Proceedings of the Workshop on

Intelligent Tutoring Systems for Ill-Defined Domains

Held during the

8th International Conference on Intelligent Tutoring Systems

Tuesday, June 27
Jhongli, Taiwan

Vincent Aleven, Kevin Ashley, Collin Lynch and Niels Pinkwart
Editors

Preface

Intelligent Tutoring Systems have made great strides in recent years. Robust ITSs have been developed and deployed in arenas ranging from mathematics and physics to engineering and chemistry. Over the past decade intelligent tutoring systems have become increasingly accepted as viable learning tools in academia and industry. Most of the ITS research and development to this point has been done in well-defined domains. Well-defined domains are characterized by a basic formal theory or clear-cut domain model. Such domains are typically quantitative, and are often taught by human tutors using problems where answers can unambiguously be classified as correct or incorrect. Well-defined domains are particularly amenable to model-tracing tutoring systems. Operationalizing the domain theory makes it possible to identify study problems, provide a clear problem solving strategy, and assess results definitively based on the existence of unambiguous answers. Help can be readily provided by comparing the students' problem-solving steps to the existing domain models.

Not all domains of teaching and inquiry are well-defined; indeed most are not. Domains such as law, argumentation, history, art, medicine, and design are ill-defined. Often even well-defined domains are increasingly ill-defined at the edges where new knowledge is being discovered. Ill-defined domains lack well-defined models and formal theories that can be operationalized. Typically problems do not have clear and unambiguous solutions. For this reason ill-defined domains are often taught by human tutors using exploratory, collaborative, or Socratic instruction techniques.

Ill-defined domains present a number of unique challenges for researchers in Intelligent Tutoring Systems and Computer Modeling. These challenges include: 1) Defining a viable computational model for aspects of underspecified or open-ended domains; 2) Development of feasible strategies for search and inference in such domains; 3) Provision of feedback when the problem-solving model is not definitive; 4) Structuring of learning experiences in the absence of a clear problem, strategy, and answer; 5) User models that accommodate the uncertainty of ill-defined domains; and 6) User interface design for ITSs in ill-defined domains where usually the learner needs to be creative in his actions, but the system still has to be able to analyze them. These challenges must be faced if the ITS community is ever to branch out from the traditional domains into newer arenas.

This volume constitutes the proceedings of the workshop on Intelligent Tutoring Systems for Ill-Defined Domains, held in conjunction with the Eighth International Conference on Intelligent Tutoring Systems (ITS 2006) in Jhongli, Taiwan. The papers contained in this volume demonstrate how researchers have begun to develop intelligent tutoring systems for ill-defined domains, addressing the challenges mentioned above. In addition to one overview paper, which surveys the current state of the art in the still nascent field of ITSs for ill-defined domains, this volume contains 10 research papers, which present work in a variety of domains. Some of these, like ethics, intercultural competence, legal argumentation, language learning, and creative group problem solving, demonstrate the potential of the new and largely unexplored fields for intelligent tutoring systems. Other papers deal with more classical domains such as programming languages and database modeling, but illustrate how even these are ill-defined when it comes to more creative and design-oriented tasks.

Apart from the different domains described in the papers of this volume, the methods and tutoring approaches also vary. Some papers show attempts to adapt paradigms that have been successful in well-defined domains such as model-tracing tutors or constraint-based approaches, and approaches aimed at supporting self-explanation. Others show new methods for building ITSs in ill-defined fields, including ontology-based feedback, collaborative methods, and peer reviews.

We thank the members of the workshop organizing committee for providing thoughtful reviews of the paper submissions for this workshop. Last but not least, we thank the participants of the workshop for contributing their ideas and research results.

May, 2006

Vincent Aleven
Kevin Ashley
Collin Lynch
Niels Pinkwart

Organizing Committee

Vincent Aleven, Carnegie Mellon University, USA
Jerry Andriessen, University of Utrecht, The Netherlands
Kevin Ashley, University of Pittsburgh, USA
Michael Baker, Centre National de la Recherche Scientifique, France
Paul Brna, University of Glasgow, UK
Robin Burke, DePaul University, USA
Jill Burstein, Educational Testing Service, USA
Rebecca Crowley, University of Pittsburgh, USA
Susanne Lajoie, McGill University, Canada
Collin Lynch, University of Pittsburgh, USA
Liz Masterman, Oxford University, UK
Bruce McLaren, Carnegie Mellon University, USA
Antoinette Muntjewerff, University of Amsterdam, The Netherlands
Katsumi Nitta, Tokyo Institute of Technology, Japan
Niels Pinkwart, Carnegie Mellon University, USA
Beverly Woolf, University of Massachusetts, USA

Table of Contents

Defining "Ill-Defined Domains"; A literature survey.

Collin F. Lynch
& Kevin D. Ashley
Intelligent Systems Program,
University of Pittsburgh.
Pittsburgh, Pennsylvania.
(collinl@cs.pitt.edu)
(ashley@pitt.edu)

Vincent Aleven
& Niels Pinkwart
Human Computer Interaction Institute,
Carnegie Mellon University.
Pittsburgh, Pennsylvania.
(aleven@cs.cmu.edu)
(nielsp@cs.cmu.edu)

Abstract: In order to make progress on Intelligent Tutoring in ill-defined domains it is helpful to start with a definition. In this paper we consider the existing definitions and select one for the basis of our discussion. We then summarize some of the more salient characteristics of ill-defined domains from a tutoring standpoint and some human and ITS strategies that have been employed to cope with them. We conclude with some challenges to the ITS community to spur further research.

Keywords: ITS 2006 workshops, intelligent tutoring systems, ill-structured domains, ill-defined domains

INTRO

Intelligent Tutoring Systems (ITS) have made great strides in recent years. Many of these gains have been made in well-defined domains such as geometry, Newtonian Mechanics, and system maintenance. In recent years similar gains have been made in ill-defined domains such as law, design, and composition.

In this paper we present an overview of ITSs for ill-defined domains. Our goal is to provide a framework for discussion and research in this area by synthesizing past efforts. We begin by providing a working definition of the term. We then highlight the relevant characteristics of ill-defined domains as they pertain to tutoring. Following this, we summarize past research in this area in terms of some of the human and ITS tutoring strategies that have proven successful.

In order to frame our discussion we will illustrate our points using the following domains:

Physics: What is the airspeed velocity of an unladen swallow? [1]

Ethics: Is it morally justified to raise swallows for food?

Law: If I raise swallows on my property am I liable for damage to my neighbors' property?

Architecture: Design a residential building with housing for swallows.

Music Composition: Write a fugue based upon a swallow's song.

This is not an exclusive list. Rather it is a representative sample meant to motivate further discussion.

For the purposes of this workshop we have chosen the term *ill-defined domain*. The terms *ill-structured* and *ill-defined* are used interchangeably in the literature. To avoid confusion we will only use the former. Much of AI and education literature is framed in terms of *problems* rather than *domains*. *Problem* typically connotes achieving a specific goal in a concrete scenario using methods amenable to state-space search. *Domain* typically connotes an area of study such as physics or a set of problems. For the purposes of our analysis, this distinction is immaterial. We have chosen *'domain'* in order to emphasize that the end goal of tutoring is typically general domain knowledge or problem solving skills, not problem-specific answers. At the same time, we recognize that domains like physics have both well-defined and ill-defined subdomains.

[1] See: "Monty Python and the Holy Grail" (1975) *Python et al.*

DEFINITION

In order to have a serious discussion of ill-defined domains it is necessary to define the term. This will be problematic as the term has been given a wide variety of definitions in the literature. Much of the literature on this subject grows out of either AI, or Decision-Making under Uncertainty. We will begin by discussing the historical AI-centric definition followed by a discussion of competing theories.

Ill-defined domains have a long history in AI. The earliest work by John McCarthy is referenced in Marvin Minsky's 1961 paper "Steps Toward Artificial Intelligence" [Minsky, 1995]. According to McCarthy and Minsky, a *well-defined domain* is one in which there exists a systematic way to determine when a proposed solution is acceptable. *Ill-defined domains*, by definition, lack such a procedure. This residual definition underpins most of the subsequent work in this area.

Four subsequent definitions have influenced our analysis. They vary in style and meaning based upon the questions that their authors were asked. Reitman [Reitman, 1964] followed Minsky's definition but sought to impose some structure on the implicit concept of ill-definedness to facilitate more serious research. His work was based upon the composition of fugues by an expert composer. Among other things, he noted that the sole requirement of this task was that the result "be a fugue."

Newell [Newell, 1969] asked why a given domain might appear ill-defined to one problem solver and well-defined to another. His definition was framed in terms of a solver's ability to identify a "specific" answer. Simon [Simon, 1973] by contrast sought to identify why ill-defined domains were *not* amenable to state-space search.

Simon's discussion was built upon his past work with the General Problem Solver [Newell and Simon, 1995]. He used this framework to consider classically ill-defined domains such as architecture. In this process he highlighted some of the salient characteristics of such domains including the lack of a clean decomposition and the relationship of scope to definedness. In chess, for example, the selection of a single move may be well-defined while winning an entire game is not. This analysis led to similar work in other domains such as design [Goel and Pirolli, 1992] and medicine [Pople, 1982].

This line of reasoning has found its way into the psychological literature. Voss and Post's [Voss and Post, 1988], paper considers several varying definitions of ill-definedness. These include the work of Johnson [Johnson, 1988], and Lawrence [Lawrence, 1988]. Johnsons' work drew on notions of ill-definedness in the expert problem-solving literature while Lawrence chose the domain of law. Voss and Post based their working definition on Reitman's [Reitman, 1964]. It focused on the "open-constraints" present in ill-defined problems and domains. They specifically emphasized the constraint-propagation aspects of ill-defined problem solving. We will return to this aspect below.

Most recently, Ashley and Pinkus [Ashley et al., 2004] defined ill-defined domains as having the following key characteristics: 1) they lack a definitive answer; 2) the answer is heavily dependent upon the problem's conception; and 3) problem solving requires both retrieving relevant concepts and mapping them to the task at hand. In identifying these characteristics, their goal was to motivate the development of an ITS for applied ethics. Their analysis, like ours, is driven by a set of indicative examples.

Our goal in this paper is to present a summary of ITS research in ill-defined domains with a focus on the characteristics of those domains that affect ITS design. For this reason we have chosen to follow Ashley and Pinkus' methodology in this paper.

RELEVANT CHARACTERISTICS

In this section we summarize five key characteristics of ill-defined domains that have been discussed in the literature. We give an intuitive description of each one highlighted by examples from the sample domains. This list is not intended to be exhaustive but to serve as a basis for subsequent discussion.

3.1 Verifiability

"What is the airspeed velocity of an unladen swallow?" This is a classical physics problem (according to Monty Python) that could be solved by applying the theory of Newtonian Mechanics. The answer can be calculated programmatically and verified empirically to an arbitrary degree of precision. This is not the case with domains such as law. Legal arguments may be judged functionally (win or lose) or aesthetically (good or bad) but no unambiguous standard exists. While there are valid arguments for or against some solutions there often is no one *right* answer.

Domains like architecture and music are even less verifiable. While arguments may be made for or against a given instance, such arguments are necessarily qualitative. Brolin's argument against modern

architecture [Brolin, 1976], while convincing, is based entirely on aesthetic value judgments, not on any absolute or quantitative measurements. Ultimately, in such domains, one man's masterpiece is another man's trash.

3.2 Formal Theories

Valid formal theories such as Newtonian Mechanics provide a means to determine a problem's outcome and test its validity. A formal theory in physics is considered valid if its predictions can be verified empirically, that is, it accurately describes all relevant phenomena. Physicists are engaged in a continuous process of theory formation. Their ultimate goal is to develop a single cohesive theory that explains all physical phenomena.

This process becomes difficult in new regions of physics (e.g., astrophysics) where empirical verification may be untenable. A number of competing theories may coexist and, while they stem from testable phenomena, they may not be readily falsified. In the realm of scientific discovery, argumentation may involve a degree of interpretation that approaches that in domains like law.

Lawyers, like physicists, are also engaged in a continuous process of domain structuring ([Levi, 1949], [Radin, 1933], [Schauer, 1998]). The goal however, is typically prescriptive, not descriptive ([Schauer, 1998], [Llewellyn, 1981]). More formal legal theories are typically based on statutes or case decisions and exist to prescribe what *should* be done in a specific case, not necessarily what *is*. Such theories are relatively specific and may not be expected to generalize across legal fields. While such prescriptions are normatively based, the norms may change over time. Formal attempts to model the structure of the law are usually in flux [Lehmann et al., 2005].

Physicists seek out formal theories. Lawyers invent such theories as needed but acknowledge their limitations. Architects by contrast typically shun such theories as being overly restrictive. While such theories may be accepted (e.g. [Alexander, 1977], [Alexander, 1979]), they are typically used to guide intuitions, not to dictate results [Goel and Pirolli, 1992].

3.3 Task Structure

Physics is largely a descriptive domain. Most textbook physics problems are similar to the swallow problem above. Given some information, compute a desired quantity using a formal theory. Research physicists, by contrast, seek to formulate new theories that explain observed phenomena or to observe phenomena that may be used to falsify existing theories. Both tasks involve elements of design and as such are necessarily ill-defined.

Law is both an analytical and design domain. Legal analysis includes determining what laws or theories are applicable to the current situation and what result they would prescribe ([Sergot et al., 1986], [Radin, 1933], [Schauer, 1998]). Such tasks are necessarily ill-defined much like medical diagnostics [Pople, 1982]. Legal design tasks include the formation of arguments that would necessitate or evade such analysis to achieve their desired goal. The study of architecture, by contrast, is characterized primarily by design tasks. In such cases novelty, not repetition, is the goal. While formal theories might be used to teach or guide intuitions, practitioners typically shun such programmatic analyses.

3.4 Open-Textured Concepts

Open-textured concepts are abstract concepts such as "vehicle" and "space" that have an inherent indeterminacy [Gardner, 1987] and lack any absolute definition. Such concepts are a defining characteristic of legal theories ([Ashley, 1990], [Berman and Hafner, 1985], [Sergot et al., 1986]) and architectural theories like the Pattern Language. They become problematic when they must be applied to concrete elements.

In many ways this is also true of physics. While physicists seek to describe real-world phenomena, physical theories still make use of concepts such as "time points" and "energy". While these concepts are defined within the physical theories, their applicability to new phenomena (e.g., black holes) is often a matter of debate. As with law and architecture, the application of a theory depends on the definition of its terms. This issue has been discussed by [Reitman, 1964], [Simon, 1973], and [Ashley et al., 2004].

Computational models of these domains typically handle open-textured concepts in one of four ways: impose a definition arbitrarily; require that the user provide one; reason from case examples; or ignore the issue entirely. In the last option, they are focusing on the theory alone. Each alternative constrains the system in some way either to a limited well-defined domain or an ad-hoc set of cases.

3.5 Overlapping Subproblems

Human and non-human problem solvers typically decompose a given problem into separate subproblems ([Goel and Pirolli, 1992], [Pople, 1982], [Newell and Simon, 1995]). Problems in ill-defined domains however, do not necessarily decompose into subproblems that are independent or easier to solve [Simon, 1973].

The swallow problem, for example, may be solved using backward chaining ([VanLehn et al., 2005], [VanLehn et al., 2004]). Given the sought quantity, we may identify a principle from Newtonian Mechanics that defines it in terms of other quantities such as the mass of the swallow and its acceleration. Each of these quantities may be solved independently of the others. While their values are related, it is unimportant how they are identified so long as it is done accurately.

Now consider the problem of designing a house for the swallow. We could divide it into subproblems such as choosing a site, or determining the size of the house to be built. These problems, however, are not independent. Selecting a given site limits us to houses that will fit on it. Selecting a given size in turn limits the sites we may choose. The answer to one subproblem necessarily constrains the other, and neither one may be solved without considering the effect it has on the other.

Several authors ([Pople, 1982] [Goel and Pirolli, 1992], [Simon, 1973], [Reitman, 1964]) have noted that human problem solvers often cope with this by solving the subproblems in parallel. As each problem-solving step is taken, it is evaluated both in terms of the current subproblem and the constraints it imposes on the others. Unlike the independent decomposition of well-defined problems this decomposition into interrelated subproblems does not reduce the complexity of the overall problem-solving process. Nevertheless such behavior is important in domains such as writing, as illustrated here, where the paper is organized into sections with each one framing and constraining the contents of the next.

HUMAN TUTORING STRATEGIES

Students in well-defined domains are commonly trained using batteries of practice problems followed by tests which are solved by and checked against a formal theory. While there has been some effort to move away from this approach [Callahan and Hoffman, 1995] it is still common practice. In this section we briefly summarize several strategies employed by human tutors in ill-defined domains.

4.1 Case Studies

Engineering students who study applied ethics routinely study real cases faced by engineering practitioners ([Ashley et al., 2004], [Harris et al., 1995]). Law students study how legal rules have been applied in past decisions, briefs, and arguments ([Llewellyn, 1981], [Aleven et al., 2005] [Ashley et al., 2005]). Architecture students likewise examine real buildings to identify both successes and failures ([Brolin, 1976], [Mullet and Sano, 1995]). Such examples can often highlight the constraints and nuances of an ill-defined domain far better than any abstract model. They also provide the students with analogies on which to base their own work [Llewellyn, 1981]. Williams [Williams, 1992] presents a conceptual analysis of this style of learning in both the legal and medical domains.

Examples are also used for instruction in well-defined domains. However those cases are unambiguously correct or incorrect and do not require any interpretation. By contrast, in ill-defined domains, the process of interpreting case examples is often carried out by means of a Socratic dialogue where students are guided by the instructor's questions.

This process of interpretation is often carried out by means of a Socratic dialogue where students are guided by the instructor's questions.

4.2 Weak Theory Scaffolding

Architecture students at the University of Oregon are directed to read *A Pattern Language* and *The Timeless Way of Building* ([Alexander, 1977] [Alexander, 1979]). They then describe existing buildings in terms of the language and design new ones according to its precepts. They do so with the knowledge that the theory itself is incomplete. As they advance, the theory's restrictions are faded out until students need not follow them at all.

This process is advantageous in that it gives the students a conceptual framework to structure the ill-defined domains. It enables them to ignore some challenges, while tackling others. In this way the domain is made more tractable without being entirely constrained. Similar strategies have also been employed in well-defined domains such as physics and probability [VanLehn et al., 2004].

4.3 Expert Review

Students in well-defined domains such as physics may test their results against a formal theory. In ill-defined domains no absolute theory exists. Students in these domains typically submit their work to domain experts for comment. Law students routinely engage in "moot court" sessions where they present arguments in a real case before judges or law professors, who critique the sessions and provide the students with immediate feedback. Students in architecture likewise typically submit their designs to faculty panels for review.

4.4 Peer Review/Collaboration

The "Crit" is a longstanding practice in architecture education. Like "brown-bag talks" in research fields, it is a chance for students to learn from their peers. During a crit, students present finished work to their peers for comments. While their peers may not be experts in the field, they have unique perspectives on the problem and can often provide novel feedback. While such advice may not be objectively verifiable, experience has shown that it is often quite valuable.

ITS STRATEGIES

In this section we will describe some ITS techniques that have been employed in ill-defined domains. We provide representative examples of systems employing each strategy. Although some of the strategies lack any apparent "artificial intelligence", they have proven successful in past studies. As Burke and Kass astutely noted, it is possible for a relatively ignorant system to teach complex ideas:

> "It is important to keep in mind that the domain knowledge, which must be conveyed to the student, does *not* have to be the same knowledge base that the system uses to diagnose the student's misconceptions. It is possible to build teaching systems that can effectively convey domain knowledge that is much richer and more complex than the system itself can understand. It is crucial to build upon this insight if intelligent tutoring is going to move beyond the easily-formalized domains, such as arithmetic" [Burke and Kass, 1996].

5.1 Model-Based

In a model-based ITS, instruction is based upon ideal solution models. These models represent one or more acceptable solutions to a given problem, or a general model of the domain as a whole. The model is used both to check the students' actions and provide help. Model tracing tutors may be loosely classified as either *strong* or *weak*. Strong tutors force the students to follow the model exactly so that each entry matches it in some way. The Andes ITS for physics uses this methodology. Each problem is represented by a solution model that contains all correct entries. Every student entry must be represented in the model in order to be accepted. Such systems are typically called model-tracing tutors. Weak methods use the model as a guide but do not require strict adherence to its contents.

Strong model-based tutors have proven successful in well-defined domains such as physics [VanLehn et al., 2005]. They have not yet achieved the same success in ill-defined domains. Doing so would require formalizing some model of the domain such as an analogous game [Allen and Saxon, 1998] or a subdomain model ([Ashley, 2000], [Aleven, 2003], [Gardner, 1987]). Such a tutor would then be operating in a comparatively well-defined subset of the domain. As Burke and Kass suggest, it still may be possible for that tutor to teach skills in the subset in a way that leaves open key ill-defined elements so as not to mislead students.

Such 'weaker' model-based approaches have achieved some success in ill-defined domains. The CATO system ([Aleven, 2003], [Ashley, 2000]) used a model of the domain to relate real cases and argument models to present to students. PETE [Goldin et al., 2001] uses a weak domain model to teach engineering ethics. While neither of these are model-tracing systems, they do make use of formal models as guides and show that such models may be useful in ill-defined domains.

5.2 Constraints

While model-tracing systems are based upon a complete solution or domain model, constraint-based systems are built from sets of constraints. These constraints specify what characteristics a solution should,

or should not have. These requirements may be sufficient to provide a complete solution specification or only a partial description. Constraints may be classified as either *strong* or *weak*. Strong constraints represent absolute requirements or prohibitions. Weak constraints represent preferences, or warnings. Unlike model-based systems, these constraint-based systems are not based upon complete, or necessarily consistent models of the domain in question.

Strong constraints have been employed in music [Holland, 1999] and [Brandao, 2005]. These systems permit students to define unique musical combinations but prevent them from violating well-accepted rules of harmony and tone. This work echoed Reitman's intuition [Reitman, 1964] that, while it is difficult to tell what *is* a fugue, it is often easy to tell what *is not*.

Weak constraints have been employed in ill-defined domains but to a lesser degree. We have begun work on a legal mark-up system that employs weak constraints to guide students [Pinkwart et al., 2006]. While students are largely free to specify the relationships they wish, the system will coach them on "optimal" choices using a set of solution preferences. Suthers [Suthers, 1998] used a similar approach but based his constraints upon expert solutions.

Simon [Simon, 1973], Reitman [Reitman, 1964], and Goel [Goel and Pirolli, 1992] all speak of problem solvers propagating constraints as they work. This process allows them to bound and narrow the search space to manageable dimensions. This view mirrors Alexander's comments in *A Pattern Language* and *The Timeless Way of Building* ([Alexander, 1977] [Alexander, 1979]) that individual patterns such as appropriate housing locations necessarily constrain the design of housing. The Pattern Language in effect introduces a set of constraints that a practitioner may choose to follow. As the practitioner makes decisions those decisions "activate" an individual pattern and propagate its constraints for future decisions. This similarity may be an indication that constraint-based tutoring is pedagogically more appropriate for ill-defined domains than model-tracing methods. The question remains whether a constraint-based tutor can provide all of the feedback that is appropriate.

