provide affordable access to intelligent collaborative educational functionality on a variety of desktop platforms. See Suthers & Jones (1997) for a detailed discussion of the Belvedere 2.0 architecture and related design considerations. The current architecture for Belvedere 2.1 is shown in Figure 2. Belvedere clients record all modifications to diagrams in a server database via the "Belvedere Object Request Broker Interface" (BORBI, Figure 2).[8] In Belvedere 2.1, BORBI forwards user changes to a Connection Manager, a small Java process on the server that keeps track of the clients using any given workspace, and informs other clients (via their Listener sockets) of the

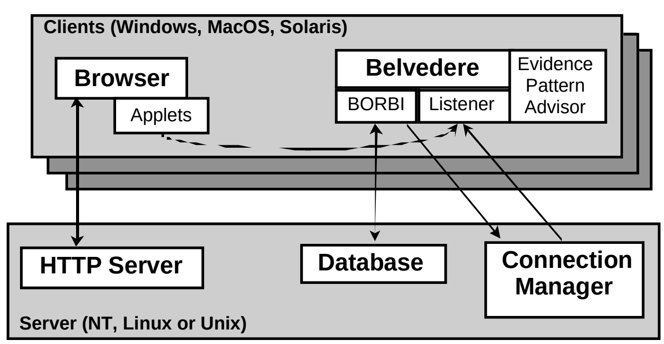


Figure 2. Belvedere 2.1 Architecture (JDBC)

[8] The database is Postgres in Belvedere 2.0 and 2.1's Unix servers; and msql in Belvedere's NT server. BORBI was CGI-based in Belvedere 2.0 and is JDBC-based in Belvedere 2.1.

changes to their workspace. This results in automatic "what you see is what I see" update of the client displays. The client includes an evidence-pattern advisor that provides advice on demand.[9] Belvedere can also operate in "stand-alone" mode, in which case a local file directory replaces the database in a manner transparent to the user, and the networked collaborative functionality is not available. Applets embedded in Web-based "science challenge" curricular materials we developed for Belvedere facilitate easy transfer of references to on-line articles into Belvedere clients through their Listeners, as shown in Figure 2. These curricular materials are part of a comprehensive classroom implementation package, described briefly in the next section.

1.1.3 Classroom Implementation

Belvedere 1.0 was initially used by students aged 12-15 working alone or in pairs in our lab, as well as by students working in small groups in a 10th grade biology classroom (Suthers & Weiner, 1995). Subsequently Belvedere 2.0 and 2.1 were used by 9th and 10th grade science classes in Department of Defense Dependents Schools (DoDDS) overseas. At this time, use in DoDDS continues, and is expanding to DoD schools in the United States, known as DDESS.

Recognizing that no software, however well designed, will impact on education if it is not well integrated into the classroom environment, we developed an integrated instructional framework for implementing Belvedere-supported collaborative inquiry in the classroom. The approach includes student activity plans worked out in collaboration with teachers. Students work in teams to investigate real world "science challenge problems,"[10] designed with attention to National Science Education Standards to match and enrich the curriculum. A science challenge problem presents a phenomenon to be explained, along with indices to relevant resources (e.g., Figure 3). The teams plan their investigation, perform hands-on experiments, analyze their results, and report their conclusions to others. Investigatory roles are rotated between hands-on experiments, tabletop data analysis, and computer-based literature review and use of simulations and analytic tools as well as Belvedere. The classroom activity plans provide teachers with specific guidance on how to manage these activities with different levels of computer resources. Teachers and students are also provided with assessment instruments designed as an integral part of the curriculum. Assessment rubrics are given to the students at the beginning of their project as criteria to guide their activities. The rubrics guide peer review, and help the teacher assess nontraditional learning objectives. See Suthers, Toth & Weiner (1997) for further information on this integrated instructional framework, as well as discussion of a third-party evaluation.

[9] In Belvedere 2.0, the advisors ran as a server-based Lisp process. The evidence pattern advisor was partially ported to Java for a client-based advisor in Belvedere 2.1.
[10] http://lilt.ics.hawaii.edu/belvedere/materials/

Figure 3. Science Challenge Problem

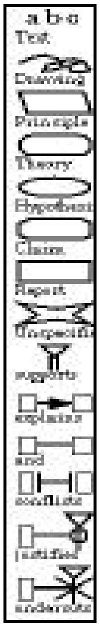
1.2 Representations and Discourse

The evolution of Belvedere's interface from Belvedere 1.0 to Belvedere 2.1 motivates our view of the roles of external representations in learning. The representations serve as stimuli, coordinators, and guides for various learning interactions between agents, including the automated advisors as well as learners. In essence, the representations help provide a loose "semantic coupling" between the activities of the agents, but by no means control or capture the full meaning of their interactions.

Belvedere 1.0 was designed under the assumptions that a visual representation language (augmented with automated advice giving) can help students learn the nuances of scientific argumentation, provided that

(a) the *language is capable of capturing all of these nuances*, and

(b) *students express their arguments in the language.*



Guided by (a), Belvedere 1.0 was provided with a rich palette of statement types and relationships. One of the Belvedere 1.0 palettes is shown to the left. Assumption (b) was motivated by the intention that the representations provide a semantic common ground for various learning activities involving students and software coaching agents. We reasoned that it would be possible to construct an artificial intelligence agent that participated in and coached argumentive discourse, provided that learners' attempts at scientific argumentation were fully expressed in a representational medium with mutually shared semantics.

1.2.1 Locus of Discourse

As indicated by assumption (b), we expected students to express all of their significant argumentation in the diagrams using primitives such as these. However, we found that much relevant argumentation was "external," arguing *from* the representations rather than arguing *in* the representations. Faced with a decision concerning some manipulation of the representations, students would begin to discuss substantial issues until they reached tentative agreement concerning how to change the representation. In the process, statements and relations we would have liked students to represent were not represented in the diagrams.

Our initial frustration soon gave way to an understanding that this is an opportunity: proper design of manipulable representations can guide students into useful learning interactions. Thus, we downplayed the originally intended roles of the representations (1) as a medium *through* which communication takes place, (2) as a complete record of the argumentation process, and (3) as a medium for expressing formal models – in favor of their role as (4) a stimulus and guide for collaborative learning discourse. The following discussion summarizes subsequent observations and further work that took place under this new view.

1.2.2 Discussion of Ontological Choices Posed by the Medium

Belvedere requires all knowledge units (statements and relations) to be categorized at the time of creation. We often observed that learners who were using Belvedere initiated discussion of the appropriate categorical primitive for a given knowledge unit when they were about to represent that unit (Suthers 1995). Although this is not surprising, it is a potentially powerful guide to learning, provided that it happens at the right time, and that discussion focuses on the underlying concepts rather than the interface widget to select. For example, consider the following interaction in which students were working with a version of Belvedere that required all statements to be categorized as either "data" or "claim." (The example is from videotape of students in a 10th grade science class.)

S1: So data, right? This would be data.
S2: I think so.
S1: Or a claim. I don't know if it would be claim or data.
S2: Claim. They have no real hard evidence. Go ahead, claim. I mean who cares? who cares what they say? Claim.

The choice forced by the tool led to a peer-coaching interaction on a distinction that was critically important for how they subsequently handled the statement. The last comment of S2 shows that the relevant epistemological concepts were being discussed, not merely which toolbar icon to press or which representational shape to use. Yet it is not always useful to confront learners with choices, even if they may become important at some point in the development of expertise. With more complex “palettes” we sometimes observed students becoming confused by choices that were not relevant at their stage of learning.

1.2.3 Refinements for Ontological Clarity

Based on these observations, we simplified Belvedere’s representational framework to focus on the most essential distinction needed concerning the epistemological source of statements: empirical (“data”) versus hypothetical (“hypothesis”). Further simplifications were motivated by observations concerning the use of relations (links). The original set of argumentation relations included evidential, logical, causal, and rhetorical relations as well as the various classifications of statements exemplified above. In exchanges similar to the previous example, we observed students’ confusion about which relation to use. Sometimes more than one applied. We felt that the ontologically mixed set of relation categories confused students about what they were trying to achieve with the diagrams, and did not help them focus on learning key distinctions. In order to encourage greater clarity, we decided to focus on evidential reasoning, and specifically on the most essential relational distinction for evidence based inquiry: whether two statements are consistent or inconsistent.

