

Simulating Tutors with Natural Dialog and Pedagogical Strategies

**Funded by the National Science Foundation (NSF)
Learning and Intelligent Systems (LIS) program**

October 1997 – October 2001

University of Memphis (unless noted)

Arthur C. Graesser, Prof. of Psychology & Mathematical Sciences

Stanley P. Franklin, Prof. of Mathematical Sciences

Max Garzon, Prof. of Mathematical Sciences

Barry Gholson, Prof. of Psychology

Douglas Hacker, Asst. Prof. of Educational Psychology

Xiangen Hu, Assoc. Prof. of Psychology

Roger Kreuz, Prof. of Psychology

William Marks, Assoc. Prof. of Psychology

Natalie Person, Asst. Prof. of Psychology, Rhodes College

Abstract

Studies indicate that human tutors provide the most effective form of instruction known (Bloom, 1984; Cohen, Kulik, & Kulik, 1982). They raise the mean performance about two standard deviations compared to students taught in classrooms. Intelligent tutoring systems offer excellent instruction, but not quite as good as human tutors. The best ones raise performance about one standard deviation above classroom instruction (e.g., Anderson, Corbett, Koedinger, & Pelletier, 1995). In other words, a human tutor can raise the student's grade by about two letter grades (e.g., from C to A) while a tutoring system can raise it by about one letter grade (e.g., from C to B). Our challenge is to create tutoring systems that are as effective as human tutors.

The biggest remaining difference between human tutors and tutoring systems is that human tutors use natural language and most tutoring systems do not. Although tutoring systems print natural language and sometimes speak it, they do not let the student type or speak in natural language. They accept only constrained language such as menu selections, mathematical expressions, key words, etc. This limits students' opportunities to generate deep explanations and have the tutor critique them. Since student generation of explanations is known to increase learning, we hypothesize that if tutoring systems could participate in natural language dialogs with students about deep explanations, then the systems would become much more effective, and possibly rival human tutors. This could have a significant impact on military, commercial and public education.

The proposed research project will (1) develop tools for building tutoring systems that conduct explanation-based natural language dialogs, (2) use the tools to develop tutoring systems for at least two task domains, and (3) evaluate their effectiveness compared to expert human tutors and to versions of the systems that use constrained language instead of natural language.

Our basic approach is to combine a shallow, statistical approach (Latent Semantic Analysis) with deep, symbolic approaches (the LCFlex parser, the Tacitus-Lite+ discourse interpreter, and the APE tutorial dialog manager). Although prototypes of these components have been developed in our earlier work, all require significant extensions to handle explanation-based tutorial dialogs. Such dialogs are less constrained than dialogs handled previously by symbolic approaches, and they require deeper processing than dialogs handled previously by statistical approaches. We believe our hybrid approach will yield both the robustness and depth of understanding that explanation-based tutorial dialogs require.

1 Introduction

Since the early days of building intelligent tutoring systems, we have been faced with the question of how to effectively communicate with students. A few early systems, such as Scholar (Carbonell, 1970) and Why (Collins & Stevens, 1982; Stevens & Collins, 1977), attempted to use natural language but their abilities were so limited that they could not be tested with target students. Communication has since been done via command lines, menus, and other formal, highly-constrained "languages" in place of natural language.

Over the years we have made substantial progress in our understanding of both tutoring and natural language. We hypothesize that these advances make it feasible to develop effective tutoring systems that can communicate via natural language dialog. Moreover, the lessons learned in developing constrained-language tutors can be profitably applied in developing tutors that communicate in natural language (NL). This is one reason to believe that the field is ready for significant advances. Another reason to expect success is that natural language understanding technology is much more robust than before. Techniques exist that can extract meaning from input that is poorly thought out, incomplete or ill-formed. A third reason for optimism is that much has been learned about human dialog and tutorial dialogs in particular. For instance, we expect reactive planning technology to yield interactive dialogs that are much more natural and pedagogically effective than older technologies. A fourth reason is that AI has discovered that there are many kinds of reasoning (e.g., deduction) that can be adequately represented and some that still present difficulties. If we tutor students on topics that can be represented with proven AI techniques, we can avoid many of the difficulties that bogged down earlier efforts at NL-based tutoring.

There are undoubtedly tutorial situations where natural language interfaces have no advantages over more constrained interfaces. For instance, when a tutor has a menu interface, the students must sometimes engage in

substantial thinking in order to figure out how to say what they want. If that thinking is actually useful pedagogically, then a menu interface may be more effective than a natural language interface. However, no such comparison has systematically been conducted. Now that natural language enterprise has advanced, we can now plan to do such comparisons.

Thus, we will address two major research questions:

- *Feasibility*: Is it possible to build good natural language interfaces for tutoring systems? What kinds of software and empirical tools can be built to help us?
- *Utility*: Under what circumstances do natural language interfaces make a tutoring systems more effective than interfaces based on menus or constrained languages?

We propose to address these issues by building a natural language-based intelligent tutoring system called Why2000. Like the original Why system (Collins & Stevens, 1982; Stevens & Collins, 1977), it will elicit explanations from students, often by asking “why” questions. In particular, a typical episode will consist of:

1. The tutor presents a situation and asks the student for an explanation of it.
2. The student types in an explanation, which can be arbitrarily long.
3. The tutor compares the student’s explanations to both correct and incorrect ones that it anticipates receiving.
4. If the explanation is incomplete, incorrect or poorly stated, the tutor conducts a tutorial dialog to correct the flaws.

In other words, students are asked to generate an explanation, and the tutor coaches them. We call this type of tutoring *coached explanation generation*. It is similar to coached problem solving, a type of tutoring that is well understood, except that the explanations are generated in natural language whereas the problems are solved in constrained languages, such as mathematics. Why2000 will also attempt to identify and correct misconceptions that the student manifests in their contribution.

Our project will take advantage of prior research conducted by the team members (reviewed below) on natural language tutors (AutoTutor, Circsim-Tutor, Atlas, BE&E tutor), coached problem solving tutors (Andes), robust parsing-based natural language understanding (LCFlex, ROSE), statistical approaches to natural language understanding and world knowledge representation (LSA, AutoTutor), natural language discourse (Coconut) and tutorial dialog management (APE).

Despite the state-of-the-art components, this project still pushes the envelop of computational linguistics. The challenge is to permit maximal freedom for student expression without making it impossible to understand them. In particular, step 2 above enables students to generate potentially complex explanations, which is known to be a useful pedagogically. Yet the tutor can still understand them because it has strong expectations based on a corpus that we will collect of hundreds of student explanations for that exercise. For the tutorial dialogs of step 4, Why2000 asks the student questions that tend to elicit short, easily understood answers. In addition to these key pedagogical design features, the tutor will understand the student’s input via both deep analyses based on compositional semantics and shallow analyses based on statistical techniques. This should also increase robustness, coverage and speed.

In order to investigate the utility of various kinds of communication, Why2000 will support 3 input modalities and 4 output modalities. The input modalities will be (a) menus and fill-in-the-blank forms, (b) typed natural language, and (c) speech (to be developed if years 4 and 5 of the project are funded). The output modalities will be (a) canned text, (b) generated text, (c) generated speech, and (d) generated speech from an animated talking head.

Although we intend to build a system that is general enough to be used with any task domain with well-formed explanations (such as deductive reasoning), we need to test it on specific task domains with actual students in actual courses. Thus, we will work with the US Naval Academy to develop explanation exercises for their introductory physics course. This means that our initial knowledge bases will be oriented strongly towards basic physical concepts such as space, movement, force, energy and the laws that govern them, which should make it much easier to add, in the later years of the proposed work, the knowledge necessary for coached explanation generation in advanced courses, such as flight dynamics, electronics, hydrology or mechanical engineering.

2 Background

Research has shown that when students generate explanations, their learning increases. However, students may sometimes require help in order to generate useful explanations. This suggests building a tutoring system that

coaches explanation. Although many tutoring systems have been built, most have coached problem solving instead of explanation, so the proposed tutoring system will break new ground. It will require extensive use of natural language processing, which also makes it unusual and challenging to build.

In this section, we review prior research on (1) the pedagogical efficacy of student-generated explanations (2) the task domain of qualitative scientific explanations (3) intelligent tutoring systems and (4) natural language processing.

2.1 Intelligent tutoring systems

Intelligent tutoring systems (ITS) have come a long ways from the early days of Scholar (Carbonell, 1970) and Why (Collins & Stevens, 1982; Stevens & Collins, 1977). There are three common varieties: Model tracers, issue recognizers and question askers. Each will be briefly discussed in this section. The next section will discuss specific tutoring systems that are similar to Why2000.

Model tracing tutors coach students who are solving problems. As students work on the problem by making entries on a graphical user interface, and the tutor compares them to entries that its model of ideal reasoning would make. If the student's entries, match, the tutor gives positive feedback. If they do not match, it gives hints or negative feedback. In addition, students can ask for a hint if they get stuck, in which case the tutor uses its model of ideal reasoning to figure out what a good next entry would be, then gives a hint intended to lead the student toward it. For instance, the Andes tutor (Conati, Gertner, VanLehn, & Druzdzel, 1997; Gertner, Conati, & VanLehn, 1998; VanLehn, 1996) coaches students who are solving a quantitative physics problems, such as "A 10 kg block slides down a frictionless plane inclined at 25 degrees. What is its acceleration?" The student solves the problem by drawing vectors, defining variables, and writing equations. If an entry is correct, Andes colors it green. If it is incorrect, Andes colors it red. If the student asks for help, Andes gives hints designed for the current context. If the student does n't understand the first hint and requests another, Andes gives increasingly more specific hints until the student eventually sees what to do. In a recent evaluation at the US Naval Academy, students who used Andes to do their homework scored significantly higher on their midterm exams than students who did the similar homework without Andes. Andes is typical of model tracing tutors both in its philosophy of giving immediate feedback (which is based on experimental studies and observation of human tutors) and in its use of hint sequences.

Issue recognizing tutors also coach students who are solving problems. They are used when it is infeasible to adequately model ideal student reasoning, and yet certain aspects of the student's reasoning, called *issues*, are well enough understood that they can be recognized from the student's behavior. If an issue is recognized and it corresponds to desirable thinking, then it may be noted for assessment purposes and the student may be congratulated. If the recognized issue is not desirable, then the tutor may attempt to remedy it. For instance, Trio (Ritter & Feurzeig, 1988) trained F-14 radar intercept operators in tactical maneuvering. If the student was about to lose radar contact with the target by turning the plane away from it, an issue recognizer would tell them so.

Although many problem solving coaches give advice and help while the student is solving a problem, as Andes and Trio do, some prefer to give advice after the problem has been solved. Such "reflective debriefings" can be done within either the model tracing architecture (e.g., Katz et al., 1998; Lesgold, Lajoie, Bunzo, & Eggan, 1992) or the issue recognizing architecture (e.g., Mitrovic & Ohlsson, 1999).

The third kind of ITS, question asking tutors, do not coach students who are solving problems or performing other cognitive skills. Instead, they are designed to enhance the students conceptual, factual and declarative knowledge. They begin by asking the student a question that is intended to provoke deep reasoning. The students usually give a less-than-perfect answer, so the tutor asks follow-up questions or makes other comments that are intended to improve the student's understanding. For instance, TAP2 (Wong, Quek, & Looi, 1998) is a recent implementation of Why (Collins & Stevens, 1982), except that it uses a constrained language instead of natural language. It teaches the student about geography and other topics using Collins' inquiry pedagogy. For instance, it may ask a student whether Nigeria can grow rice. If the students says that it can't because it lacks rainfall, then the tutor will remind them that the Yangtze Plain grows rice even though it too lacks rainfall. This should prompt the student to realize that irrigation can substitute for rainfall.

Why2000 will be a question-asker. It will start the dialog by asking the student a deep-reasoning question, and many of its subsequent contributions to the dialog will be questions. However, as the next section indicates, it will be considerably different than other question-askers that use natural language.

2.2 Intelligent tutoring systems that communicate in natural language

Research on NL-based tutoring systems fizzled after an early start, but has recently been reborn. This section briefly reviews the major projects, leaving aside intelligent foreign language tutors (see Holland, Kaplan, &

Sams, 1995), whose concerns and techniques are considerably different from tutoring systems that use NL only as a medium of communication, and not as a topic of instruction.

As mentioned earlier, the first ITS were intended to use natural language. Scholar (Carbonell, 1970) and Why (Collins & Stevens, 1982; Stevens & Collins, 1977) had extremely limited natural language processing (NLP) capabilities, so they never made it to testing with real students. Sophie (Brown, Burton, & de Kleer, 1982) was the first tutoring system to have robust enough NLP that it could be used with real students. It presented an electronic circuit with a fault and would answer student's questions about it, such as "What is the voltage across R22?", "What would it be in a working circuit?" and "I think transistor Q2 is shorted across its base-emitter junction." Sophie's natural language understanding (NLU) was done with a now-standard technology, semantic grammars. Its dialog was purely reactive—it treated each student utterance as a command to be executed. For instance, if the student hypothesized a fault, Sophie would indicate whether it was correct and whether it was consistent with the student's measurements. Sophie's text was generated by templates. Nonetheless, Sophie was used with students at a technical school in Boston. One of the main discoveries was that students began to type shorter and shorter utterances as they became familiar with Sophie, which necessitated the development of strong mechanisms for handling anaphora and ellipsis. However, Sophie's knowledge of the task domain was a "black box expert," namely, a circuit simulation package that used partial differential equations and other numerical techniques. Thus, if the student was told that their hypothesized fault was not consistent with their measurements, and the student asked for further explanation, Sophie could not provide one. This suggested that if natural language interfaces are to be used with a tutoring system, then its representation of task knowledge ought to be a "glass box expert" in that it can explain any of its reasoning if asked.

After Sophie was finished in 1978, no major NL-based tutoring systems were built until just recently. The revival seems to have begun with Circsim-Tutor (Woo et al., 1991; Zhou et al. 1999). Circsim-Tutor is a question-asking tutor that helps medical students learn how blood pressure is regulated in the human body. It first asks students to fill out a chart indicating how the blood pressure regulating system would react to a disturbance. Then, in a reflective debriefing phase, it indicates the incorrect responses and engages the students in a natural language dialog designed to remedy their misconceptions. The dialogs were intended to imitate those observed in expert human tutors. The investigators discovered many tutorial strategies and tactics, and formalized them first as finite-state machines and later as interruptible plans. In order to make the natural language understanding problem simpler, the plans asked questions that invited only short answers. Consequently, robust information-extraction techniques (see below) were able to extract meaning from the student's utterances. The combination of highly planned dialog and information extraction NLU created a tutor that generates remarkably coherent dialog. Circsim-Tutor is still under development. Although it has been used with human students, no formal evaluations have been conducted yet.

The Basic Electricity and Electronics (BE&E) tutor (Rose, Di Eugenio, & Moore, 1999) teaches technical students about basic electricity. Like a model tracer, it gives feedback and hints when the student makes an error. However, it replaces the usual hint sequence with a natural language dialog. The major focus of this work so far has been on developing robust techniques for understanding the student's contribution to the dialog. These techniques will be described later.

AutoTutor (Graesser et al., in press) is a question-asking tutor for computer literacy students. It asks an open-ended question (e.g., "What happens when you boot your computer?"), lets students answer at length, then conducts a NL dialog aimed at improving the student's answer. Unlike the other NL-based tutors we review here, AutoTutor does not try to understand the student's utterances completely, because that is not necessary for its purposes. Its pedagogical philosophy, which is founded on detailed analyses of human tutors (Graesser, Person, & Magliano, 1995) as well as the general literature on the benefits of student-generated explanation (reviewed in section 3.1 above), is to keep prompting and guiding the student for more and more explanation until the student has hit all the main points. It also attempts to recognize common misconceptions and correct them. Because it does not need an in-depth understanding of the student's explanation in order to decide whether to prompt or correct, AutoTutor uses a statistical approach to understanding the student's natural language that will be described fully later. AutoTutor is still under development. It has been used with pilot subjects in formative evaluations, but it is still too early for a summative evaluation.

Like the BE & E tutor, Atlas (Freedman, 1999) is a model tracing tutor that also replaces the usual hint sequences with natural language dialog. It uses the NLU techniques of the BE&E tutor, and the dialog-generation techniques of CIRCsim-tutor. However, Atlas is intended to be an add-on to an existing model tracing tutor. Although it is being developed initially with the Andes tutor as its host, the Atlas group is actively collaborating with the Koedinger, Corbett and Anderson—the CMU group that has developed many major model tracing tutors. A first public demonstration of Atlas occurred recently, and it will be ready for pilot testing in spring 2000.

Although there are several projects developing NL-based tutoring systems, and this project shares key personnel with most of them, the proposed tutoring system is different from any of the systems that are currently being developed. First, it is a question asking tutor, rather than a model tracing or issue recognizing coach for students solving multi-step problems. This differentiates it from Atlas and the BE&E tutor. It is somewhat like Circsim-Tutor in that it conducts a dialog designed to remove scientific misconceptions, but it is intended to be much more general. It differs from AutoTutor in that it has a much more detailed representation of the kinds of explanation it would like to hear from students, and it has more elaborate dialog plans for remedying incomplete or incorrect explanations. Although Why2000 will be different from its predecessors, it will integrate and extend their techniques in order to solve even harder problems in NL-based tutoring than the existing systems are attempting to solve.

2.3 Research on human tutorial dialog

In order to build better NL-based tutors, it is wise to understand how human tutors conduct their dialogs. Fortunately, significant research on tutorial dialog that has accumulated during the last decade. Detailed discourse analyses have been performed on small samples of accomplished tutors in an attempt to identify systematic discourse patterns and sophisticated tutoring strategies (Fox, 1993; Hume, Michael, Rovick, & Evens, 1996; McArthur, Stasz, & Zmuidzinas, 1990; Merrill, Reiser, Ranney, & Trafton, 1992; Moore, 1995; Putnam, 1987). (Lepper, Woolverton, Mumme, & Gurtner, 1993; Merrill, Reiser, Merrill, & Landes, 1995; VanLehn, Siler, Murray, & Baggett, 1998). Unskilled tutors have been investigated by (Graesser et al., 1995; Person & Graesser, 1999). Some experiments have contrasted natural tutoring with highly simplified forms of tutoring (Chi, Siler, Jeong, Yamauchi, & Hausmann, in press). Available research on tutorial dialog (as cited above) has uncovered some characteristics of the dialog patterns of unskilled and expert tutors.

(1) Tutors set most of the agenda and follows a curriculum script. It is the tutor, not the student, who sets the agenda and introduces most of the topics, questions, and problems. It is widely acknowledged that the vast majority of students are not active, self-regulated learners who are aware of their knowledge deficits and who take command of the tutorial agenda. A typical student asks only 6-8 genuine information-seeking questions per hour. The tutor invokes a “curriculum script” of topics, problems, questions, and examples in a tutor-driven fashion. Stated differently, the matter of control in the mixed-initiative dialog is heavily slanted toward the tutor.

(2) Tutors focus on only one pedagogical goal at a time. The tutor selects a particular idea to focus on (i.e., a proposition, rule, claim, part of a good answer) and works on that idea until it reaches completion in the tutorial dialog. Thus, the tutor does not hop around several unfinished ideas. Similarly, the tutor addresses only one misconception at a time. We refer to these focal ideas and misconceptions as pedagogical goals or focal content.

(3) Tutors manage an embedded dialog while accomplishing a particular pedagogical goal. Students periodically ask counter-clarification questions and information-seeking questions which need to be answered by the tutor during the course of achieving a particular pedagogical goal. The learner might express confusion, which needs to be addressed. Tutors often have a prioritized stack of dialog moves when they attempt to get a learner to articulate the focal content of a pedagogical goal: General hint first, then progressively more specific hints, then simply assert the answer, then verify the learner’s understanding by additional dialog moves. This graduated specificity occurs in expert tutors, but is not routinely implemented by unskilled tutors.

(4) Tutors follow politeness norms by giving indirect feedback when there are error-ridden learner contributions. When the student expresses a bug or misconception, the tutor does not pounce on the student and say “no you’re wrong.” This would be a face-threatening act that would discourage the student from talking. Instead, tutors signal the occurrence of the error more indirectly and manage the corrections with indirect tactics. The tactics vary with expertise. Unskilled tutors quickly give positive or neutral feedback and correct the error with an additive expression (“Okay, but also it is true that X”). Skilled tutors pause (a signal of skepticism) and give an indirect hint.

(5) Tutors need to manage dialogs when there is minimal common ground. Tutors cannot achieve a deep and complete understanding of a student’s mental model when the student contributions are fragmentary, incoherent, underspecified, and vague. It is computationally difficult, if not impossible, to induce student knowledge. Instead, misunderstandings frequently occur as the tutor scrambles to piece together a minimal understanding of the student’s knowledge and to manage the discourse. The tutor needs to manage the vagueness and lack of common ground with discourse moves that are polite but reasonably effective pedagogically.

(6) Tutors need to be trained how to use dialog tactics that embrace sophisticated pedagogy. Unskilled tutors do not use most of the ideal tutoring strategies that have been identified in education and the intelligent tutoring system enterprise. These strategies include the Socratic method (Collins, 1985), modeling-scaffolding-

fading (Collins, Brown, & Newman, 1989), reciprocal training (Palincsar & Brown, 1984), anchored situated learning (Bransford, Goldman, & Vye, 1991; Greeno, Smith, & Moore, 1993), error diagnosis and correction (vanLehn, 1990; Lesgold, Lajoie, Bunzo, & Eggan, 1992), frontier learning, building on prerequisites (Gagne, 1977), and sophisticated motivational techniques (Lepper, Woolverton, Mumme, & Gurtner, 1993). Implementing such tactics should provide learning gains over and above what the unskilled tutors deliver.