5.3 Discovery Learning

Instruction in many domains consists of reifying or formalizing domain knowledge and then transmitting it explicitly via texts or lectures. Researchers like Seymour Papert have long argued for a more constructivist approach. This approach is variously known as LOGO ([Resnick, 1998], [Holland, 1999]), Discovery Learning ([Veermans et al., 2006], [van Joolingen, 1999], [de Jong and van Joolingen, 1998] [Veermans and van Joolingen, 2004]), or Discovery Microworlds [Trafton and Trickett, 2001]. For the purposes of this paper we will group these under the heading of "Discovery Learning." De Jong [de Jong and van Joolingen, 1998] and van Joolingen [van Joolingen, 1999] have provided some useful comparative analyses of discovery learning. Their comparisons indicate that, although the method can be successful, students may have difficulties with the methodology nullifying their learning gains.

Discovery Learning approaches can be classified into three general tracks: discovery support; model exploration; and model building. *Discovery Support* systems operate by providing the user with support as they work on a task in an unconstrained domain. Such systems do not attempt to model the domain itself. Rather they seek to assist the students in exploring the domain by providing intelligent support. The HYPO system provides users with intelligent case-retrieval tools to help in issue spotting [Ashley, 1990]. Other systems have focused on providing intelligent suggestions for data mining tasks [Bernstein et al., 2005] and hypothesis support for basic science education [Suthers, 1998]. In general, any decision support system could qualify as a discovery support model so long as it helped the users better understand the domain and did not supplant their own actions.

Model Exploration systems also focus on helping students to explore a domain. Unlike discovery support systems the domain in question is represented by a formal model. Here students interact with a model of the domain rather than conducting real-world searches or experiments, drawing conclusions from it that are appropriate to the real domain. This technique has been used in some well-defined subdomains of physics ([Veermans et al., 2006], [Veermans and van Joolingen, 2004], [Trafton and Trickett, 2001]). Apart from some work in music [Holland, 1999], it has yet to be applied to any ill-defined domains, due in large part to the challenges of modeling such domains.

Model Building by contrast focuses on the development of domain models. Model building systems supply users with a suite of model building tools such as a formal development language and (optionally) some support in their use. The most prevalent such tools are descendants of Papert's original LOGO programming language such as StarLogo [Resnick, 1998] and Music Logo [Holland, 1999]. Recent work by Yoshino and Sakurai has extended this principle to law via a logic-programming lan-

guage [Yoshino and Sakurai, 2005]. Model building has a long history in psychological research including the study of problem-solving in ill-defined domains [Voss and Post, 1988].

By providing an open-ended basis for experimentation, these systems enable users to test their own intuitions about a domain and to perform arbitrary experiments. In so doing it is believed that students will better understand nuances of the domain and the challenges of modeling it that a premade system would hide. LOGO-like languages are also easier to develop than full-scale domain simulations.

Of these three methodologies, Model-Building and Discovery Support seem to hold the most promise for ill-defined domains. Both follow Burke and Kass' intuition above and permit the user to explore the nuances of a domain without limiting them or hiding the relevant aspects.

5.4 Case Analysis

As stated above, the examination of past cases is a primary teaching tool in ill-defined domains such as law, architecture and music. Systems that facilitate this process by providing cases, highlighting their relationships or otherwise facilitating analysis have long been a part of ITS research.

Ashley's HYPO [Ashley, 1990] and it's descendant CATO [Aleven and Ashley, 1996] were built on a relatively formal, though nondeterministic, model of trade secret law. This model was used to encode real cases in terms of their relevant legal factors. A hierarchy of factors was then used to categorize cases and present them to the students as they developed an argument structure. The CATO-Dial system [Ashley et al., 2002] built upon this to engage the students in court-like arguments.

Burke and Kass [Burke and Kass, 1996] also developed an automatic case-retrieval system to select cases in order to present feedback based upon the most recent student action. While their domain model was quite limited, the use of real cases enabled them to give more nuanced feedback than the system could otherwise describe.

Relatively simple techniques such as self-explanation prompts ([Schworm and Renkl, 2002], [Aleven et al., 2005], [Pinkwart et al., 2006]) have also been shown to boost student performance on case-analysis tasks. We believe that these techniques may be fruitfully combined with case retrieval systems such as those above to improve learning gains.

5.5 Collaboration

Education researchers such as Vygotsky have long argued that social support for learning is as or more important than instructional factors ([Chan and Baskin, 1990] [Chan et al., 1993]). Engaging students with similarly-situated peers can encourage their own learning and that of their peers. Using Crit-like techniques in design tasks, can give students the benefit of alternate, sometimes intensely critical, perspectives. Existing collaborative ITSs have provided this support either by casting the system as collaborator, or by using the system to facilitate interactions among human peers.

The former method has been championed by Tak-Wai Chan ([Chan and Baskin, 1990] [Chan et al., 1993]) and his colleagues. In this approach the system is equipped with a user model and cast as a student alongside the real student. The virtual student is designed to be as mistake-prone as its human peers and to learn alongside them. The challenge lies in developing a rational student model for the virtual student. While this may be easier than an expert model, it is just as critical. An irrational or unbelievable student model may, over time, cancel out any purported benefits. Although this approach has yet to be applied to ill-defined domains, it seems to us that the approach is applicable in that it resembles Socratic Dialogue.

The second approach, while relatively AI-free, has gained acceptance in recent years. The role of the system is to facilitate user interaction and leverage the users' collective knowledge for learning. This approach follows Burke and Kass' intuition that a relatively ignorant system can nevertheless support learning gains. Soller et al. [Soller et al., 2005] provides a nice summary of the state of the art in this area. In recent years systems have been developed along these lines for writing [Cho and Schunn, 2005], collaborative discovery [Suthers, 1998], linguistics [Weisler et al., 2001], and mediation [Tanaka et al., 2005]. Our own group has begun working on just such a system for legal argument formation and analysis [Pinkwart et al., 2006].

CONCLUSION

Given the distinct characteristics, teaching techniques, and ITS strategies, discussed above, it is our opinion that there are substantive differences between well-defined and ill-defined domains. Those differences

should be taken into account by researchers in this area. Having said that, we believe that the future is promising. Research in this arena has already begun to establish a solid (though ill-defined) foundation for future work. Given the nature of this arena it seems inappropriate to draw any "final" conclusions at this time. Rather we will close with two key challenges.

Firstly, the community should continue to develop new tutoring strategies and seek out ways to combine existing strategies. In our opinion, hybrid systems hold the most promise for ITSs in ill-defined domains, especially those that build upon Burke and Kass' intuitions.

Secondly, and most importantly, we urge the community to consider focusing across domains. Ill-defined domains share many common characteristics such as open-textured concepts and a lack of absolute verification. Each individual domain such as law or architecture may possess these characteristics to a different degree. Moreover, such domains are themselves quite amorphous. They contain many distinct "tasks" or subdomains, such as the formation of legal arguments, and the deciding of legal cases, that each have their own characteristics and requirements.

In our opinion, asking what ITS strategies are appropriate for "the law" is too narrow a question. Rather, researchers should focus on tutoring strategies that are appropriate for general characteristics such as overlapping subproblems and open-textured concepts. It is our belief that framing research questions in this way will lend itself to the development of strategies that can be "ported" across ill-defined and even well-defined domains. Of course, insights can be gained by looking at particular domains and tasks, but the true significance of these insights can best be appreciated by looking at characteristics that cross domains.

We do not expect this to be the last word on the subject. There are other challenges facing the ill-defined ITS community. Rather we hope that this survey will guide strategies that can spur research in new directions.

REFERENCES

[Aleven, 2003] Aleven, V. (2003).Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment.*AI*, 150:183–237.

[Aleven and Ashley, 1996] Aleven, V. and Ashley, K. D. (1996).How different is different? arguing about the significance of similarities and differences.In Smith, I. and Faltings, B., editors, *Advances in CBR: Proc. of the 3rd European Workshop, EWCBR-96, LNAI*, pages 1–15. Springer Verlag: Berlin.

[Aleven et al., 2005] Aleven, V., Ashley, K. D., and Lynch, C. (2005).Helping law students to understand supreme court oral arguments: A planned experiment.In Sartor, G., editor, *ICAIL*, pages 55–59. ACM, New York.

[Alexander, 1977] Alexander, C. (1977).*A Pattern Language*.Oxford University Press; US.

[Alexander, 1979] Alexander, C. (1979).*The Timeless Way of Building*.Oxford University Press; US.

[Allen and Saxon, 1998] Allen, L. and Saxon, C. S. (1998).The legal argument game of legal relations.*E-Law - Murdoch U. Electronic Journal of Law*, 5(3).

[Ashley, 1990] Ashley, K. D. (1990).*Modeling Legal Argument: Reasoning With Cases and Hypotheticals*.The Bradford Books, MIT Press.

[Ashley, 2000] Ashley, K. D. (2000).Designing electronic casebooks that talk back: The cato program.*Jurimetrics*, 40(3):275–319.

[Ashley et al., 2005] Ashley, K. D., Aleven, V., and Lynch, C. (2005).Teaching creative legal reasoning with examples from supreme court oral arguments.In Yoshino, H., Ashley, K. D., and Nitta, K., editors, *AILE; Proc. of the ICAIL-05 Workshop*.

[Ashley et al., 2004] Ashley, K. D., Chi, M., Pinkus, R., and Moore, J. D. (2004).Modeling learning to reason with cases in engineering ethics: A test domain for intelligent assistance. NSF Proposal.

[Ashley et al., 2002] Ashley, K. D., Desai, R., and Levine, J. M. (2002).Teaching case-based argumentation concepts using dialectic arguments vs. didactic explanations.In Cerri, S. A., Gouardères, G., and Paraguaçu, F., editors, *ITS*, volume 2363 of *LNCS*, pages 585–595. Springer.

[Berman and Hafner, 1985] Berman, D. and Hafner, C. (1985).Obstacles to the development of logic-based models of legal reasoning.In Walter, C. and Allen, L. E., editors, *Computing Power & Legal Reasoning*. West Pub Co. St Paul.

[Bernstein et al., 2005] Bernstein, A., Hill, S., and Provost, F. (2005).Intelligent assistance for the data mining process; an ontology-based approach.*IEEE Transactions on Knowledge and Data Engineering*, 17(4):503–518.

[Brandao, 2005] Brandao, M. (2005).Guided discovery tutoring and schoenberg's harmony teaching method: an investigation.

[Brolin, 1976] Brolin, B. C. (1976).*The Failure of Modern Architecture*.Van Nostrand Reinhold Company.

[Burke and Kass, 1996] Burke, R. D. and Kass, A. (1996).Interest-focused tutoring: A tractable approach to modeling in intelligent tutoring systems.Technical Report TR-96-08.

[Callahan and Hoffman, 1995] Callahan, J. and Hoffman, K. (1995).*Calculus In Context*.W.H. Freeman and Co. New York.

[Chan and Baskin, 1990] Chan, T. W. and Baskin, A. B. (1990).Learning companion systems.In Frasson, C. and Gauthier, G., editors, *ITS*. Ablex, New Jersey.

[Chan et al., 1993] Chan, T. W., Lin, C. C., Lin, S. J., and Kuo, H. C. (1993).Octr: A model of learning stages.In *Proc. of AI-Ed, Edinburgh U.K.*, pages 257–264.

[Cho and Schunn, 2005] Cho, K. and Schunn, C. D. (2005).Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system.*Computers & Education (In Press)*.

[de Jong and van Joolingen, 1998] de Jong, T. and van Joolingen, W. (1998).Scientific discovery learning with computer simulations of conceptual domains.*Review of Educational Research*, 68(2):179–201.

[Gardner, 1987] Gardner, A. V. D. L. (1987).*An Artificial Intelligence Approach To Legal Reasoning.*The Bradford Books, MIT Press.

[Goel and Pirolli, 1992] Goel, V. and Pirolli, P. (1992).The structure of design problem spaces.*Cognitive Sci.*, 16:345–429.

[Goldin et al., 2001] Goldin, I. M., Ashley, K. D., and Pinkus, R. L. (2001).Introducing PETE: computer support for teaching ethics.In *ICAIL*, pages 94–98.

[Harris et al., 1995] Harris, C. E., Pritchard, M. S., and Rabins, M. J. (1995).*Engineering Ethics*.Wadsworth Publishing Co, Boston.

[Holland, 1999] Holland, S. (1999).Artificial intelligence in music education: A critical review.*Readings in Music and AI; Contemporary Music Studies*, 20.

[Johnson, 1988] Johnson, E. J. (1988).*Expertise and Decision Under Uncertainty: Performance and Process*, chapter 7.Lawerence Erlbaum Associates: Hillsdale New Jersey.

[Lawrence, 1988] Lawrence, J. A. (1988).*Expertise on the Bench: Modeling Magistrates' Judicial Decisionmaking*, chapter 8.Lawerence Erlbaum Associates: Hillsdale New Jersey.

[Lehmann et al., 2005] Lehmann, J., Biasiotti, M. A., Francesconi, E., and Sagri, M. T., editors (2005).*LOAIT: Proc. of the ICAIL-05 Workshop*, IAAIL Workshop Series. Wolf Legal Publishers.

[Lester et al., 2004] Lester, J. C., Vicari, R. M., and Paraguaçu, F., editors (2004).*Intelligent Tutoring Systems, 7th International Conference, ITS 2004, Maceiò, Alagoas, Brazil, August 30 - September 3, 2004, Proceedings*, volume 3220 of *LNCS*. Springer.

[Levi, 1949] Levi, E. H. (1949).*An Introduction to Legal Reasoning.*U. Chicago Press; Chicago.

[Llewellyn, 1981] Llewellyn, K. N. (1981).*The Bramble Bush; On Our Law and its Study.*Oceana Publications Inc. Dobbs Ferry, New York.

[Minsky, 1995] Minsky, M. (1995).Steps to artificial intelligence.In Luger, G. F., editor, *Computation and Intelligence, Collected Readings.*, chapter 3, pages 47–90. AAAI, Menlo Park CA, and MIT Press.

[Mullet and Sano, 1995] Mullet, K. and Sano, D. (1995).*Designing Visual Interfaces*.Sun Microsystems Inc. Mountain View California.

[Newell, 1969] Newell, A. (1969).Heuristic programming: Ill-structured problems.*Progress in Operations Research*, 3:361–413.

[Newell and Simon, 1995] Newell, A. and Simon, H. A. (1995).Gps, a program that simulates human thought.In Luger, G. F., editor, *Computation and Intelligence, Collected Readings.*, chapter 16, pages 415–428. AAAI, and MIT Press.

[Pinkwart et al., 2006] Pinkwart, N., Aleven, V., Ashley, K. D., and Lynch, C. (2006).Toward legal argument instruction with graph grammars and collaborative filtering techniques.In Ashley, K. D. and Ikeda, M., editors, *ITS06, Proc. of the 2006 Conference on ITS (In-Press)*.

[Pople, 1982] Pople, H. E. (1982).*Heuristic Methods for Imposing Structure on Ill-Structured Problems: The Structuring of Medical Diagnostics*, chapter 5.Westview Press: Boulder Colorado.

[Radin, 1933] Radin, M. (1933).Case law and stare decisis concerning präjudizienrecht in amerika.*Columbia Law Review*, 27(2):199–212.

[Reitman, 1964] Reitman, W. R. (1964).*Heuristic Decision Procedures Open Constraints and the Structure of Ill-Defined Problems*, chapter 15, pages 282–315.John Wiley & Sons Inc. New York.

[Resnick, 1998] Resnick, M. (1998).*Learning About Life*, chapter 11, pages 229–241.MIT Press.

[Schauer, 1998] Schauer, F. (1998).Prediction and particularity.*Boston University Law Review*.

[Schworm and Renkl, 2002] Schworm, S. and Renkl, A. (2002).Learning by solved example problems: Instructional explanations reduce self explanation activity.In *Proc. of the 24th Annual Conf. of the Cognitive Science Society, Mahwah NJ*, pages 816–821. Lawrence Erlbaum.

[Sergot et al., 1986] Sergot, M. J., Sadri, F., Kowalski, R. A., Kriwaczek, F., Hammond, P., and Cory, H. T. (1986).The british nationality act as a logic program.*Comm. ACM*, 29(5):370–386.

[Simon, 1973] Simon, H. A. (1973).The structure of ill-structured problems.*AI*, 4:181–201.

[Soller et al., 2005] Soller, A., Martinez, A., Jermann, P., and Muehlenbrock, M. (2005).From mirroring to guiding: A review of state of the art technology for supporting collaborative learning.*International Journal of AI-Ed*, 15:261–290.

[Suthers, 1998] Suthers, D. (1998).Representations for scaffolding collaborative inquiry on ill-structured problems.

[Tanaka et al., 2005] Tanaka, T., Yasumura, Y., Kaagami, D., and Nitta, K. (2005).Case based online training support system for adr mediator.In Yoshino, H., Ashley, K. D., and Nitta, K., editors, *AILE; Proc. of the ICAIL-05 Workshop.*

[Trafton and Trickett, 2001] Trafton, J. G. and Trickett, S. B. (2001).Note-taking for self-explanation and problem solving.*Human-Computer Interaction*, 16:1–38.

[van Joolingen, 1999] van Joolingen, W. (1999).Cognitive tools for discovery learning.*International Journal of AIEd*, 10:385–397.

[VanLehn et al., 2004] VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K. G., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. (2004).Implicit versus explicit learning of strategies in a non-procedural cognitive skill.In [Lester et al., 2004], pages 521–530.

[VanLehn et al., 2005] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. (2005).The andes physics tutoring system: Lessons learned.*International Journal of AI-Ed*, 15(3).

[Veermans et al., 2006] Veermans, K., van Joolingen, W., and de Jong, T. (2006).Use of heuristics to facilitate scientific discovery learning in a simulation learning environment in a physics domain.*International Journal of Science Ed.*, 28(4):341–361.

[Veermans and van Joolingen, 2004] Veermans, K. and van Joolingen, W. R. (2004).Combining heuristics and formal methods in a tool for supporting simulation-based discovery learning.In [Lester et al., 2004], pages 217–226.

[Voss and Post, 1988] Voss, J. F. and Post, T. A. (1988).*On the Solving of Ill-Structured Problems*, chapter 9.Lawerence Erlbaum Associates: Hillsdale New Jersey.

[Weisler et al., 2001] Weisler, S., Bellin, R., Spector, L., and Stillings, N. (2001).An inquiry-based approach to e-learning: The chat digital learning environment.In *Proceedings of SSGRR-2001. Scuola Superiore G. Reiss Romoli, L'Aquila, Italy.*

[Williams, 1992] Williams, S. M. (1992).Putting case-based instruction into context: Examples from legal and medical education.*The Journal of the Learning Sciences*, 2(4):367–427.

[Yoshino and Sakurai, 2005] Yoshino, H. and Sakurai, S. (2005).A knowledge-based systems approach to educating creative legal minds.In Yoshino, H., Ashley, K. D., and Nitta, K., editors, *AILE; Proc. of the ICAIL-05 Workshop.*

A Constraint-based Assessment Approach for Free-Form Design of Class Diagrams using UML

Nguyen-Thinh Le
Department of Informatics
University of Hamburg
le@informatik.uni-hamburg.de

Abstract. For a design problem in a modeling language like UML, there is no single correct solution. Usually, there are many solutions, which satisfy a given problem specification. In principle, the solution space can be infinite. However, current approaches evaluate student's entries by comparing them with a limited set of possible solutions and errors. Some other approaches anticipate design decisions by providing students with a set of appropriate design elements to select, thus ignoring the learning objectives of mastering the object-oriented analysis and design. We are extending the ArgoUML, an UML tool, to a learning system, which enables students to design a class diagram using UML in free-form. The core component of this system is an assessment module, which evaluates a class diagram based on design guidelines. We apply the constraint-based approach to model a solution space for the given use case, which represents a design problem. This paper describes our assessment approach and the current stage of our system which is able to evaluate elements of a class diagram: classes, associations and the multiplicity of associations.

Keywords: intelligent tutoring system, UML, design patterns, constraint-based modeling.

INTRODUCTION

Class diagrams are an important modelling type of UML. They are used during the analysis phase through the design phase to the construction phase of a software development process. A class diagram can be interpreted correctly if we know for which phase it has been created. During the analysis phase, a class diagram is created to model the objects of an application domain. A class diagram created on this level is called a conceptual model and contains class names, class attributes and class associations. During the design phase, a design model is created by enriching the conceptual model with method definitions and type information. In the implementation phase, the design model is transformed into a specific programming language to create an implementation model. Our goal is to support students to learn object-oriented analysis and design by mastering the following skills according to guidelines described in (Larman, 1998)[1]:

a) To create a conceptual model:

- Identify classes, which represent domain concepts using the Concept Category List and the Noun Phrase Identification principle.
- Determine associations and class multiplicity.
- Add attributes necessary to fulfill the specification requirements.

b) To create a design model:

- Find methods for a class,
- Add type information,
- Add navigability arrows to associations and detect dependency relationships.

We want to realise the learning objectives described above by developing a system, which presents students with a list of use cases and requests students to design a corresponding conceptual model or a design model. After having evaluated the student's class diagram, our system provides students with feedback. Through iterating the process of designing a class diagram and consulting system's feedback, students should be able to improve their class diagram until it meets the specification of the given use case.

In this paper, we present a constraint-based modeling (CBM) approach, which is applied to evaluate a class diagram based on design guidelines. First, we outline current work in this domain.

RELATED WORK

Three different attempts at developing an intelligent tutoring system (ITS) for UML modeling have been mentioned in the literature. The first one (Soller & Lesgold, 2000) developed a collaborative learning

[1] Some figures in this paper are taken from the same literature.

environment for object-oriented (OO) design problems using Object Modeling Technique (OMT), a precursor of UML. Machine learning techniques are applied to monitor group members' communication patterns and problem solving actions in order to identify situations in which students effectively share new knowledge with their peers while solving OO design problems. However, this system does not evaluate the OMT diagrams themselves.

The second attempt was conducted by Blank and colleagues (Blank et al., 2005) to support students learning OO analysis and design as problem-solving skills. The evaluation component of this system observes the student's entry (class name, each attribute, each method), and tries to match them with a corresponding part of the acceptable solution(s) and possible errors. Possible solutions and errors are coded by a human tutor in advance. If the student's input is not conformed to acceptable solutions and can be mapped to a library of possible errors, the evaluation component would interpret the student's input to be erroneous. The limitation of this system is that it anticipates the student's actions by defining possible correct actions as well as possible errors in advance, but the space of errors and of acceptable solutions might be unlimited.