1.2.4 Eliminating Artifactual Distinctions

Furthermore, we eliminated directionality from Belvedere’s link representations of relations. At one time there were at least three versions of the “consistency” relation: “predicts” and “explains” (both drawn from hypotheses to data), and “supports” (drawn from data to hypotheses). Early versions of our evidence pattern coach (to be described in section 1.3.2) attempted to reason about and even enforce these semantics. However, we found that users’ use of these relations (as expressed in their links) was inconsistent and sometimes differed from the intended semantics, consistent with other research on hypermedia link categories (Marshall & Rogers, 1992; Shipman & McCall, 1994). When the users’ semantics differed from the coach’s semantics, confusion or frustration resulted. For example, one subject drew a complex map of a hypothesis with seven “supports” links leading from the hypothesis to data items. The coach, failing to see any support paths from data to the hypothesis, highlighted the hypothesis and indicated that it lacked empirical evidence. This prompted the subject to start to delete his links.

The use of “predicts,” “explains,” and “supports” links was misguided not only because different agents had different semantics for them, but also because the links were “surface” level discourse relations that did not encourage learners to think in terms of the more fundamental consistency relationships. Whether a hypothesis predicts or explains a datum is an artifact of the chronology of the datum with respect to statement of the hypothesis. Whether one uses “supports” or one of the other two links is an artifact of the focus of the discourse process by which the diagram is being constructed (argumentation about hypotheses versus explanation of data). Hence we eliminated these in favor of a single non-directional relation that expresses the more fundamental notion of evidential consistency.

1.2.5 Discussion Guided by Salience and Task

Consideration of ways in which subjects interacted with the representations led us to appreciate subtle ways in which external representations may guide discourse. For example, Figure 4 outlines a diagram state in which three statements were clustered near each other, with no links drawn between the statements. One student pointed to two statements simultaneously with two fingers of one hand, and drew them together as she gestured towards the third statement, saying “Like, I think that these two things, right here, um, together sort of support that” (Figure 4, from a videotape of an early laboratory study of Belvedere).

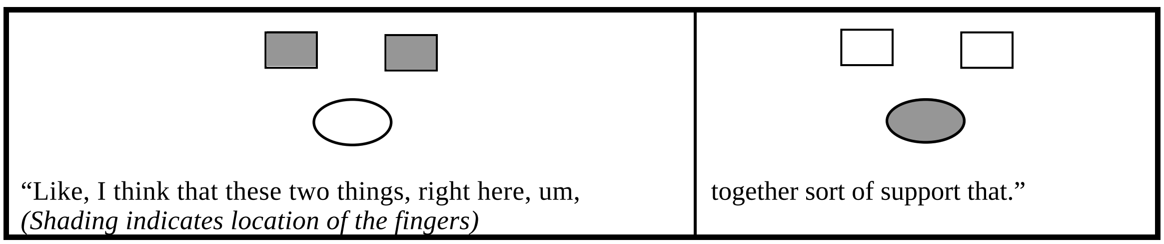


Figure 4. Gesturing to express a relationship between adjacent units.

This event was originally taken merely as an example of how external representations facilitate the expression of complex ideas (Clark & Brennan, 1991). However, this observation applies to any external representation. Reconsideration of this example led to the hypotheses that several features of the representational system in use made the student’s utterance more likely. First, elaboration on these particular statements is more likely because they (instead of others) are expressed as objects of perception in the representation. Second, this event is more likely to occur in a representational environment that provides a primitive for connecting statements with a support relation than in one that does not -- the students perceive their task as one of linking things together. Third, it may have been easier to recognize the relationship between the three statements because they happened to be spatially nearby each other (Larkin & Simon, 1987). In this example, proximity was determined by the users rather than intrinsic to the representational toolkit. However, a representational tool could constrain proximity based on potential relationships between knowledge units.

1.2.6 Roles of External Representations in Learning Interactions

The foregoing experiences led to a reconceptualization of the role of external representations in learning, particularly in collaborative learning situations. Specifically, facilities for constructing visually inspectable and manipulable external representations of learners' emerging knowledge provide cognitive, social, and evaluative support, summarized in Figure 5. The figure can alternately be read as an expression of how external representations provide a loose "semantic coupling" between different kinds of learning interactions.

1.2.6.1 Cognitive Support

Concrete representations of abstractions such as evidential arguments can *help learners "see," internalize, and keep track of abstractions* while working on complex issues, serve as *a record of what the learners have done,* and *provide an agenda of further work* (Bell, 1997; Smolensky *et al.*, 1987; Streitz *et al.*, 1989). The kind of external representation used to depict a problem may determine the ease with which the problem is solved (McGuiness, 1986; Larkin & Simon, 1987; Kotovsky & Simon, 1990; Zhang, 1997), just as appropriate design of (internal) representations for machine intelligences facilitates problem solving (Amarel, 1968) and learning (Utgoff, 1986). The constraints built into representations may make the problem very difficult to solve (e.g., the 9-dots problem; Hayes, 1989) or may enhance problem solving (Stenning & Oberlander, 1995; Klahr & Robinson, 1981).

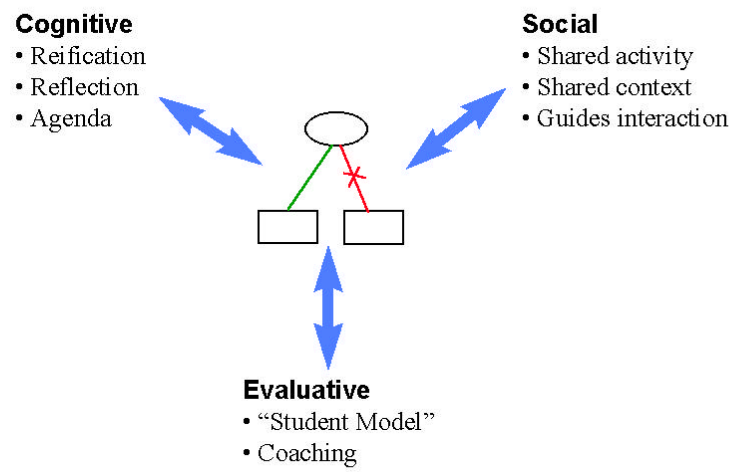


Figure 5. Learning Interactions Guided by External Representations

1.2.6.2 Social Support

The interaction of the cognitive processes of several agents is different than the reasoning of a single agent (Okada & Simon, 1997; Perkins, 1993; Salomon, 1993; Schoen, 1992; Walker, 1993), and so may be affected by external representations in different ways. Shared learner-constructed representations such as diagrams provide *shared objects of perception that coordinate distributed work,* serving as referential objects and status reminders. We often observe learners using gestures on the display to indicate prior statements and relationships. In some group configurations we have seen learners work independently, then use gesturing on the display to re-coordinate their collaboration when one learner finds relevant information (Suthers & Weiner 1995). Different representations will serve this function different ways according to their representational biases.

Also, the mere presence of representations in a shared context with collaborating agents may change each individual's cognitive processes. One person can ignore discrepancies between thought and external representations, but an individual working in a group must constantly refer back to the shared external representation while coordinating activities with others. Thus it is conceivable that external representations have a greater effect on individual cognition in a social context than they do when working alone.[11]

1.2.6.3 Evaluative Support

Shared learner-constructed representations such as diagrams provide mentors (including the teacher, peers, and the computer) with *a basis for assessing learners' understanding* of scientific inquiry, as well as of subject matter knowledge. The use of concept maps (Novak, 1990) as an assessment tool is an area of active investigation (O'Neil & Klein, 1997, Ruiz-Primo *et al.*, 1997). We are currently developing similar techniques for evidence maps. Assessment based on external representations can also support computer coaching of the inquiry process, as described in the remainder of this chapter.

1.3 Design of Computer Advisors

Ideally, we would like to have an advisor that understands the students' text as well as the domain under discussion, and provides advice based on a deep understanding of the domain of inquiry. This is not currently feasible. Although much of the technology is available, a large investment in system development and knowledge engineering is required. It is unclear which portion of this effort results in worthwhile learning gains. Instead, we have adopted the strategy of investigating how much useful advice we can get out of minimal semantic annotations before we move on to more complex approaches. In this manner we hope to better understand the cost/benefit tradeoff between knowledge engineering and added functionality.

In this section we discuss two methods of advice generation that we have implemented (Paolucci *et al.*, 1996; Toth *et al.*, 1997). First, "evidence pattern" advice strategies make suggestions from the standpoint of scientific argumentation, based solely on the syntactic structure of students' evidence maps. The strategies help the learners understand principles of inquiry such as: hypotheses are meant to explain data, and are not accepted merely by being stated; multiple lines of evidence converging on a hypothesis are better than one consistent datum; hypotheses should try to explain all of the data; one should seek disconfirming evidence as well as confirming evidence; discriminating evidence is needed when two hypotheses have identical support; etc. Second, "expert-path" advice strategies perform comparisons between the learners' diagrams and an evidence map provided by a subject matter expert. This advisor can challenge or corroborate relationships postulated by the students, or confront learners with new information (found in the expert's diagram)

[11] Micki Chi, personal communication to the first author.

that challenges learners in some way. We first briefly describe the design constraints under which we operated, and then the basic algorithms behind our advice giving methods.