(6) Skilled tutors are more prone to select dialog moves that encourage students to actively construct answers and solutions. A bad tutor lectures, whereas a good tutor gets the student to do the talking. Unskilled tutors sometimes summarize answers and solutions after the multi-turn dialogic construction of an answer/solution. Skilled tutors ask the students to summarize, to promote a more active construction of knowledge.

Although a great deal is now known about human tutorial dialogs, there is still much to learn. Part of our effort will be devoted to further studies of tutorial dialog. However, we have also learned some methodological lessons from our early work that suggest new forms of data collection and analysis.

2.4 Dialog management technology

In the last ten years, many researchers have studied the issue of dialog planning and management. These investigations generally fall into two categories. One group of researchers has concentrated on the issue of finding the right speech act to follow a given speech act in a dialog. In these systems, usually neither the structure nor the content of the dialogs is complex. For this purpose, finite state technology is generally sufficient. Expressibility is not usually an important consideration in such systems because they generally deal with concepts where a frame-based representation or just a topic name is sufficient. Domains that have been tried include travel reservations, making appointments, automating a telephone operator's job, and providing simple information about financial services. Examples of this work include Bilange (1991), Jönsson (1991), Pieraccini, Levin and Eckert (1997), Ehrlich (1999), and Papineni, Roukos and Ward (1999). These systems are often used in speech processing projects, where the real challenge is sentence-level understanding of the spoken text. They would not be appropriate in Why2000, where the focus will be on discourse and domain-level understanding of the student's explanations.

A slightly more sophisticated version is the tutor developed by Cawsey (1992), which uses Conversation Analysis (Sacks, Schegloff, & Jefferson, 1974) as the underlying dialog model. Although her approach permits some nested moves, it still assumes an overly simple domain model.

The second group of researchers, exemplified by Gerlach & Horacek (1989), Chu-Carroll and Carberry (1995), and Moore (1995), have done theoretical work on the use of various forms of first-order logic to determine the next dialog move. Although first-order logic allows for sophisticated knowledge representation, all of them have concentrated on the knowledge required for dialog handling alone rather than on its integration with realistic domain knowledge. Furthermore, the logic and knowledge representation in each of these systems is specific to the cases described and may be difficult to scale up. For instance, Smith (1992) built a tutor using a theorem prover as his reasoning engine, but his domain model is very simplified.

Neither group of researchers investigated keeping track of dialog history or the tutor's pending agenda. Thus neither has developed a sufficient model for human dialog processing, as Bratman (1987, 1990) has shown that people do not reason from scratch about each decision, but instead maintain an agenda that is changed when necessary. The closest pre-existing work is that of Jullien and Marty (1989), who represent the dialog plan as a changeable tree. Although the work of Wilkins (1988) was not developed explicitly for tutoring, it also uses tree transformations and was a significant influence on the design of the APE system discussed below.

Remediating student misconceptions requires the ability to keep track of the tutor's agenda, including nested dialog moves, as well as maintain a sophisticated representation for both dialog and control knowledge. Thus the plan-based technology developed for the Atlas system (Freedman, 1999) is more appropriate for Why2000 than either approach described here. The plan-based approach is discussed further in section 4.3.

2.5 Natural language understanding technology

In a dialog system, the natural language understanding module has two responsibilities. The first is to provide an analysis of the user's utterance that will help the dialog manager decide what to say next. The second is to update a data structure, often called the discourse or dialog history, that it will use to help it analyze subsequent user utterances.

There basically two technologies for NLU. One is based on compositional semantics. Compositional semantics refers to the fact that the meaning of a large unit of language can be computed from the meanings of its components. That is, the meaning of a paragraph is a composition of the meanings of its sentences; the meaning of a sentence is a composition of the meanings of its phrases; and the meaning of a phrase is a composition of the

meanings of its words. Technologies based on compositional semantics often build a hierarchical analysis of the user's utterance (e.g., a syntactic parse tree), then traverse it depth-first in order to build semantic structure for each unit of language by composing structures built for its constituents.

The other main NLU technology is based on classification. When a dialog manager has only a limited number of choices of what to say or do next, it need only classify the user's utterance according to what it needs to know in order to make that decision. For instance, if a simple travel advisor has just asked, "What airport will you be departing from?" then it might classify the user's utterance according to airports, cities and other geographic regions.

Often classification-based NLU is used with finite-state technology for dialog management, whereas compositional semantics is used with dialog managers based on planning or first-order logic. However, the two design choices are logically independent. For instance, the BE&E tutor used compositional semantics-based NLU with a finite-state dialog manager (Rose, Di Eugenio, & Moore, 1999).

NLU based on compositional semantics often uses two main modules. One module builds a parse tree or other hierarchical structure; the second module builds semantic structures. This two-module approach is not strictly speaking necessary, since semantic analysis can be conducted without first doing parsing (e.g., Schank, 1975). However, the two-module approach can take advantage of efficient parsing algorithms as well as broad-coverage syntactic grammars and lexicons.

Our basic approach is to use NLU based on both classification and compositional semantics, so we review both technologies below. The discussion of the compositional semantics approach is broken into two sections, one on parsing and the other on semantic interpretation.

2.5.1 NLU based on classification

Classification approaches to NLU include statistical (e.g., Charniak, 1993; Sanker and Gorin, 1993), information retrieval (e.g., Graesser, Wiemer-Hastings, Wiemer-Hastings, Harter, Person, and the TRG, in press; Wiemer-Hastings, Wiemer-Hastings & Graesser, 1999) and connectionist approaches (e.g., Miikkulainen, 1996). Notable among these approaches are those that are based on word co-occurrence patterns such as LSA (Dumais, 1994; Landauer & Dumais, 1997) and HAL (Burgess, Livesay & Lund, 1998). LSA has already proven successful for handling student input in the AutoTutor system (Graesser et al., in press; Wiemer-Hastings et al., , 1999). LSA is an attractive approach because it is trained on untagged texts and is a simple extension to keyword based techniques.

LSA has been remarkably successful at a number of natural language tasks. In the arena of query-based document retrieval (Deerwester et al., 1990; Dumais, 1994), LSA was compared to a large number of research prototypes and commercial systems. The performance of LSA varied from equivalent to the best other method to 30% better, with an average improvement of 16% over the competitors. The next success of LSA was in its modeling of human performance in the TOEFL test developed by Educational Testing Service (Landauer and Dumais, 1997). The LSA model answered 64.4% of the questions correctly, which is essentially equivalent to the 64.5% performance for college students from non-English speaking countries. Another of the recent successes is the repeated demonstration that LSA can grade the essays that college students write almost as well as human graders (Foltz et al., 1996; Landauer et al., 1998; Wolfe et al., 1998). Furthermore, LSA has been very impressive in accounting for (1) the developmental acquisition of vocabulary words, (2) the classification of words into categories, (3) the extent to which context activates the meaning of a word, (4) the amount of learning that occurs when students with varying degrees of domain knowledge read a text, and (5) the extent to which sentences in text are coherently related to each other

Although these success stories indicate the power of word co-occurrence to capture world knowledge, they don't actually test its capabilities as the NLU component in a NL-dialog system. That test is being conducted by using LSA as the workhorse for analyzing student input to AutoTutor (Graesser et al., in press; Wiemer-Hastings et al., , 1999). Although AutoTutor is still under development, experiments with an early version indicate that LSA's evaluations of college students' answers to deep reasoning questions were nearly as high as intermediate and accomplished experts of computer literacy. LSA was also capable of discriminating different classes of student ability (good, vague, erroneous, versus mute students) and in tracking the quality of contributions in tutorial dialog.

To use LSA, one must first develop an LSA space, which acts as a lexicon. The space represents the "meaning" of a word as a vector in a space of K dimensions (where K is typically 100 to 300). The space is built automatically from a training corpus. The corpus consists of a large number of "documents," where a document could be sentence, a paragraph or a longer unit of text. One computes a co-occurrence matrix that specifies the number of times that word W_i occurs in document D_j . A standard statistical method, called singular value

decomposition, reduces the large $W \times D$ co-occurrence matrix to K dimensions. This assigns each word a K -dimensional vector in the space.

Given an LSA space, the similarity (i.e., similarity in meaning, conceptual relatedness, match) between two words is computed as a geometric cosine (or dot product) between the two vectors, with values ranging from 0 to 1. The similarity between two sentences (or longer text s) is computed by first representing the meaning of each sentence as a vector that is the weighted average of the vectors for the words in the sentence, then computing the similarity between those two vectors as a geometric cosine. The match between two language strings can be high even though there are few if any words in common between the two strings. Thus, LSA goes well beyond simple keyword matches because the meaning of a language string is partly determined by the company (other words) that each word keeps.

Although it would seem that LSA is only useful for computing the similarity of two sentences, AutoTutor uses it to analyze student utterances in a dialog. Recall that AutoTutor begins by asking the student a question, then conducts a dialog that improves that answer until it is complete and correct. This means that AutoTutor must have some representation of what a complete and correct answer consists of. Such ideal answers are represented as a small (~7) set of "aspects," where each aspect is itself just a sentence or two of natural language. LSA is used to compare the student's answer to each of the aspects. If the similarity is above a threshold, then the student is deemed to have expressed that aspect. If the student's initial answer does not express all the aspects, then AutoTutor picks one and generates a prompt or hint intended to get the student to mention that aspect. This is how it handles incomplete answers. Its technique for handling partially incorrect answers is similar.

The chief advantage of AutoTutor's approach to NLU is that the space (lexicon) is built automatically and the aspects are just natural language that a trained non-programmer domain expert can enter. Thus, very little knowledge engineering is needed compared to that required by symbolic approaches. On the other hand, using LSA for NLU does not allow a deep understanding of the student's utterance. For instance, because it ignores the order of words in sentences, "X causes Y" has exactly the same representation (vector) as "Y causes X."

2.5.2 NLU based on compositional semantics: The parser

The first step to applying the compositional semantics approach to NLU is to find out what the hierarchical structure of the utterance is. Many investigators choose to use syntactic information in this phase, as it is often yields hierarchical structures that facilitate semantic interpretation. Regardless of whether degree to which syntax is used, let us call this process "parsing."

Although much of computational linguistics is concerned with parsing complex constructions or extracting and representing subtle differences in meaning, the chief issue in practical NLP applications is robustness: Developing components that can effectively handle input that is disfluent or extra-grammatical. As discovered in the Sophie (Brown et al., 1982) project and many other not-tutoring NLP projects, users not only deliberately shorten their utterances when typing them, but they express their thoughts poorly, choose wrong words, make grammatical mistakes and mistype. Traditional syntactic parsing algorithms that are designed to parse only completely grammatical input sentences are therefore unsuitable for such applications.

There are two basic approaches to robust parsing. The empirical approach acquires knowledge via machine learning or statistical processing of large collections (corpora) of text or speech. The symbolic approach uses linguistic knowledge bases, such as grammars, lexicons and semantic representations. Usually these knowledge bases are built by hand, but empirical techniques have recently begun to be used as well. The empirical and symbolic approaches will be reviewed briefly.

Empirical approaches to robust parsing include both statistical (Bod, 1998; Pietra et al., 1997; Rose & Waibel, 1997; Goodman, 1996; Miller et al., 1996; Charniak, 1993; Magerman & Marcus, 1990) and connectionist approaches (Henderson & Lane, 1998; Buo, 1996; Jain, 1991; Jain & Waibel, 1990). These approaches are trained using either supervised or unsupervised learning techniques. If they are trained on enough data, these approaches are inherently robust because they are trained to fit their behavior to the patterns that they typically encounter. Both statistical and connectionist approaches typically make their decisions based on "fuzzy" reasoning. This allows them to generalize well to new patterns, however it also places limitations on the depth of the analyses they can reliably construct.

The symbolic approach to robust parsing has been pursued by many investigators (Ait-Mokhtar & Chanod, 1997; Neumann et al., 1997; Lavie, 1995; McDonald, 1993a; Huyck & Lytinen, 1993; Hobbs, Appelt, Tyson, Bear, & Israel, 1992; Hipp, 1992; Lehman, 1989). To avoid failure when faced with extra-grammatical input, many systems relax syntax and grammar constraints. In some cases, semantic information is used to compensate for the lack of grammaticality in order to partially or completely drive the analysis process. Since our approach to robust parsing is symbolic, the existing approaches will be reviewed in some detail.

Early approaches to robust symbolic parsing involved hand-coded grammar-specific heuristics for selecting a subset of analyses to extend when each input word was processed (McDonald, 1993a, 1993b; McDonald, 1992; Hobbs et al., 1991). The downside of these approaches is that the work of making the parser robust and efficient must be redone by hand for every new grammar developed.

Perhaps the most complete approach to the problem of robust symbolic parsing is Minimum Distance Parsing (MDP) (Hipp, 1992; Lehman, 1989). When faced with input that is extra-grammatical, the goal of an MDP parser is to find the analysis of a corresponding grammatical input that is closest to the given input according to an “edit distance”. While full MDP parsers have been shown to be effective in small applications with small grammars (with two to five hundred rules), the approach does not scale up well to large applications (grammars with a thousand or more rules). As demonstrated in our previous work (Rose, 1997a; Rose and Lavie, 1997), with large coverage grammars, the search space that a full MDP parser must explore expands to magnitudes that render the approach computationally infeasible.

Recent approaches to robust parsing focus instead on shallow or partial parsing techniques (Van Noord, 1997; Worm, 1998; Mokhtar and Chanod, 1997; Abney, 1996). Rather than attempting to construct a parse covering an entire ungrammatical sentence, these approaches attempt to construct analyses for maximal contiguous portions of the input. Restrictive partial parsers are attractive because the majority of disfluencies encountered in spontaneous input can be handled effectively without the full power of MDP. The weakness of these partial parsing approaches is that part of the original meaning of the utterance may be discarded with the portion(s) of the utterance that are skipped in order to find a parsable subset. These less powerful algorithms essentially trade effectiveness for efficiency. Their goal is to introduce enough flexibility to gain an acceptable level of coverage at an acceptable computational expense.

Notable among partial parsing approaches is the vast literature on message understanding. The “message understanding” or MUC initiative, funded by DARPA, has evaluated the performance of natural language extraction systems developed in artificial intelligence and computational linguistics (DARPA, 1995, 1998; Jacobs, 1992; Lehnert, 1997). Information extraction techniques have also proven successful for input understanding in the context of the Circsim-Tutor system (Glass, 1999). Within the MUC community, there has been noticeable progress in automating many components of language analysis that lie within the span of a sentence and short discourse segments, such as identifying the correct sense of words with multiple senses, parsing sentence syntax for sentences that are short or moderate in length, linking a noun-phrase to a previous entity in the discourse history (i.e., anaphors, coreference), and extracting important information that is relevant to slots in structured templates.

Our approach (Rose & Lavie, to appear; Rose, 1999; Rose & Lavie, 1999; Rose, 1997a; Rose & Lavie, 1997) has proven effective for robust interpretation even in the face of spontaneous spoken input in a large-scale speech-to-speech translation system (Lavie et al., 1996; Suhm et al., 1994; Woszcyna et al., 1993). Spontaneous spoken language, whether transcribed by hand or by a speech recognizer, is often highly disfluent, with the meaningful portions of the utterance surrounded by a variety of phenomena that disrupt the grammaticality of the overall input stream. Our robust interpretation technology has proven effective even in the face of all of these difficulties. The application of our technology to the problem of machine translation also demonstrates the language independence of our approach, and thus the potential to easily adapt our technology to tutoring systems for speakers of languages other than English. Our LCFlex parser (Rose and Lavie, 1999) can be easily customized with respect to various types of flexibility, including: skipping over ungrammatical words and segments in the input, insertion of words or categories in specific contexts, and relaxation of grammatical constraints expressed via feature unification.

Some partial parsing approaches have been coupled with a post-parsing repair stage (Danieli and Gerbino, 1995; Rose and Waibel, 1997; Rose, 1997a; Van Noord, 1997). These two stage approaches have proven more tractable than the single stage MDP approach while achieving the same effective power. The goal behind two stage approaches is to increase the coverage over partial parsing alone at a reasonable computational cost by introducing a post-processing repair stage. The ROSE approach, introduced in (Rose, 1997a), uniquely achieves this goal without any hand coded knowledge specifically dedicated to repair. Our recent research has increased the efficiency of the original ROSE approach by at least an order of magnitude.

2.5.3 NLU based on compositional semantics: The semantic interpreter

Although the parser can build a hierarchical structure that spans a single sentence, parsers are seldom used to build a hierarchical structure that spans multiple sentences. Syntactic marking that guides parsing inside a sentence is weak or absent in discourse. Hence, in a dialog system, the emphasis during semantic interpretation is not only on building a deeper meaning structures from the shallow analyses constructed by the parser, but also on integrating the meanings of the multiple sentences that constitute the dialog.

There are two major lines of research for doing this: informational and intentional (following Hobbs). With the informational approach, the focus is on the meaning that comes from the semantic relationships between the utterance-level propositions (e.g. effect, cause, condition) whereas with the intentional approach, the focus is on recognizing the intentions of the speaker (e.g. inform, request, propose).

Work following the informational approach focuses on the question of how the correct inferences are drawn during comprehension given the input utterances and background knowledge. The earliest work tried to draw all possible inferences (Reiger, 1974; Schank, 1975; Sperber & Wilson, 1986) and in response to the problem of combinatorial explosion in doing so, later work examined ways to constrain the reasoning (DeJong, 1977; Schank et al., 1980; Hobbs, 1980). In parallel with this work, the notions of conversational implicatures (Grice, 1989) and accomodation (Lewis, 1979) were introduced. Both are related to inferences that are needed to make a discourse coherent or acceptable. These parallel lines of research converged into abductive approaches to discourse interpretation (e.g., Appelt & Pollack, 1990; Charniak, 1986; Hobbs et al., 1993; McRoy & Hirst, 1991; Lascarides & Asher, 1991; Lascarides & Oberlander, 1992; Rayner & Alshawi, 1992). The informational approach is central to work in text interpretation (e.g., MUC types of systems).

The intentional approach draws from work on the relationship between utterances and their meaning (Grice, 1969) and work on speech act theory (Searle, 1969) and generally employs AI planning tools. The early work considers only individual plans (e.g., Power, 1974; Perrault & Allen, 1980; Hobbs & Evans, 1980; Grosz & Sidner, 1986; Pollack, 1986) whereas now there is progress on modeling collaborative plans with joint intentions (Grosz & Kraus, 1993; Lochbaum, 1994). The intentional approach is central for most work on dialog since the collaborative aspect of dialog needs to be captured.

In addition to the plan-based approach we mentioned above for the intentional line of research there is also the dialog grammar method. Instead of recognizing intentions it assumes that speech acts have been assigned to utterances and then uses this knowledge in a purely expectation oriented fashion. The expectations arise from observed sequencing regularities in dialog, called adjacency pairs (Sacks et al., 1978) (e.g. questions are generally followed by answers, proposals by acceptances, etc.). One can write grammar rules based on these regularities that state sequential and hierarchical constraints on acceptable dialogs, just as syntactic grammar rules do for utterances. The analysis of the dialog then parallels what happens for sentence-level parsing. However, it isn't as parallel as one might hope. One problem is that a speech act must be recognized for each utterance. This still requires intention recognition which brings us back to a difficult problem. It is also difficult for humans to assign speech acts to utterances because they are multifunction (e.g. an utterance can both inform and reject) and this type of approach can only respond to one of those speech acts. Likewise there is no guidance for how to proceed when there are multiple paths to follow for responding.

As most researchers allow that both intentional and informational knowledge can underlie the meaning of a discourse, there is recent work to combine aspects of both (Kehler, 1995; Moore & Pollack, 1992; Thomason & Hobbs, 1997). Although the informational approach relies on inference and abduction tools, these tools can also be used for intentional approaches as well. In the COCONUT project, we implemented aspects of the intentional approach using an abduction tool, Tacitus-Lite+ (Thomason & Hobbs, 1997).

Certainly for an explanation tutor that interacts with its student via dialog both the intentional and informational aspects of the dialog will be important. The informational aspect for understanding the explanation and the intentional aspect for having a successful mixed-initiative interaction with the student. The abductive tool, Tacitus-Lite+ will be used to address the informational aspect and the implemented planning axioms described in (Thomason & Hobbs, 1997) to address the intentional.

2.5.4 The key role of expectations

Regardless of which approach one uses for NLU, success depends strongly on being able to anticipate or constrain the relevant content of the user's utterances. For example, in the MUC initiative, the relevant content was constrained to be about terrorism kidnappings or finance, so only meaning representations for those topics were originally stored and anticipated by the computer system. It is beyond the capabilities of current NLU systems to construct new microworlds, templates, and mental models from scratch, and to intelligently handle completely novel answers and contributions.

Fortunately, however, both expert and novice tutors normally anticipate one or a set of particular answers/solutions when a question/problem is presented to the learner. This is an important feature of tutoring that makes it feasible to handle natural language understanding. Indeed, Cirsim-Tutor manages to simplify the NLU problem considerably by asking questions that invited only short answers (Freedman, 1997). However, if we ask only short-answer questions, we may lose some of the pedagogical advantages of NL-based tutoring. The challenge is to maximize the freedom that the students have to express themselves without making the NLU infeasible.

In summary, the major problem in practical NLU is robustness. There are two basic approaches. The symbolic approach requires more knowledge engineering, but extracts more meaning from the student's utterance. The other approaches, and LSA in particular, conduct only a shallow analysis, but require significantly less knowledge engineering. Needless to say, practical system should include both techniques, so that is exactly what Why2000 will do.

2.6 Natural language generation technology

Reiter (Reiter, 1994; Reiter & Dale, 1997) summarizes the belief of many researchers in natural language generation that text generation can be divided into the basic subtasks of (1) content determination, (2) paragraph and sentence planning, and (3) lexical insertion and surface generation. There is some flexibility at the edges of the subtasks. In an NL-based ITS, the content determination is carried out by the dialog planner and the rest of the ITS. The second subtask, paragraph and sentence planning, becomes turn planning because the NLG system decides how to phrase the tutor's turn in the dialog. Thus, we will review here only turn planning and surface generation.