The third attempt for an ITS for UML was introduced by Baghaei and Mitrovic (Baghaei & Mitrovic, 2005). According to this work, students' class diagrams are evaluated based on constraints, which model the syntax of the domain UML and the semantics of the given task. However, this system does not enable students to invent their own identifiers for classes, associations, methods and attributes. Instead, a name is selected from highlighted phrases of the problem text. The decision to use a noun from the problem text should be used as a class attribute or a class name is already made by this system. Such an interface design restricts concept identification and the assignment of self-explanatory names to simple cases. Furthermore, this system does tutor UML but not design strategies which are the most important learning issue. Thus, a major learning goal from the objectives mentioned above is not considered by this system..

We apply the CBM technique to develop a system to assess class diagrams submitted by students. The CBM approach was introduced by Ohlsson in (Ohlsson, 1992) and has been proven successfully in building many Intelligent Tutoring Systems (ITS) for SQL and database design (Mitrovic, 2001), natural language (Menzel, 2006), and has also been researched in the domain of teaching UML (Baraghei & Mitrovic, 2005), data structures (Warendorf & Tan 1997) and logic programming (Le & Menzel, 2005). Our system differs from the other ones by three points:

1. Class diagrams for a design task are created in a free form.
2. The system does not only tutor UML but also design guidelines.
3. The system focuses on examining task related requirements leaving the syntactic diagnosis to an UML design tool like ArgoUML (http://argouml.tigris.org).

THE FREE-FORM ASSESSMENT APPROACH

We provide students with a list of use cases and ArgoUML, an UML design tool, which enables students to create appropriate class diagrams. There is no cook book, which instructs how to construct a correct class diagram. However, there are books (Larman, 1998; Fowler, 2003; Jacobson et. al 2000), which recommend best practices for designing good class diagrams.

According to (Larman, 1998), classes and attributes are identified based on the Noun Phrase Identification principle and the Concept Category List. Nouns and noun phrases in a textual use case description are good candidates to be modelled as conceptual classes or attributes. Indications for classes, attributes and associations are described in details in the books concerned above.

In addition to using the Noun Phrase Identification principle to identify classes, we should apply design patterns to solve recurring problems. Certain solutions to design problems have been expressed as a set of principles. Patterns are named problem-solution formulas that codify exemplary design principles. At present, many object-oriented software designers know fundamental object-oriented design patterns, which are also the learning objectives our system should support. We refer to design patterns, the Noun Phrase Identification principle to identify classes and attributes, and other good design practices as design guidelines.

Our problem is to analyse the student's solution and to evaluate whether the class diagram elements (classes, attributes, methods, associations and the multiplicity of associations) meet the requirements of a given use case. To solve this problem, we need 1) to understand what diagram elements of the student's solution represent and 2) to determine whether the diagram elements of the student's solution adhere the specification of the given use case. The two steps assessment for class diagrams has been realized in a module which is integrated into ArgoUML.

The shortened use case **Buy Item** from (Larman, 1998) is used to illustrate the diagnosis approach.

***Buy Item**: This use case begins when a Customer arrives at a POST (point-of-sale terminal) checkout with items to purchase. The Cashier records the Universal Product Code (UPC) from each item. The POST determines the item price and adds the item information to the running sales transaction. The description and price of the current item are presented. On completion of item entry, the Cashier indicates to the POST that item entry is complete. The POST calculates and presents the sale total.*

The solution space for a given design task

The activity of designing a class diagram rests on the given use case under consideration. In principle, design decisions are based on:

1. Design view: conceptual model, design model or implementation model[2],
2. Design guidelines for finding classes, attributes, associations, multiplicities for associations,
3. The context of the use case,
4. Individual justification.

Design views are not a part of the UML specification. The notion of design view helps us to interpret a class diagram correctly. The content of a conceptual model is different than of a design model and the content of the implementation model requires more information than the design model. If a class diagram is given to be assessed, we need to know from which view it should be evaluated. Therefore, it is important to declare an agreement in advance.

There are numerous design guidelines for class diagrams. A conceptual model or a design model cannot be assessed to be absolutely correct or wrong, but more or less useful according to the given use case. To assess the usefulness of a class diagram, design guidelines are the foundation. For example, under consideration of the use case above, one may have specified a *POSTNumber* attribute in the *Cashier* type (Figure 1). According to the design guidelines "Not Attributes as Foreign Keys", this is undesirable because its purpose is to relate the *Cashier* to a *POST* object. The better way to express that a *Cashier* uses a *POST* is with an association, not with a foreign key attribute. We cannot say that the former design is wrong, but according to design guidelines it is less useful with respect to simplification and clarification.

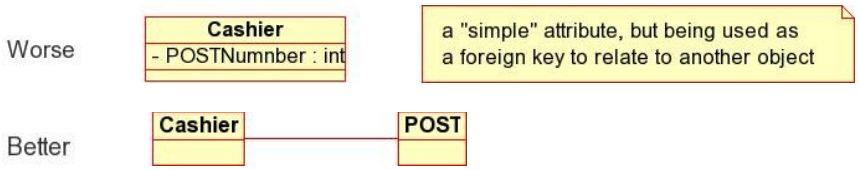

Figure 1: Design decision: relate a type using an attribute or an association.

The design decision for a class diagram is mainly derived from the context of the given use case. In the UML, the multiplicity value is context dependent. The example of *Person* and *Company* in the *Works-for* association from (Rumbaugh, 1991) indicates if a *Person* instance works for one or many *Company* instances. That is dependent on the context of the model; the tax department is interested in *many*; a union probably only *one*. For the ***Buy Item*** use case, one has to specify the multiplicity for the association *Records-sale-of* association between *SalesLineItem* and *Item*. Each line item might records a separate item sale, for example, one tofu package. However, it is also possible for a cashier to receive a group of like items, for example, six tofu packages, enter the UPC once, and then enter a quantity. Consequently, an individual *SalesLineItem* can be associated with more than one instance of an *Item* (Larman, 1998). As our use case does not state that the cashier is able to record the UPC from each item as well as to enter the quantity of the same item, we have to decide for the former design.

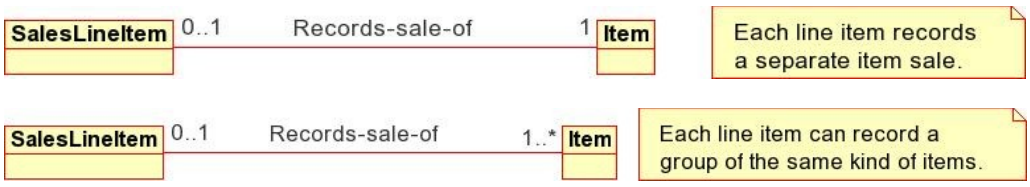

Figure 2: Design decision for the multiplicity of the association between SalesLineItem and Item.

The design of a class diagram depends not only on design guidelines, the requirements but also on the individual justification. For instance, there are two strategies to find associations for a set of concepts (Larman, 1998): 1) The knowledge of the relationship between concepts needs to be preserved for some duration can be modeled as an association ("need-to-know" association), 2) Associations derived from the *Common Associations List*[3]. We note that the ability to justify an association in terms of need-to-know is dependent on the requirements in the given use case. However, a strict need-to-know criterion for maintaining associations generates a minimal "information model" of what is needed to model the problem domain under consideration of the current requirements. Such models, which should play the role of a communication tool, do not convey a full understanding of the domain. For example, in Figure 3, which illustrates a conceptual model for the use case **Buy Item**, although on a strict need-to-know basis it might not be necessary to record *Sale Initiated-by Customer*, its absence leaves out an important aspect in understanding the domain, that a customer generates sales. Therefore,

[2] For our purposes, we just investigate conceptual model and design model.

[3] The Common Associations List contains a list of associations which are often used for conceptual model.

it is recommended to find associations somewhere in the middle between a minimal need-to-know criteria and one which illustrates every conceivable relationship.

As a result, there is no single correct model. All models are approximations of the domain we attempt to understand. A good model captures the essential abstraction and information required to understand the domain in the context of the current use case. The space of good models for the given use case is therefore unpredictably open-ended and thus, the task of designing class diagrams using UML is an ill-defined domain.

The constraint-based model of a solution space

We employ the constraint-based approach and use the ideal diagram (Figure 3) to model the space of solutions, which meet the requirements of the given use case. A constraint consists of two parts: a relevance part and a satisfaction part. The first part identifies the problem state, for which a constraint is relevant. The latter examines whether these elements satisfy the conditions of a constraint. If a constraint is relevant to the student's diagram, then it must satisfy the constraint. For example, we specify a constraint to express a design guideline that designing two concepts and an association is better than designing an attribute for a concept (Figure 1) if the attribute is neither of a simple type (string, integer, boolean) nor of a pure data value type[4] (or Data Types in UML terms).

Constraint 1:

IF the ideal diagram has a class X, a class Y, an association XY between X and Y, AND neither X nor Y represent a data type AND

The student's diagram has classes X', where X' is identical to X.

THEN the student's diagram ought to have a class Y' which is identical to Y and an association between X' and Y'.

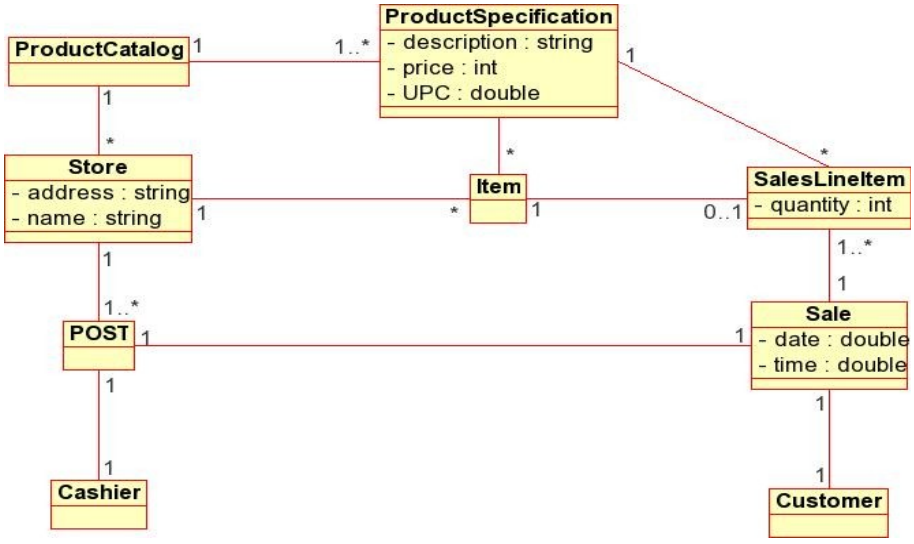

Figure 3: The conceptual model for the use case Buy Item

In addition to the existence of two classes, an association between them, Constraint 1 requires additional information about the function of the classes: none of the two classes are data types. We encode the function for each class of the ideal diagram explicitly: *domain concept, data type, generalization concept and specification concept*. The function of a class is *domain concept* if it is found in the given use case. Basically, one has to find concepts from the domain described in the use case and use the existing names in the territory and do not add things that are not there. A class has the function *data type,* if it is composed of separate sections (e.g. phone number, name of person) or there are operations usually associated with it, such as parsing or validation. A class has the function *generalization concept* if there are two concepts which share same attributes and we need a new class which represents the generalization of those two concepts. The *specification concept* is used for a class if there is a need to maintain the description of things and to reduce redundant information. For example, *ProductSpecification* is a specification concept required to describe many same items or products.

A constraint can be used to reflect the requirements of a use case. For example, the use case Buy Item requires that "*The Cashier records the Universal Product Code (UPC) from each item.*" For this purpose, we specify the following Constraint 2:

[4] Attributes are pure data values if unique identity is not meaningful for them.

Constraint 2:

IF the ideal diagram has a class X, a class Y, an association XY between X and Y, the multiplicity at X is M and the multiplicity at Y is N AND

the student's diagram has a class X' and a class Y', an association X'Y' between X' and Y' which are identical to classes X, Y and to the association XY

THEN the student's diagram ought to have an association X'Y' where the multiplicity for X' and Y' are M and N.

A constraint can not only be specified to express a requirement or a design guideline but also can be used to express different possible solutions for a certain problem. For example, classes which represent pure data values may be shown in the attribute section of another concept. But since it is a non-primitive type, with its own attributes and associations, it may be illustrated as a concept in its own box (Figure 4). For this case we specify Constraint 3.

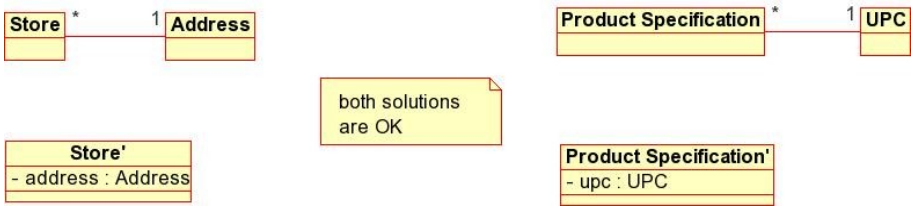

Figure 4: Pure data values can be shown in the attribute section or as their own concepts

Constraint 3:

IF the ideal diagram has a class X, a class Y, an association XY between X and Y, where Y represents a data type AND

the student' diagram has a class X' which is identical to X

THEN the student's diagram ought to have a class Y' which is identical to Y and an association X'Y'

OR the class X' has an attribute of type Y' which is identical to Y.

We enrich the constraint specification proposed by Ohlsson with more information: a penalty and a feedback, which are returned to students in case a constraint is violated. The penalty ranges from zero to ten, which represent the lowest severity and the highest severity of a constraint, respectively. The feedback can be used to mention the requirements of a use case, to describe design guidelines or to justify a design decision.

Beside constraints which are specified by the system engineer to model the domain of UML class diagrams, we enable exercise authors to specify constraints for individual justification. For instance, the exercise author might specify Constraint 4 which requires the existence of an association which expresses that a *Sale* is initiated by a *Customer*. This association can not be detected based on the need-to-know criterion. However, the exercise author might think that it is necessary to make the conceptual more useful and understandable, so he can specify this.

We specify for Constraint 4 a penalty of value five, because it represents a recommendation. Constraint 1, 2 and 3 should have a penalty of value ten because it represents a requirement. The penalty information is used to present feedback to students if constraints are violated. Feedback associated to violated constraints with higher penalty is shown first.

Constraint 4:

IF the ideal diagram has classes Sale1, Customer1 AND an association between them AND

the student' diagram has classes Sale2, Customer2 which are identical to Sale1 and Customer1

THEN the student's diagram ought to have an association between Sale2 and Customer2

Penalty: 5

Feedback: You need an association which expresses a Sale is initiated by the Customer.

We need three types of atomic constraints: class existence, association existence and multiplicity existence. Based on the penalty of the constraint, the existence of a class can be defined as optional (zero) or obligatory (ten). Similarly, the constraints for association existence and multiplicity existence ensure that a certain association or a multiplicity should exist. The constraint specifying the existence of an association can be coded as follows:

```
<constraint>
   <ae end1="SaleTransaction" end2="SalesLineItem" />
   <comment>An association between SaleTransaction and SalesLineItem is required.
   </comment>
   <penalty>10</penalty>
</constraint>
```

Based on the atomic constraints, we can define complex constraints applying conjunction, disjunction and negation operations in order to specify relationships between elements of a class diagram. For instance, the following constraint requires that if an association between the class "Item" and the class "Store" exists, then the class "Item" and the class "Shop" should have a multiplicity "1" and "0_N", respectively. The XML tag <and> conjoins two multiplicity existence requirements.

```
<constraint>
   <imply>
     <ae end1="Item" end2="Store" />
    <and>
     <me end1="Item" end2="Store" ort="end1" mult="0_N" />
     <me end1="Item" end2="Store" ort="end2" mult="1" />
   </and>
  </imply>
 <comment>A store can stock many items</comment>
 <penalty>10</penalty>
</constraint>
```

Constraints are evaluated as follows: for each constraint, the relevance part is evaluated first and if the student's diagram conforms to the relevance part, the associated satisfaction part is evaluated too. If the satisfaction part is not satisfied, i.e. the constraint is violated; the corresponding feedback and a penalty are returned. For clarity, the list of feedback messages can be sorted based on the penalty when displayed to the student.

Identifying a Diagram's Structure

Before the student's diagram can be evaluated by means of constraints, the diagram's structure needs to be identified. The identification process tries to map elements of the student's diagram to elements of the ideal diagram as Figure 5 illustrates. First, we carry out the identification process based on the Noun Phrase Identification principle.

Applying the Noun Phrase Identification Principle

We assume that students find the candidate concepts by applying the Noun Phrase Identification principle and the Concept Category List related to the current requirements under consideration. Each class of the ideal diagram has a list of alias names, which are not used for other classes. A class of the student's program is considered to be identified if either the name of the student's class matches exactly, or it is an abbreviation or a misspelled variation of the name of an ideal class or one of its aliases.

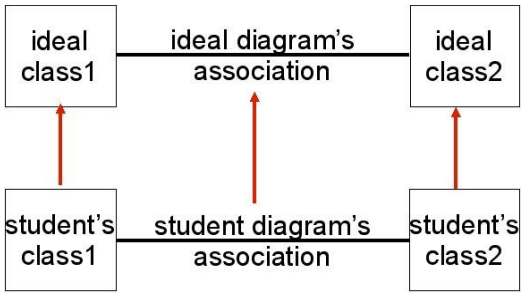

Figure 5: mapping between the student's diagram and the ideal diagram

The identification process iterates through all classes of the student's diagram. If any student's class can not be mapped, it is considered to be superfluous. Inversely, any class of the ideal diagram not paired with a class of the student's diagram is considered to be missed by the student.

After having identified the classes in the student's diagram, the identification process moves on to the level of class associations. An association of the student's diagram is mapped to the one of the ideal diagram if the two classes at the ends of the student's association correspond to the two classes at the ends of the association of the ideal diagram.

Combinatorial Matching

We continue the identification process to reduce the number of superfluous student's classes by combinatorial matching. An element from the set of superfluous student's classes is associated to a class of the ideal diagram where this class has not been associated to any student's class yet. After the class map is created, we derive the association map and evaluate constraints based on the class map and the association map. This process is continued until all mappings have been evaluated. The mapping which causes least penalty is taken to be the most plausible interpretation of the student's diagram. We clarify this combinatorial matching process by an example in Figure 6. Assuming, we have been able to map the class A' to the class A and the class B' to the class B based on the Noun Phrase Identification principle. Class X' could not be identified. We have to map the class X' to the set of classes X, Z and evaluate all constraints for each mapping. We assume that the penalty for each association existence constraint is of the same value. As result, we find two mappings: *Map1* = {A-A', B-B', X-X'} and *Map2* = {A-A', B-B', Z-X'}. The mapping X-X' satisfies constraints, which require the association existence between class X and class A and the association existence between class X and class B. The mapping Z-X' satisfies the constraint, which requires the association existence between class Z and class B, but the association between Class X' and A' will be evaluated as superfluous. Thus, *Map1* is hypothesized to be more plausible than *Map2*.

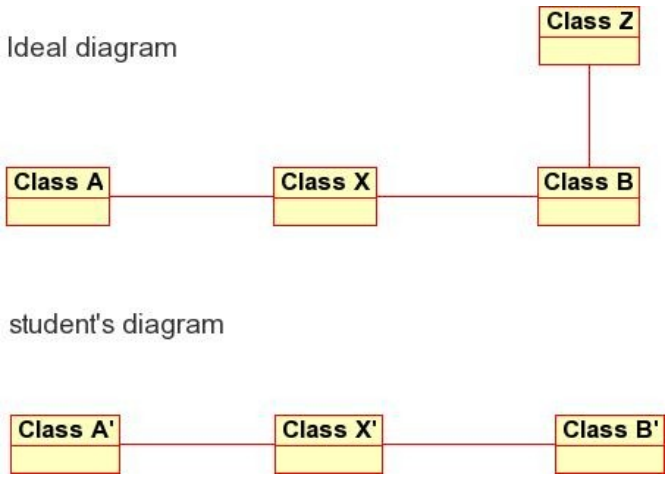

Figure 6: Combinatorial matching between a student's class and a class of the ideal diagram.

In case, some student's classes can not be identified, we can not evaluate them, because they seem to be not relevant in the solution space which is specified by constraints and the ideal diagram. We just remind the creator of these classes to consider the usefulness of them.

An Assessment Example

We demonstrate the educational capability of our system by assessing the student's diagram in Figure 7. First, the matching process is executed based on the ideal diagram. The system could not find a corresponding class in the ideal diagram for the class *Manager* in the student's diagram and returns following feedback: *The class Manager seems to be superfluous. Please, consider if you really need this class.*

Obviously, the system did not identify an appropriate class in the student's diagram for the class *POST* in the ideal diagram. But this will be evaluated by constraints. Assuming the system has only to evaluate four constraints described above.

Constraint 1 is relevant in two cases: 1) the class *Store* and the class *POST* and 2) the class *Cashier* and the class *POST* in the ideal diagram. However, Constraint 1 can not be satisfied in none of both cases because the student's diagram does not have a class which corresponds to the class *POST*. Thus, the system responds: "*According to design guidelines, the concept which represents a POST should be modeled as a class instead as an attribute because it is not a simple type.*"

Constraint 2 is violated because according to the use case the association between the class *Item* and the class *SalesLineItem* should have a multiplicity value of 1 at the class Item and 0..1 at the class SalesLineItem. Our system returns following hint because Constraint 2 has been violated: *"Based on the use case specification, one instance of SalesLineItem can be associated with one Item."*

Constraint 3 is not violated because it is possible to model the concept *UPC* as a class of data type or as an attribute.