1.3.1 Pedagogical Constraints on Advice

We believe that the most important kind of advice is that which stimulates and scaffolds constructive activity on the part of the students. Our design of the advisors to be discussed was guided in part by the following constraints.

1.3.1.1 Maintain the student-initiated character of Belvedere's environment.

Belvedere encourages reflection by allowing students to see their evidential argumentation as an object. They can point to different parts of it and focus on areas that need attention. They can engage in a process of construction and revision, reciprocally explaining and confronting each other. We felt that an advisor that is not aware of these discourse processes should not intervene excessively or prematurely. Students should feel free to discard an advisor's suggestions when they believe them to be irrelevant or inappropriate. Also, students should be free to introduce information that is not known to the system. The advisors should still be able to provide feedback.

Anderson and colleagues have substantial empirical evidence in favor of immediate feedback in tutoring systems for individual learning in domains such as Lisp programming, geometry, and algebra (Anderson *et al.* 1995; Corbett & Anderson, 1990; McKendree 1990). We take a less tightly coupled approach to feedback for two reasons. First, we are dealing with ill-defined problems in which it is not always possible to identify the correctness of a learner's construction. Second, we want students to develop skills of self and peer critiquing in a collaborative learning context. A computer advisor that intervened in an authoritative manner would discourage students' initiative in evaluating their own work (Nathan, 1998).

1.3.1.2 Address parts of the task that are critical to the desired cognitive skill.

Research on "confirmation bias" and hypothesis driven search suggests that students are inclined to construct an argument for a favored theory, sometimes overlooking or discounting discrepant data (Klayman & Ha 1987; Chinn & Brewer 1993). Also, they may not consider alternate explanations of the data they are using. An advisor should address these problems. For example, it should offer information that the student may not have sought, including information that is discrepant with the student's theory.

1.3.1.3 Be applicable to problems constructed by outside experts and teachers.

The advisor should be able to give useful advice based on a simple knowledge base that an expert or a teacher might construct. Belvedere has been used for topics as different as evolution, mountain formation, mass extinctions, AIDS, and social psychology. It is not feasible to develop, for each topic, a representation of the knowledge needed to deal with the argumentation in which students could potentially engage. We were instead interested in a general approach in which either no knowledge engineering is required or a teacher can construct the knowledge base.

Hence a “minimalist” AI approach was taken, in which we implemented an advisor that can provide reasonable advice with no domain specific knowledge engineering. Advice was provided only on request. Identification of specific needs and consideration of the cost of meeting these needs then motivated extensions to this advisor.

1.3.2 Evidence Pattern Strategies

The first approach we implemented gives advice in response to situations that can be defined on a purely syntactic basis, using only the structural and categorical features of the students’ argument graphs. (The students’ text is not interpreted.) Principles of scientific inquiry are instantiated as patterns to be matched to the diagram and textual advice to be given if there is a match. Example advice is shown in Figure 6, and example advice patterns from our Belvedere 2.0 implementation are given in Figure 7. This Lisp implementation used representation and retrieval facilities from the Loom knowledge representation system (Bates & MacGregor, 1987). When the solid-lined portions are present and the dashed portions are missing, the corresponding advice can be given. Objects that bind to variables in the patterns (the shaded boxes in Figure 7) are highlighted in yellow during presentation of advice to indicate the target(s) of definite references such as “this hypothesis.” For example, Figure 6 shows the “one-shot hypothesis” advice in Belvedere 2.1. Some advice patterns not shown in the figure include:

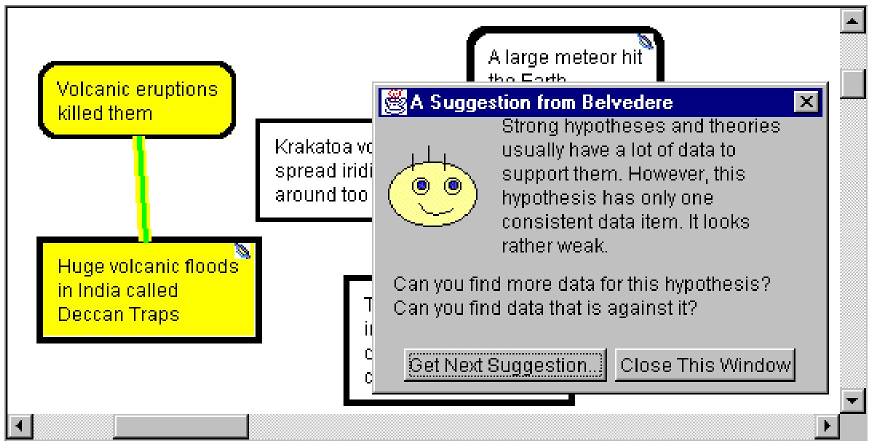
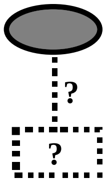


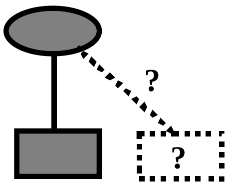
Figure 6. Example Evidence Pattern Advice

Alternate hypothesis: When only one hypothesis is stated, asks whether there is another hypothesis that provides an alternate explanation for the data (pointing out that it is important to consider alternatives so as not to be misled).

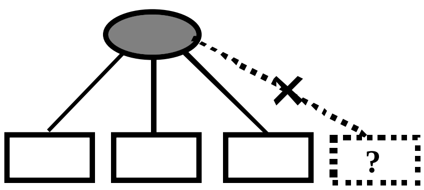
Attend to discrepant evidence: Motivated by research showing that people sometimes ignore discrepant evidence, this counterpart to the confirmation bias advice detects hypotheses that have consistent and inconsistent data, and asks whether all the data are equally credible.



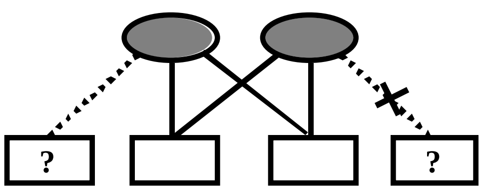

```
(def-advice ‘HYPOTHESIS-LACKS-EMPIRICAL-EVIDENCE
  :query ‘(retrieve (?h) (and (hypothesis ?h) (No-Evidencep ?h)))
  :advice (“Can you find data that are for or against this hypothesis?
A scientific hypothesis is put forward to explain observed data. Data
that a hypothesis explains or predicts count *for* it. Data that are
inconsistent with the hypothesis count *against* it.”)
  :subsequent-advice (“Can you find some data for or against this
hypothesis?”)
  :advice-types ‘(incompleteness))
```




```
(def-advice ‘ONE-SHOT-HYPOTHESIS
  :query ‘(retrieve (?d ?h)
             (and (data ?d) (hypothesis ?h)
                  (Consistent-HypoP ?d ?h)
                  (fail (Exists-Multiple-Consistent-DataP ?h))
                  (fail (Exists-Inconsistent-DataP ?h))))
  :advice (“Strong hypotheses and theories usually have a lot of data
to support them. However, this hypothesis has only one consistent data
item. It looks rather weak. Can you find more data for this
hypothesis? Can you find data that is against it?”)
  :subsequent-advice (“This hypothesis has only one consistent data
item. Could you find more data for (or against) this hypothesis?”)
  :advice-types ‘(evaluative incompleteness))
```




```
(def-advice ‘CONFIRMATION-BIAS
  :query ‘(retrieve (?h)
             (and  (hypothesis ?h)
                   (Exists-Multiple-Consistent-DataP ?h)
                   (Multiply-LinkedP ?h)
                   (fail (Exists-Inconsistent-DataP ?h))))
  :advice (“You’ve done a nice job of finding data that is
consistent with this hypothesis. However, in science we
must consider whether there is any evidence *against* our
hypothesis as well as evidence for it. Otherwise we risk
fooling ourselves into believing a false hypothesis. Is
there any evidence against this hypothesis?”)
  :subsequent-advice (“Don’t forget to look for evidence
against this hypothesis!”)
  :advice-types ‘(cognitive-bias))
```




```
(def-advice ‘DISCRIMINATING-EVIDENCE-NEEDED
  :query ‘(retrieve (?h1 ?h2)
             (and (hypothesis ?h1) (hypothesis ?h2)
                  (not (same-as ?h1 ?h2))
                  (Exists-Consistent-DataP ?h1)
                  (Exists-Consistent-DataP ?h2)
                  (fail (Consistent-HypoP ?h1 ?h2))
                  (Identical-EvidenceP ?h1 ?h2)))
  :advice (“These hypotheses are supported by the
same data. When this happens, scientists look for
more data as a \”tie breaker\” -- especially data
that is *against* one hypothesis. Can you produce
some data that would \”rule out\” one of the
hypotheses?”)
  :subsequent-advice (“Can you produce some data that
might support just one of the hypotheses?”)
  :advice-types ‘(incompleteness evaluative))
```

Figure 6. Evidence Pattern Advice

Contradicting links: When both a "for" and "against" link have been drawn between the same two statements, asks if this was intended.