2.6.1 Turn planning

Perhaps the most important function of turn planning is deciding what *not* to say. Leaving out less relevant material can make it significantly easier to understand what the tutor is trying to say. Consider the following example from GUIDON (Clancey, 1987, p. 276), which used a template-based approach to generating the tutor's turn:

- T: What are the organisms that might be causing the infection?
 S: Neisseria meningitidis.
 T: What facts about this case tell you that Neisseria meningitidis is one of the organisms that might be causing the infection? (Enter one factor per line.)
 S: Age.
 T: None of the factors that you mentioned are appropriate. The age of J. Smith is not evidence for the conclusion that Neisseria meningitidis is one of the organisms that might be causing the infection. After gaining more evidence about this case, the age of J. Smith will enable us to conclude that the organisms that might be causing the infection are E. coli (considering whether the infection was acquired while the patient was hospitalized), Pseudomonas aeruginosa, ...

Clancey's own commentary is clearer:

- T: The age of the patient will be relevant later, but it's premature to look at it now.

The turn planner should plan the sentence actually issued by Clancey, and not the one produced by GUIDON's templates.

A second major purpose of turn planning is to decide on issues such as what aspect of a proposition to put in the focal position, use of discourse markers words as "however" or "anyway", how much hedging to do, politeness, tense, anaphora, ellipsis and many other linguistic devices.

No general purpose paragraph or sentence planning systems are currently available, although several text generation systems have been developed that contain a special-purpose paragraph or sentence planner as a major component. One large-scale example is by Callaway & Lester (1995). Therefore, our turn planning module will be tuned to the genre of tutorial dialog and will be tailored to the domain knowledge being tutoring.

2.6.2 Surface generation

The surface generator takes as input a description of the propositions that the turn planner has decided are worth saying, along with information about their order, what parts of the proposition to put in focal positions, degree of hedging, tense, aspect, etc. The surface generator, outputs text to be printed, and it adds syntactic and semantic structures to the discourse history. The latter are necessary, because the student's next turn may refer to things that the tutor just said. For instance, pronouns in the student's turn often corefer with noun phrases in the preceding tutor turn.

Surface generation also takes care of the annoying "trivia" of text generation, such as getting plurals and tenses right. Template-based generation systems often become amazingly complex and fragile in trying to do this.

The two publicly available, highly developed surface generation systems are FUF/SURGE, (Elhadad and Robin, 1992, no date-a, no date-b) and KPML, developed by Bateman and his colleagues (Bateman, 1996; Bateman, 1997). While FUF expects a user intention as input, KPML input must be more closely matched to the linguistic categories of the desired output. Other surface generation systems are available that just do the conversion to English syntax and require that all other processing be done in an earlier subtask.

3 Research objectives

Stated abstractly, our research objectives are:

- To develop software tools for building NL-based ITS.
- To develop empirical methods for building NL-based ITS.
- To test the utility of our tools and methods by building an advanced NL-based ITS.
- To test the hypothesized advantages of the NL interface by comparing versions of the tutor with the NL interface to versions that use a conventional, constrained language interface instead.

These goals are our primary objectives. However, in building a tutoring system, we must choose specific task domains, student activities and pedagogies. The remainder of this section presents and justifies our choices.

3.1 The student activity: Generating explanations

As section 2.1 indicated, all intelligent tutoring systems have the students do some kind of activity, and they help the students do it. Most tutoring systems have the student solve a problem (e.g., an algebra word problem) or perform a cognitive skill (e.g., intercept a hostile aircraft with an F-14), and provide criticism and help either during or after the performance. Some tutoring systems ask the students deep questions, then help them formulate a complete, correct answer. For our purposes, we want to choose an activity where an NL interface may cause more learning than a constrained language interface. We would also like the activity to present a challenge but not an insurmountable challenge to the existing technology. These considerations have led us to pick explanation generation as the student activity. This section explains why.

A large body of research has shown that learning is increased by encouraging students to generate explanations themselves. When students are presented with examples in problem-solving domains, those who explain the examples thoroughly to themselves learned more (Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Ferguson-Hessler & de Jong, 1990; Pirolli & Bielaczyc, 1989). Students learned more when they were instructed to explain examples (Bielaczyc, Pirolli, & Brown, 1995; Lovett, 1992; Renkl, in press) or texts (Chi, 1996) to themselves. When students participated in small group discussions, students who gave explanations learned more than students who requested those explanations (O'Donnell et al., 1990; Webb, 1989). Classroom interventions that substitute student-generated explanation for teacher-generated explanation are usually more effective (Brown, Kane, & Echols, 1986; Palinscar & Brown, 1984; Pressley et al., 1992). When students are just prompted to explain a text, they learn just as much as students who read the same text with the aid of a human tutor (Chi et al., in press). In short, there is ample evidence that getting students to generate their own explanations significantly increases their learning.

However, students are often unable to generate explanations without help. If a student is unable to explain some point during training or offers an incorrect explanation, that flaw often persists and can be detected with post-training tests (Renkl, in press).

This suggests developing instruction where an instructor or a computer tutor helps the student generate explanations. Such an explanation coach would provide information when the student needs it, being careful not to provide too much. It would correct the flawed portions of students' explanation, but only when the students are unlikely to do such corrections themselves.

This would not be the first tutoring system to coach explanations. At least one other has been built: The Self-Explanation Coach (Conati, Larkin, & VanLehn, 1997; Conati & VanLehn, 1999a; Conati & VanLehn, 1999b), which is a part of the Andes system. The SE Coach helped students explain physics examples, each consisting of a quantitative physics problems and its solution. The students click on lines they wish to explain, then enter their explanation via menus. The coach provides immediate feedback and help as the students enter their explanations. The coach also suggests lines for the student to explain if explaining that line would involve using physics knowledge that the student has not yet mastered. In a recent evaluation, students using the SE Coach learned slightly more than students who explain twice as many examples without the aid of the coach.

Although further studies are clearly needed, it appears that coaching explanations is a useful pedagogical activity even when the explanations are not expressed in natural language. We hypothesize that coaching natural language explanations might be even more effective.

3.2 The task domain: Qualitative analysis of physical systems

Although our techniques will be general, we want to start with one task domain and then add more as the technology develops. The task domains should be chosen so that NL-interfaces are likely to aid learning, that the knowledge can be feasibly represented with current AI techniques, and the domain is difficult to learn given existing pedagogical methods.

Fortunately, there is a class of task domains with these properties. In virtually every discipline of science, mathematics and engineering, research has documented the existence of persistent misconceptions. In mechanics, for instance, many students believe that when an object is moving, there must be force propelling it along even if it is moving at constant speed (Viennot, 1979). This is exactly the opposite of Newton's first law. In electricity, to take another task domain, students believe that when a circuit is cut, it may take a moment for the electrical current to stop and some electrons may even spill out the ends of the cut wires (Chi, Slotta, & de Leeuw, 1994). In probability, students believe that when a fair coin has come up heads in many consecutive tosses, it is more likely to come up tails on the next toss (Tversky & Kahneman, 1974). The ubiquity of such misconceptions is startling: an early bibliography of research on misconceptions had several hundred entries (Pfundt & Duit, 1991), and more work has been done since then.

In many cases, existing instruction fails to remove the misconceptions. For instance, many misconceptions about force and motion persisted after a semester of college physics, even among students who received A grades (Halloun & Hestenes, 1985; McCloskey, Caramazza, & Green, 1980). A standard test of mechanics conceptions (the Force Concept Inventory, see Hestenes, Wells, & Swackhamer, 1992) has been developed and used in hundreds of courses, including many with innovative instruction, but rarely do students gain more than 70% (Hake, under review). Conventionally taught courses typically had much smaller gains, on the order of 25% (Hake, under review). Scientific misconceptions is clearly an area where existing pedagogical approaches are not completely effective, so new approaches may yield a significant improvement.

Scientific misconceptions are almost always elicited during *qualitative* rather than quantitative analyses of situations. A qualitative analysis involves explaining a situation without using equations, graphs or other mathematical tools. For instance, suppose students are presented with the situation of a big truck colliding head-on with a small car. Both vehicles were originally moving at the same speed. A qualitative analysis would simply ask which vehicle exerted the greater force on the other. A quantitative analysis might give the vehicle's masses, their initial speeds and the duration of the collision, then ask students to calculate the forces acting on each vehicle. Although the qualitative analysis involves applying only one law (Newton's third law—the forces are equal because they are an action-reaction pair), and the quantitative analysis requires 3 laws (Newton's second and third laws, and the definition of acceleration), students are more likely to get the quantitative problem right. Almost all the documented misconceptions occur during such qualitative analysis. Thus, if these misconceptions are to be adequately addressed, students must be engaged in qualitative analysis.

Much research has been devoted to why students have such faulty beliefs, and why they are so resistant to conventional instruction (e.g., Chi et al., 1994; Smith, diSessa, & Roschelle, 1993). Often these investigations suggest specific kinds of remediation. For instance, Chi's ontological account of robust misconceptions suggests that students' need prior instruction on a certain kind of abstraction before they can take advantage of instruction on specific misconceptions (Slotta, Chi, & Joram, 1995). Although much of this work is still controversial, they agree on one thing: whatever instruction we use to remedy misconceptions, it will involve doing qualitative analyses and not just quantitative ones. Many educators (e.g., Hunt & Minstrell, 1994; Mazur, 1993) have observed that conventionally taught science, mathematics and engineering classes give students ample practice on quantitative analysis, but very little practice on qualitative analysis. Cognitive task analyses also indicate that the inferences used to do qualitative analysis sometimes overlaps very little with the inferences used to do quantitative analysis, so just about any theory of learning predicts that training on one will not transfer to competence in the other (e.g., Ploetzner & VanLehn, 1997). Thus, it seems clear that the only way to improve students' understanding and remove misconceptions is to have them practice doing qualitative analysis and provide them with whatever extra instruction is suggested by research on misconceptions and by experienced instructors.

Unfortunately, there are practical impediments to increasing practice on qualitative analysis. Currently, students get little practice simply because instructors can not afford the time to give them feedback on their solutions. Even though the solution to a qualitative analysis problem seems simple (about a paragraph of text), the ideas are usually expressed quite poorly, so it may require substantial rereading in order to recognize their merit. Moreover, once the instructor understands the flaw in the student's argument, the instructor may have to write a paragraph-long response to indicate exactly what was wrong and what the correct reasoning should have been. That

is, marking a single qualitative problem is like grading an essay, so it may take as long as marking a whole quantitative problem set.

The proposed system will do exactly what instructors do not have the time to do. It will understand student qualitative analyses and attempt to remedy any misconceptions it finds. Moreover, rather than simply writing a paragraph-long response to the student, it will conduct a natural-language dialog with the student, which should be much easier for the student to understand.

It might seem that natural language is unnecessary, because students could express their qualitative analyses in a constrained language, which would make the tutor's job much easier (or the instructors, for that matter). Indeed, AI has invented notations and tools that substitute qualitative mathematics for the standard mathematics that underlies quantitative analysis (Weld & de Kleer, 1990). For instance, one widely used approach substitutes three values (positive, zero and negative) for the real numbers, and systems of "confluences" for systems of partial differential equations (de Kleer & Brown, 1984). However, these analytic systems require as their input the same idealizations that quantitative analyses requires. For instance, in the case of the truck-car collision, the solver must first recognize that the two forces are an action-reaction pair in order to apply Newton's third law. It doesn't matter whether the law is written as a confluence or an equation, because the misconception lies in the failure to recognize its applicability. Although AI tools for qualitative analysis may relieve the student of the distraction of manipulating complex systems of equations, they don't get at the fundamental problems of getting students to idealize physical situations correctly.

Another possibility is to have students practice qualitative analysis by getting feedback from other students rather than the instructor or a tutoring system. This is the essential idea behind "Interactive Engagement," a class of instructional activities "which yield immediate feedback to the students through discussion with peers and/or instructors." (Hake, under review, pg. 4). In one form, pioneered by Mazur (1993), Interactive Engagement consists of taking five minutes out of a lecture to present students with a qualitative analysis problem and have them discuss it with students seated nearby in the lecture hall. The instructor then collects each group's explanation via a show of hands or hand-held electronic devices. The instructor then presents the correct explanation and perhaps discusses why some of the other explanations given by students were wrong. Interactive engagement appears more effective than conventional instruction. A survey of 48 physics courses using Interactive Engagement methods indicated that 41 of them were significantly more effective at removing misconceptions than 14 conventionally taught courses (Hake, under review). However, none of the Interactive Engagement classes gained more than 69% on the Force Concept Inventory, so there is still much room for improvement. This makes sense, as instructors usually use only one or two qualitative analysis problems per lecture. Clearly, even more practice is necessary. We believe our tutoring system can provide that practice in a way that expedites learning.

We plan to start with one task domain and add others as the technology matures. The first task domain we plan to address is mechanics, the initial topic in most physics courses. From earlier work in this task domain (e.g., Ploetzner & VanLehn, 1997), we are confident that it poses no intractable knowledge representation problems. Moreover, there is a standard instrument for measuring success (the Force Concept Inventory) and a huge amount of baseline data, which makes it much easier to evaluate the proposed system. A large set of qualitative analysis problems and common erroneous solutions have been developed as part of the Diagnoser project (Hunt & Minstrell, 1994). Although the problems and misconceptions are expressed in English and must be converted to a formal representation, just having the content is a huge savings in time. Lastly, we have developed contacts with the physics education community via our work on Andes and Atlas (e.g., Bob Shelby at the US Naval Academy) that should enable us to develop our tutoring systems in ways that meet the needs of real instructors.

As a second task domain, we are currently considering both flight dynamics and electrical circuits, and would be interested in considering other task domains that are important to ONR.

3.3 An illustration: A good physics student interacting with the tutor

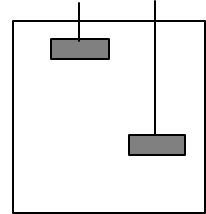
In order to illustrate what the proposed coach should be able to do, this section presents hypothetical scenarios of Why2000 tutoring a physics student.

In order to keep our first scenario short, let us assume that the student gives a good answer. Thus, the tutor has only minor corrections to make. The scenario begins with the tutor presenting a diagram of two identical bricks suspended in a tank of water. One is near the surface, the other is near the bottom. The tutor asks the student to explain which of these two bricks requires more force to hold it up. The student types in an explanation. Then the student and tutor discuss it. Figure 1 shows the screen at this point. The student's explanation is in the top box, and the tutorial dialog occurs in the lower box, with "T" marking the tutor's turn and "S" marking the student's turn. For expositional purposes, we have numbered each turn. The bottom box scrolls, but the top box does not; thus, as

the student and tutor discuss the student's explanation, it remains visible. Not shown in the figure is that after turn T3, the student edits the explanation. This satisfies the tutor, which says "Great! That's an excellent explanation. Let's move on."

Question: Two identical bricks are suspended in a tank of water. Which requires more force to hold it up, and why?

Explanation: The pressure is proportional to the depth. Thus, there is more pressure on the bottom of the left bricks than the top. So there is a net boyant force proportional to the thickness of the brick. Same thing for the right brick. So the boyant forces are the same, because the thickness is the same. Thus, the forces holding them up are thesame.



T1: Excellent! You've gotten all the main ideas. However, I have a minor comment. Is the difference in pressure really what physicist call a buoyant force?
 S1: I think so. Its upward.
 T2: Right, and physicists do use "buoyant" for the upward force exerted on objects that are floating on the surface of a fluid. But is this an individual force or a combination of individual forces?
 S2: Oh. Its the difference.
 T3: Right. It's the vector sum of two individual forces. It not a simple contact force or field force, so physicists would not dignify it with a name. Could you revise the language of your explanation?

Figure 1: A dialog between a good student and the tutor

From the student's point of view, the task is to enter a good qualitative analysis of the given situation into the top box. In this case, the student didn't ask for help while generating the explanation, so the tutor kept silent until the student was finished. Had the student asked for help, perhaps after typing in a partial explanation, the tutor would have offered some.

The tutor's first task was to interpret the student's explanation. Even though this is a remarkably good explanation, the tutor must still deal with:

- low level problems, such as misspellings ("boyant" and "thesame") and grammatical mistakes (e.g., agreement errors)
- non-grammatical sentences, such as "Same thing for the right brick."
- ellipsis, such as: "There is more pressure on the bottom of the brick than the top [of the brick]."
- anaphora: In "the forces holding them up are the same," "them" refers to the two bricks.
- reference: In "the pressure is proportional to the depth," the noun phrases do not refer to the given situation, but to a generic situation. In "The buoyant forces are the same because the thickness is the same," the noun phrase "the thickness" refers to the thicknesses of both blocks.

Despite these difficulties, the tutor must understand the student's explanation. In particular, it should recognize that the student applied the pressure law 4 times (top and bottom of both bricks) and Newton's first law twice (once for each brick). The vector summations were done properly. The student abbreviated by not mentioning irrelevant forces (e.g., the pressures on the sides of the bricks; the weights of the bricks). Underlying this apparently simple explanation is an impressive amount of reasoning. In order to completely understand it, the tutor uses deep, symbolic analyses (see section 2.5.2).

In the process of interpreting the explanation, the tutor notices that the adjective "buoyant" is not a semantically valid modifier for the concept of "difference." This discrepancy triggers an issue recognizer (see section 2.1), which posts a tutorial goal of correcting the student's use of terminology. Although this is a low-priority goal, no other tutorial goals are elicited by this nearly pristine explanation. Notice that it is highly unlikely that a menu-oriented system would have the opportunity to detect such misuse of language, as it would not offer "buoyant force" as a menu choice in this context.

At any rate, the tutor chooses to implement its goal by adopting the plan of simply asking the student whether the error was intentional (turn T1). This gives the student a chance to defend the error (turn S1), something that advanced students often want to do (Rose, 1997b). In general, such defenses are nearly impenetrable, and even human tutors tend to process them shallowly. In this case, suppose that symbolic techniques fail to recognize the

student's defense, so the tutor falls back on statistical techniques. It uses LSA to recognize that the student has part of the basic idea of a buoyant force right, so it splices in a correction, namely the full definition of buoyant forces. However, LSA also indicates that the student has said nothing about the key point, which is that the modified object is a vector sum of two forces rather than an individual force. Thus, the tutor asks about it point blank: "But is it an individual force or a combination of forces?" The natural language generator was instructed to mention the concept and use abstract language ("combination") instead of concrete language ("vector sum") so that the student would have to think in order to answer tutor's question. This is an appropriate general tactic for competent students, but not for weaker students, so the tutor consulted its student model before deciding to use this tactic.

On turn S2, the student finally gets the point, and the tutor's interpretation routines manage to recognize that despite the ellipsis. Thus, it considers the tutorial plan to have been successfully completed. As always when remediation has concluded, the student is asked to revise the original faulty explanation. This is necessary in order to make sure that the student actually understood the remediation. The revised explanation should and does pass inspection, and the tutor moves on to another explanation generation exercise. When all the exercises assigned by the instructor are done, the student's explanations are emailed to the course drop-box.

This particular exercise, with identical objects suspended at different depths in a fluid, is widely used in conceptually oriented physics instruction (Hunt & Minstrell, 1994; Mazur, 1993) because it elicits a remarkable number of misconceptions. The following are a few frequent explanations, taken verbatim from the Diagnoser web site:

- At a deeper level, objects will sink more easily, because there is more down ward pressure.
- At a deeper level, the water will push upward harder.
- If there is more water under an object than above it, there will be a greater pressure from below.
- If there is more water above an object than under it, there will be a greater pressure from above.

The tutor should be able to recognize common misconceptions whenever it has tutorial plans for dealing with them. LSA may suffice for this recognition task, although deep analysis might be necessary for discriminating among the others (compare the last two explanations listed above, which differ by only by switching "above" and "below"). In order to illustrate how the tutor detects and remedies such misconceptions, let us use a new scenario.

3.4 Another illustration: A poor electricity student interacting with the tutor

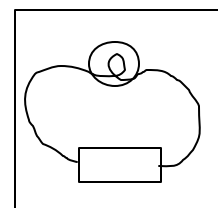
In this scenario, the tutor begins by presenting a simple battery and bulb circuit, and asking the student to explain why the bulb lights. This simple question is known to elicit a wide variety of misconceptions. Figure 2 shows the tutor handling a particularly simple misconception. This explanation was adapted from one produced by a novice student (Johsua & Dupin, 1987, pg. 122), although the rest of the dialog is fabricated.

This student has a common misconception, which is that there are two kinds of current, and that when they clash, energy is released. The tutor recognizes the misconception and starts a tutorial dialog plan which lasts until T4. Notice that the tutor asks only short-answer questions at T1, T2 and T3. At T4, the tutor prompts the student to integrate the tutor's explanation with the student's background knowledge. This is likely to generate a long answer, but the tutor can handle it because it is expecting two possible objections from the student, based on what has been observed with other students. One objection is given by this student: batteries die from electron depletion. The other common objection is that the batteries are labeled with + and - symbols because there are two kinds of current. The tutor has plans for dealing with both objections. In general, remedying a misconception requires much more than simply telling the student the correct version. All the students' reasons for believing the misconception must be rooted out and dealt with.

Although we have stressed the use of natural language, it may sometimes be pedagogically effective to accompany the language with graphical aids. In this case, some students might find the analogy easier to understand if the tutor presented a diagram or video clip of the pump and paddle wheel. Because we are already committed to presenting a diagram as part of the exercise statement, the linguistic problems of deictic reference need to be handled. Thus, adding an occasional graphical aid should not introduce much more complexity.

=====
Question: A battery is connected to a bulb by two wires. The bulb lights. Why?