Constraint 4 is relevant and violated because the exercise author specified the requirement explicitly that an association between the class *Sale* and the class *Customer* should exist. A corresponding hint, which was

specified by the exercise author, will be forwarded to the student: "*You need an association which expresses a Sale is initiated by the Customer.*"

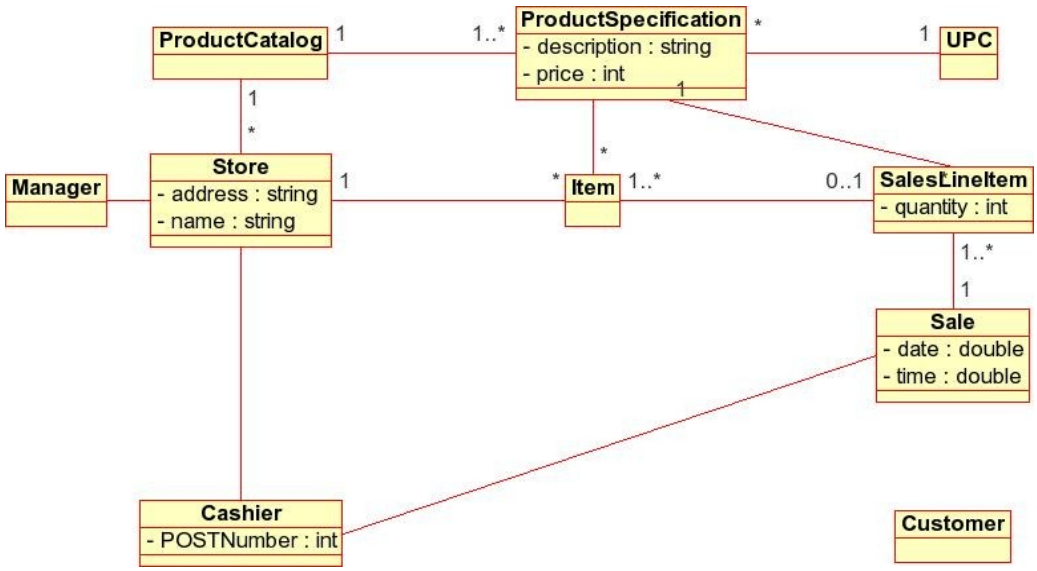

Figure 7 a student's diagram for the use case Buy Item

CONCLUSION AND FUTURE WORKS

Designing a class diagram using UML is a creative work. Thus, we provide students with a design tool like ArgoUML with which they can create class diagrams in a free-form. For a design problem in form of a use case, there are many solutions, which meet the requirements. To evaluate a solution in this domain, we apply the constraint-based approach and use an ideal diagram representing the requirements of the given use case. Constraints are not only used to specify requirements but also to describe the application of a design pattern. If the student's diagram violates a constraint, then it indicates that the student's diagram does not meet a requirement of the given use case or a design guideline has not been considered in the student's diagram. By enabling exercise authors specify constraints, the assessment can be enhanced with individual justification. Our assessment module, which has been integrated into the ArgoUML tool, supports students to develop their creativity and skill to design a class diagram and to master design patterns. At present, our assessment module is able to evaluate the following elements of a class diagram: classes, associations and the multiplicity of associations. The structure identification is currently based on the Noun Phrase Identification principle.

We plan to enrich the structure identification component with the combinatorial matching algorithm. Our goal is to extend our assessment module to be able to evaluate other elements of a class diagram: attributes, class methods, type information, and association navigation.

REFERENCES

ArgoUML homepage: http://argouml.tigris.org

Baghaei, N. and Mitrovic, A. (2005) *COLLECT-UML: Supporting individual and collaborative learning of UML class diagrams in a constraint-based tutor. Accepted for presentation at KES, 2005.*

Blank, G., Parvez,S., Wei, F., and Moritz, S. (2005) *A web-based ITS for OO design. Workshop on adaptive systems for web-based education tools and reusability.* 12th Int. Conference on AIED, Amsterdam.

Fowler, M. (2003) *UML Distilled. A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley.

Jacobson, I., Booch, G. and Rumbaugh, J. (2000) *The unified software development process.* Addison-Wesley.

Larman, C. (1998) *Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design.* Prentice Hall PTR.

Le, N.T. and Menzel, W. *Constraint-based Error Diagnosis in Logic Programming*. In Proceedings of 13th International Conference on Computers in Education 2005.

Menzel, W. (2006) *Constraint-based modeling and ambiguity*. In International Journal of Artificial Intelligence in Education, volume 16.

Mitrovic, A. et al. (2001) *Constraint-based tutors: a success story*. In L. Monostori and J. Vancza, Proceeding of the 14th International Conference on Industrial Engineering Application of AI and Expert Systems, 931-940, Budapest.

Ohlsson, S. (1992). *Constraint-based Student Modeling.* In International Journal of Artificial Intelligence in Education 3(4), 429-447.

Rumbaugh, J. (1991) *Object-oriented modelling and design*. Englewood Cliffs, NJ: Prentice-Hall.

Soller, A. and Lesgold, A. (2000) *Knowledge Acquisition for Adaptive Collaborative Learning Environments.* AAAI Fall Symposium: Learning How to Do Things.

Warendorf, K. and Tan, C. (1997) *Constraint-based student modeling - a simpler way of revising student errors.* In Proceedings of ICICS, 2, 1083-1087.

Language Learning: Challenges for Intelligent Tutoring Systems

Michael Heilman
Language Technologies Institute,
Carnegie Mellon University
mheilman@cs.cmu.edu

Maxine Eskenazi
Language Technologies Institute,
Carnegie Mellon University
max@cs.cmu.edu

Abstract. We describe the challenges presented by the assessment and presentation of knowledge components in the language learning domain, with particular attention to vocabulary acquisition. This paper first discusses the fact that the meaning of words is not as well formalized as many topics in better-defined domains such as mathematics. There follows a comparison of the number of knowledge components in language to the much lower amounts in other domains. An Intelligent Tutoring System for language must adopt different presentation and assessment strategies to confront the specific challenges of the domain. We describe REAP, a system that confronts these issues, and present empirical results that demonstrate its effectiveness.

Keywords: Intelligent Tutoring Systems, Computer-Assisted Language Learning

INTRODUCTION

Language learning is a multi-level task that integrates elements such as words, syntax, pronunciation, and culture. For one part of language learning, learning how to read, it is very different from more structured domains in that there are tens of thousands of knowledge components to be learned rather than a few hundred. In mathematics, knowledge components consist of particular formulae and theorems along with methods for their application. In learning to read, however, the set of knowledge components includes not only all of the grammatical rules in a language as well as exceptions to those rules, but also all of the lexical items in a language. Dictionaries, vocabulary lists, and other lexical resources define word meaning in an informal and limited capacity, and are not well suited for direct study. Knowledge of each word has not, or perhaps cannot, be defined as explicitly and formally as in other domains. Different teaching and assessment strategies must therefore be employed.

Description of the REAP Tutoring System

We begin with a presentation of the REAP reading tutoring system (Collins-Thompson and J. Callan, 2004 and Brown and Eskenazi, 2004), the development of which fuels our discussion of the many challenges of language tutoring. The goal of the REAP system is to furnish appropriate, authentic texts to students to help in reading and vocabulary learning. The tutoring system will incorporate grammatical constructions in the near future, but until now has focused primarily on teaching vocabulary. In REAP, a student sees short reading passages that contain a number of words (usually ranging from two to four) from his or her list of target words to be learned from context. The passages are Web documents of about one to two pages in length, covering a wide variety of topics.

In the current iteration of the system, a student user has a list of target words that he or she needs to learn over the course of a semester. We need to generate a list of words that appear in documents of the proper grade level but which the student has not previously learned. The student takes a preliminary test in which one question is presented for each word in a list of words that are assumed to be just above reading level for that student. This method takes a great deal of time, and brings up several issues related to assessment that we will discuss below.

Finding and identifying appropriate documents for these reading passages is also a significant challenge because of constraints on length, readability, topic, context, and text quality, as well as the preference for documents that contain multiple target words. We have found that less than one percent of the documents containing any target words are actually suitable for the students when we use the above-mentioned constraints. We will discuss the criteria for finding good documents in greater detail below.

The target words in a reading passage are highlighted to draw attention to them. In addition, students can look up these target words, as well as any other unknown words in the passage, by using an electronic version of the Cambridge Advanced Learner's Dictionary (Woodford and Jackson 2004) integrated into REAP. All dictionary use by students is tracked.

After each reading passage, the student works through exercises that facilitate construction and refinement of knowledge components for the target words. In later sections we discuss the various problems we have encountered that are related to the creation and evaluation of these exercises. These exercises are also a way to assess student knowledge, and can be used to select the subsequent reading material. It is difficult to accurately assess vocabulary knowledge, however, because of various issues that are specific to the language domain. We discuss these assessment-related challenges in the next section.

ISSUES RELATED TO STUDENT KNOWLEDGE OF VOCABULARY

A major issue we have encountered while creating this tutoring system centers on the assessment of a student's knowledge of words. This is essential in order to present readings that sufficiently challenge the student and provide effective instruction. Before choosing documents to present, the system must have an idea of what the student needs to learn. The REAP system therefore presents a pre-test to determine which words, from a chosen list, the student does not know. Once the tutoring begins, if a student has already learned a target word from a prior reading, then it is not efficient to next present a document with that word. Conversely, if a student has not learned a word from several prior readings, then it may not be worthwhile to present a document with that word in a new context. Also, it is important to have a model of the student's overall vocabulary knowledge. If a student's overall vocabulary knowledge is overestimated, the system will consistently search for readings that are too difficult and impede learning. Conversely, if the vocabulary is underestimated, the system will search for readings that are not sufficiently challenging.

Considering individual words, beyond the morphological relations between some words e.g., “select”, “selection”, “selecting”), there is little or no overlap among word usage patterns that allows for prediction of knowledge from synonyms or related words. Two semantically related words may appear in very different contexts. For instance, it is difficult to accurately predict whether a student knows the meaning of “industrious” from the fact that he or she knows the meaning of “hard-working”. Also, language courses cover hundreds of words—and beyond the classroom students learn thousands more; testing all these words is not usually feasible. In contrast, a course in a well-defined domain such as mathematics may have a curriculum consisting of fifty knowledge components, making it easier to test each and every one of them reliably.

Another reason for the difficulty of assessing individual words is that there are multiple levels of knowing a word. It is easier for a student to recognize the meaning of a word in a sentence than to produce a sentence of his or her own using that word. Stahl (1986) proposed a model of word knowledge with three levels with various degrees of knowledge. In the first level, a student is unfamiliar with the word. In the second level, the student has passive knowledge of the word and can understand it when reading or listening but cannot produce it. In the third level, the student has active knowledge and can produce the word in novel contexts. We assess passive word knowledge in REAP with a variety of multiple choice questions (Brown, et al. 2004). Synonym and antonym questions are generated by using WordNet (Fellbaum 1998) in conjunction with frequency statistics so that neither overly rare nor common words appear in exercises. Cloze questions, examples of which are shown in Figure 1, are generated automatically as well by extracting passages that contain the target word in an informative context. Finally, students are asked to produce novel sentences demonstrating knowledge of words. Thus REAP generates exercises that assess the various levels of knowledge of a word, from passive to active. The sentence production items currently have to be hand-graded, and are used only as post-test items.

He could never ___ the success he had enjoyed with his first record.
acknowledge comprise induce reproduce

Recently, the software company became a(n) ___ of a large corporation.
index subsidiary transmission interval

He answered the first question correctly, though he got all ___ questions wrong.
subsequent empirical identical legal

Figure 1: Example multiple-choice cloze questions generated automatically in REAP

A student may also know a word’s meaning but not the set of words with which it is used conventionally. Words often occur in set phrases and also in collocations, which are pairs of words that co-occur more frequently than would be expected from semantic constraints alone. With collocations, the meaning of words is not necessarily compositional, such that the collocation "white wine" does not refer to wine that is white but rather yellow-colored wine made in a certain way. Students thus may know the individual meanings of words, but then use them improperly. For example, a non-native speaker might describe tea as "powerful" and a car as "strong,"

whereas a native speaker would assign the adjectives in the opposite way though they are basically synonymous. (Halliday, 1966). These collocations are difficult to identify in a tutoring system because there is no comprehensive list available electronically. However, collocations can be identified automatically by employing statistical measures of co-occurrence (Church and Hanks 1989), including the Chi-square statistic, likelihood ratio, and mutual information of two words. These measures can be calculated for a given pair of words from a corpus of text using the frequencies of co-occurrence of these words within a small window (a few words long), the frequencies of the words separately, and the total number of words in the corpus. For instance, using the REAP corpus of Web documents, the likelihood ratio of "exceedingly difficult" is 61.9, while the ratio for "usually difficult" is a non-significant 4.1. This indicates that the former is a collocation while the latter is not. We have found that useful collocations usually have values over 50 for the likelihood ratio statistic. We have developed a prototype version of REAP which highlights significant collocations that have been identified by co-occurrence statistics and part of speech information. The system also creates multiple choice questions to assess student knowledge of which pairs of words collocate and which occur together more or less by chance.

REAP as a Tool for Teachers and Researchers

It is often assumed that students at a given level have encountered and learned a set of words associated with that level. The teacher assumes that an intermediate student knows words like "say" and "give" without explicitly testing these words. Of course, presumption of prior knowledge occurs in any domain--calculus students should know long division, for example--but it is usually not possible for students in such domains to reach the given level without that prior knowledge. It is very likely, however, that an advanced student of a foreign language might have "gaps" or "holes" in his vocabulary. For instance, a word like "dinosaur," which is known by any first grade student, might be unknown to a second language learner because it was never encountered in any lesson. Electronic dictionary use by students using the REAP tutoring system provides evidence of these "gaps" in student vocabulary knowledge. Although these students are at a language learning level approximately equivalent to eighth grade in an American school, they often look up words that are commonly learned before sixth or even fourth grade. We used the Living Word Vocabulary (LWV) to define first language grade levels for words (Dale and O'Rourke, 1981). In the LWV, the grade levels assigned to a word is the grade in American schools by which most students know that word. A chart of the proportion of the total number of dictionary accesses for words of each LWV level is shown in Figure 2. The data do not sum to one because there is a small percentage of looked up words for which there is no level defined in the LWV. While the probability of a student looking up words increases with that word's grade level, lower level words occur much more frequently than the rarer high-level words. Most words in a document are therefore lower level words normally acquired by sixth grade by native speakers. As a result, more than a third of words looked up by students were fourth or sixth grade words according to the LWV list. This indicates that there are often gaps in the lexical knowledge of students. Thus, any estimation of vocabulary knowledge based on a subset of words will be prone to error.

In the REAP system, we have developed tools that allow teachers and researchers to track the gaps in student vocabulary. We integrated Automatically updated reports show teachers which words that students are looking up while reading, and in which documents these words are looked up. For a given student or for a whole class, teachers and researchers can easily compare data on looked-up words to performance on post-reading exercises, the time spent per document by a student, or any other data that are tracked in the REAP system. These tools allow teachers to track the gaps in student vocabulary and address them in class. Researchers can also look for patterns in the gaps that might correlate with the native language of the students or other factors.

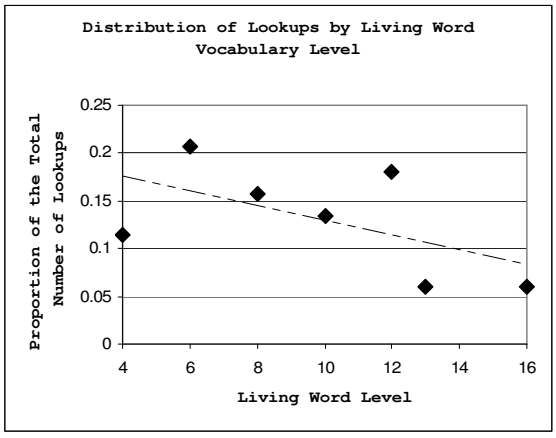

Figure 2: Proportion of total lookups by Living Word Vocabulary level

We also employ various heuristics in assessing student knowledge of individual target words. It is unclear how often, at what times, or in how many different contexts a word must be presented in order for a student to learn that word properly, though some research has been done on the topic (Pavlik, 2005 and Zahar, 2001). To refine word knowledge, we feel it is important to present words multiple times in authentic documents with various contexts. It has been shown that students acquire better knowledge of words when they are presented in multiple contexts (Zahar, et al., 2001). We chose to present each target word three times in order to facilitate knowledge refinement. There are many alternatives to such a scheme. In addition, it is unclear what exercise types most accurately assess student knowledge of words. The REAP system provides a way for researchers to determine the optimal number and timing for presentation of new vocabulary, as well as the best assessment strategies for vocabulary.

CHALLENGES OF PRESENTATION

The proper presentation of words is a major issue we have encountered in the REAP system. There are thousands of words that a language student must learn. Short of asking them to memorize lists of words and definitions, it is not feasible to present each individually. A language tutoring system must attempt to teach multiple items in the same reading passage, taking into account the uncertainties about what a student knows that result from the assessment problems detailed above. We will discuss this issue from the point of view of vocabulary teaching, although it applies to grammar as well.

When multiple words must be presented in the same reading, it is important to know the optimal number of unknown words to present in any given passage—which is called the vocabulary "stretch". If too many words in a reading are unknown, then the student is likely to become confused and learn none of them since it would be hard to understand the meaning of the passage. Research has indicated that a reader must know 98% of the words in a passage for it to be comprehensible without access to a dictionary (Hsueh-chao, et al. 2000). It is unclear what percentage of words must be known if a dictionary is available, and whether this threshold depends on the topic or individual. Also, it is not clear what percentage of words must be known in order for a student to learn vocabulary from contextual clues alone. A current study using REAP is aimed at clarifying some of these issues.

In our tutoring system, we use a readability measure based on language modeling techniques (described in Collins-Thompson and Callan 2004) in order to find documents at the appropriate level of difficulty. Collins-Thompson and Callan gathered a corpus of documents labeled with reading level for first through twelfth grade levels (in an American school). The unigram frequencies of individual words, as well as the rate of out of vocabulary words at each grade was used to identify the words that appear frequently and infrequently in texts at certain grades. The frequencies of words in a new text can then be compared to the models for each of grades in order to identify which grade level it is most likely the new text corresponds to. The accuracy of this measure compares favorably to that of other readability measures, especially on Web documents. Even though the REAP system can fairly accurately predict the reading level of a document, the number of words that students look up in a document varied from zero to thirty or more per reading in a current study. In the study, the mean number of words looked up was about 4.0, and the standard deviation was 3.5. Finding documents that contain the optimal number of unknown items while adhering to other criteria, such as reading level, is a challenge.

We can estimate the percentage of unknown words in a document by summing the number of target words and other words that a student looked up, and dividing by the total number of distinct word tokens. These estimates show that the REAP system is fairly successful at selecting passages of appropriate difficulty because for most documents less than three percent of the words are either target words or additional unknown words the student looked up while reading the document. In addition, student feedback gathered after each reading indicates that students find the documents neither too difficult nor too easy.

The optimal passage length for a reading is another issue for REAP. Currently, we present documents of approximately a thousand words to students. These documents provide a great deal of context in the hopes that students will acquire robust knowledge of the new vocabulary items. It may, however, be more effective to present new words with less contextual information. A single paragraph, or a single sentence, may be sufficient for a student to learn a word accurately. Presenting words in shorter passages would probably allow for greater control and efficiency in moving the student through a curriculum, but these shorter passages may not provide enough surrounding context for the words to be learned accurately. The REAP system will allow us to test these hypotheses in the near future.

Drawing focus to target vocabulary words affects student learning of those words. When presenting a word in a passage of any length, the student may not focus on and learn the target word if he or she is not prompted to do so. A student may very well skip over a target word because it is unknown and not necessary for comprehension of the passage. Highlighting target words is a way of increasing the student's 'noticing' of those words, which has been shown to be very important in second language learning (Schmidt, 1990). Research has shown mostly

positive effects of highlighting target words (De Ridder, 2002), and REAP has highlighted target words. This choice to draw attention to target words may have negative effects, of course. For instance, the student may choose to focus solely on the highlighted words and ignore the rest of the reading. In practice, we find that this is sometimes the case. Some students go through long documents in a few minutes because they only look at the highlighted target words. There are two problems that arise from students focusing only on highlighted words in this way. First, the surrounding context, which may be crucial to learning nuances of word meaning, is ignored. Second, there are likely a number of other new words in a passage that the student has the opportunity to learn. Although it is desirable to guide a student to focus on certain parts of a reading passage and on specific items, learning opportunities may be missed if the rest of the passage is ignored.

Dictionary accessibility is related to the prior issue of drawing attention to target words. In the REAP system, students can click on highlighted words to see a definition, and can also easily look up any other unknown words from the reading. Dictionary access can also lead to a student using only the dictionary definition to learn a word's meaning, instead of looking at the surrounding context as well. We feel, however, that it is more valuable for a student to be able to engage in a form of exploratory learning by looking up meanings for non-target words. Also, the dictionary allows the student to better grasp the surrounding context of a target word if that context contains unknown words as well. Students ignoring contextual information in favor of dictionary definitions can be seen as a form of "gaming the system," which is a common problem for intelligent tutoring systems (Baker, et al., 2004). Dictionary use and the highlighting of target words can have negative effects, but student behavior can be monitored either by teachers or automatically, to stop students from "gaming the system" and missing opportunities for learning.

Quality of Authentic Passages as Reading Material

Independent of passage length and difficulty, not all documents are of equal pedagogical value. The REAP system focuses on using authentic materials because they both improve student motivation and help in overcoming the cultural barriers to language acquisition (Bacon and Finnemann, 1990). In order to find a sufficient number of passages automatically, the system uses the Web as a corpus from which authentic reading materials can be gathered, but the quality of Web documents as readings is a crucially important issue. Many documents on the Web consist of long lists of names or words, with few well-formed sentences. Common sense tells us that such documents without well-formed sentences are not valuable as readings. Documents consisting primarily of coherent and cohesive sentences and paragraphs are generally much more engaging and valuable. Human filtering and selection of reading material is possible for a small-scale tutoring system, but in REAP all filtering is automatic. The first person to see much of the material used in REAP is the student. Automatic filtering of authentic materials greatly decreases the ratio of development time to instructional time, which is an important issue for intelligent tutoring systems.

We have implemented a text quality predictor in REAP that effectively filters out the large number of documents that are useless as reading material. At first, we attempted to identify useful documents by analyzing HTML structure in order to tell how much text might be contained in tables and lists of items. Examining the widely varying structure of Web pages, however, is not feasible because there is no set of consistent formatting criteria for Web pages. Instead, we base our measure of text quality on the probabilistic context-free grammar (PCFG) scores from a natural language parser that produces parse trees representing grammatical structure for each sentence in a document. These sentence-level PCFG scores are log-likelihood estimates for the most likely parse tree for the given sentence. Lists, menus, and other features that appear often in poor quality documents consist of series of noun phrases for which there is no likely syntactic parse tree (e.g., "electronics cameras computers games appliances mobile phones"). Likewise, other poor reading material such as bulletin board postings often contain incomplete sentences for which there is no likely parse. We use the Stanford parser to generate these sentence-level PCFG scores (Klein and Manning, 2002). Significantly fewer than half of the documents gathered during crawls of the Web pass the text quality filter, illustrating that good reading passages for students are difficult to find and identify automatically. Of course, the cohesiveness and quality of contextual information is something that is difficult to define quantitatively, and so we had to set a threshold for the level of quality that is acceptable for a document to be presented to students.