Data supports conflicting hypotheses: Asks if this configuration makes sense; if so, suggests a search for discriminating data.

Explain all the data: Matching to a hypothesis that has explained some of the data but has no relation to other data, points out the importance of attempting to explain all the data and asks whether the hypothesis is consistent or inconsistent with the as of yet unrelated datum.

Many objects and no links: After acknowledging that it's OK to be gathering data and hypotheses, suggests that the user begin to consider the relationships between them. *Nothing in diagram:* Suggests that a theory or hypothesis be formulated when none is present in the evidence map. Provides basic instructions on use of the toolbar icons.

1.3.2.1 Evidential Paths and Cache

The evidence patterns make heavy use of predicates for paths of consistency relations, such as "Exists-Inconsistent-Data-P" and "Exists-Multiple-Consistent-DataP" (Figure 6). These predicates are efficiently computed by using a "consistency cache" which is filled on an as-needed basis. The first time a consistency predicate is called on a given statement, a search algorithm is run that finds all hypotheses and data that are connected to the statement by a path of one or more consistency relations, or by a path of any number of consistency relations with one inconsistency relation on either end of the path. (We found that advice drawn from a composition of two or more inconsistency relations is difficult to understand.) The cache for each statement consists of four sets: consistent-hypotheses, consistent-data, inconsistent-hypotheses, and inconsistent-data. The statement itself is excluded from these sets. The sets are represented as ordered lists (sorted on internal object IDs), enabling fast set comparison with list equality. For example, the "Identical-Evidence-P" predicate merely requires a list equality comparison of the consistent-data and inconsistent-data ordered lists.

1.3.2.2 Advice Selection

Typically, several advice patterns will match an evidence map, sometimes with multiple matches per pattern. This is more than a student can be expected to absorb and respond to at one time. It is necessary to be selective in a context sensitive manner. For example, Figure 7 (a) shows an evidence map with 6 matches, called Advice Activation Records (AARs), to three advice patterns.

Selection is performed by a preference-based quick-sort algorithm, following a mechanism used by Suthers (1993) for selecting between alternate explanations. Preferences (Table 1) take into account factors such as prior advice that has been given, how recently the object of advice was constructed and by whom, and various categorical attributes of the applicable advice. Given an ordered pair of AARs, a preference will return ">," "<," or "=" indicating whether it prefers one over the other. For example, given two AARs, the first of which binds a variable to an object created by the current user and the second of which does not, "created-by-user" will return ">." The sort algorithm is given a prioritized list of preferences, as exemplified in Figure 7 (b). Our variation of the quicksort algorithm first partitions the set of AARs into equivalence classes under the first (highest priority) preference on the list. The equivalence classes are ordered with respect to each other. It then calls itself recursively on each equivalence class with the remaining list of preferences. When the list of preferences becomes empty on a recursive call involving a nontrivial set of AARs, the AARs are ordered randomly for variety. Finally, the sequence of equivalence classes that is returned by the recursive sorts is concatenated to yield the prioritized list of AARs.

There are three advice selection strategies for early, mid, and late phases of an evidence map. The phases are defined in terms of the complexity of the diagram: The user is "getting started" if there is no data, no hypothesis, or only one evidential relation. The user is late in the process if there are at least two hypotheses and the number of data items and evidential relations is at least 4 each and greater than the number of hypotheses. Otherwise the strategy shown in Table 1 is used. Strategies are expressed as different priority orderings of the preferences. For example, the preference "new-advice" is applied first to partition the AARs into those that have been given before

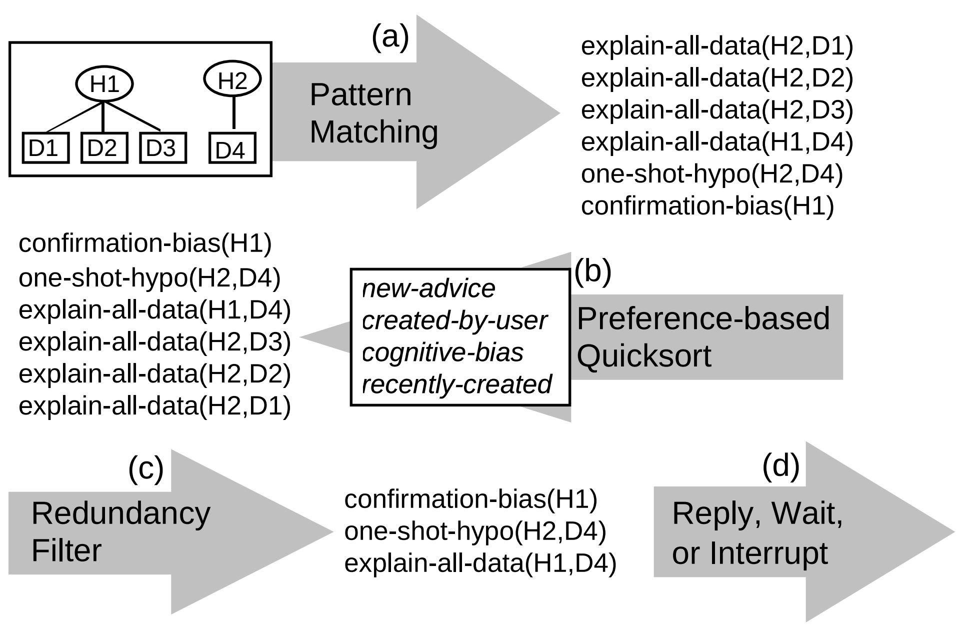


Figure 7. Advice Selection

Preference Name	***Prefers AARs …***
New-Advice	… that have not been given before (based on a bounded history of prior communications).
Expert-Path	… that were created by the expert-path advisor (described in next section)
Created-by-User	… that bind variables to objects created by the user to be advised.
Interrupting-Advice	… that are marked as worth an interruption (interrupting advisor only)
Cognitive-Bias	… for advice types that address problematic cognitive biases.
Incompletness	… for advice types concerned with ways the user can engage in constructive activity.
Incoherence	… for advice types that address semantically incoherent diagram configurations.
Many-Siblings	… for advice patterns that have many instantiations (AARs).
Recently-Created	… that bind variables to objects recently created (by anyone).
Evaluative-Advice	… for advice types that address, in part, the evaluation of hypotheses based on the data (this preference is high priority in the "late" strategy).
Getting-Started	… for advice useful to someone learning to use the evidence mapping tool (this preference is high priority in the "early" strategy).

Table 1. Prioritized Preferences

and those that have not. Then "created-by-user" partitions each of these into ordered subpartitions, and so on down the list. In the example of Figure 7, the "late" strategy applies, although for simplicity of presentation only 4 of the preferences are shown in the figure. Suppose all of the AARs are new (have not been presented); that one user created all of the objects; and that object D4 was created most recently. Preferences "new-advice" and "created-by-user" have no effect: all AARs go into one equivalence class. Preference "cognitive-bias" creates two equivalence classes: the "confirmation-bias" AAR, and all the rest. Finally, "recently-created" is applied to each of these equivalence classes, resulting in the reordering of AARs according to recency.

After sorting, a "redundancy filter" is applied that removes all but one of multiple instantiations of a given advice pattern, retaining the highest priority instantiation. This provides the final prioritized list of advice, as exemplified in Figure 7 (c). The advice-on-demand version of the advisor then sends the first AAR on the list to the requesting client (d). If further advice is requested before the diagram changes, subsequent advice instances on the sorted list are used without reanalysis.

We have been experimenting with an intrusive advisor that differs from the on-demand advisor in step (d) of Figure 7. This advisor recomputes the list of advice after every user action. It then examines the top N (usually we set N=1) AARs on the advice list, and determines whether the advice merits an interruption, based on two considerations. First, only certain categories of advice are deemed to be sufficiently important to merit an interruption. Second, each AAR is given a delay factor to allow the user sufficient "time" (measured by counting modifications to the diagram) to anticipate and address the issue that would be raised by the advice. For example, one would not want the advisor to interrupt with advice "hypothesis lacks empirical evidence" every time one creates a hypothesis. It takes two steps to create a data object and link it to the hypothesis. Hence this advice pattern is given a delay of 2, meaning that AARs for this advice pattern are filtered until they recur 3 times, allowing for the creation of the hypothesis, the data, and the link.