Explanation: There are two currents which come out, one current positive and one negative. They combine in the bulb to produce light and heat.



T1:	I see. Which current is composed of moving electrons?
S1:	I'm not sure. Both?
T2:	Right, both are. An electrical current is just moving electrons. Which direction do the electrons move in your two currents?
S2:	It think the electrons move out from the battery to the bulb. The two currents meet there and produce energy.
T3:	Are the electrons actually destroyed in the bulb?
S3:	I'm not sure.
T4:	Actually, electrons are not destroyed or lost in an electrical current. Instead, they go out one side of the battery, go through the bulb and go back in the other side of the battery. Thus, your two currents are actually the same flow, only the electrons are leaving the battery for one and returning to the battery for the other. Given everything you know about batteries and bulbs, does this make sense?
S4:	But I thought the battery dies when it runs out of electrons.
T4:	Batteries die when they run out of the chemical energy that they use to pump electrons around the circuit. The battery is like a water pump that produces a powerful current, and the bulb is like a paddle wheel that is driven by the water current. When the water pump runs out of energy, the current slows to a stop. As this happens, does the number of electrons in the battery change?
S2:	Maybe a little. But it stays about the same.
T3:	Exactly right! I think you understand electrical current better now. See if you can write a better explanation.

Figure 2: A dialog between a novice student and the tutor

Chi and Slotta (Chi et al., 1994; Slotta et al., 1995) hypothesize that some misconceptions cannot be easily remedied even with extensive argumentation. These are misconceptions whose removal requires learning of a new ontological category, which they call constraint-base interactions. Fortunately, they also showed that students who are taught this ontological category in relatively familiar domains (e.g., diffusion of liquids) can transfer their knowledge to less familiar domains, including electricity. Since it would take us rather far afield to incorporate the Chi-Slotta instruction as a subgoal or prerequisite to our tutorial dialogs, we may simply choose explanation exercises that avoid these especially difficult misconceptions.

3.5 The research objectives, specified more precisely

Having illustrated the proposed tutoring system, several points should be clear. First, the initial system has such a simple user interface—text, a single read-only diagram, and two boxes for the user to type in—that a programmer can create it in a few hours, and a user can learn it in a few minutes. On the other hand, the proposed system has a great deal of intelligence. Not only can it understand multi-line input texts and generate coherent dialog, it also knows a great deal about its subject matter (physics, electricity) and pedagogical tricks for teaching it. With these illustrations as a backdrop, more precise research objectives can be listed below and elaborated in next section:

- To develop software tools for building NL-based ITS, and in particular for automating or at least semi-automating the process of developing domain specific:
 - lexicons
 - grammars
 - concept recognizers
 - correct explanations that deep interpretation will match to the student's explanations.
 - vectors that LSA will use to interpret student explanations and other utterances
 - tutorial plans
 - general knowledge for interpreting the above
 - curriculum scripts that organize a corpus of explanation generation exercises
- To develop empirical methods for building NL-based ITS, including:
 - methods for obtaining hundreds or even thousands of student explanations for tuning the lexicons, grammars, explanation bases, and other resources.
 - methods for obtaining effective tutorial plans
- To test the utility of our tools and methods by building an advanced NL-based ITS.
 - The tutor will coach students in qualitative analyses of physical situations.

- To test the hypothesized advantages of the NL interface by comparing the tutor to:
 - a version of the tutor that use a conventional, constrained language interface, and
 - to human tutors.

Clearly, this is an ambitious project that may seem beyond the state of the art of computational linguistics. However, the use of prefabricated explanation exercises allows us to collect a corpus of student explanations, which in turn allows us to formulate strong expectations for what students will type in the top box. During the tutorial dialogs of the bottom box, the tutor asks the student questions that tend to elicit short, easily understood answers. Thus, the design of the explanation activity maximizes the student's freedom to generate explanations, which is known to be effective pedagogically, while also allowing the system to have anticipated the student's utterances, which is important for making the computational linguistics feasible. Moreover, we intend to use a combination of shallow analyses based on LSA and deep analyses based on semantic compositions. This hybrid should allow us to achieve robustness and coverage as well as depth.

4 Technical approach

4.1 Corpus development and the client-server architecture

As described below, we plan to develop several tools that use machine learning and statistical techniques to automate the development of the lexicon, meaning representation specification, concept recognizers, LSA space, and other knowledge bases. All these tools require as input dialogs from real students using the tutor. In order to collect such a corpus, Why2000 will be used as both a data collection device and a tutor, and it will have a client-server architecture.

Students will use the interface shown in Figures 1 and 2, which will be implemented as a lightweight client (probably just a Java applet to be used inside a web-browser). Whenever they have finished their turn of the conversation, it will be sent to a server. Eventually, the server will be a heavyweight application located on the students' machines. During development, the server will be located on a different machine, so that a human can supervise it. The supervisor, who will usually be an expert human tutor, reads the students' contribution and the response that Why2000 has generated. If the supervisor is satisfied with Why2000's response, it is sent off to the student. If not, the supervisor enters a response, which is sent to the student instead. During the early stages of development, only the supervisor will generate responses, a set-up that is often called a Wizard of Oz study.

In order to collect the amount of data that we will need, it will be necessary to hire at least two full-time tutors, and to offer this tutoring service to a very large number of students. We are considering making the service available over the Web to any student who is interested and enrolled in an appropriate course.

As a side-effect of this procedure, our human tutors will become extremely experienced. We will encourage them read the educational literature in their domain and to try different remedial techniques with their tutees. This will allow them to give us expert advice on what tutorial techniques really work—a bonus that few tutoring projects have enjoyed.

In the later stages of development, the emphasis will be less on collecting a corpus and more on fixing defects in the knowledge bases that drive the tutoring system. Each time the human supervisor overrides the response that Why2000 generates, the development team needs to find out why and fix the problem. Every successful tutoring system has undergone this kind of tuning, albeit not perhaps so methodically. For instance, when the semantic grammars of Sophie were being developed, the system would email every student utterance that would not parse to Richard Burton, who would fix the grammars overnight in preparation for the next day's subjects. In the Andes tutoring system, verbal protocols are routinely recorded as part of the log files and replayed by a research assistant with a masters degree in physics. The RA finds episodes when Andes confused the subject, and logs them for attention by the development team.

Lastly, this set-up provides one simple evaluation of the system: the percentage of times when our super-expert tutors override Why2000. Of course, this evaluation will have to be supplemented with others, to be described later, as it does not measure the amount of learning.

4.2 Overall server architecture

The overall architecture of the server is quite simple. Outer loop consists of the following steps

1. The tutor presents a situation and asks the student for an explanation of it.
2. The student types in an explanation into the top box, which can be arbitrarily long.
3. The tutor tries to interpret the explanation in terms of the correct explanations that it knows.

4. If the explanation is correct, complete and concise, the tutor congratulates the student and they move on to another episode. If it is complete and correct, but poorly expressed, the tutor may congratulate the student and restate the explanation concisely. If the explanation is partially correct, incorrect or unrecognizable, the tutor devises a tutorial plan for eliciting the missing parts of the explanation, and starts executing it. The resulting tutorial dialog is conducted in the lower box.
5. If the plan succeeds in eliciting an acceptable explanation, perhaps over the course of many turns, then the tutor congratulates the student and asks him or her to type a revised version of the explanation into the top box. This version is handled just like the original version.
6. If the plan fails, then the tutor devises another plan to elicit a correct explanation and tries again.

There are several challenging steps in this process. One is to analyze the student's explanation in order to find out what, if anything, deserves remediation. This step can be done by the NLU module. The second challenging step is to conduct the tutorial dialog. The dialog manager is in control here, but it calls upon NLU to analyze the student's turns and NLG to generate the tutor's turns.

NLU is done by a combination of classification and semantic compositionality approaches. Compositional approaches are tried first, as they yield a deeper analysis when they work. If they fail, then LSA is tried. Compositional NLU is implemented by the traditional two modules, a parser and a semantic interpreter, which also handles discourse integration. As will be seen later, the three NLU modules actually work together to analyze the student's utterances. In summary, the main modules of Why2000 are:

- the tutorial dialog module
- the LSA module
- the parser
- the semantic interpreter
- the natural language generator

These modules will be discussed in separate sections in a moment.

One task that Why2000 will *not* do is to intelligently choose which explanation exercise the student should address. Instead, students will be allowed to pick which topic they want to study, and the tutor will simply march down a list of explanation exercises for that topic, except that students may skip exercises if they have already seen them or if they have not been assigned by their instructor. A more elaborate way of selecting exercises is to assess the student's mastery of various pieces of domain knowledge and choose an exercise that utilizes only mastered pieces of knowledge and a few mastered pieces of knowledge whose prerequisites have been mastered. Such mastery-based selection of exercise is fairly standard in the ITS literature. It results in students repeating exercises on a topic until it is mastered, a practice called mastery learning; (Bloom, 1984). Mastery learning is known to significantly increase learning, as measured by post-test scores. However, it means that some students must do much more work than others, and that often goes against the accepted practices. Sometimes, tutors with mastery learning capabilities have those capabilities turned off when they are used in real classrooms (Koedinger, Anderson, Hadley, & Mark, 1995).

By turning over the choice of exercise to the students and teachers, Why2000 is relieved of a major responsibility: it does not have to assess which pieces of knowledge the student has mastered. In fact, it has no need to do student modeling at all. Student modeling, which is the most computationally expensive and complex part of Andes and other model tracing tutors, involves two functions: assessment of knowledge and plan recognition. As just discussed, Why2000 has no use for knowledge assessments. It also has no use for plan recognition. When the student is generating an explanation, Why2000 doesn't even watch the process, so it has no need to recognize the student's plan. It only cares about the product—the explanation that the student produced. When the student and tutor are engaged in a tutorial dialog, Why2000 controls what plan the dialog is following, so plan recognition is again unnecessary. Thus, Why2000 has no need for a student model, which greatly simplifies its architecture.

Of course, a crude indication of the student's overall competence might be useful, as indicated in the first scenario discussed earlier. A single number would do, perhaps. This can be initialized fairly easily by asking the student a few "have you ever heard about X" questions at the beginning of the session, just as human tutors do. This initial indicator can be then be refined and maintained by simply seeing what flaws occur in the student's explanations and by the LSA-based techniques used by AutoTutor. Indeed, there is considerable evidence that human tutors maintain only such a crude indicator of competence, and not the elaborate ones used by model tracing tutors (Chi et al., in press; Graesser et al., 1995; Merrill, Reiser, Ranney, & Trafton, 1992; Putnam, 1987).

4.3 Dialog planning and management

In order to achieve a tutorial goal using multi-step teaching strategies, the tutor must plan and keep track of its plan as the situation changes. It is the tutor's responsibility to ensure that the student is able to work through the steps of the teaching strategy regardless of any wrong answers the student may give during the process. To remediate wrong answers, the tutor may need to counter a common student misconception with a standard response, begin a nested subdialog, or drop a subdialog and replace it by one more suited to the student.

Control structures used in earlier tutoring systems, such as the finite-state machines used in Cirsim-Tutor v. 2 (Zhou et al., 1999), AutoTutor (Graesser et al., 1999) or the Basic Electricity and Electronics tutor (Rose et al., 1999), are not appropriate for Why2000 because they do not keep track of either history or future goals. Traditional AI planners are not suitable either because they produce a single plan which is not expected to change during execution. It is wasteful to plan future conversational turns in detail when the conversation may take a different tack before those turns are reached. Furthermore, Anderson (1998) has shown that it is not feasible to create a partial-order plan in advance and update it as circumstances change. Finally, in order to maintain a helpful and coherent conversation, the tutor must occasionally make responding to the student a higher priority than continuing with the plan.

In the past few years, planners suitable for dynamic environments have been developed and have become known as *reactive planners* (Bratman, Israel & Pollack, 1988; Georgeff & Ingrand, 1989; Georgeff et al., 1998). In addition to interleaving planning and execution, reactive planners keep track of unsatisfied goals and can revise their goals after every student turn. Planners suitable for conversation planning tend to rely more heavily on hierarchical decomposition (Yang, 1990; Erol, Hendler & Nau, 1994) than on means-end reasoning. Hierarchical decomposition is appropriate for applications where the path taken by the planner (i.e. the generated conversation) is more important than the state the planner is in. Hierarchical decomposition is extremely rapid compared to other types of planning. The philosophy behind these planners has been elucidated by Bratman (1987, 1990).

Our approach will be to use the reactive planner that was developed for the Atlas project. APE (Atlas Planning Engine) is a hierarchical-decomposition style reactive planner designed specifically to make common tutorial planning operators easy to implement. APE is a fast, domain-independent system that obtains domain knowledge by communicating with a host system. APE implements all of the tutorial planning operations described above, including the ability to handle nested dialogs, drop one subdialog and replace it by another, and add and delete topics from the agenda. It can change its agenda after every student turn.

APE has an open API for specifying the preconditions of plan operators so that any data available to the host can be used in a precondition, including operators to access the host system's GUI or other databases. Preconditions to plan operators can be specified at varying levels of detail depending on the level of detail the author of the tutorial content wants to use to classify student inputs.

As a complement to APE, we have implemented a set of fundamental plan operators that can be used to implement mixed-initiative processing in any APE-based tutoring system. Thus APE can handle full multimodal, mixed-initiative processing. The degree to which these characteristics are available in a specific tutoring system that uses APE depends on the modalities available in the host tutor, the tutoring policies in force (some authors don't want students to be able to break out of a subdialog), and the plan operators written by the system authors.

We have used APE to implement a prototype Andes/Atlas system that adds a limited tutorial dialog capability to the Andes physics tutor. In particular, it uses short-answer questions or more open-ended questions whose answers we can classify into a small number of categories. Since Andes is a model-tracing tutor, the prototype generates dialogs to fix errors identified one at a time through model tracing. Since Why2000 is a question-asking tutor, it will produce longer dialogs that attempt to remediate several errors. This will use more of the planning aspects (as opposed to the execution aspects) of reactive planning.

4.4 LSA-based natural language understanding

The LSA versions of NLU will be an extension of AutoTutor's approach. It will be used for both understanding the student explanation (in the top box) and understanding the student's turns of the tutorial dialogs.

The basic idea is to decompose expected explanations and answers into sets of aspects. There are both correct aspects, which represent ideas that we would like student to mention, and incorrect aspects, which represent misconceptions that student might mention. In the LSA approach, all domain knowledge is represented as vectors in LSA space regardless of whether it is a word, an explanation or an aspect. When the student enters an explanation, each word is looked up in the LSA space, and its vector retrieved. The vector to represent the whole explanation is computed as the weighted average of the vectors representing the words. This vector is then compared to the aspects (this process is described in more detail below). If the explanation vector is close enough in LSA space to

the aspect vector, then the tutor infers that the student has expressed the basic idea behind that aspect. If the aspect is incorrect, it notifies the planner that there is misconception flaw that needs to be remediate. If the explanation does not match a correct aspect, then the planner is told that there is a missing-concept flaw that needs to be remedied.

Overall, the process is remarkably simple. However, we have glossed over few points that deserve fuller discussion, namely (1) constructing the LSA space, (2) calculating the match between two vectors, and (3) representing aspects.

An LSA space with K dimensions can be developed automatically given a sufficiently large corpus of texts. If we use the development of AutoTutor as a guide, we would need two textbooks per task domain and about 10 expert-written explanations per explanation exercise. However, a somewhat larger corpus may be needed if performance is not adequate. Analyses will be performed that assess the relationship between corpus size and performance.

There are three steps in computing the K dimensions for a corpus of texts with LSA. These steps are specified below.

(1) Preparation of a Word by Text rectangular matrix. LSA first prepares a large rectangular co-occurrence matrix that specifies the number of times that word W_i occurs in text T_j . A cell in the matrix is designated as $fr(W_i, T_j)$. The matrix is extracted from all of the words and texts in the entire corpus. It is possible to define a basic text unit (i.e., a document) as either a sentence or paragraph in the corpus. We will start out defining each paragraph and problem in the corpus as a text document, but we may need to assess performance when the more fine-grained sentence is adopted.

(2) Transformation of cell values. Each cell frequency is transformed in two ways. First, the frequency (plus 1.0) is converted to its logarithm: $\log [fr(W_i, T_j) + 1]$. Second, there is a computation that estimates the relative distinctiveness of the word to a particular text, relative to the alternative texts. For example, the information-theoretic measure of entropy is computed for each word ($-\sum p \log p$) over all entries in its row and then the cell entry is divided by the row entropy value. This value increases to the extent that a word appears in a particular text and not in alternative texts.

(3) Singular Value Decomposition (SVD). SVD decomposes the large rectangular Word X Text matrix into the product of three component matrices. We refer to the large matrix as $\{X\}$ and the three component matrices as $\{W\}$, $\{S\}$, and $\{P\}$. LSA determines a best-fit set of component matrices that approximately reproduces $\{X\}$. That is, $\{X\} = \{W\}\{S\}\{P\}$. The $\{W\}$ matrix maps the set of words onto the set of K dimensions (i.e., functional features, factors). If there are N words and K dimensions, this would be an N by K matrix with each cell having a weight for a word-dimension combination (capturing the extent to which a word possesses a functional feature). $\{S\}$ is a vector with K values that weights the generic importance of each of the K dimensions. $\{P\}$ is a K by T matrix that maps the K dimensions onto the set of T texts. Therefore, the Word by Text matrix is reduced to K dimensions that serve as functional factors/features in the domain knowledge. It should be noted that there is an optimal number of dimensions that fits data in tests of LSA. For example, in Landauer's tests on the corpus of encyclopedia articles and the TOEFL data, 300 dimensions provided better fits to the data than 100 dimensions and 500 dimensions. AutoTutor used a space of 300 dimensions in the domain of computer literacy.

After the LSA space is constructed, one can compute the conceptual relatedness between any two bag of words i.e., $\text{sim}(A,B)$. A bag contains one or more words, without any information about the order of the words in the text. In some LSA applications, the function words and other high frequency words (e.g., the, are, it) are removed because they are not distinctive words that discriminate texts. The results are frequently unaffected when these nondistinctive high frequency words are retained, but the role of high frequency words is still a matter of debate and research.

The two most common ways of computing relatedness are the cosine match and the dot product (Rehder et al., 1998). These two methods normally provide very similar results. Consider the relatedness of two words, X and Y. There is a K-dimensional vector for word X that is extracted from the $\{W\}$ matrix: $\underline{X} = (x_1, x_2, \dots, x_k)$. There is also a K-dimensional vector for word Y: $\underline{Y} = (y_1, y_2, \dots, y_k)$. The dot product ($\underline{X}\underline{Y}$) is the inner product of the two vectors: $\underline{X}\underline{Y} = x_1y_1 + x_2y_2 + \dots + x_ky_k$. The dot product is a scalar (not a vector) that reflects the extent to which the two words are conceptually related. The length of vector X, designated as $\underline{X}\underline{X}$, is the square root of the dot product of vector X with itself: $\underline{X}\underline{X} = x_1x_1 + x_2x_2 + \dots + x_kx_k$.^{1/2} Similarly, there is a length of vector Y, designated at $\underline{Y}\underline{Y}$. The cosine match between X and Y is computed as follows: $\cos(X,Y) = \underline{X}\underline{Y}/(\underline{X}\underline{X} * \underline{Y}\underline{Y})$. When the cosine match is used, the values can vary from -1 to 1 (but values vary in practice from 0 to 1). The computations are readily extendible to cases when two or more words are in each bag of words. A bag of 2 or more words is a weighted average of the vectors of the words it contains.

Aspects are represented internally as vectors in LSA space. However, it would be impossible to construct these vectors by hand. Instead, human authors write one short text per aspect, and LSA computes the aspect vector in the usual way, by taking the weighted average of the vectors of the words in the text. Compared to normal knowledge engineering practices, writing short texts is an extremely simple way to decompose an explanation into correct components and to specify misconceptions.

4.5 The parsing module

The parsing module will be an extension of the one being developed for Atlas. It is composed of several domain-independent components, one of which is a parser, and a domain-specific component. The domain-independent components constructs deep syntactic analyses of input sentences paired with shallow semantic representations, from which the domain-specific component generates a domain-specific propositional representation.

Separating the domain-specific components from the domain-independent ones allows us to invest once in broad coverage general linguistic knowledge sources that will give us a great deal of leverage towards building domain specific representations. We therefore minimize the effort required to build domain specific input understanders for new systems, because they can be built on top of our domain-independent components.

Functional Grammar (Halliday, 1985) and Lexical Functional Grammar (Bresnan, 1982) provide the linguistic framework. Functional syntactic analyses represent predicate-argument relationships, head-modifier relationships, and control relationships. These representations have been demonstrated to translate straightforwardly into quasi logical formulas (van Genabith & Crouch, 1996) and thus provide useful input for semantic interpretation (Halvorsen, 1987; Dalrymple et al., 1993; Dalrymple, 1999). In the spirit of (Dalrymple, 1999), semantic constructor functions are linked into the Atlas lexicon that build semantic representations from constituent representations according to their grammatical function. This organization allows the parser to build up semantic representations at parse time while the parsing grammar is kept free of any semantic information. Thus, the lexicon serves as an interface between deep syntactic analyses and shallow semantic representations built up in parallel with them at parse time.

Input understanding within the core system proceeds in four distinct stages:

1. The lexical preprocessing stage performs morphological analysis on each word in the input sentence and retrieves all lexical entries that match the root form of each word from the lexicon.
2. The parsing stage performs a syntactic and shallow semantic analysis with these lexical entries in so far as the parser's flexibility settings will allow.
3. When necessary, a recovery stage is responsible for assembling the pieces of a fragmentary parse when no complete parse is possible.
4. The domain-specific stage translates the domain independent meaning representations into domain specific ones.

Subsequent sections discuss each of these components individually.