Student interest in reading material is another important issue in language learning. Prior research has shown that personalization and choice in tutoring systems can facilitate learning in other domains (Cordova and Lepper, 1996). While tutoring systems for well-defined domains such as mathematics have little control over the context in which material is presented to students, language learning material can be taught in almost any context. The great majority of words (grammar as well) is not specific to any given topic or context. So while an algebra tutoring system might be forced to present a given theorem in the context of a business transaction, a language tutoring system can present a word such as "specific" in any context, for example in passages referring to "a specific sports team," "a specific food," "a specific car," etc. The great variety of topics in which language teaching may be situated is a great advantage to tutoring systems. While a human teacher usually gives a single

reading that may interest only a subset of students, a computer tutoring system can provide individual instruction tailored to each student's interests. Although our system does not currently incorporate topic detection and tracking, we feel that personalization and choice are important goals for any language tutoring system. We have created a text categorization system based on Support Vector Machines (Burges 1998) using SVM-Light (Joachims 2002) that assigns one of ten topic labels from the top level of the topic hierarchy of the Open Directory Project (ODP, http://dmoz.org) with 78% accuracy. The topic labels include "Arts," "Science," "Business," and others. The system was trained and tested on a set of 10,000 labeled documents gathered from the ODP in early 2006. The corpus was split randomly into a training set of 8,000 documents and a test set of 2,000 documents. In an upcoming study, REAP will use these topic labels to provide documents to students according to their individual interests.

Automatically finding appropriate reading material is not a trivial task as one might at first assume. Although finding arbitrary documents that contain single target words is simple with modern search engine technology, these documents are rarely useful as reading passages. It is even more difficult to find pairs or groups of specific words since the likelihood of rare words occurring together is so low. For example, if two words each occur in one in a thousand documents, then unless they are strongly related they will occur together in about one in a million documents. What further complicates the matter is that most documents do not satisfy length, readability, and text quality constraints either—not to mention topic constraints. In generating our database of reading passages, we found that only about 0.5% of documents pass through our filters. More than half of documents are too long, about a third are out of the appropriate range of estimated reading difficulty, and about two-thirds of documents consist mainly of lists and menus rather than cohesive text. It is therefore very difficult for an intelligent tutoring system to select appropriate reading passages to present to students, even independent of the nature of the presentation.

EMPIRICAL RESULTS FROM A STUDY INVOLVING REAP

In this section we will present results from a recent study that validate the approach used in the REAP system. The study was conducted in the Spring of 2006 at the English Language Institute of the University of Pittsburgh. Thirty-two English as a Second Language (ESL) students used the system once a week in eleven forty-minute sessions over the course of the semester. Twenty-two of these students successfully completed the post-test. Some did not show up for the post-test because it was voluntary rather than for a grade, and a few experienced minor technical difficulties in one portion of the test, so their scores were excluded from the results we present. The subjects were international students from a variety of countries—including Saudi Arabia, Korea, Japan, and France—who were studying English in order to enter American universities. The students were at an intermediate ESL level corresponding to about eighth grade in a U.S. elementary school.

The REAP system supplemented their coursework in an English reading skills course. A list of 216 target vocabulary words was created for the study. These words were chosen from the Academic Word List (Coxhead 2000), which consists of words that are essential for reading and writing University-level English text. These words are normally learned at a level above the ESL level of the students, and none appeared in the course's regular materials. It is thus unlikely, although still possible, that the students would learn these words during the semester outside of the REAP system. Each student took a 45 minute pre-test consisting of multiple choice cloze (that is, fill-in-the-blank) exercises in order to assess which of these words they did and did not know. Most students completed about half of the list of possible pre-test items. The words for which responses were incorrect were added to individual student focus word lists, on which the REAP system would provide training.

During sessions with REAP, the students read Web documents selected by REAP containing up to four of the words from a particular student's focus word list. These documents also passed the various automatic filters in REAP for text quality, length, and reading level. The REAP system attempted to show each focus word up to three times in three different documents. As discussed above, students were able to look up any additional unknown words using an electronic dictionary. After each reading, students completed multiple-choice cloze exercises related to focus words from the document just read. In most but not all cases, students received training on their entire list of focus words.

At the end of the semester, one week following the final reading session, students took a post-test to assess their progress made while using REAP. The post-test had two sections. In the first section, students were asked to produce novel sentences demonstrating knowledge of the meaning of a word for ten of the focus words on which they received training. The second part of the post-test consisted of forty previously unseen multiple-choice cloze exercises that were similar to the pre-test and post-reading exercises. The first part of the test—the goal of which was to assess transfer of knowledge to a novel task—was conducted before the other section so that the cloze exercises did not give away sentences which could be used to answer the transfer items. In a separate test conducted a few days later, eight of the students were asked to produce sentences demonstrating knowledge of looked-up words. The words for this test were not on a student's focus word list, but instead were words

looked up in the electronic dictionary while the student used REAP. All of the sentence production items were graded using the following three-point system: one point was given for proper grammar usage of the word, one point was given for the word's semantic meaning fitting into the sentence regardless of whether knowledge was demonstrated (e.g., "The student *demonstrated* his success."), and a third point for clearly demonstrating knowledge of the word (e.g., "The student *demonstrated* his knowledge to the teacher by writing a sentence").

The results from the post-test for focus words are presented in Figure 3. The pre-test scores for all of these words are essentially zero because students answered pre-test exercises for these words incorrectly. For the pre-test multiple-choice cloze exercises, there was a twenty-five percent chance of guessing the answer correctly since there were four possible choices. For the sentence production tasks, the chance of producing a valid answer was considerably lower, although if the part of speech of the word was known, a student might receive partial credit for a grammatical but meaningless sentence. The students performed very well on the multiple-choice cloze items, indicating that using REAP helped them to learn their focus words. Although performance on the transfer task was lower, there is some evidence that the knowledge gained on focus words while using REAP transferred to a novel situation. Teachers of the course predicted low performance on the sentence-production transfer task beforehand due the level of the students and the comparative difficulty of production tasks over recognition tasks.

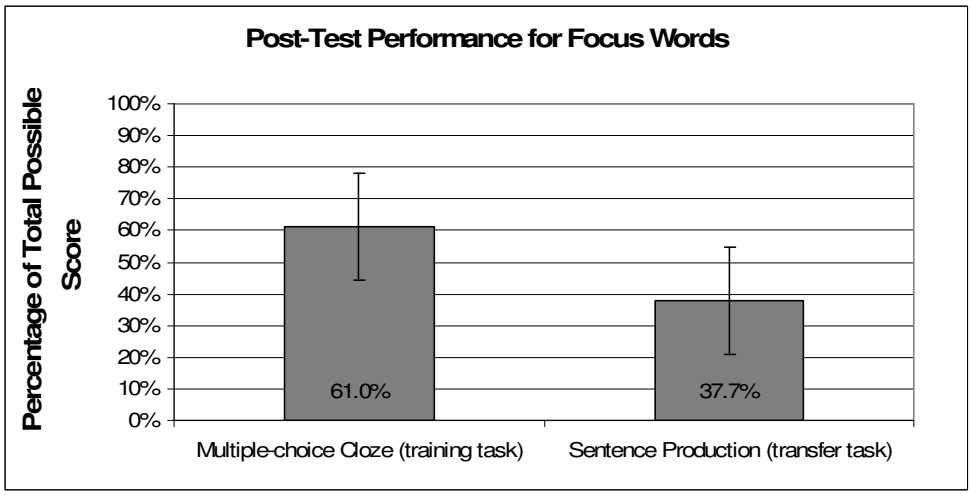

Figure 3: Student post-test performance on focus words for the training task and a transfer task

The production scores for looked-up words are presented for comparison with the scores for focus words in Figure 4. From the scores for looked-up words, students appear to have learned a small but significant portion of the words beyond their individual focus word lists. Several students looked up more than one hundred additional words during reading, so even if a small percentage of these words were learned, it would be valuable for accelerated learning.

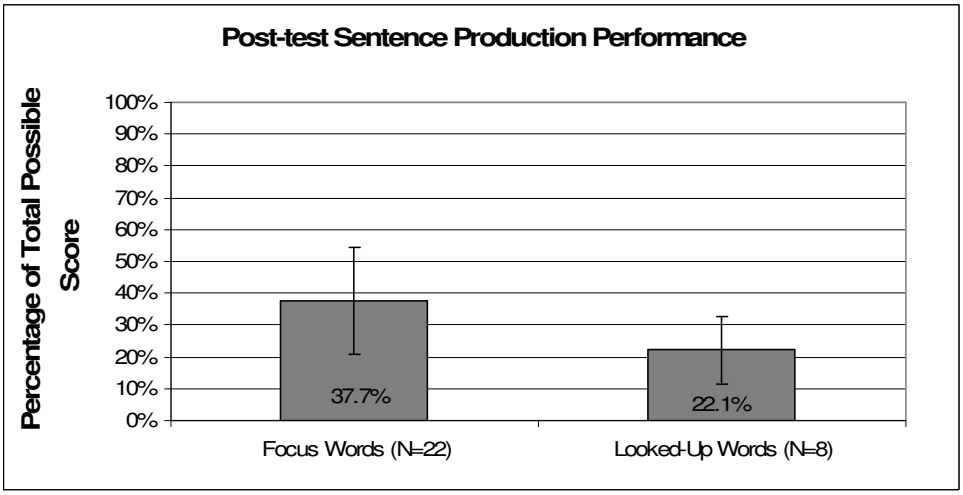

Figure 4: Sentence production performance for focus words compared to looked-up words

The students opinions of the system also validate REAP. Prior to one of the final reading sessions, students took an exit survey asking ten questions on a Likert scale from one to five. A value of one indicated strong disagreement with a given statement, while a value of five indicated strong agreement. The results of this exit survey are shown in Figure 5, with bars for standard error. On the positive side, students found the system very easy to use, and most felt that REAP should be used in future classes. They also believed that they had learned a lot of the focus words, corroborating the quantitative evidence from the post-test. Also, the students did not feel that the documents selected by REAP were too difficult for them to understand, which indicates that our automatic filters work properly to provide high-quality, authentic documents. On the negative side of things, students did not find the documents particularly interesting, and would have liked to select the documents or at least the general topics of the documents. Overall, however, the exit survey results were very positive, as were the additional free-response comments of the students.

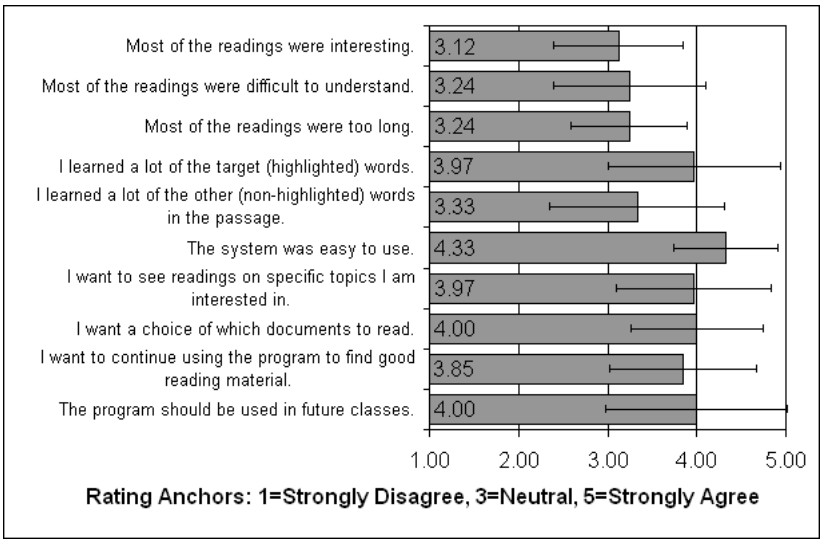

Figure 5: Exit Survey Results from 32 students who used the REAP system

CONCLUDING REMARKS

Language tutoring systems face a unique set of problems that arise from the nature of the language domain. The set of items, either grammatical or lexical, that a student must learn is very large. Also, it is often difficult to accurately assess the knowledge of a single item because of the various contexts in which words may occur. Choosing the number, nature, and timing of assessment exercises is thus a great challenge. In addition, knowledge components must often be presented together in a single reading passage because teaching items individually is not feasible. A language tutor must choose the optimal number of items to present in a passage, the length of the passage, and whether to draw attention to these items, in order to make learning efficient. When using authentic material, which is desirable in language learning situations to increase motivation and transfer to real-life situations, documents must also be filtered for text quality to identify useful passages that consist of cohesive sentences and paragraphs. Finally, language learning materials can cover a wide variety of topics such that there are unique opportunities for personalization and choice in a language tutoring system. Although similar issues often arise in other learning domains, their unique combination provides a number of interesting challenges for intelligent language tutoring systems.

The REAP system addresses many of these challenges of the language learning domain, and empirical results validate our approach. We are planning to develop better automatic question generation techniques, perhaps including techniques for evaluating free responses, so that we can better assess students' knowledge. It may also be useful for us to estimate the "holes" in students' vocabulary by using language background and dictionary access information. We are investigating the optimal length and scheduling of the passages that REAP shows to students. In the near future, the REAP will also include options for personalization and choice of reading topics in order to make students more interested and engaged while using the system. The system will eventually expand to include grammar instruction along with lexical practice.

ACKNOWLEDGMENTS

The authors thank Jamie Callan, Kevyn Collins-Thompson, Jon Brown, and James Sanders for their work on the REAP project. We also thank Alan Juffs and Lois Wilson at English Language Institute at the University of

Pittsburgh for using REAP in the classroom. We thank Vincent Aleven for some early guidance for the paper, as well as the anonymous reviewers for providing useful feedback.

This material is based on work supported by NSF grant IIS-0096139. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsor.

REFERENCES

Bacon, S., and Finnemann, M. (1990). A study of attitudes, motives, and strategies of university foreign language students and their disposition to authentic oral and written input. *Modern Language Journal, 74*. 459-473.

Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System". Proceedings of ACM CHI 2004: Computer-Human Interaction (2004) 383-390.

Brown, J. and Eskenazi, M. (2004) "Retrieval of authentic documents for reader-specific lexical practice." In Proceedings of InSTIL/ICALL Symposium 2004. Venice, Italy.

Brown, J., Frishkoff, G., and Eskenazi, M.. (2005). "Automatic question generation for vocabulary assessment." In Proceedings of HLT/EMNLP 2005. Vancouver, B.C.

Burges C.J.C. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2): 121-167.

Church, K. W., and Hanks, P. (1989). Word Association norms, mutual information and lexicography. In *ACL 27*, pp. 76-83.

Collins-Thompson, K. and Callan, J. (2004) "Information retrieval for language tutoring: An overview of the REAP project" (poster description). In Proceedings of the Twenty Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffield, UK.

Collins-Thompson, K. and Callan, J. (2004) "A language modeling approach to predicting reading difficulty." In Proceedings of the HLT/NAACL 2004 Conference. Boston.

Cordova, D. I., & Lepper, M. R. (1996) Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. Journal of Educational Psychology, 88, 715-730.

Coxhead, A (2000) A new academic word list. TESOL Quarterly, 34, 2: 213-238.

Dale, E., and O'Rourke, J. (1976, 1981) *The living word vocabulary*. Chicago: World Book/Childcraft International.

De Ridder, I. (2002) Visible or invisible links: Does the highlighting of hyperlinks affect incidental vocabulary learning, text comprehension, and the reading process? Language Learning and Technology 6: 123–146.

Fellbaum, C., editor. (1998). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA.

Halliday, M.A.K. (1994) Lexis as a linguistic level. In C.E. Bazell, J.C. Catford, M.A.K. Halliday, and R.H. Robins (eds.), *In memory of J.R. Firth*, pp. 148-162. London: Longmans.

Joachims, T. (2006). SVM light, An implementation of Support Vector Machines (SVMs) in C. http://svmlight.joachims.org/.

Klein, D. and Manning, C. D. (2002) Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15 (NIPS 2002), December 2002.

Open Directory Project (2006). http://dmoz.org.

Pavlik, P. I. and Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. Cognitive Science, 29, 559-586.

Schmidt, R. (1990) The role of consciousness in second language learning. *Applied Linguistics*. 11, 129-158.

Steven A. Stahl. (1986) Three principals of effective vocabulary instruction. Journal of Reading, 29.

Woodford, K and Jackson, G. (2003). *Cambridge Advanced Learner's Dictionary*. Cambridge University Press.

Zahar, R., Cobb, T., & Spada, N. (2001) Acquiring vocabulary through reading: Effects of frequency and contextual richness. The Canadian Modern Language Journal, 57(4), 541-572.

The challenges in adapting traditional techniques for modeling student behavior in ill-defined domains

Amy Ogan, Ruth Wylie, Erin Walker
Human Computer Interaction Institute, Carnegie Mellon University
aeo, rwylie, erinwalk@andrew.cmu.edu
5000 Forbes Ave.
Pittsburgh, PA 15213
(412) 268-1208

Abstract. Designing cognitive tutors and modeling behavior for ill-defined domains require innovative methods and techniques. We combine a top-down, theoretical approach with a bottom-up, empirical approach to develop a student model for the selection of aspect in French verbs. In performing this task, we design a new representation applicable to feature-driven ill-defined problem spaces and utilize tutoring scaffolds in order to elucidate the student thought process. We then evaluate and refine our model based on empirical data collected through student think-alouds. We plan to use our preliminary results to design and evaluate a fully-developed cognitive tutor, and hope to generalize our tutor development process to other ill-defined domains.

Keywords: Ill-defined domains, student modeling, passé compose and imparfait

MOTIVATION

A completely well-structured problem is a problem in which the starting statement contains all relevant information, and there exist a limited number of relatively easily-formalized rules used to reach an unambiguously correct or incorrect solution [Jonassen, 1999]. These problems are often found in mathematical domains; to an expert, the simple arithmetic question "26+38=?" has a clear set of steps from the initial state to the goal state. Most real-world problems are not so well-structured. In particular, domains like design and cultural education contain problems that are not so tidy. When the start state, rules, or goals of a problem are not easily formalized, the problem is ill-structured [Ormerod, 2006]. For example, writing an essay is completely ill-structured: the start-state is underspecified, there is no predefined set of rules for completing the task, and it is difficult to know when a satisfactory result has been attained. Not all domains are so ill-defined. Some domains lie somewhere in the middle. Their problems may have well-structured start and goal states, but ill-structured rules because 1) there are multiple representations of knowledge with complex interactions and 2) the ways in which the rules apply vary across cases nominally of the same type [Spiro, 1991]. Unfortunately, when rules and the conditions in which they are applied are difficult to formalize, it is also difficult to form a model of student and expert performance for that domain.

This modeling issue is particularly problematic when trying to build a cognitive tutor for an ill-defined domain. A cognitive tutor is an intelligent tutoring system that compares student action to a model of correct and incorrect behavior and provides context-sensitive feedback and problem selection. Cognitive tutors have been effective at increasing student learning in real-world settings by as much as one standard deviation over traditional classroom instruction [Koedinger, 1997]. However, most successful tutors have been limited to well-defined domains like algebra and physics. Ill-defined domains may also benefit from cognitive tutoring, but this area is only beginning to be explored. Cognitive tutors rely on the existence of a formal domain model as the basis for other stages of tutor development like identifying problem content or designing tutor feedback, thus the absence of a model for a given ill-defined domain makes developing a tutor problematic. Alternative solutions have been suggested such as simply forcing a structure on the domain [Simon, 1973], or forgoing a structure altogether and focusing on non-model-based learning tasks [Davis & Tessier, 1996]. We believe, however, that by adapting current cognitive tutor modeling techniques, it is possible to develop a cognitive model for an ill-defined domain that provides structure while not over simplifying the complexity of the domain.

In this paper, we focus on the ill-defined domain of determining aspect in French second language learning. We discuss the difficulties of applying traditional modeling techniques in ill-defined domains as well as our solutions for adapting these techniques. Finally, we examine the model created using this procedure and outline future plans for developing a tutor. Our results suggest that while traditional modeling techniques are inadequate for ill-defined domains, adaptations in knowledge representation, problem presentation, and experimental design lead to effective solutions.

ASPECT IN FRENCH LANGUAGE LEARNING

Problems in language learning fall at all points on the continuum of well-defined to ill-defined tasks. Successful cognitive tutors based on student models of observable behaviors have been implemented in well-defined language areas, for example the Capit system [Mayo & Mitrovic, 1997] teaches students capitalization and punctuation rules [see Gamper & Knapp, 2002, for review]. However, not as much work has been done in more ill-defined areas (e.g. the acquisition of grammatical gender) even though formal instruction in these areas is a necessary component of second language education [Norris & Ortega, 2000]. In particular, the distinction between the passé composé and the imparfait tenses in French is a prototypical ill-defined language learning problem that has clear start and goal states, but ill-structured rules and conditions for applying them [DeKeyser, 2005]. Mastering the distinction between these is a difficult task for both beginning and advanced French students and is reviewed often throughout programs of French instruction. This distinction is acquired by learning and understanding the concept of aspect.

Aspect is the relation between a situation and its associated interval of time [Comrie, 1976]. Students must know the role aspect plays in a sentence in order to both understand temporal qualities of actions and be able to accurately produce novel utterances. When speaking in the past in French, aspect is conveyed through the use of two tenses: the passé composé and the imparfait. The passé composé involves a completed action, as if viewed from an external perspective, while the imparfait involves an ongoing action, as if viewed from an internal perspective [Salaberry, 1998]. Examples of uses of the passé composé, translated into English, are "I went to the store on Tuesday" and "I stopped at the store," while uses of the imparfait include "I went to the store every Tuesday" and "I was at the store". The phrases "on Tuesday" and "stopped at" indicate that the action is finished, while "every Tuesday" and "was at" indicate ongoing actions. Students must use features of the sentence, like lexical semantics and calendric expressions [von Stutterheim, 1991], to infer properties of the action such as its duration that are relevant to making this aspectual decision [Ishida, 2004]. Because the past tense in English is 'completely ambivalent', this task is novel and particularly difficult for native English speakers learning French [Salaberry, 1998].

Aspect is difficult to learn, because it belongs to a class whose problems "express highly abstract notions that are extremely hard to infer, implicitly or explicitly, from the input" [DeKeyser, 2005]. Aspect has a relatively clear goal state: Experts might disagree on the aspect of a sentence, but only in certain ambiguous cases where the intention of the speaker is not clear. Difficulties in identifying aspect tend to arise because apsect is a problem with ill-structured operators: rules are hard to formalize, the conditions under which they apply are too specific and numerous to be described, and rules may be applied in parallel. It is difficult to find a formal description of the rules for identifying aspect, and descriptions often do not correspond to one another. Additionally, instructional texts often confuse rules, which will always return a correct answer when applied correctly (e.g., "If the action occurs a single time the tense is passé composé"), with heuristics, which may be easier to apply but are not always correct (e.g., "If the sentence contains a word like 'once' the tense is passé composé"). Instructional texts also generally do not cover the whole problem space, leaving students with cases that are difficult to classify. In fact, since rules that do cover the space are abstract, describing all the conditions under which they apply is important but impractical. To understand what is meant by a completed action, the student must be aware that it has a start, an end, or a specific duration. "Has a start" is then broken down into many sub-cases, such as verbs that imply a beginning action, and as these conditions become more specific, enumerating them amounts to identifying particular examples. To complicate the situation, there are sentences where multiple rules apply, and the student must perform conflict resolution. For example, the sentence, "All of a sudden, the sky was blue," appears to be both an ongoing description of circumstances and a completed change of state. The student has to know that this sentence is in fact not a description, but an event (as signified by *all of a sudden*) and the passé composé should be used. Because aspect is an ill-defined domain which is challenging and important for students to master, it is an ideal candidate for our attempts to model ill-structured problems.