1.3.2.3 Evaluations of the Evidence Pattern Advisor

The evidence pattern advisor provides advice about abstracted patterns of relationships among statements, but has nothing to say about the contents of these statements. Its strengths are in its potential for pointing out principles of scientific inquiry in the context of students' own evidential reasoning, and its generality and applicability to new topics with no additional knowledge engineering.

Empirical evaluation of this advisor took two forms: it was made available in DoD dependent school (DoDDS) classrooms in Germany and Italy; and laboratory studies of expert advisors were conducted. At this writing a third study, a controlled comparison of intrusive and nonintrusive strategies, is underway.[12]

Although distance prevented detailed classroom observations, data available to us from DoDDS in the form of limited personal observations, third party observations, videotapes, and computer logs indicates that (1) the on-demand advisor was almost never invoked, although the advice icon was readily available on the toolbar; (2) there were situations where students did not know what to do next, situations in which the advisor would have helped if it were invoked; and (3) the advice and its relevance to the students' activities was sometimes ignored as if not understood. Items (1) and (2) indicate that in spite of our reluctance to interfere with students' deliberations, unsolicited advice is sometimes needed. In response to this need, we have implemented and begun laboratory experimentation with the intrusive version of the advisor described previously.

There are several explanations for the third observation. The wording may require some simplification and shortening. The current strategy is to give a general principle and interpret this in terms of the diagram, for example:

> *Principle:* "… in science we must consider whether there is any evidence *against* our hypothesis as well as evidence for it. Otherwise we risk fooling ourselves into believing a false hypothesis.
> *Specific Advice:* Is there any evidence against this hypothesis?"

Students may become confused by the more abstract justification and never read or process the specific suggestion, or the advice may simply be too long. A modality mismatch may also be a factor: students are working with diagrams, but the advice is textual. We would like to modify the advice presentation to temporarily display the suggested additional structure directly in the students' diagram, perhaps using dashed lines as was done the left column of Figure 6.

In the laboratory studies (Katz & Suthers, 1998) we used the distal "chat" facility to enable subject matter experts – geologists[13] – to coach pairs of students working on the Mass Extinctions issue. The geologist for a given session could only see what the computer advisor sees, namely the

[12] John Connelly's thesis work.
[13] Dr. Jack Donahue, and graduate students John Dembosky and Brian Peer.

user's changes to the diagram. However, we allowed students to ask the geologist questions in natural language. Categorization of the geologists' communications for four sessions showed that most advice giving was concerned with domain specific knowledge rather than the general principles applied by the evidence pattern advisor, although there were some clear examples of the latter as well. Many communications either (1) introduced relevant information or suggested that students search for new relevant information, or (2) commented on the correctness of evidential relations that the students drew. These results confirmed what we knew all along: that the evidence pattern advisor would be too limited. However they also helped guide the next direction taken in our incremental approach: the addition of simple techniques with low knowledge engineering costs that would yet enable the machine to (1) introduce or suggest new information and (2) evaluate students' evidential relations.

1.3.3 Expert-Path Advice Strategies

The expert-path advisor was designed to offer specific information that the student may not discover on her own. It makes the assumption that a correspondence can be found between statements in a student's evidence map and those in a pre-stored expert's evidence map. The path advisor searches the latter "expert graph" to find paths between units that students have linked in their evidence maps, and selects other units found along those paths that are brought to the students' attention. Our claim is that this enables us to point out information that is relevant at a given point in the inquiry process without needing to pay the cost of a more complete semantic model of that information, such as would be necessary in traditional knowledge-based AI&ED. The only costs incurred are in the construction of the "expert diagram" consisting of semantic units that are also available to the student, and the additional mechanisms needed to identify the correspondence between statements in the student and expert diagrams.

1.3.3.1 Constructing and Using Expert "Snippets"

A teacher or domain expert first authors HTML-based reference pages to be used by the students. Each page consists of one or more semantic units, which we call "snippets." A snippet is a short text describing a hypothesis or an empirical finding such as a field observation or the results of an experiment. "Reference-this buttons" – the icons in the HTML page on the right of Figure 8 – are then attached to each snippet. These buttons invoke Java code that presents a dialog by which users can send statements containing references to the snippets into Belvedere. An example dialog is shown on the left of Figure 8. The dialog requires users to summarize snippets in their own words.

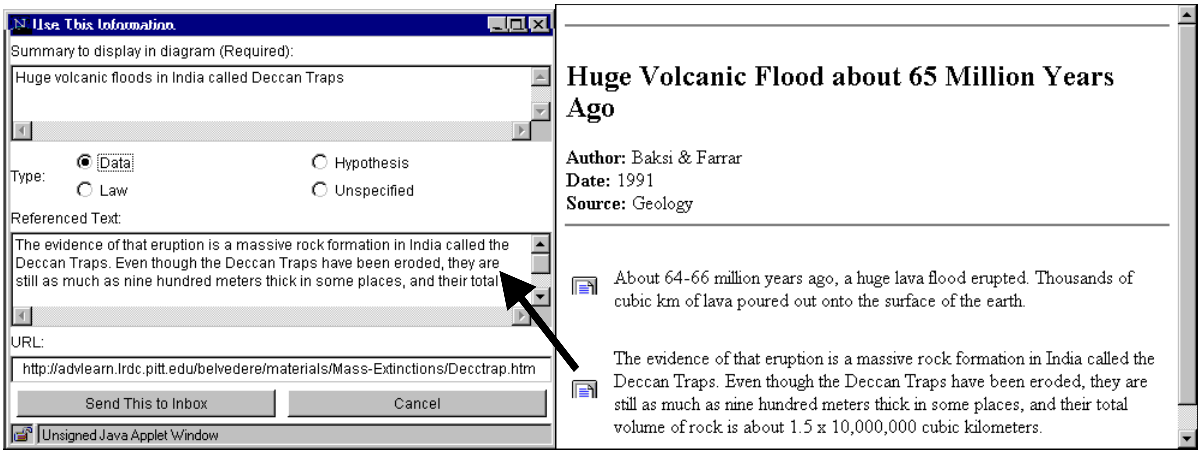


Figure 8. Generating a reference to a Snippet

The inset to the right shows the “data” statement that would be created by this dialog. As shown, the user’s wording is displayed in the diagram. The “link” icon in the upper right corner of the “data” shape indicates that it has a URL reference to a source page. One can reinvoke the web browser on this page by clicking on the link icon.

After authoring the snippet-annotated reference pages, teachers or domain experts can then construct an expert evidence map in Belvedere by using the buttons to send in references and connecting them with links. This map is converted and stored as an “expert graph.”

Then, during student sessions, students can use the “reference-this buttons” to send references to snippets into their diagrams, where they may express evidential relationships between the snippets. (Thus, “reference-this buttons” are the mechanism by which we obtain a correspondence between statements in users’ evidence maps and those in an expert graph.) The expert-path advisor will then compare consistency relations in the student’s evidence map with paths of consistency relations between the same statements in the expert graph. Mismatches in the polarity of these paths and/or the presence of extra information on the expert’s paths are be used to provide advice, as described below.

1.3.3.2 Computing Expert-Path Advice

The Belvedere 2.0 expert-path advisor was implemented in Lisp (along with one version of the evidence pattern advisor). One server-based advisor process serves multiple clients. Expert diagrams are read from the Postgres server into a Loom knowledge base and instantiated as Loom objects. During a session the expert diagram is read-only and not visible to the students. Each time a change occurs in a student diagram, the expert advisor notes the change, and the Loom knowledge base is updated with the new information.

As students construct an evidence map, they may include references to expert "snippets." The expert-path advisor is utilized only when a student assigns a relationship between two of these references with a "for," "against," or "and" link. The expert-path advisor has no advice on statements that did not reference snippets, but can work with diagrams containing such statements. The evidence-pattern advisor can respond to such non-snippets.

After an initial experimental implementation using a marker-passing algorithm in Belvedere 1.0 (Paolucci *et al.*, 1996), the expert advisor was implemented with an A* best-first heuristic search (Nilsson 1980) in Belvedere 2.0 (Toth *et al.* 1997). The search finds an optimal path from the start node to the goal node in the expert diagram according to the following cost heuristics. (The start and goal statements in the student diagram must be snippets and must also exist in the expert diagram.)