4.5.1 The lexical preprocessing stage

The basic job of the lexical preprocessing stage is to convert a sequence of words into a sequence (actually, a “chart”) of lexical entries. The key to robustness at this stage is the coordination of morphological analysis, spelling correction and lexical lookup.

Lexical preprocessing is built around a large scale lexicon. Our lexicon is built on top of the broad coverage COMLEX lexicon available from the Linguistic Data Consortium (Grishman et al., 1994). Our lexicon contains two kinds of entries: regular lexical entries for individual words, and construct entries for idiomatic and otherwise non-compositional expressions. In general, we have taken a compositional approach to semantic interpretation, building up the meaning of an expression from the meanings of its constituent parts. However, idiomatic expressions are not correctly interpreted compositionally, so construct entries are needed to associate their meaning with the whole expression.

4.5.2 The LCFlex parser

The key to robustness in the parsing stage is the LCFlex robust parser (Rose and Lavie, 1999). It applies a broad coverage syntactic grammar by introducing various types of flexibility when needed. It makes use of a number of types of flexibility including:

- skipping over ungrammatical words and segments in the input,

- insertion of words or categories in specific contexts, and
- relaxation of grammatical constraints expressed via feature unification.

This section gives a very brief account of how LCFlex accomplishes this. LCFlex will be used with the Atlas grammar, which was adapted and significantly extended from a grammar developed at the University of Delaware (Schneider & McCoy, 1998).

Shallow semantic analyses are constructed in parallel with deep syntactic analyses at parse time. As constituents are assigned to syntactic roles with respect to their syntactic head, LCFlex's built in semantic interpretation framework inserts that constituent's semantic interpretation structure into the syntactic head's semantic interpretation structure by calling the semantic constructor function associated with the syntactic head in the lexicon. For example, the semantic field of the verb "come" contains a call to the semantic constructor function *move1*. The arguments in the function call specify (1) that a new semantic object of type *<move>* should be created, (2) that the semantic object representing the noun phrase that is filling the *subj* syntactic role should be placed in the *theme* slot of the new *<move>* object, and (3) that the semantic object of the noun phrase in the *obj* syntactic role should be placed in the *goal* slot of the new *<move>* object. Thus, semantic interpretation proceeds in parallel with syntactic interpretation. Note that the grammar does not contain any semantic information in it. Instead, the lexicon serves as an interface between syntax and semantics.

Because the lexicon references a large variety of semantic constructor functions, a tool has been built to make it easier to define them. The functions are compiled automatically from a meaning representation specification which defines semantic types and relationships between types. Two example entries are displayed in Figure 3. One semantic constructor function is built for each semantic type. The compilation code supports multiple inheritance via the *isa* field. The *spec* field specifies where instantiated variables are inserted into the resulting structure.

```
(:type <*event>
:isa (<>)
:vars (agent patient theme source goal instrument time)
:spec ((event +)(time <*time> time))

(:type <move>
:isa (<*event>)
:spec ((frame *move)(theme <*object> theme)(path <*path> goal)))
```

Figure 3: A fragment of the meaning representation specification

4.5.3 The repair component

When the flexibility allowed at parse time is not sufficient to construct an analysis of a sentence deviating too far from the grammar's coverage, a fragmentary analysis is passed into the repair module. The ROSE repair module was first introduced in (Rose, 1997a; Rose & Levin, 1998). It assembles the fragments returned by the parser into complete meaning representation structures.

We have recently reimplemented it within our semantic interpretation framework yielding an order of magnitude speedup. Each semantic object constructed during parsing contains a pointer to the constructor function that built it. Thus, the fragments returned by the parser contain implicit knowledge about how they can be combined with one another. The repair module uses this information to quickly assemble the fragments returned by the parser.

4.5.4 Domain-specific component

The domain-specific component translates the semantic structures generated by the parser into whatever semantic structures are used by host system, which in this case is a pedagogical dialog manager (described below). We assume that the host system semantic structures are composite structures that are built from primitive units of knowledge such as propositions, objects or functions. To accomplish the translation, each primitive unit of knowledge is given a *semantic concept detector*. The semantic concept detectors match against patterns of semantic and syntactic features inside the analyses produced by the core component and generate instances of the primitive units of knowledge. In Why2000, the concept detectors transform the domain independent analyses produced by the core component into domain specific propositional analyses that are usable by the planner within a specific domain.

4.5.5 Tool development

We plan to explore both hand coded and automatic methods for extending the coverage of the meaning representation specification. The purpose of the meaning representation is to collapse together sentences with similar meanings so that they can be mapped as a class onto domain specific categories. We are considering alternative previously proposed meaning representations such as Jackendoff's Lexical Conceptual Structures (Jackendoff, 1990; Jackendoff, 1983) and Halliday's transitivity classes (Halliday, 1985). As an alternative to these theory-driven but still hand-coded approaches, we plan to explore an automatic approach based on methods previously used for the word sense disambiguation task to identify domain appropriate synset classes from Wordnet (Miller, 1990).

Extending the domain-specific components means adding more concept detectors. Currently these concept detectors are coded by hand, but we would like to build a tool that will allow these concept detectors to be learned automatically from labeled examples. We plan to take a text classification approach similar to (Riloff, 1996). Where in Riloff's work classification is always based on features at a single level of representation (i.e., syntactic argument structure), we plan to learn which level or combinations of levels of representation is most appropriate for identifying each semantic concept (i.e., morphology, surface syntax, deep syntax, shallow semantics) using a decision tree learning approach (Utgoff et al., 1997; Martin, 1997; Helmbold and Schapire, 1997; Quinlan, 1990) or an explanation based learning approach (Minton et al., 1990; Dietterich & Flann, 1997).

4.6 Semantic interpretation

The module described in this section is responsible for interpreting the student's current contribution, which is either an explanation typed into the top box or a turn in the tutorial dialog of the lower box. We call this semantic interpretation, although discourse interpretation is an equally good name. The semantic interpreter's main input is propositions created by the semantic concept detectors of the grammar-based sentence understander. It must combine the propositions from multiple utterances to form larger units of meaning. Moreover, it must do so relative to the context in which the language was used; the context in this case is the conceptual knowledge that is the tutoring target and what was previously said in the dialog. By interpreting the students' turn relative to the context, we indirectly form larger units of meaning from multiple turns in the dialog.

The semantic interpreter must also handle other discourse level problems such as coreference; set, class and part-of relations (Prince, 1981; Karttunen, 1976; Grosz, 1977; Heim, 1982; Hawkins, 1978; Passonneau, 1994) and ellipsis. The main components of the semantic interpreter are an extensive knowledge base, a discourse history and an inference engine. We discuss the inference engine first.

4.6.1 Discourse interpretation based on abductive inference

The semantic interpreter will be based on abductive interpretation (Hobbs, Stickel, Appelt, & Martin, 1993). Abductive interpretation takes utterances as bits of evidence and builds up explanations or meanings from this evidence by backward chaining through axioms or rules. To piece the utterances together into an explanation, abductive inference often has to make assumptions. For instance, to interpret the turn S1 in Figure 1 "It's upward," we have to assume that the pronoun refers to a specific object that was mentioned earlier in the discourse. We make assumptions at all levels in language (e.g., what is the referent of this pronoun, what is the missing argument in this sentence, how do these two utterances relate to one another).

Often utterances have multiple logically possible interpretations, but hearers prefer only one. In such cases, there are multiple assumptions that abductive inference can make. We will assign different costs to different assumptions, and prefer the interpretations or proofs that are the least costly.

To make the explanation building more practical, it helps to start with hypotheses about what we are trying to prove. When understanding an explanation, these are the pre-fabricated explanations that knowledge engineers familiar with the task domain have encoded. Each explanation exercise may have several correct and incorrect explanations authored for it. When the understander is processing the student's turn during the tutorial dialog, it has the expected answers, which are part of the current tutorial dialog plan. These expectations about the interpretations of explanations and answers can significantly speed up the abductive inference process.

As a further increase in speed, LSA can be used to prioritize the expected explanations. That is, if the words in the student's utterance are more similar to the semantic primitives in explanation A than to those in explanation B, then the abductive inference engine should start by trying to prove A. Thus, when it starts to prove B and the costs exceed those of A, it can stop early. Indeed, the quantitative nature of LSA's similarity judgements might be useful at multiple points during the proof in order to prune unpromising subproofs.

The Tacitus message understanding system which incorporates abductive interpretation and which was evaluated during some of the original MUC trials, was notoriously slow compared to more tailored approaches. However, it did sentence interpretation using abduction in addition to the discourse interpretation that we propose to use it for. For the COCONUT project (COCONUT Project, 1999), we worked with a rudimentary version of Tacitus, which we called Tacitus-Lite+, and we used it for both discourse interpretation and discourse planning. We found its performance adequate. For Why2000, we will only use it for discourse interpretation and not dialog management, and thus expect its speed to be quite acceptable.

To further improve the performance of Tacitus-Lite+, we experimented with heuristics for pruning the search space, thresholds at which to end the search and small knowledge representation enhancements for axiom writing (e.g. we reduced the number of axioms and the search space by allowing classes for arguments in addition to instances). For the Why2000 project, we will further improve the performance of Tacitus-Lite+ by incorporating heuristics that are tailored to the tutoring domain. Moreover, we will integrate Tacitus-Lite with a terminological knowledge representation system, which is discussed in the next section, should also improve speed as well as the robustness and extensibility of the axioms.

4.6.2 Knowledge Representation

We propose to use a combination of knowledge representation tools for the Why2000 project. We will use a terminological knowledge representation system such as CLASSIC (Brachman et al., 1991) to represent objects and axioms in Tacitus-Lite+ to represent the propositions and actions. There are many advantages to using different kinds of knowledge representations for objects and propositions/actions. First, a system like CLASSIC is good for representing definitions of concepts. It provides inheritance, classification, subsumption and simple forward chaining. Second, terminological KR systems can also be used to build knowledge engineering tools that manage system software and knowledge resources (Yen, 1991; Devanbu & Litman, 1991). Terminological languages have also been used in NLP applications for lexical representation (Burkert, 1995), and grammar representation (Brachman & Schmolze, 1991), and to assist in the acquisition and maintenance of domain specific lexical semantics knowledge (Ayuso et al., 1987; Jordan, 1996). Also, a system such as CLASSIC provides feedback about inconsistencies in definitions and allows querying so that one can find out if a concept has already been defined (Brachman et al., 1991).

However, CLASSIC is not good at representing and providing the reasoning tools needed for complex relationship between events such as transitive relationship where events are at different levels of detail and necessary parts of the definition of the event are missing (as with the “kick the ball” example above where “kick” is at one level of representational detail and “win” is at another e.g. kick is a step in scoring points, and winning a game is at a higher level of abstraction). At the same time the axiom representations in Tacitus-Lite+ would benefit from a concept definition language so that we can write one axiom for a class of arguments. In this way, we do not have to add reasoning about definitions directly to Tacitus-Lite+.

As we mentioned earlier, knowledge representation is a difficult problem and we can make our problem approachable by limiting what we do to precisely what we need for explanation coaching. First, we will constrain the details of our representation by building it from the bottom up instead of the top down. That is, will we do a corpus analysis of the dialogs we collect via our web-interface to find the words and concepts that we need to represent. For example, if dialog participants only use the words “east”, “west”, “north” and “south” for direction and never use “left”, “right”, “up” or “down” then we would not build as abstract of a representation. For instance we might not need to represent a direction as having an x-component and y-component that are perpendicular to one another where the axes of these components have positive or negative values that are opposites of one another. Likewise the relations that we build between concepts will also be driven by the corpus. If we don’t have to talk about opposite directions then we don’t need this relation and the associated inferencing that would be needed to realize that “left” and “right” are opposites and that the same is true of “north” and “south”. Similarly for the Tacitus-Lite+ axiom writing we will use the expected good and bad explanations for guidance.

4.6.3 The discourse history and reference resolution

The discourse history is a knowledge source that is used by the semantic interpreter mainly to help resolve references of various kinds (discourse entity relationship, ellipsis, anaphora, definite noun phrases, etc.). It is an organized record of everything that the student might refer to that has been said recently by either the tutor or the student. If the tutor chooses to display a diagram or video, the discourse history will also include a high-level record of what was displayed (e.g., Mittal et al., 1995).

When searching the discourse history in order to resolve a referring expression, most discourse interpreters use some kind of salience analysis in order to prioritize the candidates. By limiting processing to the most salient part of the discourse history, we can improve the efficiency and results of the discourse level abductive interpretation. The tutorial dialog manager also needs salience analysis to support discourse planning and content selection, so that we know which pieces of knowledge need to be repeated in order to make the interaction with the student more productive.

As a basis for the salience analysis, we propose to use a combination of the attentional account of discourse structure (Grosz & Sidner, 1986), recency and frequency of usage of propositions (Walker, 1993) and conversational threads (Poesio, 1993). The salience should be organized by the task structure (Grosz & Sidner, 1986) but since we are dealing with what we hope will be dialogs that are more balanced with respect to initiative and information sharing (e.g. Grosz & Sidner, 1986 analyzed text and advisory dialogs), it is more practical to organize by task actions than by trying to recognize higher-level discourse segment purposes (this is similar to Poesio's conversational threads). This structure allow for the more tangled character of dialog. As an action is revisited, the whole thread becomes accessible. We would also want a recency/frequency window on the threads and the discourse history as well. What is salient is the union of the recency/frequency window of the active conversational thread and what is in the recency/frequency window for the dialog history. (Jordan & Walker, 1996) found that using a snapshot of the speaker's attentional state to determine what is salient for the hearer and considering the tradeoffs between communicative effort and reasoning effort when deciding what to repeat, improved the overall task performance compared to always repeating or never repeating mutually known knowledge.

4.7 Natural language generation

As mentioned earlier, natural language generation is often divided into content planning, turn planning and surface generation. In Why2000, the content planning is carried out by the tutorial dialog manager. It maintains an overall plan for the dialog. The turn planner is a presentation planner responsible for organizing a single tutorial turn. It must figure out a simple, easily comprehended way to present the content selected by the tutorial dialog manager. The surface generator takes care of phrasing individual sentences. It chooses the words, decides when to use anaphora and gets the details of agreement and punctuation right. This section describes our approach to turn planning, surface generation, and the talking head.

4.7.1 Turn planning

The main job of turn planning is to convert the intentions chosen by the dialog planner into an effective rhetorical pattern for expressing an idea. This may involve deciding what *not* to say. For instance, the semantic structure for a force might have lots of descriptive slots—its type, the object it acts on, the object causing it, its direction and its magnitude. What is the minimal combination of features that the tutor can mention and yet still expect the student to resolve the reference correctly?

Turn planning is still a black art, in that little is known in general about how to do it. Current work [e.g., Cawsey; Green] has produced task specific solutions. However, it is safe to say that we will have to invent our own heuristics by trial and error.

Although a variety of frameworks have been used for turn planning—from sophisticated constraint satisfaction algorithms to table look-up—it is not clear which is best for our project because it is not clear yet what the relevant turn planning heuristics are. Thus, we intend to use APE as the framework for our first forays. As we discover more about the relevant heuristics, we may design a more tailored framework.

4.7.2 Surface generation

As mentioned earlier, two publicly available, highly developed surface generation systems are available. We intend to use FUF/SURGE, (Elhadad and Robin, 1992, no date-a, no date-b), because its input is closer to the representations that APE uses.

4.7.3 The talking head

In recent years we have witnessed the emergence of a number of animated pedagogical agents that are capable of communicating with the learner in face-to-face interactions (Johnson, Rickel, & Lester, in press; Paiva & Machado, 1998). These efforts are a natural extension of the more general mission to create embodied animated agents that exhibit the features of human communication (Cohen & Massaro, 1994; Cassell & Thorisson, 1999). The primary challenge is to appropriately coordinate the agent's facial expressions, speech, gestures, and body

movements in real time. Pedagogical agents have the added burden of facilitating the learning process. We plan to experiment with animated agents in our proposed system.

Animated agents have been used before in the context of tutoring. AutoTutor has a talking head (a Microsoft Agent) that incorporates some important properties of a pedagogical agent (McCauley et al., 1998; Person, Klettke, Link, Kreuz, & TRG, 1999). The facial expressions and intonation in the immediate short feedback (positive, negative, versus neutral) are sensitive to the quality of the assertions in the student's most recent turn. The manner of delivering hints, prompts, corrections, and questions have distinct intonation contours that signal the function of these speech act categories. The parameters of the facial expressions and intonation are generated by fuzzy production rules. A talking head is an presumably an important enhancement to AutoTutor because it concretely grounds the conversation between the tutor and student.

A talking head also provides separate channels of cues for providing mixed feedback to the learner. When a student's contribution is incorrect or vague, for example, the speech is often positive and polite whereas the face has a puzzled expression; this conflicting message satisfies both pedagogical and politeness constraints so it is preferable to a threatening speech message that says "That's wrong" or "I'm having trouble understanding you." The nonverbal facial cues are known to be an important form of backchannel feedback during tutoring (Fox, 1993; Graesser et al., 1995; Person et al., 1994), as well as other contexts of conversation (Clark, 1996). Similarly, pitch, pause, duration, amplitude, and intonation contours are among the variety of intonation cues that signal backchannel feedback, affect, and emphasis (Brennan & Williams, 1995). Some of these intonation parameters have been implemented successfully in synthesized and digitized speech (Cowley & Jones, 1992). These intonation parameters are important to tutoring because they qualify the backchannel feedback ("Uh huh", "Okay") and substantive contributions of the tutor (Fox, 1993; Graesser et al., 1995). For example, the tutor frequently pauses after a vague student contribution, or pounces in quickly after an obvious error-ridden student contribution.

One drawback of having a face in Why2000 is that it potentially distracts the student's attention from the substantive content on the compute display, such as a physics problem. Thus, the student would be frequently moving the eyes back and forth between the talking head and the elements of the physics problem. If this is the case, it may be more prudent to have Why2000's dialog moves be delivered by just synthesized speech. Moreover, recent speech recognition systems (such as Dragon) have demonstrated improving performance, particularly when there is a limited number of frozen speaker expressions (e.g., "I don't understand", "What did you say?", "Please repeat", "Go on."). Thus, the students could have a spoken dialog with AutoTutor as they focus their eyes on the physics problem and their hands on the keyboard. We will experiment with these various forms of communication media during years 4 and 5 of the MURI grant.

5 Evaluation

We plan to conduct 3 kinds of evaluation: formative evaluations, summative evaluations and componential evaluations. Each are discussed below.

The formative evaluations are intended to guide our development of the system by detecting bugs, design flaws, pedagogical mistakes and so on. As described earlier, we will employ at least 2 human tutors who initially will act like Why2000 and provide us with a corpus of data. However, when Why2000 is ready, it will be used in a supervised mode. Before the system's turn is sent to the student, it is presented to the human tutor for approval. If the human tutor thinks that Why2000's text is pedagogically and linguistically acceptable, the text is sent unchanged. Otherwise, the tutor can edit it or send an entirely new response. The whole session is logged, and events where the tutor overrides the system are diagnosed in order to find out how to improve Why2000. The percentage of turns that are overridden serves as ongoing evaluation of the improvement in the system's performance.

The summative evaluations are designed to benchmark the overall performance of the system. Each evaluation will consist of a 3-condition experiment. Why2000 will be compared to a version of it that has a constrained language interface, and to human tutors. Data will be collected from college students. The 3 conditions will be compared for learning gains and differences in language:

Learning gains. Items from the Force Concept Inventory, a standard test of conceptual physics, as well as experimenter-generated items will be administered both before and after the students complete the tutoring sessions.

Differences in dialog. Assuming there is a difference in learning outcomes, then it will be appropriate to find out why. To this end, we will determine if the language actually used during the experiment was different in different conditions. We will code the dialogs 4 different ways:

- The dialog moves and plans of the tutors, such as immediate feedback, hint, assertion, analogical plan, Socratic plan, and so on.

- The domain concepts mentioned in the dialog (as in VanLehn et al., in press)
- The linguistic features such as the syntactic complexity, selection of content words, coherence, use of discourse markers, speech disfluency, and so on.
- The quality of the dialogs as rated by experts in discourse or pedagogy (cf. the evaluations of AutoTutor reported in Weimer-Hastings et al, in press).

The frequency distributions of these categories will be used to determine if the 3 types of tutoring differ in meaningful ways. We will also determine which categories correlate with learning outcomes.

A second method of summative evaluation is to conduct a modified Turing test. Experts will be presented the transcripts from Why2000 and from a human tutor and will decide which one was generated by the computer. An alternative to this 2-alternative forced choice format is to ask them to rate single alternatives on a 6-point scale, using a standard signal detection paradigm that collects hit rates and false alarm rates. In both cases, we can compute d' discrimination scores as a function of the different dialog move categories. The d' scores should approach zero to the extent that good dialog moves are being generated. In essence, expert judges cannot distinguish the turns of a computer versus a human tutor.

In addition to determining the overall performance of the tutor, we would like to evaluate the performance of the individual component modules. Such an evaluation could be called a componential evaluation. The basic idea is to break out the performance of a module and compare it to humans doing the same task. For instance, an analysis of anaphoric coreference would collect a sample of noun phrases from the tutorial dialogs that refer to a previous constituent in the dialog history. Expert humans would make a decision on what the appropriate referent of each noun-phrase and this would be compared with the decision of Why2000. Such componential evaluations can be done as the modules are developed, so this technique is useful for both formative as well as summative evaluations.

In some cases, we have two or more modules that use different techniques to achieve the same ends. For instance, we have both LSA and symbolic approaches to NLU, and we have text, speech and the talking head for output. We will apply componential evaluation techniques to compare them. In short, our intention is not just to build a tutoring system, but to understand the technology of NL-based tutoring—which parts work, when and why.