TRADITIONAL MODEL-BUILDING AS APPLIED TO ASPECT

A cognitive model is a formal description of a problem-solving process. It includes both expert and novice behavior, and correct and incorrect actions. It is required when building a cognitive tutor so that the tutor can provide contextual feedback on problem-solving. Cognitive models are generally developed using a combination of theory-driven and data-driven approaches. Although a theory-driven approach can identify what is relevant about a task and pinpoint thought processes that are not necessarily visible through behavioral observation, it may not ultimately reflect the steps novices take to solve a problem. A data-driven approach can highlight problem-solving strategies and misconceptions in actual users, refining the initial formal model.

We modeled the specific task of identifying the aspect of a verb. Students were presented with a French sentence containing a verb in its infinitive form and asked to indicate whether the sentence should use the passé composé or imparfait. For example, a student was given the French translation of "While I was doing my homework, the telephone _____ (to ring)" and asked to select the appropriate tense of the sentence. We first performed a rational task analysis and then enhanced our model through think-aloud protocols from experts and students.

Rational Task Analysis

In a rational task analysis, a formal specification of the task is combined with consultation with experts to produce a model of ideal performance and identify places where novices may make errors. The result is generally a set of production rules (see Anderson et. al., 2004) arranged in a sequential and deterministic structure. For example, this approach was used by Siegler (1976), who proposed a decision tree representing children's ideal performance on balance scale problems, identified subsets of the tree representing novice performance, and validated his model empirically. A properly developed theoretical model forms a basis for the design of an effective cognitive tutor.

A model of performance on determining aspect in French was developed. Logical analysis of the model suggested that the task involves production rules of the following structure: "IF the sentence contains feature X, AND feature X is a member of class Y, AND class Y indicates tense Z, THEN the tense of the sentence is Z." An example of one of these rules is, "If the sentence contains an expression of time, and the expression of time indicates a one-time action, and a one-time action indicates the passé composé tense, then the tense of the sentence is the passé composé." Because the instructional texts and experts we consulted tended to disagree on the formalization of the rules, we chose a set of five that were agreed upon by the majority of sources and covered the full problem space. We then attempted to identify how these rules might be applied in sequence to efficiently reach a correct result. The full decision tree is shown in Figure 2. In our model, students ask themselves a sequence of increasingly abstract and more difficult questions. Notice that the default question is not a yes or no question but a catch-all which covers the full problem space. It requires students to make a high-level judgment about the action in the sentence without resorting to the heuristics targeted in previous questions that are easy to answer but not always accurate. The model predicts that experts (and novices) go through a sequential process of querying the sentence for features and use the first positive response to arrive at a decision. We intended to deal further with the problem's ill-structuredness by providing scaffolding (described in later sections) for the identification of features in the resulting tutor.

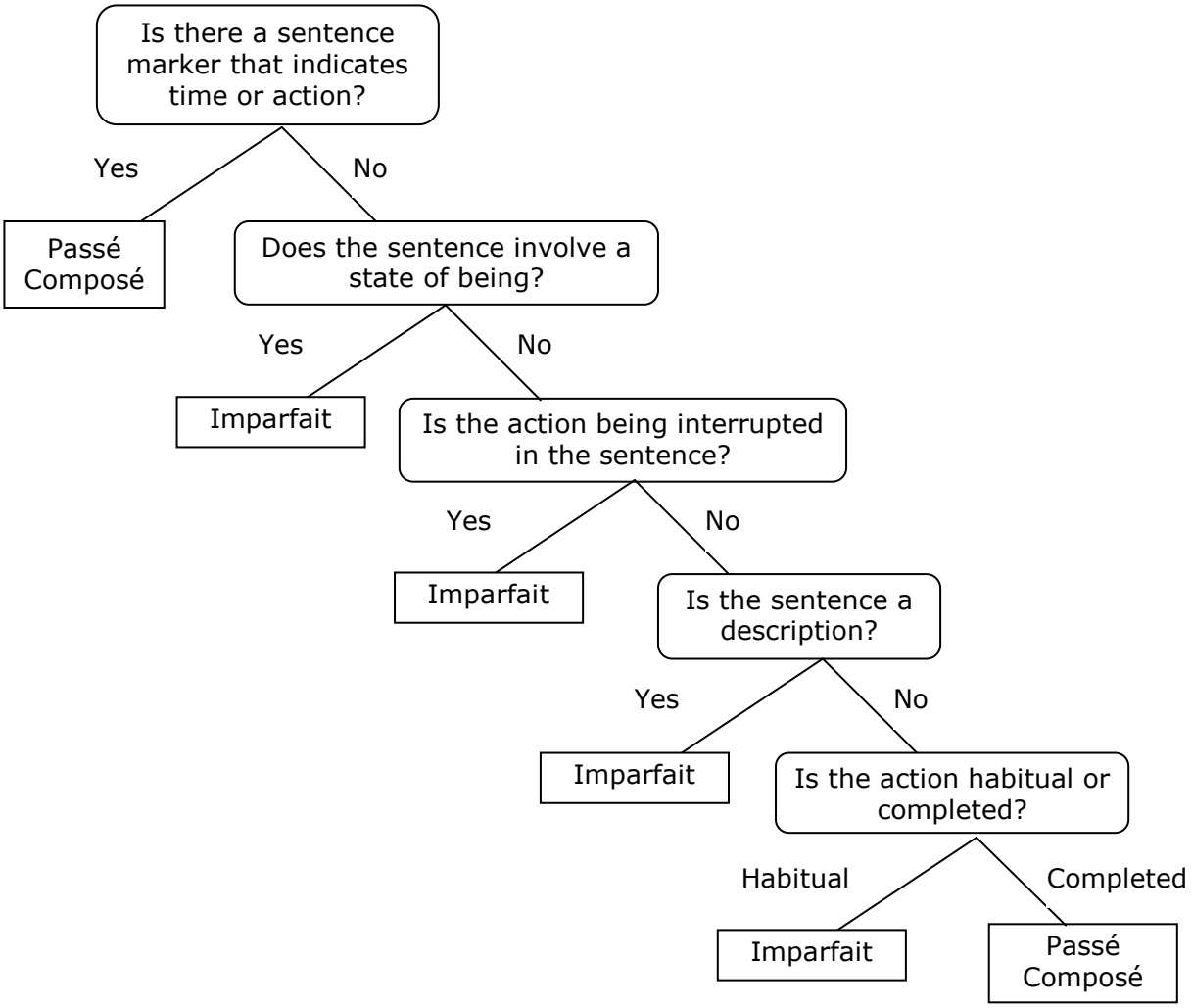

Figure 1: Decision Tree used in original model

Think-alouds

Our next step was to collect data using a think-aloud procedure (Ericsson & Simon, 1984). In a think-aloud, individuals are asked to verbalize all thoughts and actions while solving a problem. Participants are not asked to explain or justify what they are doing nor are questions posed by the experimenter during the process. The result is a stream of conscious record of the actions being performed and is thus valuable for collecting data to build or refine a model. Think-alouds have been shown to not significantly interfere with the problem-solving processes (Schooler et. al., 1993). We performed think-alouds on both expert and novice subjects.

We attempted to refine our model of expert performance on the aspect identification task by conducting a think-aloud with four French professors. We generated twenty initial problem sentences, which were simple and without context, and prompted experts to think out loud as they solved the problems. The results did not support our model. Experts did not appear to go through a sequential process of decision making or base their choice on the rules in our model. There was never a response in the explanation that indicated they were taking a 'no' branch of the decision tree. Secondly, the first three rules of the tree were never mentioned. Finally, their explanations were holistic, often using phrases such as "This sentence is ..." rather than referring to individual parts of the sentence.

We used three subjects for the novice think-aloud. All had completed elementary French instruction so in theory should have been able to complete the task. However, their level of expertise made a great difference in performance. Our first subject had one semester of French and didn't remember conjugations other than the present tense. The second subject had the most French experience, and completed the task with a high level of automaticity. He could not explicitly articulate the process he followed when making the aspectual decisions. The third subject had sufficient French experience, but since time had elapsed since her last instruction she had difficulty completing some of the sub-steps (e.g. understanding the sentence, identifying the tense, conjugating the verb) of the task. Due to the variation in student performance, we were limited in the amount of process data we were able to collect.

Discussion

Our expert and novice think-aloud protocols revealed three main issues with our model and techniques. First, there were problems with the data-collection methodology. Our task was structured so that our novice think-aloud participants did not provide verbal protocols that could decisively confirm or disconfirm our model. In fact, there seemed to be a narrow window of participant expertise where the process of solving this type of task can be verbalized; two of our participants were too inexperienced and thus did not have the necessary knowledge to complete the task let alone explicitly state the rules being applied, and one of our participants was too expert and relied on implicit knowledge to complete the task. One solution is to include additional scaffolding that would elicit more information about participant problem-solving strategies at all levels. Despite the limitations of the problem design, we were able to collect some data from our expert and novice think-alouds. However, the data did not appear to confirm our model. In our design of the decision tree, we did not differentiate between rules, which are correct all the time but can be difficult to apply, and heuristics, which are not always correct but can be easier to apply. More seriously, it appears that the decision tree representation is not appropriate for this ill-defined domain. Based on the expert performance, a task analysis which incorporates a representation that includes the application of rules in any order and allows the holistic processing of sentences would be more appropriate.

ADAPTING MODEL-BUILDING TECHNIQUES

Rational Task Analysis

Our first step in adapting traditional modeling methodology was to develop an alternative representation that is more suitable for use in this class of ill-defined domains. Instead of a decision tree, we adopted a model loosely based on scientific experimentation that better explains the data from our initial exploration and better fits the parameters of this ill-structured task. In this model, as individuals begin to solve the problem, they keep in mind two competing hypotheses: *H1* – The sentence should use the passé composé, and *H2* – The sentence should use the imparfait. The individual applies methods (or rules) for gathering evidence that support one of the hypotheses with a certain degree of weight. The weight depends on the difficulty of the problem and the experience of the student. After students have gathered enough evidence in favor of a hypothesis (i.e., the level of supporting evidence crosses a certain threshold), they make a decision. Notice that this representation can be translated into English production rules such as: 1) IF method *A* yields result *B*, THEN increment evidence for *H1* by *C*, and 2) IF *D* is the amount of evidence for *H1*, *E* is the threshold for accepting *H1*, and $D > E$, THEN accept *H1*.

Using this representation, it is easier to formalize the rules and heuristics relevant to identifying the aspect of a sentence. Experts read a sentence from beginning to end and apply the three methods in Table 2 simultaneously. The evidence provided by these methods leads them to select the tense of the sentence. For an expert, at least one of the methods always provides evidence strong enough to confirm one of the hypotheses. It

is important to note that although Table 2 represents a complete mapping of the problem space, individual expert knowledge ranges from this level of specificity to a much more general representation (e.g., one containing only the complete/incomplete distinction, but covering all cases).

Method	Evidence for Passé Composé	Evidence for Imparfait
Determine the manner of action	Action is one time Action is repeated (specific number of times)	Action is habitual Action is repeated (generalized)
Determine the duration of action	Action has a start Action has an end Action has a specific duration Action is completed	Action is incomplete Action is in progress
Determine the role of action in sentence	Action is an event Action is a change	Action is a description Action is a context

Table 2: Available expert problem-solving methods

In this model, an expert might read the sentence, "It was raining" and apply the three methods simultaneously. The manner and the duration of the action are not entirely clear, and would produce results that weakly confirm the hypothesis that the action is in the imparfait. However, the role of the action is clearly a description, and strongly confirms the hypothesis. The expert would therefore choose the imparfait tense. It is important to note that we are treating these evidence-gathering methods as black boxes, and avoid describing exactly how one determines that a sentence is a description.

This evidence-gathering approach to task analysis also gives some insight into where novices may make errors. There are four areas where novice performance might differ from expert performance. First, novices may be unaware of all the methods they can use to determine the aspect of the sentence. For example, they may not know that the duration of the action relates to the aspect. Second, novices may know that they should be using a method, but may be unable to do so. For example, they may be unable to identify the duration of an action. Third, novices may understand that a given method provides evidence, but be unsure of what the evidence means. For example, they may know that the duration of the action is important, but forget whether it provides evidence for the passé composé or imparfait tense. Finally, novices may be aware of a method and able to apply it correctly to support a particular hypothesis, but they may be unclear to what extent the evidence supports the hypothesis.

In addition, novices may compensate for their lack of expertise by using imperfect methods for gathering evidence. Three such methods are listed in Table 3. Although these heuristics can supplement expert methods for determining the tense, they sometimes yield erroneous results. If the two pieces of evidence gathered are conflicting, the novice uses the weighting of each piece of evidence to resolve conflicts and determine whether the tense is passé composé or imparfait.

Method	Evidence for Passé Composé	Evidence for Imparfait
Identify a temporal keyword	Word signifies a completed action or an action which occurs a specific number of times	Word signifies a habitual or incomplete action
Identify verb type	There is an action verb in the sentence	There is a state verb in the sentence
Identify an interruption	There is an interrupting clause	There is a interrupted clause

Table 3: Supplementary novice problem-solving heuristics

We believe that this model addresses the concerns that arose from the decision tree-based model. Namely, unlike the previous model which made no mention of implicit knowledge, now implicit knowledge is accounted for by the existence of evidence gathering methods at different levels of abstraction. Expert knowledge can be represented as a single evidence gathering method ("Identify the completeness of the action"), rather than the more specific methods we have described. Moreover, rules can be applied in any order, represented by specifying a list of evidence gathering methods than can be used, rather than a sequence of actions that must be performed to solve the problem. Finally, the approach handles uncertainty by specifying a threshold for accepting a given hypothesis, and allowing different evidence gathering methods to be weighted based on characteristics of the problem and the skill of the problem-solver. This uncertainty also allows heuristics, or

rules that do not necessarily work all the time, to be counted as valid evidence gathering methods that can be used by novices. Following traditional modeling procedure, this model was then refined using data from additional think-aloud protocols.

Think-aloud Protocols

As evidenced by the first think-aloud study, simply observing student output fails to capture the underlying processes by which the student arrived at his or her final decision. As such, we supplemented our primary task with secondary scaffolding tasks that elicited richer responses from the student.

The first task we added was requiring students to provide an explanation for their answer in addition to the answer itself. When students are required to explain their decisions, the tutor learns the rule the student is trying to apply. We experimented both with rule-based and freeform explanations. Further, self-explanation, both free-form and rule-based, has been shown to increase student learning in intelligent tutoring systems [Aleven, 2002].

Unfortunately, simply providing an explanation does not necessarily yield information regarding the features of the sentence that lead students to initially apply the rule. For this information, we asked students to identify a sentence comparable to the problem sentence with respect to the features for choosing aspect. This process highlights the features students use to make their decision. With the added scaffolding, not only is the student required to actively process the text but insight into the process being used to derive the answer is now available to the tutor; they are no longer simply making a binary decision. Forcing students to make comparisons between examples is particularly helpful for teaching feature discrimination in ill-defined domains [DeKeyser, 2005].

Finally, we added some instruction at the beginning of the task, based on our model of expert performance, in order to remind students of the uses of the aspects and insure that all participants shared a common foundation for the domain. Additionally, based on interviews with experts, we determined that context is critical for removing ambiguity in the selection of aspect. For that reason, we opted to situate the individual problem sentences within a paragraph. See Figure 2 for a screenshot detailing the types of scaffolding provided.

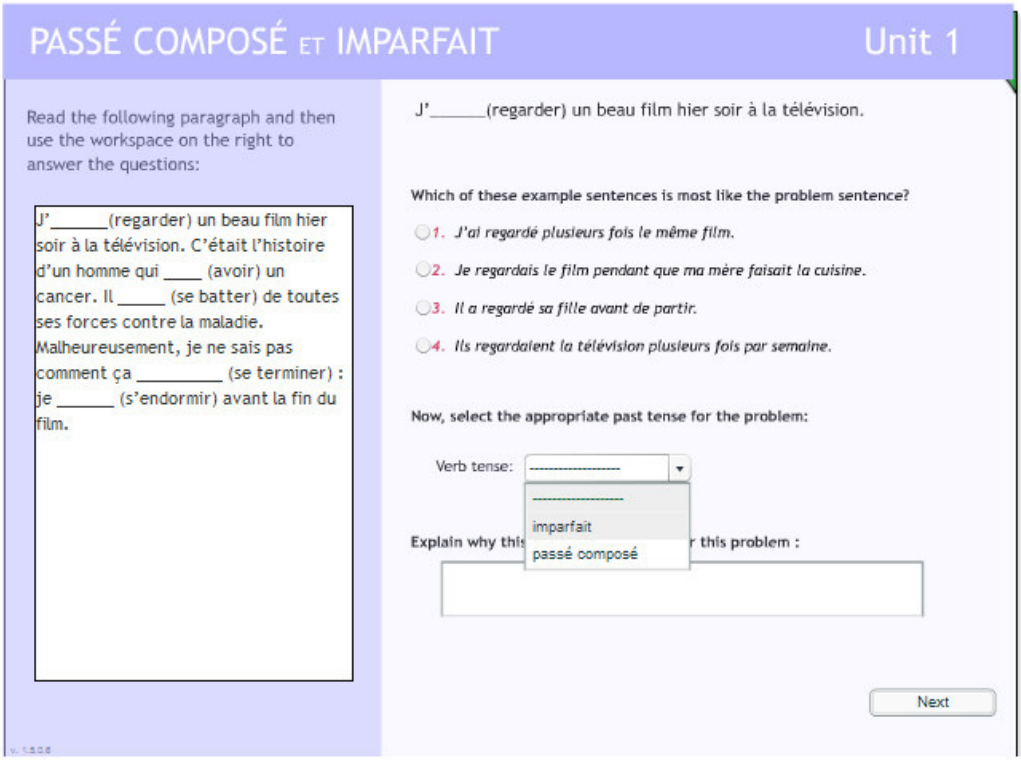

Figure 2: Screenshot of tutor interface with context and scaffolding

We collected think aloud data from six novice French speakers between the ages of 18-29. All participants had recently completed at least the first semester of college French (or equivalent) but had not continued beyond the second semester. Participants had differing levels of exposure to the passé compose and the imparfait. Students were given a pretest, instruction regarding the use of aspect, and then asked to think out loud as they solved the task. Students completed this activity for four different paragraphs with four different types of scaffolding (no scaffolding, comparison example, self-explanation, combined comparison example and self-explanation). During this phase, students were given immediate feedback on their performance. Each session concluded with students completing a post-test and transfer assessment.

Results

Much of the behavior we saw supported our original model. Since students were primed with the rules and heuristics of the model during the instruction phase, it is not surprising that the language they used when explaining their reasons was similar to that which was presented. However, the fact that they were able to acquire and successfully use the model given only brief exposure (average time reading instruction = 4.5 minutes) might suggest that our model doesn't deviate much from students' existing models. We also found some direct evidence for the idea that students were in fact conducting evidence gathering when making their decisions. When presented with conflicting evidence, students would verbalize the conflict and look for other features to support one aspect or the other. Even when conflicts were absent, some students were reluctant to make a decision based on limited information. For example, P1's behavior and statement "[the sentence] doesn't say exactly when it happened, no specific time, but it's an event" suggest that she was trying to gather more evidence that the sentence was in the passé compose before making a final decision.

Perhaps more interesting is our model also accurately predicted areas where students would have difficulty. We proposed four main areas where novice behavior may differ from expert behavior. Each of these are re-examined and supported by samples of student behavior:

1. *Unawareness of evidence gathering methods* -- It was common for participants to rely on a handful of well-known rules and attempt to classify the aspect based on this small subset only. For example, P1 never used description as a way of determining aspect, even when it would have been appropriate, suggesting that she was not aware of that method for gathering evidence.
2. *Inability to apply evidence gathering methods correctly* -- Students also failed to correctly apply the methods even when it was clear that they correctly understood the concepts behind them. For example, when determining the aspect of the following sentence: "Nous *avons dû* ranger les affaires en vitesse", the student failed to recognize that "en vitesse" is a time keyword. We know that the student was looking for a keyword because transcripts of the session show the student incorrectly acknowledging that there is "no specific mention of time".
3. *Inability to link the results of methods to particular hypotheses* -- Only one participant, P4, exhibited frequent incorrect mappings between the evidence in the sentence and the hypothesis being supported. In one exercise, she used the explanation, "a one time action" to explain both uses of the imparfait and passé composé. Later, the incorrect mappings between results and hypotheses became more evident when she commented that specific duration and completed action meant using the imparfait when in fact the passé compose should be used.
4. *Lack of understanding of how much evidence contributes to a given hypothesis* -- We also saw evidence that suggested novices placed too much weight on some heuristics, often failing to continue examining the sentence. For example, P2, heavily weighed the use of time keywords. Upon noticing one in the sentence, she automatically chose the passé composé even when other evidence in the sentence suggested otherwise.

Participants used several heuristics when solving the problem, only a handful of which were identified in the previous model. A table of the heuristics including the participants who used them follows:

Method	**Passé Composé**	**Imparfait**
Identify the type of verb	Action verb (2, 4, 6)	State verb (1, 3, 7) Not an action verb (2)
Identify the specificity of the action	Specific (1, 4, 6)	Not specific (1, 4, 6) Vague statement (1)
Identify the suddenness of the activity	Happened all of a sudden (2) Interruption word (4, 7) Happened once (2, 3, 6, 7)	Something that occurs frequently (1) Ongoing Activity (3) Habit (3, 4, 6, 7)
Identify a mention of times	Time keyword (2, 3, 6)	No mention of exact time (1, 2) Unspecified number of times (3, 4, 7)
Look for a comparable verb	The tense of the other verb is passé composé (2, 6)	The tense of the other verb is imparfait (2, 6)

Table 4: Heuristics used by novices in second Think-Aloud

Discussion

The preliminary results suggest that the model developed using a combined theory-driven and data-driven approach is a decent representation of student behavior, and an improvement over the previous model. The new model is based on a representation that is suitable for ill-defined domains: It incorporates holistic sentence processing, the application of rules in any order, and allows both rules and heuristics to be employed by students.

In addition, we improved our data-collection methodology so that student data could better inform the model. The added prompts for self-explanation and comparison to examples lead students to talk more about the rules and sentence features they were using to solve the problem, and exposed some misconceptions that would not otherwise have been clear. Ultimately, the data collected fits the model. Students show an evidence gathering approach to identifying aspect, and use the rules and heuristics that we have described to help them arrive at the correct answer. It is important to note that this model is preliminary, and serves as a plausible interpretation of the data. Further validation and specification of the model will be necessary.