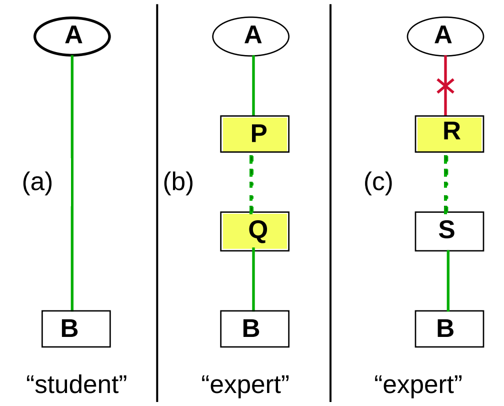


Figure 9. Comparison of student to expert graph

1. Shorter paths are given lower costs, based on the heuristic that more direct relationships are less likely to lead to obscurely related information. This heuristic takes precedence over the following two.
2. If the student has indicated a "for" link, all paths in the expert diagram that contain a single "against" link will be assigned lower costs than paths with only "for" links. Likewise, if a student has indicated an "against" link, all paths in the expert diagram that contain only "for" links will be assigned lower costs than paths with "against" links. This addresses the confirmation bias by seeking information that might contradict the student's link.
3. Paths with more than one against link are given higher costs than other paths. As previously noted, experience showed that the meaning of such paths is unclear to users.

Once a lowest-cost path is found between the start and the goal statements, advice is generated as follows:

- ♦ When the expert diagram has a direct link between the start and the goal, simple feedback is generated based on a comparison to the student's link:
  - If a student has indicated a "for" link between the start and goal, and the expert diagram has an "against" link between them, return an AAR (advice activation record) that would ask the student to reconsider the link.
  - If a student has indicated an "against" link between the start and goal and the expert diagram has a "for" link between the start and goal, return an AAR that would ask the student to reconsider the link.

- If the links agree, return an AAR that would indicate agreement.

- ♦ When a nontrivial path is found between the start and the goal, the advisor can confront the student with information that may contradict or corroborate the student's link as follows:
  - If the student has connected two snippets with a "for" link (e.g., Figure 9a), and the lowest cost path in the expert evidence map has an "against" link in it, identify the statement connected by the "against" link that is internal to the path (e.g., node R of Figure 9c), and return an AAR that would bring this statement to the attention of the student
  - If the student has connected two snippets with an "against" link, and the lowest cost path in the expert evidence map that consists entirely of "for" links, return an AAR that would bring the student's attention to statements in that path (e.g., if Figure 9a were an inconsistency link, communicate nodes P and Q of Figure 9b).
  - If the student's path is of the same polarity as the expert's path, return an AAR that would agree with the student's link, but elaborate on it by presenting an internal node (e.g., P and Q of Figure 9b in response to Figure 9a).

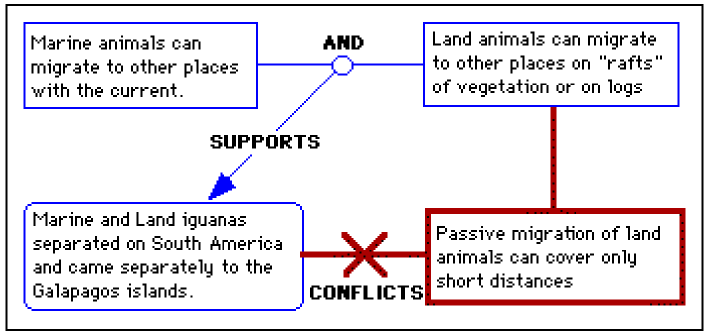
Our implementation presents the selected snippet in a pop-up dialog. A better approach might be to show users the web page containing the source information, or, for students requiring more scaffolding, to temporarily display the relevant portion of the expert graph. Presentation could also be sensitive to whether or not the student has viewed the source web page.

All of the above strategies are advice *generators*; it remains for the preference mechanism discussed previously to decide when the generated advice is actually worth giving. One preference was added to promote expert path advice over others, because this advice is more specific to the situation at hand than the evidence-pattern advice. This arbitration scheme can easily be extended to manage additional sources of advice.

1.3.3.3 Formative Experiments

Although the expert-path advisor has not been deployed in classrooms, formative evaluation took place during development. We conducted two experiments with Belvedere 1.0's version of the expert-path advisor (Paolucci *et al.* 1996). In the first experiment we were interested in testing consistency relations that we expected to be difficult or that required some inferential power. We used a subset of the "iguana problem" knowledge base used in some of the studies with students, comprised of 19 nodes, 14 consistent and inconsistent relations, and 2 and-links. Three of the present authors made judgments of consistency between pairs of statements corresponding to the nodes. Then we compared our judgments with the advisor's judgments. In all the relations about which all three authors agreed, the advisor made the same judgment. The only disagreements were on relations about which the authors disagreed. These cases were all characterized by the lack of a connecting path between the tested nodes. Either the search process was blocked by an inconsistency link, or a critical link was missing in an intermediate step of the process.

In the second experiment, we were concerned with the advice that would be given in a real interaction with students. We constructed a consistency graph of 90 statements and 73 relations from the materials used in one of the sessions with students, and performed path analyses on each link from two student sessions. The performance was similar to the previous experiment. We always agreed with the system's judgment, and the intermediate steps were sequences of coherent proofs. On most of the links the advisor agreed with the students (these were among our best students). In one case only, the advisor gave a different judgment: see the support link in Figure 10. (This study was performed with the earlier representational toolkit that differentiated "supports," "explains," and "predicts.") The path the advisor constructed starts at the "and" node, crosses the upper right and lower right nodes (not displayed in the students' graph), and ends at the lower left node. The advisor recognizes that this path (shaded) crosses an inconsistency link, and so conflicts with the students' support link. If the students would ask the advisor for a critique of their argument, the advisor would highlight the link and display the node on the lower right (the only information on the path that they have not seen), confronting them with the conditions for land animals' migration which they overlooked.[14]



Thin lined statements and links are in the students' diagram; thick lined items are only in the expert graph.

Figure 10. Example of expert-path advisor

Although we have selected an appropriate level of representation, the snippet, to allow the student to access domain-relevant material, we have also considered the pedagogical value of both a finer and a coarser grain size. A finer grain would reduce ambiguity and increase the accuracy of feedback. On the other hand, a coarser grain, i.e., at the level of a normal paragraph, or of a typical Web document, would enable quicker authoring of the Web-based materials described earlier. The model of advising with a larger grain size would be an "FYI" advisor, which would function like a research librarian forwarding new information to those likely to be interested in it. It would still be possible to specify "for" and "against" relations in a general sense, just as a paper can give evidence for or against a particular view. However, coarse-grained representation has obvious limitations. For example, it is important for students to learn that one can often extract evidence for a view from a

[14] However, Dr. Ellen Censky has evidence that land iguanas migrated between Caribbean islands 200 miles apart on trees downed during a hurricane in 1995.

paper that is generally unfavorable to that view. Indeed, scientific papers are obliged to take note of divergent views and limitations.

1.3.4 Comparison of Advisors and Future Directions

Table 2 summarizes a comparison between the two advisors. The evidence-pattern advisor can make suggestions to stimulate students' thinking with no knowledge engineering required on the part of the teacher or domain expert. However, the advice is very general. It could better address the confirmation bias by confronting students with discrepant information they may be ignoring. The expert-path advisor can provide students with assistance in identifying relevant information which they may not have considered (perhaps due to the confirmation bias), and which may challenge their thinking. The pattern-based advisor cannot provide this assistance, because it requires a model of evidential relationships between the units of information being manipulated by students. With the expert-path advisor, we have shown this assistance can be provided without deep modeling of or reasoning about the domain.

An attractive option is to combine the two advisors reported herein. Patterns could be matched to both student and expert diagrams to identify principled ways in which students might engage in additional constructive inquiry along with information that is relevant to that inquiry. For example, if the pattern matches the expert's graph but one pattern component is missing in the student's graph, the advisor could then present this information as indicated by the missing component's role in the pattern.

In both advisors, the knowledge engineering demands on educators who prepare materials for students are very low. Clearly, a minimal semantic approach has limitations. For example, the advisor cannot help the student in the construction of an argument, find a counter argument that

	Evidence Pattern Advisor	**Expert Path Advisor**
Knowledge Required	Principles of scientific inquiry (author once for many domains)	Expert evidence map (author for each area of inquiry)
Inference Required	Pattern matching	Search for and compare paths
Advantages	Expresses general principles in terms of student's constructions	Can point out relevant information
	Very general; widely applicable without additional knowledge engineering	No special training needed for authoring
Functional Limitations	Cannot point out relevant information due to lack of model of domain.	Shallow domain model does not support advice on causal or temporal reasoning

Table 2. Comparison of Belvedere's Advice Strategies

attacks her theory, or engage the student in a scientific discussion of causal or mathematical models underlying the theories. It cannot infer the goals of the student, in particular which theory she is trying to build or support. However, continued investigations of the utility of advice obtained from these minimal semantic annotations will provide insight into the cost-benefit tradeoff between knowledge engineering and educational gains, and point the way toward further artificial intelligence approaches that may be worth pursuing.