6 Milestones

The first year will be devoted to system integration and corpus analysis. As the discussion above indicated, many of the pieces of Why2000 exist already in the AutoTutor, Atlas and Coconut projects. However, this code was written in different languages with wildly different assumptions about how it will be used. Moreover, there are some components (e.g., turn planning) that are still in the early design stages. In addition to forging an integrated framework, we will collect a large corpus of tutorial dialog. This will be used in limited ways during the first year (e.g., building the LSA space; augmenting the COMLEX lexicon).

The second year will be devoted to tool development and scaling up the first task domain. For instance, the machine learning tool for creating semantic concept detectors will be developed and applied in order to populate the domain-specific part of the sentence understander. Tools for facilitating the entry of tutorial dialog plans will be developed and a large knowledge base of plans will be engineered, using our super-tutors as expert informants. Midway through the second year, we will begin running pilot subjects and giving demos. Corpus collection on the second task will also begin.

The third year will be devoted to tuning the system and evaluating it on the first task domain. We will begin implementing knowledge bases for the second task domain.

If the 2 year extension is granted, we will complete development of the second task domain and evaluate it. If low-level speech recognition technology has improved sufficiently, then we will investigate a speech-only system.

7 Personnel and organization

Kurt VanLehn will lead the University of Pittsburgh group, and Art Graesser will lead the University of Memphis group. The Pitt group will be organized into 3 subgroups, each led by a post-doctoral research associate: Parsing (Carolyn Rose), semantic interpretation (Pam Jordan) and tutorial dialog management and NLG (Reva Freedman). Each subgroup will include a programmer and as many graduate students as are appropriate. The semantic interpretation group will be assisted by a consultant, Prof. Richmond Thomason, director of the Coconut project and an expert in knowledge representation and abductive approaches to discourse. The corpus collection will be done by the two tutors, and data analysis will be done by a linguist.

The Memphis group will be organized into 4 subgroups, each led by one or more faculty members: LSA (Xiangen Hu and Peter Wiemer-Hastings), tutorial dialog and natural language processing (Natalie Person and Peter Wiemer-Hastings), empirical evaluation (Natalie Person), and speech recognition and talking heads (Xiangen Hu). Graduate students in computer science and psychology will be assigned to each of the four groups. There also will be a full-time programmer in Memphis who will integrate software modules and coordinate efforts with the Pittsburgh group.

In addition to the professors, all the post-doctoral research associates (except Pam Jordan) have advised graduate students. We intend to train as many as we can, using both the funding in our budgets (6/yr.), MURI fellowships (3/yr.) and AASSERT grants. We believe that the advanced character of this research will attract and retain qualified students.

8 Scientific impact

The proposed MURI project will advance scientific theories and empirical research in several fields: Education. The field of education has accepted both the constructionist thesis and the social collaboration thesis. The first emphasizes the importance of learners actively constructing their knowledge, as opposed to being passive recipients of information delivery systems. The second emphasizes the importance of social interaction, conversation, and collaboration. Whereas these two theses are widely accepted in the education community, the central challenge has been to precisely specify *what* construction strategies and *what* discourse patterns are responsible for learning gains at both deep and shallow levels. Indeed, these remain a mystery to developers of intelligent tutoring systems and other learning technologies. Our attempt to integrate a tutorial dialog facility with an ITS will advance this research at the fine-grained level. Our exploration and comparison of human and computer tutors will help us design ITS's more like expert human tutors, and will help us identify the advantages and disadvantages of each.

Cognitive science. This project will explore and test a number of computational models, theories of representation, and architectures of human cognition. Some of these systems are statistical (such as Bayesian models, fuzzy systems, and latent semantic analysis), some are symbolic (planners, production systems, parsers, behavior networks, conceptual graph structures), and others are hybrids. Also, we will investigate the role of explanations and collaborative interaction in promoting deep comprehension and learning.

Artificial intelligence. We will investigate some of the most challenging problems in artificial intelligence, such as NLU, NLG, the representation of world knowledge, constraint satisfaction, reactive planning, and the development of ITS's. We will investigate the limitations and strengths of symbolic computation models, probabilistic statistical models, and hybrids of these two approaches. Rigorous performance evaluation will test the alternative intelligent systems.

Discourse processing. Tutoring is an intrinsically interesting discourse context for several reasons. There is often minimal common ground between learner and tutor so researchers can explore discourse patterns as a function of varying degrees of shared knowledge. The pragmatic ground-rules are sufficiently constrained to track the pragmatic goals, plans, assumptions, and discourse patterns of the speech participants. Why2000 provides a testbed for exploring a wide range of other problems that are topical in discourse processing, such as speech acts, given-new contrasts, discourse focus, discourse markers and connectives, coherence, turn-taking, discourse disfluency, repair, backchannel feedback, intonation, and metacommunication. This project will be one of very few attempts to simulate smooth conversation, as opposed to merely analyzing conversation.

Computational linguistics in general. This is the first project that will systematically integrate world knowledge (as captured by latent semantic analysis) with other language modules that have traditionally been investigated in computational linguistics. It will also be the first that rigorously evaluates the performance of language and dialog modules in an ITS system.

9 References

- Abney, S. (1996). Partial parsing via finite-state cascades. In Proceedings of the Eighth European Summer School in Logic, Language and Information, Prague, Czech Republic.
- Ait-Mokhtar, S. & Chanod, J. (1997). Incremental finite-state parsing. In Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, D.C.
- Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4(2), 167–207.

- Anderson, S. D. (1998). Issues in interleaved planning and execution. Workshop on Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments, held in conjunction with the Fourth International Conference on Artificial Intelligence Planning Systems (AIPS '98), Pittsburgh. AAAI Technical Report WS-98-02.
- Appelt, D. & Pollack, M. (1990). Weighted abduction for plan ascription. Menlo Park, CA: SRI International. Technical Note 491.
- Ayuso, D., Shaked, V., & Weischedel, R. (1987). An environment for acquiring semantic information. In Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL '87), Stanford (pp. 32-40).
- Bateman, J. A. (1996). KPML development environment: Multilingual linguistic resource development and sentence generation. Sankt Augustin: German National Research Center for Information Technology (GMD). GMD-Studie 304.
- Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering* 3(1), 15-55.
- Bielaczyc, K., Pirolli, P., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving. *Cognition and Instruction*, 13(2), 221-252.
- Bilange, E. (1991). A task independent oral dialogue model. In Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin (pp. 83-88).
- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4-16.
- Bod, R. (1998). Spoken dialogue interpretation with the DOP model. In Proceedings of the 17th COLING/36th ACL (COLING-ACL '98), Montreal.
- Brachman, R., McGuinness, D., Patel-Schneider, P., & Resnik, L. (1991). Living with CLASSIC: When and how to use a KL-ONE-like language. In J. Sowa (Ed.), *Principles of Semantic Networks* (pp. 401-456). San Mateo, CA: Morgan Kaufmann.
- Brachman, R. & Schmolze, J. (1991). Overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9, 171-216.
- Bransford, J. D., Goldman, S. R., & Vye, N. J. (1991). Making a difference in people's ability to think: Reflections on a decade of work and some hopes for the future. In R. J. Sternberg & L. Okagaki (Eds.), *Influences on Children* (pp. 147-180). Hillsdale, NJ: Erlbaum.
- Bratman, M. E. (1987). *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.
- Bratman, M. E. (1990). What is intention? In P. R. Cohen, J. Morgan & M. E. Pollack, *Intentions in Communication*. Cambridge, MA: MIT Press.
- Bratman, M. E., Israel, D. J., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4), 349-355.
- Brennan, S. E. & Williams, M. (1995). The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language*, 34, 383-398.
- Bresnan, J. (1982). *Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.
- Brown, A. L., Kane, M. J., & Echols, C. H. (1986). Young children's mental models determine analogical transfer across problems with a common goal structure. *Cognitive Development*, 1, 103-121.
- Brown, J. S., Burton, R. R., & de Kleer, J. (1982). Pedagogical, natural language and knowledge engineering techniques in Sophie I, II and III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems*. New York: Academic Press.
- Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in Context Space: Words, Sentences, Discourse. *Discourse Processes*, 25, 211-257.
- Buo, F. D. (1996). Feaspar—A feature structure parser learning to parse spoken language. In Proceedings of the 15th International Conference on Computational Linguistics (COLING '94), Kyoto.
- Burkert, G. (1995). Lexical semantics and terminological knowledge representation. In P. Saint-Dizier & E. Viegas (Eds.), *Computational Lexical Semantics*. Cambridge: Cambridge University Press.
- Callaway, C. & Lester, J. (1995). Robust Natural Language Generation from Large-Scale Knowledge Bases. In Proceedings of the Fourth Bar-Ilan Symposium on Foundations of Artificial Intelligence, Jerusalem (pp. 96-105).
- Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4), 190-202.
- Cassell, J., & Thorisson, K. R. (1999). The power of a nod and a glance: Envelope versus emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, 13, 519-538.

- Cawsey, A. (1992). *Explanation and Interaction: The Computer Generation of Explanatory Dialogues*. Cambridge, MA: MIT Press.
- Charniak, E. (1986). A neat theory of marker passing. In Proceedings of the Fifth National Conference on Artificial Intelligence, Philadelphia (AAAI '86) (pp. 584–588).
- Charniak, E. (1993). *Statistical Language Analysis*. Cambridge: Cambridge University Press.
- Chi, M. T. H. (1996). Constructing self-explanations and scaffolded explanations in tutoring. *Applied Cognitive Psychology, 10*, S33–S49.
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145–182.
- Chi, M. T. H., de Leeuw, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439–477.
- Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. G. (in press). Learning from tutoring: A student-centered versus a tutor-centered approach. *Cognitive Science*.
- Chi, M. T. H., Slotta, J. D., & de Leeuw, N. (1994). From things to processes: A theory of conceptual change for learning science concepts. *Learning and Instruction, 4*, 27–43.
- Chu-Carroll, J. & Carberry, S. (1995). Response generation in collaborative negotiation. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge, MA (pp. 136–143).
- Clancey, W. J. (1987). *Knowledge-Based Tutoring: The GUIDON Program*. Cambridge, MA: MIT Press.
- Clark, H. H. (1996). *Using Language*. Cambridge: Cambridge University Press.
- COCONUT Project. (1999). Available from <http://www.isp.pitt.edu/~intgen/coconut.html>.
- Cohen, M. M. & Massaro, D. W. (1994). Development and experimentation with synthetic visible speech. *Behavior Research Methods, Instruments, and Computers, 26*, 260–265.
- Cohen, P. A., Kulik, J. A., & Kulik, C.-L. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19*, 237–248.
- Collins, A. (1985). Teaching reasoning skills. In S. F. Chipman, J. W. Segal, & R. Glaser (Eds.), *Thinking and Learning Skills* (vol. 2, pp. 579–586). Hillsdale, NJ: Erlbaum.
- Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser* (pp. 453–494). Hillsdale, NJ: Erlbaum.
- Collins, A. & Stevens, A. (1982). Goals and methods for inquiry teachers. In R. Glaser (Ed.), *Advances in Instructional Psychology*, v. 2. Hillsdale, NJ: Erlbaum.
- Conati, C., Gertner, A., VanLehn, K., & Druzdzel, M. (1997). On-line student modeling for coached problem solving using Bayesian networks. In A. Jameson, C. Paris, & C. Tasso (Eds.), *User Modeling: Proceedings of the Sixth International Conference (UM '97)*. New York: Springer.
- Conati, C., Larkin, J., & VanLehn, K. (1997). A computer framework to support self-explanation. In Proceedings of the Eighth World Conference of Artificial Intelligence in Education.
- Conati, C. & VanLehn, K. (1999a). A student model to assess self-explanation while learning from examples. In Proceedings of UM '99, 7th International Conference on User Modeling, Banff, Canada.
- Conati, C. & VanLehn, K. (1999b). Teaching meta-cognitive skills: Implementation and evaluation of a tutoring system to guide self-explanation while learning from examples. In S. P. Lajoie & M. Vivet (Eds.), *Artificial Intelligence in Education (Proceedings of AI-ED '99, Le Mans)* (pp. 297–304). Amsterdam: IOS Press.
- Cowley, C. K., & Jones, D. M. (1992). Synthesized or digitized? A guide to the use of computer speech. *Applied Ergonomics, 23*, 172–176.
- Dalrymple, M. (1999). *Semantics and Syntax in Lexical Functional Grammar*. Cambridge, MA: MIT Press.
- Dalrymple, M., Lamping, J., & Saraswat, V. (1993). LFG semantics via constraints. In Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL '93), Utrecht.
- Danieli, M. & Gerbino, E. (1995). Metrics for evaluating dialogue strategies in a spoken language system. In Working Notes of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation.
- DARPA (1995). Proceedings of the Sixth Message Understanding Conference (MUC–6). San Francisco: Morgan Kaufmann.
- DARPA (1998). Proceedings of the Seventh Message Understanding Conference (MUC–7), Available from http://www.muc.saic.com/proceedings/muc_7_proceedings/overview.html.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*, 391–407.
- DeJong, G. (1977). Skimming newspaper stories by computer. New Haven, CT: Department of Computer Science, Yale University. Research Report 104.

- de Kleer, J. & Brown, J. S. (1984). A qualitative physics based on confluences. *Artificial Intelligence*, 24, 7–83.
- Devanbu, P. & Litman, D. (1991). Plan-based terminological reasoning. In J. Allen, R. Fikes, & E. Sandewall (Eds.), *KR '91: Principles of Knowledge Representation and Reasoning* (pp. 128–138). San Mateo, CA: Morgan Kaufmann.
- Dietterich, T. G. & Flann, N. S. (1997). Explanation-based learning and reinforcement learning: A unified view. *Machine Learning*, 28(2/3).
- Dumais, S. T. (1994). Latent semantic indexing (LSI) and TREC–2. In D. Harman (Ed.), *National Institute of Standards and Technology Text Retrieval Conference*. NIST special publication.
- Ehrlich, U. (1999). Task hierarchies representing Sub-dialogues in speech dialog systems. In *Proceedings of Eurospeech '99*, Budapest.
- Elhadad, M. & Robin, J. (1992). Controlling content realization with functional unification grammars. In R. Dale, E. Hovy, D. Rosner & O. Stock (Eds.), *Aspects of Automated Natural Language Generation* (pp. 89–104). New York: Springer 1992. LNAI 587.
- Elhadad, M. & Robin, J. (no date-a). FUF Manual. Available from <ftp://ftp.cs.bgu.ac.il/pub/people/elhadad/nlg/fufman.ps>.
- Elhadad, M. & Robin, J. (no date-b). SURGE: a comprehensive plug-in syntactic realization component for text. Available from <ftp://ftp.cs.bgu.ac.il/pub/people/elhadad/surge2.ps>.
- Erol, K., Hendler, J. and Nau, D. S. (1994). HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI '94)*, Seattle, WA.
- Ferguson-Hessler, M. G. M. & de Jong, T. (1990). Studying physics texts: Differences in study processes between good and poor solvers. *Cognition and Instruction*, 7, 41–54.
- Foltz, P. W., Britt, M. A., & Perfetti, C. A. (1996). Reasoning from multiple texts: An automatic analysis of readers' situation models. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society* (pp. 110–115). Mahwah, NJ: Erlbaum.
- Fox, B. (1993). *The Human Tutorial Dialog Project*. Hillsdale, NJ: Erlbaum.
- Freedman, R. (1997). Degrees of mixed-initiative interaction in an intelligent tutoring system. *AAAI 1997 Spring Symposium: Computational Models for Mixed Initiative Interaction*.
- Freedman, R. (1999). Atlas: A plan manager for mixed-initiative, multimodal dialogue. *AAAI '99 Workshop on Mixed-Initiative Intelligence*, Orlando.
- Gagné, R. M. (1977). *The Conditions of Learning* (3rd ed.). New York: Holdt, Rinehart, & Winston.
- Georgeff, M. P. and Ingrand, F. F. (1989). Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI '89)*, Detroit, MI, 972–978.
- Georgeff, M., Pell, B., Pollack, M. E., Tambe, M. and Wooldridge, M. (1998). The belief-desire-intention model of agency. In N. Jennings, J. Muller, and M. Wooldridge, *Intelligent Agents V*. Springer.
- Gerlach, M. & Horacek, H. (1989). Dialog control in a natural language system. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester (pp. 27–34).
- Gertner, A. S., Conati, C. & VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI '98)*, Madison. Menlo Park, CA: AAAI Press.
- Glass, M. (1999). Broadening Input Understanding in an Intelligent Tutoring System. PhD thesis, Illinois Institute of Technology.
- Goodman, J. (1996). Parsing algorithms and metrics. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz.
- Graesser, A. C., & Clark, L. C. (1985). *Structures and Procedures of Implicit Knowledge*. Norwood, NJ: Ablex.
- Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31, 104–137.
- Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9, 359–387.
- Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and the TRG (in press). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*.
- Greeno, J. G., Smith, D. R., & Moore, U. L. (1993). Transfer of situated learning. In D. K. Detterman and R. J. Sternberg (Eds.), *Transfer on Trial: Intelligence, cognition, and Instruction* (pp. 99–167). Norwood, NJ: Ablex.
- Grice, H. P. (1969). Utterer's meaning and intentions. *Philosophical Review*, 68(2):147–177.

- Grice, H. P. (1989). *Studies in the Ways of Words*. Cambridge, MA: Harvard University Press.
- Grishman, R., Macleod, C., and Meyers, A. (1994). COMLEX syntax: Building a computational lexicon. In Proceedings of the 15th International Conference on Computational Linguistics (COLING '94), Kyoto.
- Grosz, B. J. (1977). The representation and use of focus in dialogue understanding. Menlo Park, CA: Artificial Intelligence Center, SRI International. Technical Report 151.
- Grosz, B. & Kraus, S. (1993). Collaborative plans for group activities. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI '93), Chambéry, France (vol. 1, pp. 367–373).
- Grosz, B. J. & Sidner, C. L. (1986). Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12, 175–204.
- Hake, R. R. (under review). Interactive-engagement vs. traditional methods: A six-thousand student survey of mechanics test data for introductory physics courses.
- Halliday, M. A. K. (1985). *Introduction to Functional Grammar*. London: Edward Arnold.
- Halloun, I. A. & Hestenes, D. (1985). Common sense concepts about motion. *American Journal of Physics*, 53(11), 1056–1065.
- Halvorsen, P. K. (1987). Situation semantics and semantic interpretation in constraint-based grammars. Technical Report 101, Stanford, CA: CSLI, Stanford University.
- Hawkins, J. A. (1978). *Definiteness and Indefiniteness*. Atlantic Highlands, NJ: Humanities Press.
- Heim, I. (1982). Semantics of Definite and Indefinite Noun Phrases. PhD thesis, University of Massachusetts, Amherst.
- Helmbold, D. P. and Schapire, R. E. (1997). Predicting nearly as well as best pruning decision tree. *Machine Learning*, 27(1).
- Henderson, J. and Lane, P. (1998). A connectionist architecture for learning to parse. In Proceedings of the 17th COLING/36th ACL (COLING-ACL '98), Montreal.
- Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *Physics Teacher*, 30, 141–158.
- Hipp, D. R. (1992). Design and Development of Spoken Natural-Language Dialog Parsing Systems. PhD thesis, Dept. of Computer Science, Duke University.
- Hobbs, J. (1980). Selective inferencing. In Proceedings of the Third National Conference of the Canadian Society for Computational Studies of Intelligence, Victoria, BC (pp. 101–114).
- Hobbs, J. R., Appelt, D. E., Bear, J., and Tyson, M. (1991). Robust processing of real-world natural-language texts. Technical report, SRI International.
- Hobbs, J. R., Appelt, D., Tyson, M., Bear, J., & Israel, D. (1992). SRI International: Description of the FASTUS System used for MUC-4. In Proceedings of the Fourth Message Understanding Conference (MUC-4) (pp. 268–275). San Mateo, CA: Morgan Kaufmann.
- Hobbs, J. & Evans, D. (1980). Conversation as planned behavior. *Cognitive Science* 4(4), 349–377.
- Hobbs, J., Stickel, M., Appelt, D., & Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence* 63(1–2), 69–142.
- Holland, V. M., Kaplan, J. D., & Sams, M. R. (1995) (Eds.). *Intelligent Language Tutors*. Mahwah, NJ: Erlbaum.
- Hume, G. D., Michael, J. A., Rovick, A., & Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, 5, 23–47.
- Hunt, E. & Minstrell, J. (1994). A cognitive approach to the teaching of physics. In K. McGilly (Ed.), *Classroom Lessons: Integrating Cognitive Theory and Classroom Practice* (pp. 51–74). Cambridge: MIT Press.
- Huyck, C. R. & Lytinen, S. L. (1993). Efficient heuristic natural language parsing. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI '93), Washington, D. C. (pp. 386–391).
- Jackendoff, R. S. (1983). *Semantics and Cognition*. Cambridge, MA: MIT Press.
- Jackendoff, R. S. (1990). *Semantic Structures*. Cambridge, MA: MIT Press.
- Jacobs, P. S. (1992) (Ed.). Text -based intelligent systems: Current research and practice in information extraction and retrieval. Hillsdale, NJ: Erlbaum.
- Jain, A. N. (1991). PARSEC: A Connectionist Learning Architecture for Parsing Speech. PhD thesis, School of Computer Science, Carnegie Mellon University.
- Jain, A. N. and Waibel, A. H. (1990). Incremental parsing by modular recurrent connectionist networks. In Tourterzky, D. S., editor, *Advances in Neural Information Processing 2*. Morgan Kaufmann.
- Johnson, W. L., Rickel, J., & Lester, J. C. (in press). Animated pedagogical agents: Face-to-face interactive learning environments. *International Journal of Artificial Intelligence in Education*.
- Johnsua, S. & Dupin, J. J. (1987). Taking into account student conceptions in instructional strategy: An example in physics. *Cognition and Instruction*, 4(3), 117–135.