In particular, there are still unanswered questions with respect to the completeness of the model. Because we provided students with the rules involved in the model when we gave them instruction on identifying aspect, we could not fully evaluate whether the rules map to actual novice performance. However, these rules are similar to instruction that students actually receive on the difference between the passé composé and the imparfait, so they do represent knowledge that novices should have already been exposed to, and represent knowledge that students are expected to learn. Therefore, the fact that students employed the rules after limited exposure to them might suggest that a student model for solving the problem is similar to the theory-driven model that we described. Additionally, we did not do a full analysis of the "helping" strategies that students used to solve problems such as translation of the sentence or looking for context in surrounding sentences. In the future, we intend to incorporate those types of heuristics into our model. Finally, we limited our analysis to a high level of abstraction, and did not look at how students parse the sentence to identify features that may be relevant for the conditions of the rules in our model. We believe that scaffolding and tutoring mechanisms can be built from our model which do not require such a fine grain of analysis to be effective.

FUTURE DIRECTIONS

Developing an accurate student model marks the beginning of full tutor development. With our current model, we plan to create a full cognitive tutor to be deployed and evaluated in an online French course. The tutor will likely incorporate the scaffolds used during the think-aloud procedure, but future studies are also planned to identify the exact combination of scaffolds that lead to the greatest learning gains. During this process, we will continue to refine and evaluate our model for identifying aspect in the French past tense. We hope that our model, representation, and techniques can be generalized to other ill-defined domains where the rules are ill-structured but the start and goal states are easily formalized.

ACKNOWLEDGMENTS

Thanks to Dr. Christopher Jones, Dr. Vincent Aleven, Dr. Ken Koedinger, Anne Catherine Delmelle, and Alida Skogsholm for their help with this project.

REFERENCES

Aleven, V., & Koedinger, K. R. (2002). An Effective Meta-cognitive Strategy: Learning by Doing and Explaining with a Computer-Based Cognitive Tutor. *Cognitive Science, 26*(2), 147-179.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . (2004). An integrated theory of the mind. *Psychological Review 111,* (4). 1036-1060.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. W. (2002). Learning from examples: Instructional principles from the worked examples research. Review of Educational Research.

Cadierno, T. (1995) Formal Instruction from a Processing Perspective: An Introduction to the French Past Tense. Modern Language Journal. 79. 179-193.

Comrie, B. (1976). Aspect. Cambridge: Cambridge University Press.

Davis, Tessier, (1996). Authoring and Design for the WWW. Advisory Group on Computer Graphics.

DeKeyser, R. (2005) What Makes Learning Second Language Grammar Difficult? A Review of Issues. *Language Learning, 55*, 1-25

Ericsson & Simon (1984). Ericsson, K. A, & Simon, H. (1984). *Protocol analysis: Verbal reports as data.* Cambridge, MA: MIT Press.

Gamper, J. & Knapp, J. (2002). A review of intelligent CALL systems.Computer Assisted Language Learning, 15(4):329–342.

Ishida, M. (2004) Effects of Recasts on the Acquisition of the Aspectual Form te i(ru) by Learners of Japanese as a Foreign Language. Language Learning 54:2, 311-394.

Jonassen, D.H., Tessmer, M., & Hannum, W.H. (1999). Task analysis methods for instructional design. Mahwah, NJ: L. Erlbaum Associates.

Koedinger, K. R.; Anderson, J. R.; Hadley, W. H.; and Mark, M. A. (1997). Intelligent Tutoring Goes to School in the Big City. Journal of Artificial Intelligence in Education 8(1): 30–43.

Mayo, M., & Mitrovic, A. (2001). Optimising ITS behavior with Bayesian networks and decision theory. *IJAIED*, *12*(3), 124-153. Project homepage at http://www.cosc.canterbury.ac.nz/~tanja/capit.html

Norris, J., and Ortega, L. (2000) Effectiveness of L2 Instruction: A Research Synthesis and Quantitative Meta-analysis. *Language Learning, 50*(3), 417-528.
Ormerod, T.C. (2006). Planning and ill-defined problems. Chapter in R. Morris and G. Ward (Eds.): The Cognitive Psychology of Planning. London: Psychology Press.
Robert M. DeKeyser. (2005) What Makes Learning Second Language Grammar Difficult? A Review of Issues. Language Learning 55:s1, 1-25.
Salaberry, R. (1998). The development of aspectual distinctions in L2 French classroom learning. The Canadian Modern Language Review, 54, 508-542.
Schooler, J.W., Ohlsson, S., and Brooks, K. (1993). Thoughts beyond words: When language overshadows insight. *Journal of Experimental Psychology: General 122*, 166-183.
Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology,* 8, 481-520.
Simon, H. (1973). The structure of ill-structured problems, Artificial Intelligence, 4:181-201.
Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1995). Cognitive Flexibility, constructivism, and hypertext: Random access instruction for advance knowledge acquisition in ill-structured domains. In P. Stele, & J. Gale, *Constructivism in education* (pp. 85-108). Hillsdale, NJ: Erlbaum.
von Stutterheim, C. (1991). Narrative and description: Temporal reference in second language acquisition. In Crosscurrents in second language acquisition and linguistic theories (pp. 385-403). Philadelphia: Benjamins.

Using Prolog Design Patterns to Support Constraint-Based Error Diagnosis in Logic Programming

Nguyen-Thinh Le
Department of Informatics
University of Hamburg
le@informatik.uni-hamburg.de

Abstract. Logic programming provides many possibilities to implement a task. Solutions can be realized by applying a variety of different design strategies and programming techniques. Using constraint-based modeling (CBM) we are in a position to cover the solution space for a given programming task. In this paper, we investigate the CBM approach for diagnostic purposes in the domain of logic programming. We address the ill-structuredness of this domain and the complexity of constraints modelling variations of design strategies. We propose an approach to structure the domain of logic programming by using Prolog patterns and to relieve the complexity of constraints by hypothesizing the design strategy in the student's solution.

Keywords: diagnosis, constraint-based modeling, logic programming, programming techniques, Prolog patterns.

INTRODUCTION

Prolog is one of the most widely used logic programming language. Prolog is considered to be difficult to learn because of the simple syntax and the concept of recursive programming which is the most important programming technique (Taylor 1999; Taylor & Boulay 1987). In addition, the domain of logic programming is infinite. For a given programming task, there is no single solution, but many strategies to design a solution. For a strategy, there are many ways to implement it. This causes students to search for a correct program for the given task.

Over the last two decades, numerous error diagnosis approaches in the domain of programming languages have been devised, such as program transformation (Vanneste, 1994; Xu and Chee, 2003), program verification (Murray, 1988), plan and bug library (Weber, 1996), model tracing (Anderson and Reiser, 1985) and constraint-based modeling (Ohlsson, 1994; Mitrovic et al, 2001). Among these, model tracing is used by cognitive tutors which are some of the most successful ITS today (Koedinger, Anderson, Hadley & Mark 1997).

The model tracing approach keeps track of the student's programming process and tries to guide him towards expert programming behavior. Possible actions a student might take are described by means of production rules. By tracing the actions of the student with a collection of these rules, model tracing systems are able to return feedback immediately, whenever students perform a "bad" action. Model tracing approach has been applied to build a tutor system for Lisp (Anderson&Reiser 1985). However, the model tracing tutor is unable to trace the student's input when a student follows an unexpected but correct strategy.

While the model tracing technique has been widely applied for developing cognitive-motivated tutor systems (Martin 2001), recently, the constraint-based modeling (CBM) approach has been showing great promise as a diagnostic approach (Mitrovic et. al., 2001) which focuses on static cognitive states rather than problem solving processes. This approach has been employed successfully to build an SQL tutor system (Mitrovic & Ohlsson, 1999), natural language (Menzel, 2005), in the domain of data structures (Warendorf & Tan 1997) and has also been researched in the domain of UML (Baraghei, 2005).

In this paper, we investigate the CBM approach for diagnostic purposes in the domain of logic programming. We address the ill-structuredness of this domain and the complexity of constraints modelling variations of design strategies. We propose an approach to structure the domain of logic programming by using Prolog patterns and to relieve the complexity of constraints by hypothesizing the design strategy in the student's solution.

LOGIC PROGRAMMING

We intend to support students of Computer Science doing homework of logic programming course by using our tutoring system. The system provides students with task assignments and requests them to submit their program for the given task in a free form. A logic program consists of the definition of a main predicate and several auxiliary predicates which are necessary to solve sub-problems in the main predicate. A predicate definition is comprised of clauses and each clause has a clause head and several goals as the predicate `reverse/2` in Figure 1 illustrates. One of the main execution techniques of logic programs is unification. Unification is the way Prolog

does its matching. Two terms match, if they are equal or if they contain variables that can be instantiated in such a way that the resulting terms are equal. (Blackburn, Bos and Striegnitz, 2001). Co-reference relationship is referred as a special case of unification which applies for arguments and functors.

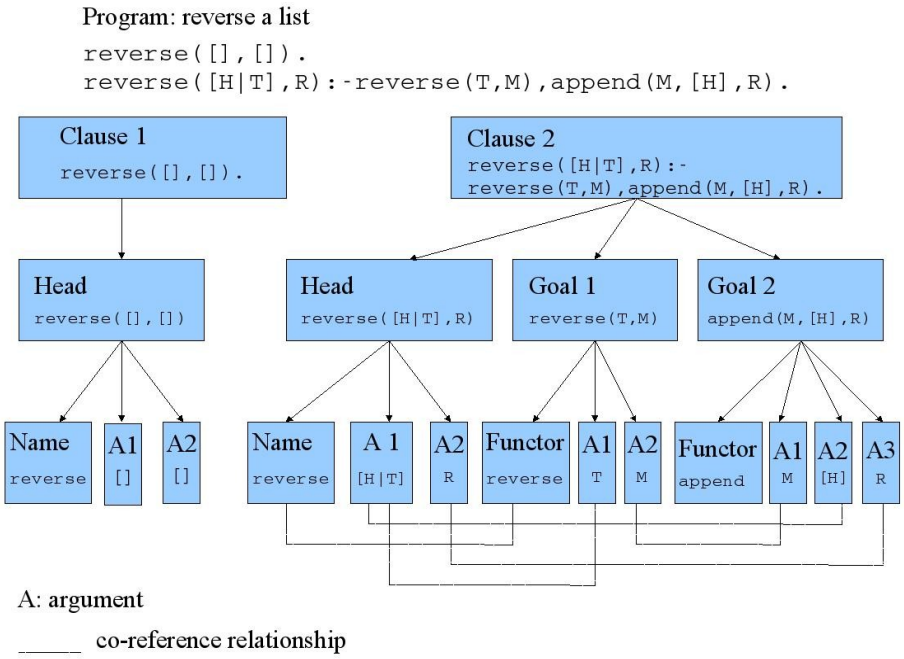

Figure 1 Structure of a Prolog predicate definition

Logic programming is an open-ended domain. The solution space for a programming task is infinite. There are three cases where solutions can be varied:

1. Different algorithms and design strategies provide various implementations. For instance, for the task of reversing a list we can apply different design strategies among which the common ones are e.g. naive recursion, inverse naive, railway-shunt and accumulator. Those design strategies are explained later on in this paper. For different strategies and algorithms, there are different implementations as the following example shows.

Naive	*Inverse naive*

`reverse([],[]).`
`reverse([H|T],R):-`
`  reverse(T,M), append(M,[H],R).`
	`reverse([],[]).`
	`reverse(L,[H|T]):-`
	`  append(M,[H],L), reverse(M,T).`

2. Different programming techniques can be applied for the same purpose. For instance, to realize a unification, we can choose between implicit unification and explicit unification. To implement explicit unification different operators can be applied. In the following clause, the second argument of the clause head unifies with the third argument of the call "`append`" implicitly:
   `reverse([H|T],R):-reverse(T,M),append(M,[H],R).`
   We also can implement that unification explicitly using **`R1`** and **`R2`** for different arguments and the unification between them is coerced by the operator **`"="`**.
   `reverse([H|T],R1):-reverse(T,M), append(M,[H],R2),R1=R2.`

3. Auxiliary predicates can be defined according to individual needs. There are two reasons to define an auxiliary predicate:
   1. To execute a necessary subtask and enable the re-usability. The *Naive* definition for `reverse/2` above needs the auxiliary predicate `append/3` in order to concatenate an element to the end of a list. For the purpose of composition an element to the end of a list, one can use `append/3` or define an auxiliary predicate `enqueue/3` which queues a single element at the back of a list. Although the our system provides students the possibility to select appropriate built-in predicates like `append/3`, sometimes students still want to define their own auxiliary predicate like in the following *Naive* definition for `reverse/2`:
      ```
      reverse([],[]).
      reverse([H|T],R):-reverse(T,M), enqueue(H,M,R).
      enqueue(X,[],[X]).
      ```

```
enqueue(X,[H|T],[H|R]):-enqueue(X,T,R).
```

2. To keep the code in the main predicate definition simple. For example, the task of determining a list of persons whose age is greater than 18 can be implemented in `adult/2` with or without using an auxiliary predicate, where each element of the person list is a pair of name and age. The test subgoal A>=18 in Solution 1 is replaced by the call of an auxiliary predicate `greater18/1` in Solution 2.

Solution 1:	*Solution 2: using auxiliary predicate*

`adult([],[]).`
`adult([(N,A)|T], [(N,A)|R]):-`
`  A>=18, adult(T,R).`
`adult([(N,A)|T], R):-adult(T,R).`
	`adult([],[]).`
	`adult([(N,A)|T], [(N,A)|R]):-`
	`  greater18(A), adult(T,R).`
	`adult([(N,A)|T], R):-adult(T,R).`
	
	`greater18(X):-X >= 18.`

For a programming task, the number of applicable algorithms and programming techniques is limited. However, the variants of design strategies are numerous and the space of auxiliary predicates which might be defined by each individual is not predictable. Thus, the domain of logic programming is ill-defined. We apply the CBM approach to model the domain of logic programming and specify Prolog patterns to structure this domain.

CONSTRAINT-BASED MODELING

The CBM approach is proposed in (Ohlsson, 1994) to model general principles of a domain as a set of constraints. A constraint is represented as an ordered pair consisting of a relevance part and a satisfaction part:

`Constraint C = <relevance part, satisfaction part>`

where the relevance part represents circumstances under which the constraint applies, and the satisfaction part represents a condition that requires to be met for the constraint to be satisfied.

A constraint is used to describe a fact, a principle or a condition which must hold for every solution contributed by the student. For example, the following principle can be formulated as a constraint:

Example 1: a Prolog principle an arithmetic expression of a logic program, e.g. "A is X+Y" can only be evaluated if all variables of the right hand side are bound.

Constraint 1:
Relevance: X is a variable of the right hand side of an arithmetic expression,
Satisfaction: X ought to be bound.

Constraints are not only used to circumscribe facts, principles or conditions of a domain, they can also be used to specify the requirements of a task or to handle solution variations. Using the relevance part, constraints can be tailored according to an ideal solution, which represents the requirements of the given task. Ideal solutions enables us to check whether the student has answered the problem correctly, looking at the semantics. Additional requirements, which have to be satisfied in that specific situation, can be specified in the satisfaction part. We take an example from (Suraweera, Mitrovic and Martin, 2005) and specify the constraint 2 to examine the operator required in the task.

Example 2: a task requirement
"When I went to the shop to buy two loafs of bread, I gave the shopkeeper a $5 note and he gave me $1 as change. Write an expression to find the price of a loaf of bread using x to represent the price". It can be represented as 2x + 1 = 5 or 2x = 5 - 1.

Constraint 2:
Relevance: the left hand side of the ideal solution has a +,
Satisfaction: in the student's solution, the left hand side has a + OR the right hand side has a -

If a constraint is violated, it indicates that the student solution does not hold principles of a domain or it does not meet the requirements of the given task. In order to be able to evaluate constraints, we need to define a formal representation for constraints.

Constraint's formal representation

We define a constraint as a tuple: (Type, Relevance, Satisfaction, Severity, Position, Hint).

- *Type* is "pattern", "exercise" or "general". Constraint types are used to control the diagnosis process;
- *Relevance* is a relevance part;
- Satisfaction is a satisfaction part;

- *Severity* indicates the severity of the constraint, it ranges between zero if the constraint is very important and one if the constraint is just informative;
- *Position* is the position of the structural elements which are considered in the relevance part. Position is only instantiated when the constraint is relevant and is used to indicate the error location in the student's solution in case the satisfaction part is not fulfilled;
- *Hint* is an instructional message explaining a principle of the domain or a requirement of the given task.

Syntactically, the relevance part and the satisfaction part are logical expressions, i.e. conjunctions of propositions about a problem state (Ohlsson, 1993). However, a problem state cannot be identified directly from a (partial) structure of a Prolog program due to two reasons:

First, the structural elements (clauses, goals, arguments, and functors) of a Prolog program are related to each other not only horizontally but also vertically (Figure 1). A Prolog program can be parsed of three levels: clause level, goal level and argument/functor level. From the horizontal view, the clause order and the goal order determine the relationship between clauses and goals, respectively. The unification and the argument order determine the relationship between arguments themselves and between them and functors. From the vertical view, the existence of an argument within a goal and the existence of a goal within a clause indicate the relationship between an argument and a goal, between a goal and a clause, respectively. The relationship between structural elements of a Prolog program is so complex that a partial structure of a Prolog program can not reflect a problem state sufficiently.

Second, more information can be detected, e.g. the instantiation state of an argument or the type of an argument, if a structural element is observed on a whole picture of its corresponding program. The instantiation state of an argument is obtained by using the predicate declaration and inferring instantiation states for all arguments from the begin of a clause to the occurrence of the argument being considered. Such information cannot be read off if only a partial structure of a Prolog program is considered.

As a result, we need to extract information about horizontal and vertical relationships between structural elements from a given Prolog program and a given predicate declaration in order to create three types of facts to serve the relevance part and satisfaction part:

headargument(Creator, ClauseIndex, ClauseType, AuxiliaryPredicateList, HeadName, HeadLength, ArgumentIndex, ArgumentType, ArgumentValue)[1]

bodyargument(Creator, ClauseIndex, SubgoalIndex, Functor, SubgoalType, SubgoalLength, ArgumentIndex, ArgumentType, ArgumentValue)

argumentmode(Creator, ClauseIndex, SubgoalIndex, ArgumentIndex, Value, InstantiationState)

The facts *headargument* and *bodyargument* contain information about each argument in the clause head and in the clause body, respectively. If a fact of type *argumentmode* exists, it expresses that the argument is bound, after its corresponding subgoal has been executed.

Relevance parts and satisfaction parts can be specified as conjunctions of the facts of *headargument, bodyargument* and *argumentmode*. If additional functions are necessary for constraint evaluation, they can be added into the specification of relevance parts and satisfaction parts. The constraint evaluation is carried out as follows: First, the relevance part of the constraint is matched against the facts extracted from a Prolog program. If there is a match, i.e. the constraint is relevant to the program, and then the satisfaction part is matched against the facts. If the satisfaction part is fulfilled, then the Prolog program is considered to be correct regarding to that constraint. Otherwise, it indicates a shortcoming in the program and the corresponding information will be returned for instructional purposes: the position of the structural element considered in the relevance part, the constraint severity and the hint encoded in the constraint.

Applying the CBM for Prolog

We categorize constraints into two classes: Prolog general constraints and exercise specific constraints. The former class represents general principles and conditions of Prolog. The constraint 1 mentioned above is an example of this class. The second class considers the specific requirements of the given task. Constraint 2 above is an instance of this type.

Each exercise specific constraint requires an ideal solution which encapsulates the requirements of a task. We extract facts (*headargument, bodyargument, argumentmode*) from ideal solutions to specify constraints. The information we want to attain from ideal solutions is: What kind of elements exist? What kind of relationship exists between different elements? In logic programming, for a design strategy, there are many various implementations. Therefore, it is necessary to define a canonical normal form for ideal solutions. Using ideal

[1]Creator: the program being considered is created by the student or the human tutor; ClauseIndex: the index of the clause within the program; ClauseType: the clause being considered is a base case or a recursive case; AuxiliaryPredicateList: a list of predicates which are used for the current clause; HeadName: the name of the clause head; HeadLength: the number of arguments the clause head consists of; ArgumentIndex: the index of the argument within the clause head; ArgumentType: an argument can have one of the types: variable, anonymous variable, list, atom, number, peano number, arithmetic or arbitrary; ArgumentValue: the structural representation of the argument itself; SubgoalIndex: the index of the subgoal within the clause body, beginning with number one; Functor: the name of the subgoal's functor; SubgoalType: a subgoal can have one of the types: arithmetic test, calculation, list manipulation, term test, user defined, relation, recursion or unknown; SubgoalLength: the number of arguments the subgoal consists of; InstantiationState: an argument is bound or free.

solutions, we can specify constraints which are in a position to cover solution variations. In Prolog, there are four levels of solution variations. 1) variation of operations over arguments, e.g. unification can be used implicitly or explicitly or a term comparison can be expressed using: either `X<Y` or `Y>X` or `not(X>=Y)`; 2) variation of the subgoal order, e.g. two subgoals can be transposed without changing the correctness of a program; 3) variation of the clause order, e.g. the order of two clauses can be changed while the semantics is preserved; 4) variation of implementation strategies for the same programming problem.

The following constraint which is able to cover different unification implementations for list arguments in a base case is a constraint of the first variation level.

Constraint 3:
Relevance:
the predicate of the ideal solution is a single recursion AND
there exists 1 base case AND
its clause head contains 2 arguments [H|R] at position Pos1 AND [H|T] at position Pos2 AND
the predicate of the student's solution is a single recursion AND
there exists 1 base case AND
its clause head contains 2 arguments [X1|Y1] at position Pos1 AND [X2|Y2] at position Pos2
Satisfaction:
in the student's solution the value of X1 must be the same as the value of X2 OR
there must exist a subgoal "X1 = X2" or "X2 = X1" in the body of the base case.

Similarly, we can specify constraints to cover solution variations of the subgoal level and the clause level. However, the CBM seems to encounter limitations to handle solution variations on the design strategy level. Exercise specific constraints modelling solution variations on this level are very complex. If the specification of the constraint is too complex, then it might happen that due to a slightly deviation the constraint will no longer be relevant to that solution.

The complexity problem

From Constraint 2 and Constraint 3 we derive a general formula for exercise specific constraints which should cover different implementations:

Formula 1:

C_r(IS, SS, relevantobject) → SS = C_s(IS) v SS = variant(C_s(IS))

where the left hand side and the right hand side of the → represent the relevance part and the satisfaction part, respectively[2]. C_r(IS, SS, object) refers to the problem state which is described by the `relevantobject` in the ideal solution and the student's program. The `relevantobject` can be an operation, a goal, a clause or a design strategy. SS = C_s(IS) v SS = variant(C_s(IS)) means the student's solution matches the state required in the ideal solution or a variant of that state.

Deriving from Formula 1 we define Formula 2 for constraints which cover solution variations of the design strategy level.