1.4 Alternative Approaches to Artificial Intelligence and Education

We have discussed our changing view of the role of representations in supporting learning interactions, and our adoption of an incremental approach to the design of minimal automated advisors that can yet usefully contribute to these learning interactions. In this work, those of us who are trained in Artificial Intelligence have found new ways to apply the methods and sensitivities of our field to education. The chapter concludes with a summary of these alternative approaches.

1.4.1 "Strong" AI&Ed

The phrase "Artificial Intelligence and Education" most immediately brings to mind the endeavor to build smart machines that teach. Ideally, under this vision, such machines would "know" a great deal about a particular subject matter, being able to both articulate concepts and principles and engage in expert level problem solving behavior (Clancey & Letsinger, 1981; Reiser *et al.*, 1985). They would also know about pedagogy, being able to track the progress of individual students and choose the best feedback strategies and trajectory through a curriculum for a particular student (VanLehn, 1988). This vision of AI&ED might be termed "strong AI&ED."

"Strong"[15] approaches to AI&ED have been behind work resulting in major contributions to Artificial Intelligence, and (less often) education. For example, Clancey's efforts to transform a rule-based expert system, MYCIN, into a teaching machine, drawing upon the clinical knowledge supposedly embodied in MYCIN, led to fundamental insights into the limitations of rule-based systems for supporting explanation and the need for causal, conceptual, and strategic knowledge structures (Clancey, 1983; 1986). Early work on instructional simulations on the SOPHIE and STEAMER projects have led a long and fruitful research program in automated qualitative reasoning (de Kleer & Brown, 1984; Forbus, 1984) resulting in simulations with new pedagogical capabilities (Forbus, 1997; Forbus & Whalley, 1988).

Some criticize "strong AI&ED" approaches to computer-supported learning, questioning whether computers can know enough about the student (Self, 1988) the domain, or teaching; or

[15] "Strong AI&ED" versus "minimalist AI&ED" is not identical to "strong methods" versus "weak methods," although there is a relationship. Strong methods are domain specific procedures that are justified by, if not compiled from, a great deal of domain knowledge. Weak methods are domain independent, may require encoding of significant domain knowledge to be applied, and may engage in a great deal of inferencing. Strong AI&ED makes significant use of at least one of these two. Minimalist AI techniques minimize both knowledge and inferencing.

questioning whether observed learning gains are actually due to the artificial intelligence elements, or to contextual factors (Nathan, 1988). Skepticism concerning the potential of "strong" approaches is warranted. However, in our opinion some such efforts are worthwhile for the synergistic interaction of AI and education that benefit further understanding in both fields, provided other approaches that promise to yield more immediate benefits are pursued as well.

1.4.2 "Minimalist" AI and Education

Contributions are also being made by others who take an approach we will characterize as "minimalist" AI&ED (Nathan, 1998; Schank & Cleary, 1995). The advisors discussed in this chapter are an example of "minimalist" AI&ED. Instead of attempting to build relatively complete knowledge representations, reasoning capabilities and/or pedagogical agent functionality, this alternative approach provides machines with minimal abilities to respond (in a manner believed to be educationally relevant) to the semantics of student activities and constructions; tests the educational value of these minimal abilities, and adds functionality as needed to address deficiencies in the utility of the system. An incremental approach interleaved with evaluation keeps the work focused on technologies with educational relevance. It also provides a viable research strategy, ensuring that we evaluate the capabilities and limitations of each representational and inferential device unencumbered by the simultaneous complexities of an attempted complete pedagogical agent.

The feedback provided by a minimalist approach may be characterized as "state-based" rather than "knowledge-based" (Nathan, 1988): the software helps students recognize important features of their problem solving state. A minimalist approach is consistent with instructional approaches in which students are expected to take greater responsibility for management of their learning, including diagnosis.

1.4.3 "Residual" AI and Education

The design history of Belvedere's representational tools suggests to us that the relevance of AI for education goes beyond attempts to build reasoning machines, even of the minimalist sort. Artificial Intelligence offers concepts and techniques that can be applied to the design of software that would not itself be considered an artificial intelligence at any level, yet which constitutes a contribution of AI to education, and potentially even a source and test-bed of AI ideas. This kind of application can be seen most clearly in the design of representational systems. An artificial intelligence sensitivity to the expressive and heuristic utility of formal representations for automated reasoning can be applied to the analysis and design of external representations for human reasoning as well as machine reasoning (Larkin & Simon, 1987; Stenning & Oberlander, 1995). External representations for learning and problem solving can differ in their expressiveness and in their heuristic bias – the perceptual salience of different kinds of information. Such differences can be exploited to design interactive learning environments that guide individual and group learning activities. The AI "in" software systems built under this approach is residual, influencing the design but being a run-time factor only for human rather than artificial agents. Examples of work in this category include Kaput (1995), Koedinger (1991), Reusser (1993), and Suthers (1999).

1.5 Acknowledgments

The authors thank Violetta Cavalli-Sforza for conceiving and programming the predecessor to Belvedere, and for contributions to the design of Belvedere 1.0; Johanna Moore for contributions to Belvedere 1.0; Dan Jones and Kim Harrigal for work on the Belvedere 2.x server and clients; Sandy Katz for empirical studies that informed the design of the expert-path advisor; Cynthia Liefeld for her recent assistance on empirical studies; and Micki Chi for discussions concerning the role of representations in learning. Work on Belvedere was funded by DoDEA's Presidential Technology Initiative (1997-1998); DARPA's Computer Aided Education and Training Initiative (1995-1997); and the NSF Applications of Advanced Technology program (1992-1995).

1.6 References

Amarel, S. (1968). On representations of problems of reasoning about actions. *Machine Intelligence, 3*, 131-171.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences,* 4(2), 167-207.

Bates, R., & MacGregor, R. (1987). *The Loom Knowledge Representation Language* (ADA 183415 RS-87-188). Marina del Rey, CA: Information Sciences Institute, University of Southern California.

Bell, P. (1997). Using argument representations to make thinking visible for individuals and groups. In *Proceedings of the Computer Supported Collaborative Learning Conference '97,* pp. 10-19. University of Toronto, December 10-14, 1997.

Brown, A. L. & Campione, J. C. (1994). Guided discovery in a community of learners. In *K. McGilly (Ed), Classroom Lessons: Integrating Cognitive Theory and Practice.* Cambridge: MIT Press, 1994. pp. 229-270.

Chan, T. W., & Baskin, A. B. (1988, June 1-3). *Studying with the prince: the computer as a learning companion.* Paper presented at the Proceedings of the International Conference on Intelligent Tutoring Systems, Monteal.

Chi, M.T.H., Bassok, M., Lewis, M., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science,* 13:145-182, 1989.

Clancey, W. J., & Letsinger, R. (1981). *Neomycin: Reconfiguring a rule-based expert system for application to teaching.* Paper presented at the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, August, 1981.

Clancey, W. J. (1983). The epistemology of a rule-baed expert system: A framework for explanation. *Artificial Intelligence, 20*(3), 215-252.

Clancey, W. J. (1986). From GUIDON to NEOMYCIN and HERACLES in twenty short lessons. *AI Magazine*, *7,* 40-60.

Clark, H.H. & Brennan, S.E. (1991). Grounding in Communication. In L.B. Resnick, J.M. Levine and S.D. Teasley (eds.), *Perspectives on Socially Shared Cognition,* American Psychological Association, 1991, pp. 127-149.

Corbett, A. T., & Anderson, J. R. (1990). *The effect of feedback control on learning to program with the Lisp tutor.* Paper presented at the Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.

Craik, F. I. M., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior.* 11: 671-684.

de Kleer, J., & Brown, J. S. (1984). A physics based on confluences. *Artificial Intelligence, 24*, 7-83.

Dunbar, K., & Klahr, D. (1989). Developmental differences in scientific discovery processes. In D. Klahr & K. Kotovsky (Eds.), *Complex Information Processing: The Impact of Herbert Simon* (pp. 109–143). Hillsdale, NJ: Lawrence Erlbaum Associates.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence, 24,* 85-168.

Forbus, K. (1997). Using qualitative physics to create articulate educational software. *IEEE Expert, 12*(3).

Forbus, K. D., & Whalley, P. B. (1998). Using qualitative physics to build articulate software for thermodynamics education: A preliminary report. *Journal of Interactive Learning Environments,* 1(1), 19-32.

Greeno, J. G., Collins, A. M., & Resnick, L. B. (1996). Cognition and Learning. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of Educational Psychology* (pp. 15-46). New York: Simon & Schuster Macmillan.

Hayes, J. R. (1989). *The complete problem solver*. Hillsdale, NJ, Lawrence Erlbaum Associates.