- Jönsson, A. (1991). A dialogue manager using initiative-response units and distributed control. In Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics, Manchester (pp. 233–238).
- Jordan, P. W. (1996). Using terminological knowledge representation languages to manage linguistic resources. ACL '96 Student Session, Santa Cruz, CA.
- Jordan, P. W. and Walker, M. A. (1996). Deciding to remind during collaborative problem solving: Empirical evidence for agent strategies. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI '96), Portland, OR.
- Jullien, C. & Marty, J.-C. (1989). Plan revision in person-machine dialog. In Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics, Manchester (pp. 153–160).
- Karttunen, L. (1976). Discourse referents. In J. McCawley (Ed.), *Syntax and Semantics*, vol. 7. New York: Academic Press.
- Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., & Greenberg, L. (1998). Sherlock 2: An intelligent tutoring system built on the LRDC framework. In C. P. Bloom & R. B. Loftin (Eds.), *Facilitating the Development and Use of Interactive Learning Environments* (pp. 227–258). Hillsdale, NJ: Erlbaum.
- Kehler, A. (1995). Interpreting Cohesive Forms in the Context of Discourse Inference. PhD thesis, Harvard University.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1995). Intelligent tutoring goes to school in the big city. In J. Greer (Ed.), *Proceedings of the 7th World Conference on Artificial Intelligence and Education* (pp. 421–428). Charlottesville, NC: AACE.
- Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*.
- Landauer, T. K., Foltz, P. W., Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25, 259–284.
- Lascarides, A. & Asher, N. (1991). Discourse relations and defeasible knowledge. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL '91), Berkeley, CA (pp. 55–62).
- Lascarides, A. & Oberlander, J. (1992). Temporal coherence and defeasible knowledge. *Theoretical Linguistics*, 19.
- Lavie, A. (1995). A Grammar Based Robust Parser For Spontaneous Speech. PhD thesis, School of Computer Science, Carnegie Mellon University.
- Lavie, A., Gates, D., Gavalda, M., Mayfield, L., Waibel, A., and Levin, L. (1996). Multi-lingual translation of spontaneously spoken language in a limited domain. Proceedings of the 16th International Conference on Computational Linguistics (COLING '96), Copenhagen.
- Lehman, J. F. (1989). Adaptive Parsing: Self-Extending Natural Language Interfaces. PhD thesis, School of Computer Science, Carnegie Mellon University.
- Lehmann, F. (1992) (Eds.). *Semantic Networks in Artificial Intelligence*. New York: Pergamon.
- Lehnert, W. (1997). Information extraction: What have we learned? *Discourse Processes*, 23, 441–470.
- Lepper, M. R., Woolverton, M., Mumme, D. L., & Gurtner, J.-L. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 75–105). Hillsdale, NJ: Erlbaum.
- Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. H. Larkin & R. W. Chabay (Eds.), *Computer-assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches* (pp. 201–238). Hillsdale, NJ: Erlbaum.
- Lewis, D. (1979). Scorekeeping in a language game. *Journal of Philosophical Logic* 6, 339–359.
- Lochbaum, K. (1994). Using Collaborative Plans to Model the Intentional Structure of Discourse. PhD thesis, Harvard University.
- Lovett, M. C. (1992). Learning by problem solving versus by examples: The benefits of generating and receiving information. In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.
- Magerman, D. M. and Marcus, M. P. (1990). Parsing a natural language using mutual information statistics. In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI '90), Boston.
- Martin, J. K. (1997). An exact probability metric for decision tree splitting. *Machine Learning* 28(2).
- Mazur, E. (1993). *Peer Instruction: A User's Manual*. Cambridge, MA: Harvard University Press.
- McArthur, D., Stasz, C., & Zmuidzinis, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction*, 7, 197–244.

McCauley, L., Gholson, B., Hu, X., Graesser, A. C., and the Tutoring Research Group (1998). Delivering smooth tutorial dialog using a talking head. In Proceedings of the Workshop on Embodied Conversation Characters (pp. 31–38). Tahoe City, CA: AAAI and ACM.

McCloskey, M., Caramazza, A., & Green, B. (1980). Curvilinear motion in the absence of external forces: Naive beliefs about the motion of objects. *Science*, 210(5), 1139–1141.

McDonald, D. (1990). Robust partial-parsing through incremental, multi-level processing: Rationales and biases. In P. S. Jacobs (Ed.), Proceedings of the AAAI Spring Symposium on Text-Based Intelligent Systems: Current Research in Text Analysis, Information Extraction, and Retrieval. Schenectady, NY: GE Research and Development Center. Technical report 90CRD198.

McDonald, D. (1992). An efficient chart-based algorithm for partial-parsing of unrestricted texts. In Proceedings of the Third Conference on Applied Natural Language Processing, Trento.

McDonald, D. (1993a). Efficiently parsing large corpora. In Proceedings of the ACL Workshop on Very Large Corpora: Academic and Industrial Perspectives.

McDonald, D. (1993b). The interplay of syntactic and semantic node labels in partial parsing. In Proceedings of the Third International Workshop on Parsing Technologies.

McRoy, S. & Hirst, G. (1991). An abductive account of repair in conversation. AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation, Asilomar, CA (pp. 52–57).

Merrill, D. C., Reiser, B. J., Merrill, S. K., & Landes, S. (1995). Tutoring: Guided learning by doing. *Cognition and Instruction*, 13(3), 315–372.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *Journal of the Learning Sciences*, 2(3), 277–305.

Miikkulainen, R. (1996). Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science*, 20, 47–74.

Miller, G. A. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 235–312.

Miller, S., Stallard, D., Bobrow, R., and Schwartz, R. (1996). A fully statistical approach to natural language interfaces. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics.

Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzioni, O., and Gil, Y. (1990). Explanation-based learning: A problem solving perspective. In J. G. Carbonell (Ed.), *Machine Learning: Paradigms and Methods*. Boston: MIT Press.

Mitrovic, A. & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence and Education*, 10.

Mittal, V., Roth, S., Moore, J. D., Mattis, J., & Carenini, G. (1995). Generating explanatory captions for information graphics. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI '95), Montreal. Menlo Park, CA: AAAI Press.

Moore, J. D. (1995). *Participating in Explanatory Dialogues*. Cambridge, MA: MIT Press.

Moore, J., & Pollack, M. (1992). A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4), 537–544.

Neumann, G., Backofen, R., Baur, J., Becker, M., and Braun, C. (1997). An information extraction core system for real world German text processing. In Proceedings of the Fifth Conference on Applied Natural Language Processing.

O'Donnell, A. M., Dansereau, D. F., Hall, R. H., Skaggs, L. P., Hythecker, V. I., Peel, J. L., & Rewey, K. L. (1990). Learning concrete procedures: Effects of processing strategies and cooperative learning. *Journal of Educational Psychology*, 82(1), 171–177.

Paiva, A., & Machado, I. (1998). Vincent, an autonomous pedagogical agent for on-the-job training. In Proceedings of the Fourth International Conference on Intelligent Tutoring Systems (pp. 584–593). New York: Springer.

Palincsar, A. S., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition & Instruction*, 1, 117–175.

Papineni, K. A., Roukos, S., & Ward, R. T. (1999). Free-flow dialog management using forms. In Proceedings of Eurospeech '99, Budapest.

Passonneau, R. J. (1994). Protocol for coding discourse referential noun phrases and their antecedents. Technical report, Columbia University.

Perrault, C. & Allen, J. (1980). A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3–4), 167–182.

- Person, N. K. & Graesser, A.C. (1999). Evolution of discourse in cross-age tutoring. In A. M. O'Donnell and A. King (Eds.), *Cognitive Perspectives on Peer Learning* (pp. 69–86). Mahwah, NJ: Erlbaum.
- Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences*, 6, 205–229.
- Person, N., Klettke, B., Link, K., Kreuz, R., & TRG (1999). The integration of affective responses in AutoTutor. In Proceedings of the International Workshop on Affect in Interactions. New York: Springer.
- Pfundt, H. & Duit, R. (1991). *Bibliography: Students' Alternative Frameworks and Science Education*. Kiel, FRG: Institute for Science Education.
- Pieraccini, R., Levin, E., & Eckert, W. (1997). AMICA: The AT&T mixed initiative conversational architecture. In Proceedings of Eurospeech '97, Rhodes.
- Pietra, S., Epstein, M., Roukos, S., and Ward, T. (1997). Fertility models for statistical natural language understanding. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics.
- Pirolli, P. & Bielaczyc, K. (1989). Empirical analyses of self-explanation and transfer in learning to program. In Proceedings of the Eleventh Annual Conference of the Cognitive Science Society (pp. 459–457). Hillsdale, NJ: Erlbaum.
- Ploetzner, R. & VanLehn, K. (1997). The acquisition of informal physics knowledge during formal physics training. *Cognition and Instruction*, 15(2), 169–206.
- Poesio, M. (1993). A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues, In P. Aczel, D. Israel, Y. Katgiri, and S. Peters, editors, *Situation Theory and its Applications* (vol. 3, pp. 339–374).
- Pollack, M. (1986). A model of plan inference that distinguishes between the beliefs of actors and observers. In Proceedings of 24th Annual Meeting of the Association for Computational Linguistics, New York (pp. 207–214).
- Power, R. (1974). A Computer Model of Conversation. PhD. thesis, University of Edinburgh, Scotland.
- Pressley, M., Wood, E., Woloshyn, V. E., Martin, V., King, A., & Menke, D. (1992). Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitates learning. *Educational Psychologist*, 27, 91–109.
- Prince, E. F. (1981). Toward a taxonomy of given-new information. In *Radical Pragmatics*, pages 223–255. New York: Academic Press.
- Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal*, 24(1), 13–48.
- Quinlan, J. R. (1990). Probabilistic decision trees. In Y. Kodratoff and R. Michalski (Eds.), *Machine Learning: An Artificial Intelligence Approach*. San Mateo, CA: Morgan Kaufmann.
- Rayner, M. & Alshawi, H. (1992). Deriving database queries from logical forms by abductive definition expansion. In Proceedings of the Third Conference of Applied Natural Language Processing, Trento, Italy (pp. 1–8).
- Rehder, B., Schreiner, M. E., Wolfe, M. B. W., Laham, D., Landauer, T. K., & Kintsch, W. (1998). Using latent semantic analysis to assess knowledge: Some technical considerations. *Discourse Processes*, 25, 337–354.
- Reiger, C. (1974). *Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances*. Stanford, CA: Stanford Artificial Intelligence Laboratory. Memo AIM-233.
- Reiter, E. (1994). Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible?. In Proceedings of the Seventh International Workshop on Natural Language Generation, Kennebunkport, ME (pp. 163–170).
- Reiter, E. & Dale, R. (1997). Building applied natural-language generation systems. *Journal of Natural-Language Engineering* 3, 57–87.
- Renkl, A. (in press). Learning from worked-examples: A study on individual differences. *Cognitive Science*.
- Riloff, E. (1996). Using learned extraction patterns for text classification. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches for Natural Language Processing*. New York: Springer.
- Ritter, F. & Feurzeig, W. (1988). Teaching real-time tactical thinking. In J. Psozka, L. D. Massey, & S. A. Mutter (Eds.), *Intelligent Tutoring Systems: Lessons Learned* (pp. 285–301). Hillsdale, NJ: Erlbaum.
- Rose, C. P. (1997a). Robust Interactive Dialogue Interpretation. PhD thesis, School of Computer Science, Carnegie Mellon University.
- Rose, C. P. (1997b). The role of natural language interaction in electronics troubleshooting. In Proceedings of the Energy Week Conference and Exhibition.
- Rose, C. P. (1999). A genetic programming approach for robust language interpretation. In L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. Angeline (Eds.), *Advances in Genetic Programming 3*. Cambridge, MA: MIT Press.

- Rose, C. P., Di Eugenio, B., & Moore, J. D. (1999). A dialogue based tutoring system for basic electricity and electronics. In S. P. Lajoie & M. Vivet (Eds.), *Artificial Intelligence in Education (Proceedings of AI-ED '99, Le Mans)* (pp. 759–761). Amsterdam: IOS Press.
- Rose, C. P. & Lavie, A. (1997). An efficient distribution of labor in a two stage robust interpretation process. In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.
- Rose, C. P. & Lavie, A. (1999). Balancing robustness and efficiency in unification augmented context-free parsers for large practical applications. In J. C. Junqua & G. V. Noord (Eds.), *Robustness in Language and Speech Technologies*. Dordrecht: Kluwer.
- Rose, C. P. & Lavie, A. (to appear). A domain independent approach for efficiently interpreting extragrammatical utterances. *Journal of Natural Language Engineering*.
- Rose, C. P. & Levin, L. S. (1998). An interactive domain independent approach to robust dialogue interpretation. In Proceedings of the 17th COLING/36th ACL (COLING-ACL '98), Montreal..
- Rose, C. P. & Waibel, A. (1997). Recovering from parser failures: A hybrid statistical/symbolic approach. In J. Klavans & P. Resnik (Eds.), *Balancing Act: Combining Symbolic and Statistical Approaches to Language Processing*. Cambridge, MA: MIT Press.
- Sacks, H., Schegloff, E. A. & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking in conversation. *Language*, 50(4), 696–735.
- Sanker, A. & Gorin, A. (1993). Adaptive language acquisition in a multi-sensory device. *IEEE Transactions on Systems, Man, and Cybernetics*.
- Schank, R. (1975). *Conceptual Information Processing*. New York: Elsevier.
- Schank, R., Lebowitz, M., & Birnbaum, L. (1980). An integrated understander. *American Journal of Computational Linguistics*, 6(1).
- Schneider, D. and McCoy, K. F. (1998). Recognizing syntactic errors in the writing of second language learners. In Proceedings of the 17th COLING/36th ACL (COLING-ACL '98), Montreal..
- Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press.
- Slotta, J. D., Chi, M. T. H., & Joram, E. (1995). Assessing students' misclassifications of physics concepts: An ontological basis for conceptual change. *Cognition and Instruction*, 13(3), 373–400.
- Smith, J. P., diSessa, A. A., & Roschelle, J. (1993). Misconceptions reconceived: A constructivist analysis of knowledge in transition. *Journal of the Learning Sciences*, 2(2), 115–164.
- Smith, R. W. (1992). Integration of domain problem solving with natural language dialog: The missing axiom theory. In *Applications of Artificial Intelligence X: Knowledge-Based Systems* (pp. 270–275). SPIE v. 1707.
- Sperber, D. & Wilson, D. (1986). *Relevance: Communication and Cognition*. Cambridge, MA: Harvard University Press.
- Stevens, A. & Collins, A. (1977). The goal structure of a Socratic tutor. In Proceedings of the National ACM Conference. New York: ACM.
- Suhm, B., Levin, L., Coccaro, N., Carbonell, J., Horiguchi, K., Isotani, R., Lavie, A., Mayfield, L., Rose, C. P., Dykema, C. V.-E., and Waibel, A. (1994). Speech-language integration in a multi-lingual speech translation system. In Proceedings of the AAAI Workshop on Integration of Natural Language and Speech Processing.
- Thomason, R. & Hobbs, J. R. (1997). Interrelating interpretation and generation in an abductive framework. AAAI Fall Symposium on Communicative Action in Humans and Machines, Cambridge, MA.
- Tversky, A. & Kahneman, D. (1974). Judgments under uncertainty: Heuristics and biases. *Science*, 185, 1124–1131.
- Utgoff, P. E., Berkman, N. C., and Clouse, J. A. (1997). Decision tree induction based on tree restructuring. *Machine Learning*, 29(1).
- van Genabith, J. & Crouch, D. (1996). Direct and underspecified interpretations of LFG f-structures. In Proceedings of the 16th International Conference on Computational Linguistics (COLING '96), Copenhagen.
- VanLehn, K. (1990). *Mind Bugs: The Origins of Procedural Misconceptions*. Cambridge, MA: MIT Press.
- VanLehn, K. (1996). Conceptual and meta learning during coached problem solving. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *ITS '96: Proceedings of the Third International Conference on Intelligent Tutoring Systems*. New York: Springer.
- VanLehn, K., Siler, S., Murray, C. & Baggett, W. (1998). What makes a tutorial event effective? In: M. A. Gernsbacher & S. J. Derry (Eds.), Proceedings of the Twenty-first Annual Conference of the Cognitive Science Society (pp. 1084–1089). Hillsdale, NJ: Erlbaum.
- Van Noord, G. (1997). An efficient implementation of the head-corner parser. *Computational Linguistics*, 23(3).

- Viennot, L. (1979). Spontaneous reasoning in elementary dynamics. *European Journal of Science Education*, 1, 205–221.
- Walker, M. A. (1993). Informational Redundancy and Resource Bounds in Dialogue. PhD thesis, University of Pennsylvania.
- Webb, N. M. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research*, 13, 21–40.
- Weld, D. & de Kleer, J. (1990). *Readings in Qualitative Reasoning about Physical Systems*. Menlo Park, CA: Morgan Kaufmann.
- Wiemer-Hastings, P., Graesser, A., Harter, D., and the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. In B. Goettl, H. Halff, C. Redfield, & V. Shute (Eds.), *Intelligent Tutoring Systems: Fourth International Conference (ITS '98)* (pp. 334–343). New York: Springer.
- Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. *Artificial Intelligence in Education (Proceedings of AI-ED '99, Le Mans)* (pp. 535–542). Amsterdam: IOS Press.
- Wilkins, D. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. San Mateo, CA: Morgan Kaufmann.
- Wolfe, M. B. W., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., Landauer, T. K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes*, 25, 309–336.
- Wong, L. H., Quek, C., & Looi, C. K. (1998). TAP-2: A framework for an inquiry dialog-based tutoring system. *International Journal of Artificial Intelligence in Education*, 9.
- Woo, C. W., Evens, M. W., Michael, J. A., & Rovick, A. A. (1991). Dynamic instructional planning for an intelligent physiology tutoring system. In *Proceedings of the 4th Annual IEEE Computer-Based Medical Systems Symposium*, Baltimore (pp. 226–233). Los Alamitos, CA: IEEE Computer Society Press.
- Worm, K. (1998). A model of robust processing of spontaneous speech by integrating viable fragments. In *Proceedings of the 17th COLING/36th ACL (COLING-ACL '98)*, Montreal.
- Woscyna, M., Coccaro, N., Eisele, A., Lavie, A., McNair, A., Polzin, T., Rogina, I., Rose, C. P., Sloboda, T., Tomita, M., Tsutsumi, J., Waibel, N., Waibel, A., and Ward, W. (1993). Recent advances in JANUS: A speech translation system. In *Proceedings of the ARPA Human Languages Technology Workshop*.
- Yang, Q. (1990). Formalizing planning knowledge for hierarchical planning. *Computational Intelligence* 6(1), 12–24.
- Yen, J. (1991). CLASP: Integrating term subsumption systems and production systems. *IEEE Transactions on Knowledge and Data Engineering* 4(1), 25–31.
- Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., Evens, M. W. (1999a). Delivering hints in a dialogue-based intelligent tutoring system. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI '99)*, Orlando, FL.

10 Vitae

VITA - Kurt VanLehn

Education

- B.S. Mathematics, Stanford University, 1974.
- M.S. Computer Science, Massachusetts Institute of Technology, 1978.
- Ph.D. Computer Science, Massachusetts Institute of Technology, 1983.

Employment

- Research Assistant, MIT Artificial Intelligence Laboratory, 1977.
- Research Associate, Bolt Beranek and Newman Inc., 1978 (January-October).
- Research Associate, Xerox Palo Alto Research Center, 1978-1985.
- Assistant Professor, Depts. of Psychology and Computer Science, Carnegie-Mellon University, 1985-1990.
- Associate Professor (tenured), Department of Computer Science, University of Pittsburgh, 1990-1998
- Professor, Department of Computer Science, University of Pittsburgh, 1998-present
- Senior Scientist, Learning Research and Development Center, University of Pittsburgh, 1990-present
- Co-director, Intelligent Systems Program, University of Pittsburgh, 1994-1996

Awards and Honors

National Science Foundation Graduate Fellow, 1974-1977.
 Spencer Foundation Research Fellow, October 1986 to August 1988.
 Resident Fellow, Center for Philosophy of Science, University of Pittsburgh, 1990-present
 Best Paper (with Joel Martin), 1993 World Conference for Artificial Intelligence in Education.
 Resident Fellow, Center for Advanced Study in the Behavioral Sciences, 1996-1997.
 Best Paper (with Cristina Conati, Abigail Gertner & Marek Druzdzal) 1997 User Modeling Conference.
 Outstanding Paper Award (with Conati, C.). 9th World Conference of Artificial Intelligence and Education, Le Man, France, 1999

Research Grants and Contracts

- 14 single-investigator grants and contracts since 1982 from ONR, DARPA, NIH, AFOSR. Combined average is \$214,135 per year.
- PI of 2 multi-investigator grants: ONR/DARPA \$11,575,053 from 1986 to 1991. Mellon/Sage \$1,000,000 from 1996 to 1998.
- Director of NSF funded Center for Intedisciplinary Research in Constructive Learning Environments, \$4,997,797 over 5 years starting January 1, 1998.

Selected publications relevant to the grant

Chi, M. T. H. & VanLehn, K. (1991) The content of physics self-explanations. *Journal of the Learning Sciences*, 1(1), pp. 69-106.

VanLehn, K. (1991) Rule acquisition events in the discovery of problem solving strategies. *Cognitive Science*, 15(1), pp. 1-48.

VanLehn, K., Jones, R.M. & Chi, M. T. H. (1992) A model of the self-explanation effect. *Journal of the Learning Sciences*, 2(1), pp. 1-60.

Martin, J. & VanLehn, K. (1993). OLAE: Progress toward a multi-activity, Bayesian student modeller. In S. P. Brna, S. Ohlsson & H. Pain (eds.) *Artificial Intelligence in Education, 1993: Proceedings of AI-ED 93*. Charlottesville, VA: Association for the Advancement of Computing in Education. Pp. 410-417. Winner of Best Paper award for the conference.