Kaput, J. (1995). Creating Cybernetic and Psychological Ramps from the Concrete to the Abstract: Examples from Multiplicative Structures. In D. Perkins, J. Schwartz, M. West, & M. Wiske (Eds). *Software Goes to School: Teaching for Understanding with New Technologies,* pp. 130-154. New York: Oxford University Press 1995.

Katz, S., & Suthers, D. (1998). Guiding the development of critical inquiry skills: Lessons learned by observing students interacting with subject-matter experts and a simulated inquiry coach. Paper presented at the American Educational Research Association 1998 Annual Meeting, April 13-17 1998, San Diego, CA.

Klahr, D. and M. Robinson (1981). “Formal assessment of problem-solving and planning processing in school children.” *Cognitive Psychology* 13: 113-148.

Koedinger, K. (1991). On the design of novel notations and actions to facilitate thinking and learning. *Proceedings of the International Conference on the Learning Sciences,* pp. 266-273. Charlottesville, VA: Association for the Advancement of Computing in Education. 1991.

Koedinger, K. R., Suthers, D. D., & Forbus, K. D. (1998). *Component-Based Construction of a Science Learning Space.* Paper presented at the Proceedings of Intelligent Tutoring Systems 98, San Antonio, TX.

Kotovsky, K. and H. A. Simon (1990). “What makes some problems really hard: Explorations in the problem space of difficulty.” *Cognitive Psychology* 22: 143-183.

Larkin, J. H. & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science 11(1):* 65-99. 1987.

Marshall, C. C., & Rogers, R. A. (1992). Two Years before the Mist: Experiences with Aquanet. Paper presented at the *Proceedings of the Fourth ACM Conference on Hypertext*, Milano, November 30-December 4, pp. 53-62.

McGuiness, C. (1986). “Problem representation: The effects of spatial arrays.*” Memory & Cognition* 14(3): 270-280.

McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human-Computer Interaction,* 5(4), 381-413.

Nathan, M. J. (1998). Knowledge and situational feedback in a learning environment for algebra story problem solving. *Interactive Learning Environments*, *5*, 135-159.

Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Palo Alto: Tioga.

Novak, J. (1990). Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching 27(10)*: 937-49.

Okada, T. & Simon, H. A. (1997). Collaborative discovery in a scientific domain. *Cognitive Science 21(2):* 109-146.

O’Neill, D. K., & Gomez, L. M. (1994). The Collaboratory Notebook: A distributed knowledge-building environment for project-enhanced learning. In *Proceedings of Ed-Media ‘94,* Vancouver, BC.

O’Neil, H. F., & Klein, C. D. (1997). *Feasibility of Machine Scoring of Concept Maps* (CSE Technical Report 460): National Center for Research on Evaluation, Standards, and Student Testing.

Paolucci, M., Suthers, D., & Weiner, A. (1996). Automated advice-giving strategies for scientific inquiry. *Intelligent Tutoring Systems, 3rd International Conference,* Montreal, June 12-14, 1996.

Perkins, D.N. (1993). Person-plus: A distributed view of thinking and learning. In *G. Salomon (Ed). Distributed Cognitions: Psychological and Educational Considerations* pp. 88-111.Cambridge: Cambridge University Press.

Perkins, D. N., Crismond, D, Simmons, R., & Unger, C. (1995). Inside Understanding. In *D. Perkins, J. Schwartz, M. West, & M. Wiske (Eds) Software Goes to School: Teaching for Understanding with New Technologies*, pp. 70-87. New York: Oxford University Press 1995.

Reiser, B. J., Anderson, J. R., & Farrell, R. G. (1985). *Dynamic student modeling in an intelligent tutor for LISP programming.* Paper presented at the Ninth International Joint Conference on Artificial Intelligence, Los Angeles.

Resnick, L., & Chi, M. T. H. (1988). Cognitive psychology and science learning. In M. Druger (Ed.), *Science for the Fun of it: A Guide to Informal Science Education* (pp. 24-31): National Science Teachers Association.

Reusser, K. (1993). Tutoring systems and pedagogical theory: Representational tools for understanding, planning, and reflection in problem solving. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 143-177). Hillsdale, NJ: Lawrence Erlbaum Associates.

Roschelle., J. (1994). Designing for cognitive communication: Epistemic fidelity or mediating collaborative inquiry? *The Arachnet Electronic Journal on Virtual Culture,* 2(2), 1994.

Ruiz-Primo, M. A., Shavelson, R. J., & Schultz, S. E. (1997*). On the Validity of Concept Map-Base Assessment Interpretations: An Experiment Testing the Assumption of Hierarchical Concept Maps in Science* (CSE Technical Report 455). Los Angeles, CA: University of California, Los Angeles.

Salomon, G. (1993). No distribution without individuals' cognition: A dynamic interactional view. In *G. Salomon (Ed). Distributed cognitions: Psychological and Educational Considerations* pp. 111-138.Cambridge: Cambridge University Press.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *Journal of the Learning Sciences, 3*(3), 265-283.

Schank, R. C., & Cleary, C. (1995). *Engines for Education*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schoen, D. (1992). Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems Journal - Special Issue on AI and Design,* Vol. 5, No. 1, 1992, pp. 3-14.

Self, J. (1988). Bypassing the intractable problem of student modeling. *Proceedings of the First International Conference on Intelligent Tutoring Systems,* Montreal, June 1-3, pp. 18-24.

Shipman, F. M., & McCall, R. (1994). Supporting Knowledge-Base Evolution with Incremental Formalization Design Evaluation. Paper presented at *the Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems,* Boston, April 24-28, pp. 285-291.

Smolensky, P., Fox, B., King, R., & Lewis, C. (1987). Computer-aided reasoned discourse, or, how to argue with a computer. In R. Guindon (Ed.), Cognitive science and its applications for human-computer interaction (pp. 109-162). Hillsdale, NJ: Erlbaum.

Stenning, K. & Oberlander, J. (1995). A cognitive theory of grahical and linguistic reasoning: Logic and implementation. *Cognitive Science 19(1):* 97-140. 1995.

Streitz, N. A., Hannemann, J., & Thuring, M. (1989). From ideas and arguments to hyperdocuments: Traveling through activity spaces. Paper presented at *Hypertext '89*, Pittsburgh, PA.

Suthers, D. (1993). *Preferences for Model Selection in Explanation.* Paper presented at the Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France.

Suthers, D. (1995). Designing for internal vs. external discourse in groupware for developing critical discussion skills. *CHI'95 Research Symposium.* Denver, May 1995.

Suthers, D. D. (1999). Representational Support for Collaborative Inquiry. Paper to be presented at the *Hawai'i International Conference on System Sciences,* Maui, Hawai'I, January 5-8, 1999.

Suthers, D. & Jones, D. (1997). An architecture for intelligent collaborative educational systems. *AI-Ed 97, the 8th World Conference on Artificial Intelligence in Education,* Kobe Japan, August 20-22, 1997.

Suthers, D., Toth, E., and Weiner, A. (1997). An Integrated Approach to Implementing Collaborative Inquiry in the Classroom. *Proc. 2nd Int. Conf. on Computer Supported Collaborative Learning (CSCL'97),* Toronto, December 10-14, 1997. pp. 272-279.

Suthers, D. and Weiner, A. (1995). Groupware for developing critical discussion skills. *CSCL '95, Computer Supported Cooperative Learning,* Bloomington, Indiana, October 17-20, 1995.

Suthers, D., Weiner, A., Connelly, J. and Paolucci, M. (1995). Belvedere: Engaging students in critical discussion of science and public policy issues.AI-Ed 95, the 7th World Conference on Artificial Intelligence in Education., August 16-19, 1995, Washington DC

Toth, J., Suthers, D., and Weiner, A. (1997). Providing expert advice in the domain of collaborative scientific inquiry. *8th World Conference on Artificial Intelligence in Education (AIED'97),* Kobe, August, 1997.

Utgoff, P. (1986). Shift of bias for inductive concept learning. In *R. Michalski, J. Carbonell, T. Mitchell (Eds.) Machine Learning: An Artitificial Intelligence Approach, Volume II,* Los Altos: Morgan Kaufmann 1986, pp. 107-148.

VanLehn, K. (1988). Student Modeling. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems* (pp. 55-78). Hillsdale, NJ: Erlbaum.

Walker, M. A. (1993). *Informational Redundancy and Resource Bounds in Dialogue*. Ph.D. Thesis, University of Pennsylvania, Philadelphia.

Webb, N. & Palincsar, A. (1996). Group proxesses in the classroom. In *Handbook of Educational Psychology*, D. Berlmer & R. Calfee, Eds. Simon & Schuster Macmillian, New York 1996.

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21(2)*: 179-217, 1997.