Jones, R. M. & VanLehn, K. (1994). Acquisition of children's addition strategies: A model of impasse-free, knowledge-level learning. *Machine Learning*, 15(1&2), pp. 11-36.

VanLehn, K., Ohlsson, S. & Nason, R. (1994). Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education*, 5(2), pp. 135-175.

Martin, J. & VanLehn, K. (1995) Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, 42, pp. 575-591.

Ur, S. & VanLehn, K. (1995) Steps: A simulated, tutable physics student. *Journal of Artificial Intelligence in Education*, 6(4), pp. 405-437.

Ploetzner, R. & VanLehn, K. (1997) The acquisition of informal physics knowledge during formal physics training. *Cognition and Instruction*, 15(2), 169-205.

Conati, C., Gertner, A., VanLehn, K. & Druzdzal, M. (1997) On-line student modelign for coached problem solving using Bayesian networks. In A. Jameson, C. Paris & C. Tasso (eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*. Vienna; springer. Winner of "Best Paper" award. pp. 231-242.

VanLehn, K. & Martin, J. (1998) Evaluation of an assessment system based on Bayesian student modeling. *International Journal of Artificial Intelligence and Education*, vol. 8.2.

VanLehn, K. (1998) Analogy events: How examples are used during problem solving. *Cognitive Science* 22(3), pp. 347-388.

VanLehn, K. (1998) Student modeling. In M. Polson & J. Richardson (Eds.) *Foundations of Intelligent Tutoring Systems*. Hillsdale, NJ: Erlbaum. Pp. 55-78.

VanLehn, K. (1999) Rule learning events in the acquisition of a complex skill: An evaluation of Cascade. *Journal of the Learning Sciences* 8(1), pp. 71-125.

Conati, C. & VanLehn, K. (1999) Teaching meta-cognitive skills: Implimentation and evaluation of a tutoring system to guide self-explanation while learning from examples. In *Proceedings of AIED '99, 9th World Conference of Artificial Intelligence and Education*, LeMan, France. Winner of the "Outstanding Paper" award.

VanLehn, K., Siler, S., Murray, C., Yamauchi, T. & Baggett, W. B. (in press) Human tutoring: Why do only some events cause learning? *Cognition and Instruction*.

VITA - Arthur C. Graesser

Degree History and Professional Experience

1968-1972 B.A. in Psychology, Florida State University
 1972-1977 Ph.D. in Psychology, University of California, San Diego
 1976-1985 Assistant, Associate, and Full Professor with Tenure at California State University, Fullerton, Department of Psychology.
 Fall, 1983 Research Associate at Yale University, Departments of Psychology & Computer Science.
 Spring, 1984 Visiting Professor at Stanford University, Department of Psychology; instructor of graduate seminar on discourse comprehension.
 1985-present Full professor at The University of Memphis, Departments of Psychology and Mathematical Sciences. Co-director of Institute for Intelligent Systems, Director of the Center for Applied Psychological Research

Publications Most Closely Related to the Proposed Project

Graesser, A.C., & Black, J.B. (1985) (Eds.). The psychology of questions. Hillsdale, NJ: Erlbaum.
 Graesser, A.C., & Clark, L.C. (1985). Structures and procedures of implicit knowledge. Norwood, NJ: Ablex.
 Graesser, A.C., & Bower, G.H. (1990) (Eds.). The psychology of learning and motivation: Inferences and text comprehension. New York: Academic Press.
 Lauer, T., Peacock, E., & Graesser, A. C. (1992) (Eds.). Questions and information systems. Hillsdale, NJ: Erlbaum.
 Britton, B.F., & Graesser, A.C. (1996) (Eds.). Models of understanding text. Hillsdale, NJ: Erlbaum.
 Hacker, D.J., Dunlosky, J., & Graesser, A.C. (1998)(Eds.). Metacognition in educational theory and practice. Mahwah, NJ: Erlbaum.
 Golding, J. M., Graesser, A. C., & Millis, K. K. (1990). What makes a good answer to a question?: Testing a psychological model of question answering. Discourse Processes, 13, 305-325.
 Graesser, A. C., & Franklin, S. P. (1990). QUEST: A cognitive model of question answering. Discourse Processes, 13, 279-303.
 Graesser, A. C. & Hemphill, D. (1991). Question answering in the context of scientific mechanisms. Journal of Memory and Language, 30, 186-209.
 Graesser, A. C., Lang, K. L., & Roberts, R. M. (1991). Question answering in the context of stories. Journal of Experimental Psychology: General, 120, 254-277.
 Graesser, A. C., Gordon, S. E., & Brainerd, L. E. (1992). QUEST: A model of question answering. Computers and Mathematics with Applications, 23, 733-745.
 Graesser, A. C., Langston, M. C., & Lang, K. L., (1992). Designing educational software around questioning. Journal of Artificial Intelligence in Education, 3, 235-241.
 Graesser, A. C., & McMahan, C. L. (1993). Anomalous information triggers questions when adults solve problems and comprehend stories. Journal of Educational Psychology, 85, 136-151.
 Langston, M.C., & Graesser, A.C. (1993). The Point and Query Interface: Exploring knowledge by asking questions. Journal of Educational Multimedia and Hypermedia, 2, 355-368.
 Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. American Educational Research Journal, 31, 104-137.
 Graesser, A.C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. Psychological Review, 101, 371-95.
 Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. Learning and Individual Differences, 6, 205-29.
 Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. Applied Cognitive Psychology, 9, 359.1-28.
 Person, N. K., Kreuz, R. J., Zwaan, R., & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. Cognition and Instruction, 13, 161-188.

- Graesser, A.C., Baggett, W., & Williams, K. (1996). Question-driven explanatory reasoning. Applied Cognitive Psychology, *10*, S17-S32.
- Graesser, A.C., Swamer, S., & Hu, X. (1997). Quantitative discourse psychology. Discourse Processes, *23*, 229-263.
- Graesser, A.C., & Bertus, E.L. (1998). The construction of causal inferences while reading expository texts on science and technology. Scientific Studies of Reading, *2*, 247-269.
- Graesser, A.C., Kessler, M.A., Kreuz, R.J., & McLain-Allen, B. (1998). Verification of statements about story worlds that deviate from normal conceptions of time: What is true about *Einstein's Dreams*? Cognitive Psychology, *35*, 246-301.
- Williams, K.E., Hultman, E., & Graesser, A.C. (1998). CAT: A tool for eliciting knowledge on how to perform procedures. Behavior Research Methods, Instruments, & Computers, *30*, 565-572.
- Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., and the Tutoring Research Group (in press). AutoTutor: A simulation of a human tutor. Journal of Cognitive Systems Research.
- Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and the TRG (in press). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. Interactive Learning Environments.
- Magliano, J., Trabasso, T., & Graesser, A.C. (in press). Strategic processing during comprehension. Journal of Educational Psychology.
- Graesser, A. C., McMahan, C. L., & Johnson, B. K. (1994). Question asking and answering. In M. Gernsbacher (Ed.), Handbook of Psycholinguistics (pp. 517-538). San Diego, CA: Academic Press.
- Graesser, A.C., Millis, K.K., & Zwaan, R.A. (1997). Discourse comprehension. In J.T. Spence, J.M. Darley, and D.J. Foss (Eds.), Annual Review of Psychology, Vol. 48. Palo Alto, CA: Annual Reviews Inc.

VITA - Peter M. Wiemer-Hastings

Degree History and Professional Experience

1979-1984	B.S. in Computer Science, Michigan State University
1984-1989	Research Scientist, National Security Agency
1986-1988	M.S. in Computer Science, Johns Hopkins University
1988-1989	Visiting Researcher, Center for Machine Translation, Carnegie Mellon University
1989-1994	Ph.D. in Computer Science, University of Michigan (Steven Lytinen, chair)
1990-1991	Research Scientist, EDS Center for Advanced Research
1994-1996	Postdoctoral Research Fellow, University of Michigan
1996-present	Postdoctoral Research Fellow, University of Memphis

Publications most closely related to the proposed project

- Wiemer-Hastings, P. How Latent is Latent Semantic Analysis? *IJCAI-99*, Stockholm, Sweden, 1999.
- Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A., Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. *AI in Education '99*, Le Mans, France, 1999.
- Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A., Approximate natural language understanding for an intelligent tutor, *Proceedings of the 12th International Florida Artificial Intelligence Research Symposium*, Menlo Park, CA: AAAI Press, 1999.
- Wiemer-Hastings, P., Graesser, A., Harter, D., and the Tutoring Research Group. The foundations and architecture of Autotutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, San Antonio, Texas (pp. 334-343). Berlin: Springer-Verlag, 1998.
- Graesser, A., Franklin, S., Wiemer-Hastings, P., and the Tutoring Research Group. Simulating smooth tutorial dialogue with pedagogical value, *Proceedings of the 11th International Florida Artificial Intelligence Research Symposium*, Menlo Park, CA: AAAI Press, 1998.

Representative Publications

- Wiemer-Hastings, P., Graesser, A., and Wiemer-Hastings, K. (1998). Inferring the meaning of verbs from context, *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 1142-1147). Mahwah, NJ: Lawrence Erlbaum Associates.
- Hastings, P., Use of Context in an Automatic Lexical Acquisition System, *Proceedings of the IJCAI workshop on Context in Natural Language Processing*, L. Iwanska, Chair, 1995.

Hastings, P. *Automatic Acquisition of Word Meaning from Context*, doctoral dissertation, The University of Michigan, 1994

Hastings, P. & Lytinen, S., The Ups and Downs of Lexical Acquisition, *Proceedings of the 12th National Conference on Artificial Intelligence*, Cambridge, MA: MIT Press, 1994.

Hastings, P. & Lytinen, S., Objects, Actions, Nouns, and Verbs, *Proceedings of the 16th Meeting of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.

VITA - Natalie K. Person

EDUCATION

1991 - 1994 The University of Memphis, Ph.D. in Cognitive Psychology
 1989 - 1990 Memphis State University, M.S. in General Psychology
 1983 - 1987 University of Mississippi, B.A. in Psychology

PROFESSIONAL HISTORY

August 1994 - present Assistant professor, Rhodes College, Department of Psychology
 September 1997- Co-PI, National Science Foundation grant
 September 2000 Simulating Tutors with Natural dialog and pedagogical strategies
 August 1988- Teacher, Lamplighter Montessori School, Memphis, TN 38106
 May 1990

RELATED PUBLICATIONS

Person, N.K., Klettke, B., Link, K., Kreuz, R.J., and the TRG (in press). The integration of affective responses into AutoTutor. *Proceedings for the International Workshop on Affect in Interactions*. Springer-Verlag.

Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N. K., and the TRG (in press). Using latent semantic analysis to evaluate the contributions of students in AutoTutor.

Interactive Learning Environments.

Person, N.K., & Graesser, A.C. (1999). Evolution of discourse in cross-age tutoring. In A.M. O'Donnell and A. King (Eds.), *Cognitive perspectives on peer learning* (pp. 69-86). Mahwah, NJ: Erlbaum.

Graesser, A. C., Bowers, C. A., Hacker, D. J., & Person, N. K. (1997). An anatomy of naturalistic tutoring. In K. Hogan & M. Pressley (Eds.), *Scaffolding student learning: Instruction approaches and issues* (pp. 145-184). Cambridge, MA: Brookline Books.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring sessions. *Applied Cognitive Psychology*, 9, 1-28.

Person, N. K., Kreuz, R. J., Zwaan, R., & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. *Cognition and Instruction*, 13, 161-188.

Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences*, 6, 205-229.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, 31, 104-137.

Graesser, A. C., Person, N. K., & Johnston, G. S. (1993). Three obstacles in empirical research on aesthetic and literary comprehension. In R. J. Kreuz & M. S. MacNealy (Eds.), *Empirical approaches to literature and aesthetics*. Norwood, NJ: Ablex.

Graesser, A. C., Person, N. K., and Huber, J. D. (1993). Question asking during tutoring and in the design of educational software. In M. Rabinowitz (Ed.), *Cognitive science foundations of instructional software*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Graesser, A. C., Person, N. K., and Huber, J. D. (1992). Mechanisms that generate questions. In T. Lauer, E. Peacock, & A. C. Graesser (Eds.), *Questions and information systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

VITA- Xiangen Hu

Education

- 1982: B. S, Mathematics, Huazhong University of Science and Technology, PRC.
 1985: M.S, Applied Mathematics. Huazhong University of Science and Technology, PRC.
 1991: M.A, Social Sciences, University of California, Irvine.
 1993: Ph. D, Psychology, University of California, Irvine.

Selected Journal Articles, Book Chapters, Refereed published conference proceedings

Hu, X. (in press) Extending General Processing Tree Models to Analyze Reaction Time Experiments. Journal of Mathematical Psychology.

Hu, X. (in press) Multinomial processing tree models: An implementation. Behavior Research Methods, Instrumentation, and Computers.

^aGraesser, A. C., Bowers, C., Bayen, U. J., & Hu, X. (in press). Who said what? Who knows what? Tracking speakers and knowledge in narrative. In W. van Peer, E. Andringa & S. Chatman, (Eds.), Narrative Perspective: Cognition and Emotion. Albany, NY: SUNY Press.

Hu, X. & Phillips, G. A. (1999) GPT.EXE: A Powerful tool for the Visualization and Analysis of General Processing Tree Models. Behavior Research Methods, Instrumentation, and Computers.31(2),220-234.

^bHu, X., Graesser, A.C., and the Tutoring Research Group (1998). Using WordNet and latent semantic analysis to evaluate the conversational contributions of learners in the tutorial dialog. Proceedings of the International Conference on Computers in Education, Vol. 2.(pp. 337-341). Beijing, China: Springer.

Graesser, A., Swamer, S, & Hu, X. (1997) Quantitative discourse psychology, Discourse Processes, 23(3), 229-263.

Crowther, C., Batchelder, W. H, & Hu, X. (1995). A measurement-theoretic analysis of Massaro's fuzzy logic model of perception. Psychological Review, 102(2), 196-408.

Batchelder, W.H., Riefer, D. M., & Hu, X. (1994). Measuring memory factors in source monitoring: Reply to Kinchla. Psychological Review, 101(1). 172-176.

Hu, X. & Batchelder, H. W. (1994). Statistical analysis of general processing tree models with the EM algorithm, Psychometrika, 59(1). 21-47.

Riefer, D. M., Hu, X., & Batchelder, W. H. (1994). Response strategies in source monitoring, Journal of Experimental Psychology: Learning, memory & cognition. 20(3). 680-693.

^aBatchelder, W.H., Hu, X., & Riefer, D. M. (1993). Analysis of a model for source monitoring. In G. H. Fischer & D. Laming (Eds.), Contributions to mathematical psychology, psychometrics, and methodology. New York: Springer-Verlag.

^aBook chapter. ^bRefereed published conference proceedings.

VITA - Pamela W. Jordan**Professional History:**

- 1994-present: ABD in Intelligent Systems specialty in Discourse Processing
 Summer 1994: Intern, Mitsubishi Electric Research Laboratories, Cambridge MA
 1992-1994: Graduate Student Researcher, Center for Machine Translation, Carnegie Mellon University
 1992-1994: M.S. in Computational Linguistics, Carnegie Mellon University
 1987-1994: Systems Engineer, Artificial Intelligence Technical Center, MITRE
 1988-1991: M.S. in Computer Science, George Mason University
 Summer 1987: Summer Linguistics Institute, Stanford University
 1981-1987: Senior Software Analyst, E-Systems
 1976-1980: B.S. in Computer Science, University of Virginia

Selected Publications:

Barbara Di Eugenio, Pamela Jordan, Richmond Thomason and Johanna Moore. The Agreement Process: An Empirical Investigation of Human-human Computer-Mediated Collaborative Dialogues, International Journal of Human-Computer Studies, to appear 2000.

Pamela Jordan. "An Empirical Study of the Communicative Goals Impacting Nominal Expressions", In the Proceedings of the ESSLLI workshop on The Generation of Nominal Expressions, 1999.

Barbara Di Eugenio, Pamela Jordan, Johanna Moore and Richmond Thomason. "An Empirical Investigation of Proposals in Collaborative Dialogues", In the Proceedings COLING-ACL'98. Montréal, Canada, 1998.

Pamela Jordan and Barbara Di Eugenio. "Control and Initiative in Collaborative Problem Solving Dialogues", In the proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative, Stanford, 1997.

Pamela Jordan and Richmond Thomason. "Refining the Categories of Miscommunication", In Proceedings of AAAI Workshop on Detecting, Repairing, and Preventing Human-machine Miscommunication, Portland, OR, August, 1996.

Pamela Jordan and Marilyn Walker. "Deciding to Remind During Collaborative Problem Solving: Empirical Evidence for Agent Strategies", In the proceedings of AAAI-96, 1996.

Pamela Jordan. "Using Terminological Knowledge Representation Languages to Manage Linguistic Resources", In proceedings of ACL96, Student Session, Santa Cruz, CA, 1996.

Marilyn Walker and Pamela Jordan. "Design-World: A testbed of communicative action and resource limits", In ACM SIGART Special Issue on Artificial Intelligence Education, Kumar & Hearst (editors), Vol 6, No. 2, 1995.

Pamela Jordan. "Determining the Temporal Ordering of Events in Discourse", Masters Thesis for CMU Computational Linguistics Program, 1994.

Kathryn Baker, Alexander Franz, and Pamela Jordan. "Coping with Ambiguity in Knowledge-based Natural Language Analysis", In the Proceedings of FLAIRS, May 1994.

Pamela Jordan, Bonnie Dorr and John Benoit. "A First-Pass Approach for Evaluating Machine Translation Systems", Machine Translation, Vol 8, 1993.

Pamela Jordan, Karl Keller, Richard Tucker, and David Vogel. "Software Storming: Combining Rapid Prototyping and Knowledge Engineering", Computer, Vol 22, No 5, May 1989.

VITA - Carolyn Penstein Rosé

Education:

Ph.D., Language and Information Technologies, Carnegie Mellon University, December 1997.

M.S., Computational Linguistics, Carnegie Mellon University, May, 1994.

B.S., Information and Computer Science (Magna Cum Laude), University of California at Irvine, June 1992.

Positions Held:

1997- *Research Associate, Learning Research and Development Center, University of Pittsburgh.*

1992-1997 *Research/Teaching Assistant, Language Technologies Institute, Carnegie Mellon University.*

1993 *Summer Research Internship, Apple Computer.*

Awards and Honors:

Carnegie Scholar Award, Carnegie Mellon University, 1994-1997.

Phi Beta Kappa, University of California at Irvine, 1991.

Golden Key National Honor Society, University of California at Irvine, 1991.

Simms Memorial Scholarship, University of California at Irvine, 1991-1992.

Selected Recent Publications:

Rosé, Carolyn P. and Lavie, Alon, A Domain Independent Approach for Efficiently Interpreting Extragrammatical Utterances, to appear in the Journal of Natural Language Engineering.

Rosé, Carolyn P. and Lavie, Alon, Balancing Robustness and Efficiency in Unification Augmented Context-Free Parsers for Large Practical Applications, to appear in J. C. Junqua and G. Van Noord (eds.) Robustness in Language and Speech Technologies, Kluwer Academic Press, in press.

Rosé, Carolyn P., Di Eugenio, Barbara, and Moore, Johanna D., A Dialogue Based Tutoring System for Basic Electricity and Electronics, Proceedings of AI in Education, 1999.

Rosé, Carolyn P., A Genetic Programming Approach for Robust Language Interpretation, in Spector, L., Langdon, W. B., O'Reilly, U.-M., and Angeline, P. (eds.), Advances in Genetic Programming 3, The MIT Press, 1999.

Rosé, Carolyn P., Robust Interactive Dialogue Interpretation, Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 1997.

VITA - Reva K. Freedman

Education

Ph.D., Computer Science, Northwestern University, December 1996.

Dissertation title: *Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System*

M.A., Computer Science, Northwestern University.

B.A., Mathematics, University of Chicago.

Professional Experience

1/98–present Research Associate, LRDC, University of Pittsburgh.

1/97–12/97 Research Associate, Illinois Institute of Technology.

Relevant Publications

Freedman, R. (1999). Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. AAAI-99 Workshop on Mixed-Initiative Intelligence, Orlando, FL.

Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). Delivering Hints in a Dialogue-Based Intelligent Tutoring System. Sixteenth National Conference on Artificial Intelligence (AAAI '99), Orlando, FL.

Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., and Evens, M. W. (1999). What Should the Tutor Do When the Student Cannot Answer a Question? Proceedings of the 12th Florida Artificial Intelligence Symposium (FLAIRS '99), Orlando, FL.

Freedman, R., Brandle, S., Glass, M., Kim, J. H., Zhou, Y. and Evens, M. W. (1998). Content Planning as the Basis for an Intelligent Tutoring System. Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-9), Niagara on the Lake, Canada, demo session.

Freedman, R. (1997). Degrees of Mixed-Initiative Interaction in an Intelligent Tutoring System. *1997 AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*.

Freedman, R. (1996). Using Tutoring Patterns to Generate More Cohesive Text in an Intelligent Tutoring System. *Proceedings of the International Conference on Learning Systems (ICLS '96)*.

Freedman, R. (1996). Using a Text Planner to Model the Behavior of Human Tutors in an ITS. In M. Gasser, ed., *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*. URL <http://www.cs.indiana.edu/event/maics96/Proceedings/Freedman/freedman.html> or freedman.ps.

Freedman, R. and M. W. Evens. (1996). Generating and Revising Hierarchical Multi-turn Text Plans in an ITS. In C. Frasson, G. Gauthier and A. Lesgold, eds. *Intelligent Tutoring Systems: Third International Conference (ITS '96)*, *Proceedings*. Berlin: Springer.