Formula 2:

C_r(IS,SS,strategy) → SS=C_s(IS,strategy) v SS=variant(C_s(IS,strategy))

At this point, several questions will arise: 1) what are the elements which characterize a design strategy such that it can be distinguished from another one? 2) How can we specify the relevant part and the satisfaction part for a constraint which should cover different design strategies? 3) Assuming, we are able to specify such constraints, will they be relevant to a slightly erroneous solution, which is intended to be implemented following that strategy? In order to address these questions, we should have a look at the *Naive* implementation and *Inverse* naive implementation of the task `reverse/2` which reverses a list.

The first question can be answered by deriving from the *Naive* implementation of `reverse/2` the following description: a base case exists, a recursive case exists, the input list is decomposed in a head and a tail and the tail is decomposed recursively. The characteristic of the *Inverse naive* implementation can be described as follows: a base case exists, a recursive case exists, the input list is decomposed into a front list and a last element, the front list is decomposed recursively.

The second question can be answered by applying the descriptions above to specify the relevance part and the satisfaction part of an exercise specific constraint. The relevance part describes the characteristic of a design strategy and the satisfaction part describes additional requirements. The specification for such a constraint will be obviously very complex as the following Constraints 4 and 5 show, where **Naive** and **Inverse** are abbreviations of the description for the *Naive* strategy and the *Inverse naive* strategy above[3].

[2] the operator → is not a symbol for logical implication. Constraints are not inference rules (Ohlsson, 1993).

[3] The description for a design strategy is too long, thus we use a name to avoid repetition.

Constraint 4:
Relevance:
The ideal solution is **Naive** AND
in the recursive clause, the composition subgoal is a built-in predicate "append" AND
the student's solution is **Naive**
Satisfaction:
the student's solution ought to have a composition subgoal "append"

Constraint 5:
Relevance:
The ideal solution is **Naive**
Satisfaction:
The student's solution is **Naive** OR **Inverse**

Constraint 4 examines the application of a composition subgoal and Constraint 5 makes sure that the student's solution is implemented corresponding to either *Naive* or *Inverse.*

Before we answer the third question, we investigate two cases:

Case A: if there is a student's solution which is intended to implement the *Naive* strategy but it misses a base case, then Constraint 4 will be not relevant to the student's solution, i.e. the student's solution is considered to be correct. This diagnostic information is too vague.

Case B: if there is a student's solution which is intended to implement the *Naive* strategy but it is erroneous, Constraint 5 will be violated as expected. However, due to the nature of that constraint specification, the location of the error can not be identified.

The two cases above answer the third question negatively and support our claim that constraints specified to cover solution variations on the design strategy level are not useful for diagnostic purposes. This is attributed to the high complexity of such constraints which are specified with a lot of information so that a slight error in the student's solution might cause the relevance part or the satisfaction part easily to evaluate to false. We propose to relieve the complexity problem by breaking down such constraints into two parts: a set of constraints which describe the characteristic of a design strategy and another set of constraints which examine additional requirements of that strategy. The former one is used to hypothesize the intention hidden in the student's solution.

Prolog patterns and design strategy hypothesis

Investigating the *Naive* implementations for different tasks, we notice that they share the same pattern. For instance, the following definitions for `insertion_sort/2` and `reverse/2` have the programming techniques in common: a base case exists, in the recursive case the input list is decomposed into a head and a tail, get the first element of the input list for processing, decompose the tail recursively and finally compose the result argument.

```
insertion_sort([],[]).
insertion_sort([H|T],R):- insertion_sort(T,S), insert(S,H,R).

reverse([],[]).
reverse([H|T],R):- reverse(T,M), append(M,[H],R).
```

Prolog programming techniques are programming practices which can be found in different contexts (Brna et al, 1999). A programming technique is language dependent, but specification independent, e.g., the same technique might be used in sorting a list or in finding the maximum of two numbers. Furthermore, a technique might apply to only parts of a complete procedure, and many techniques may be combined together in a procedure. (Bowles & Brna, 1999) propose five essential Prolog programming techniques. The *same* technique is used to pass the same value between the head and the recursive subgoal of a clause. The *list head* is used when the head value is the list and the value of the recursive subgoal is its tail. The *list subgoal* technique is where the value of the recursive subgoal is the list and the head value is its tail. The *after* technique is used to indicate non recursive subgoals after the recursive one. The *before* technique is the opposite of the *after* technique. Techniques can be combined to create a new technique.

A design strategy in logic programming can be represented by a set of programming techniques which can be referred to as a Prolog pattern. We can apply the CBM approach to model Prolog patterns. The *Naive* pattern can be modelled by the following set of constraints using the formal representation of constraint(Type, Relevance, Satisfaction, Severity, Position, Hint)[4]. The constraints which are associated to a Prolog pattern are referred to as pattern constraints.

Constraint N1: *constraint(pattern, "patternname=naive", "a base case exists", 0.1, [], Hint1)*

Constraint N2: *constraint(pattern, "patternname=naive", "a recursive case exists", 0.1, [], Hint2)*

[4] For the simplicity we do not specify the error position and instructional hints in our constraint examples.

Constraint N3: *constraint(pattern, "patternname=naive and a recursive case exists", "the input list is decomposed in a head and a tail", 0.3, [], Hint3)*

Constraint N4: *constraint(pattern, "patternname=naive and a recursive case exists", "the tail is decomposed recursively", 0.3, [], Hint4)*

Constraint N5: *constraint(pattern, "patternname=naive and a recursive case exists", "the result argument is composed by the head and the process result of the tail", 0.3, [], Hint5)*

We specify the severity for constraints N1 and N2 as a value of 0.1 and for constraints N3, N4, N5 as a value of 0.3 because the constraints N1 and N2 are more important than the other ones. In addition, constraints N3, N4 and N5 assume that N2 evaluates to true. The severity is used to hypothesize the Prolog pattern implemented in the student's solution. The following formula determines how plausible the student's solution is realized corresponding to that Prolog pattern where `S1, S2...` and `Sn` are the severity of constraints which are violated if they are relevant to the student's solution.

Formula 3:

```
Plausibility(solution, patternname) = S1 * S2 * … *Sn
```

If the task `reverse/2` can be implemented by applying different Prolog patterns: *Naive, Inverse naive, Accumulator, Railway shunt* (Hong, 2004)[5], then we have to evaluate all four sets of pattern constraints for any given student's solution and to compute the plausibility for each Prolog pattern. The Prolog pattern which has the highest plausibility is considered to be the most plausible one implemented in the student's solution.

By means of Prolog patterns we are in a position to hypothesize the strategy in the student's solution and to examine the correctness of structural elements at deeper levels, even if the student's solution is not in agreement with the strategy properly. E.g. the constraint 4 above can be rewritten as the following constraint:

Constraint 4A:

Relevance:
patternname = naive AND
in the recursive clause of the ideal solution, the composition subgoal is a built-in predicate "append"
Satisfaction:
in the recursive clause of the student's solution should have a built-in predicate "append"

Prolog Patterns can be organized two dimensionally. On the horizontal dimension, Prolog patterns can be found according to classes of programming problems. In general we can have four classes of programming problems: 1) test if any element of an input collection satisfies the given property, 2) test if all elements of an input collection satisfy the given property, 3) process one element of the input collection which satisfies the given property and return a result, 4) process all elements of the input collection and returns a result (Brna, 2001).

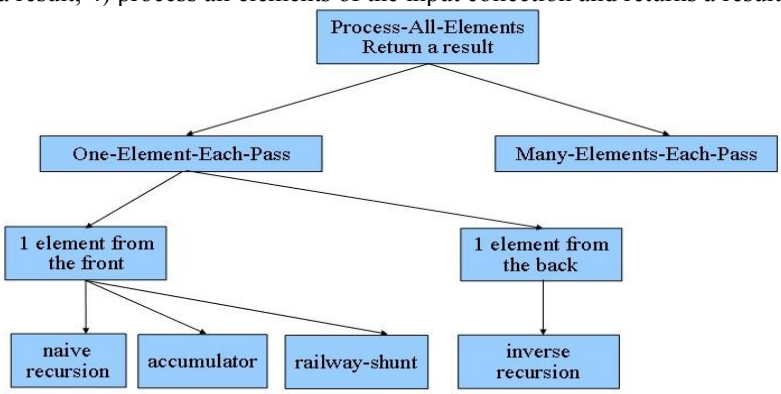

Figure 2 a small hierarchy of Prolog patterns

On the vertical dimension, Prolog patterns are differentiated by programming techniques. Figure 2 illustrates a hierarchy of Prolog patterns for the problem class of processing all elements of an input list and returning a result. The technique of list processing can be differentiated between the removal of one element or the removal of two elements from the input list for recursive processing. The technique of decomposition can be distinguished by taking an element from the front or from the back of a list. When an element has been decomposed from the front of a list, we can apply different programming techniques and have three patterns: *naive recursion, accumulator* or *railway-shunt*. If the decomposition takes place at the back of the input list, we apply the recursion on the element decomposed which specifies the *inverse* recursion pattern (Gegg-Harrison, 1993).

The benefits of Prolog patterns can be summarized in following points:

[5] Hong refers to the design strategy as a programming technique. We prefer the term Prolog pattern to avoid confusion with the notion of programming techniques in (Bowles & Brna, 1999).

1. Structure the domain of logic programming for diagnostic purpose. (see Figure 2)
2. Hypothesize the strategy implemented in a Prolog program. (see Formula 3)
3. Decrease the complexity of exercise specific constraints and thus, to increase the accuracy of the diagnostic information. (see Constraint 4A)

The problem of user-defined auxiliary predicates

If an auxiliary predicate is necessary to solve a subtask, then the ideal solution should also contain a corresponding definition of auxiliary predicates. The amount of possible design strategies to solve a subtask is countable.

However, if one wants to keep the code in the main predicate definition clear and simple, then he can define any auxiliary predicate and this is unpredictable. We need an ideal solution to encapsulate the exercise requirements to specify exercise-specific constraints. Which of the solutions should be the ideal one? Applying the unfolding/folding techniques proposed in (Tamaki & Sato, 1984) we are in a position to transform students solutions which uses auxiliary predicate to the one without auxiliary predicates, if not both the main predicate and the auxiliary predicate apply the recursion technique. The solution without auxiliary predicates is supposed to be the ideal one which has the normal form. If both the main predicate and the auxiliary predicate require recursion, then the ideal solution must contain the definition of the auxiliary predicate.

THE DIAGNOSIS PROCESS

Applying the CBM for Prolog we carry out the diagnosis in four steps:

1. transform the student's solution to the normal form
2. evaluate the pattern constraints
3. evaluate the exercise specific constraints
4. evaluate the Prolog general constraints

Violated constraints from the first step yield diagnostic information about the lack of the skill in designing a solution. Hints on this level are used to help students to master the application of Prolog patterns. The second step delivers information about application of programming techniques required for specific tasks. Diagnostic information from the third step reminds students to attend to the regularity of Prolog.

CONCLUSION AND FUTURE WORK

For a task in logic programming there is a variety of possible solutions. The amount of correct solutions for the given task is unpredictable. Thus, the task of logic programming is an ill-domain. We presented a way to structure this domain by using Prolog patterns and to model this domain applying the CBM approach. In addition, using Prolog patterns we are in a position to hypothesize the design strategy in the student's solution.

As a result, constraints are partitioned into different sets: pattern constraints, exercise-specific constraints and Prolog general constraints. For each constraint set, the instructional intention is differentiated. Pattern constraints are used to support students to master the activity of solution design by using a Prolog pattern. Both pattern constraints and exercise specific constraints can be encoded with instructional hints which help students to be familiar with programming techniques. The last set of constraints reminds students to consider the regularity of logic programming.

Until now we have implemented the evaluation component which is able to evaluate the three kinds of constraints mentioned above. We now need to verify the approach by establishing a constraint database and testing it on a wide range of programming tasks. We are also investigating the following problems:

- The problem of using user defined auxiliary predicates: folding/unfolding techniques are limited to certain types of solutions which use auxiliary predicates so that not both the main predicate and the auxiliary predicate apply recursion. For other solutions, which can not be transformed using folding/unfoldind, we need to invent another approach.
- The problem of task analysis: in many cases, students make errors because they have difficulties with task analysis. Thus, at that point, diagnostic information about semantic or syntactic errors is not relevant for students, but rather the stage of the problem solving process where the student becomes stuck in: problem analysis, solution design or implementation. We intend to develop an approach to determine how far the student has understood the task specification and to support students to analyse the given task.

REFERENCES

Anderson, J.R. and Reiser, B.J. (1985) *The Lisp Tutor*. BYTE, April, 159-175.

Baghaei, N. & Mitrovic, A. (2005) *COLLECT-UML: Supporting individual and collaborative learning of UML class diagrams in a constraint-based tutor*. Accepted for presentation at KES.

Blackburn, Bos and Striegnitz (2001) *Learn Prolog now*. http://www.coli.uni-sb.de/kris/learn-prolog-now

Bowles, A. & Brna, P. (1999) *Introductory Prolog: a suitable selection of programming techniques*. In: Brna, P., du Boulay, B., Pain, H.(eds), Learning to Build and Comprehend Complex Information Structures: Prolog as a Case Study. Ablex, pp. 167-178.

Brna, P. et al. (1999) *Programming techniques for Prolog*. In: Brna, P., du Boulay, B., Pain, H.(eds), Learning to Build and Comprehend Complex Information Structures: Prolog as a Case Study. Ablex, pp. 143-166.

Brna, P. (2001) *Prolog Programming, a First Course*.

Gegg-Harrison, T.S. (1993) *Exploiting program schemata in a Prolog tutoring system*. Ph.D. Dissertation, Technical Report CS-1993-11, Department of Computer Science. Duke University. Durham.

Hong, J. (2004) *Guided programming and automated error analysis in an intelligent Prolog tutor*. International Journal Human-Computer Studies 61, p.505-534.

Menzel, W. (2006) *Constraint-based modeling and ambiguity*. To be published in International Journal of Artificial Intelligence in Education, volume 16.

Mitrovic, A. and Ohlsson, S. (1999) *Evaluation of a constraint-based tutor for a database language*. International Journal of Artificial Intelligence in Education, 10, 238-256.

Mitrovic, A. et al. (2001) *Constraint-based tutors: a success story*. In L. Monostori and J. Vancza, Proceeding of the 14th Int. Conf. on Industrial Engineering Application of AI and Expert Systems, 931-940, Budapest.

Murray, W. (1988) *Automatic Program Debugging for Intelligent Tutoring Systems.* Los Altos, CA: Morgan Kaufmann, 1988.

Ohlsson, S. (1993) *The interaction between knowledge and practice in the acquisition of cognitive skills.* In Chipman, S. Foundations of knowledge acquisition.

Ohlsson, S. (1994) *Constraint-based student modeling.* In J. E. Greer, G.I. McCalla, Student Modelling: The Key to Individualized Knowledge-based Instruction, 167-189. Berlin.

Suraweera, P., Mitrovic, A., Martin, B. (2005) *A knowledge acquisition system for constraint-based Intelligent Tutoring Systems.* http://www.cosc.canterbury.ac.nz/tanja.mitrovic/Suraweera-AIED05.pdf

Tamaki, H. and Sato, T. (1984) *Unfold/Fold transformation of logic programs*. Proceedings of the 2nd International Logic Programming Conference, Uppsala, Sweden, 127-138.

Taylor, J. (1999) *Analysing novices analysing Prolog: what stories do novices tell themselves about Prolog*? In P. Brna, B. du Boulay, H. Pain (Eds). Learning to build and Comprehend Complex Information Structures: Prolog as a Case Study, Ablex, 43-71.

Taylor, J. and Boulay, B.D. (1987) *Studying novice programmers: why they might find learning Prolog hard.* In Rutkowska, J.C., Crook, C.(Eds), Computers, Cognition and Development: Issues for Psychology and Education. Wiley, New York.

Vanneste, P. (1994) *A Reverse Engineering Approach to Novice Program Analysis*. PhD thesis, KU Leuven Campus Kortrijk.

Warendorf, K. and Tan, C. (1997) *Constraint-based student modeling - a simpler way of revising student errors.* In Proceedings of ICICS, 2, 1083-1087.

Weber, G. (1996) *Episodic learner modelling.* Cognitive Science, (20), 195-236.

Xu, S. and Chee, Y.S (2003) *Transformation-based diagnosis of student programs for programming tutoring systems.* IEEE Transactions on Software Engineering, 29(4):360-384.

Supporting Self-explanation of Argument Transcripts: Specific v. Generic Prompts

Vincent Aleven
Niels Pinkwart
Human-Computer Interaction Institute
Carnegie Mellon University
{aleven, nielsp}@cs.cmu.edu

Kevin Ashley
Collin Lynch
Learning Research and Development Center
Intelligent Systems Program
School of Law
University of Pittsburgh
ashley@pitt.edu, collinl@cs.pitt.edu

Abstract. We are developing an intelligent tutoring system that helps beginning law students learn argumentation skills through the study of transcripts of oral argument sessions before the US Supreme Court. These transcripts exemplify complex reasoning processes in which proposed decision rules are evaluated by holding them against real and hypothetical cases. As a first step, we investigated (without computer-based support) how to design good self-explanation prompts. In well-structured domains, *generic* prompts (e.g., "Explain.") may be most effective, because they leave students more latitude in discovering deficits in their own knowledge. However, in an ill-defined domain such as legal reasoning, *specific* prompts, which ask students to interpret a transcript in terms of a specific argumentation framework, may be more likely to help them arrive at insightful interpretations. In an experiment with 17 beginning law students, we found that the less able students (as measured by LSAT scores) learned better with specific prompts, as hypothesized, but the more able students learned better with generic prompts. This interaction was seen on test items that asked students to make arguments about a legal issue similar to that encountered in one of the transcripts. There was no significant interaction on items where students were asked to interpret a transcript dealing with a new area of the law (as opposed to making arguments). Thus, for less able learners in an ill-defined domain, the advantages of specific prompts outweigh those of generic prompts. It is surprising however how quickly the balance tips in favor of generic prompts. We are currently analyzing students' self-explanations to provide a deeper interpretation of the results.

Keywords: self-explanation, argumentation, legal reasoning, ill-defined domains

INTRODUCTION

We report on a project to develop an ITS for legal argumentation (e.g., Aleven, 2003; in press; Muntjewerff & Breuker, 2001). The legal domain is ill-structured in that cases present issues for which there seldom are uniquely right answers. Instead, reasonable arguments usually support competing answers, as evidenced by dissenting opinions eventually becoming the law of the land, by decisions reversed on appeal, and by a general reluctance of legal professionals to predict the outcome of legal cases. Competing reasons can be found in conflicting precedents, the sometimes ambiguous logical structure of statutes, and alternative interpretations of abstract, open-textured concepts in legal rules. This ill structure is unavoidable. (See, e.g., Frank, 1930; Llewellyn, 1951. But see Dworkin, 1986 for an argument that legal and moral questions do have right answers.) For instance, legislators write statutes in terms of abstract legal concepts to implement underlying legal policies, but they cannot foresee all of the scenarios to which a particular statute will be applied. Further, in real-world scenarios the policies often conflict, and subtle differences in facts can lead courts to resolve otherwise similar problems in different ways.

We focus on US Supreme Court oral argument, rapid-fire exchanges in which opposing attorneys propose decision rules to decide the case at hand (and cases like it), and the Justices explore the ramifications of these proposals by posing hypothetical fact situations and asking how they should be decided according to the proposed rules. Each side in the argument has one half hour to address the court; the Justices famously interrupt an advocate with questions. Arguing one's first case before the U.S. Supreme Court is a professional milestone –some experienced advocates become famous for their skills in making such arguments. Transcripts of these arguments have been published, and are readily available on-line through Westlaw[1]® and Lexis[2]®. Audio

[1] www.westlaw.com
[2] www.lexis.com

recordings of the proceedings are becoming increasingly available through websites like OYEZ[3]. We believe that these transcripts, due to their authenticity and high drama, will be motivating materials for beginning law students.

Our main goal is to help law students understand the kinds of argumentation processes that unfold in these transcripts and to develop some of the argumentation skills that are employed in these exchanges. Eventually, our goal is to develop an intelligent tutoring system that engages and guides students in this regard. Even if the transcripts are motivating, they are very challenging materials. They are different from most "worked-out examples" (e.g., Atkinson, Derry, Renkl, & Wortham, 2000) in that they show reasoning processes in their authentic raw form, complete with the false starts, blind alleys, and tangential lines of reasoning that are typically removed from annotated materials. The oral argument transcripts are messier; the Justices interrupt and advocates fumble to regroup.

As a stepping stone toward building an ITS, we study self-explanation, which has been shown to be an effective metacognitive strategy, although primarily in well-structured domains such as physics, biology, or geometry (Aleven & Koedinger, 2002; Chi, 2000; Renkl et al., 1997; but see Schworm & Renkl, 2002). While a number of cognitive science studies have produced evidence of the effectiveness of self-explanation prompts (Chi, de Leeuw, Chiu, & Lavancher, 1994; Renkl, 2002; Schworm & Renkl, 2002), we know of no studies that have asked specifically what kinds of prompts are the most effective. Many authors (e.g., Chi, 2000; VanLehn, Jones, & Chi, 1992) seem to have assumed that generic prompts (e.g., "explain this to yourself," where "this" refers to a line in a worked-out example or a sentence or paragraph of expository text) are the most effective, presumably because they increase the chances that individual students will be able to identify gaps in their own understanding, discover deficiencies in their mental models, or generate useful inferences. Specific prompts on the other hand, specific questions about how to interpret the materials, appear to have been considered less effective, at least in well-structured domains. It is possible that they would be more helpful in getting some students to realize that they have a gap in their understanding and may even hint at how to fill the gap (e.g., VanLehn et al., 1992). But specific prompts, which typically target a particular gap, are only likely to benefit those students who have that gap. For all other students, such prompts simply ask them to explain something that they understand already, which will not greatly impact their learning. It seems that it would be difficult to prompt all students for all gaps that they might possibly have. Even worse, specific prompts may rob students of the opportunity to make a range of useful inferences because such prompts, due to their specificity, focus their attention on one specific issue.

In an open-ended and ill-structured domain such as legal reasoning, however, the trade-off between specific and generic prompts may play out differently. A basic assumption of our work is that students will develop a better understanding of legal argumentation if they interpret it as a process of hypothesis formation and testing. We have designed an argumentation framework, based on that view, which is described below. Specific prompts that ask students to interpret the transcript in terms of this framework may be more beneficial than generic prompts that merely draw students' attention to particular passages, especially if students are unfamiliar with the framework. The prompts may spur useful inferences that students would not have made otherwise (e.g., Chi, 2000). In an open-ended domain, specific prompts may be helpful in a more general sense as well. In studying the transcripts, students may make many connections with prior knowledge and may generate many inferences regarding the issues that they read about. This assumption is reasonable in light of the fact that legal cases deal with real-life events. Further, many people have at least a basic understanding of what the law says in many areas and of the legal concepts being applied (e.g., "the right to privacy"). They are likely also to bring to bear their common sense notions of what is just. Thus, they may not experience discrete "gaps" or deficiencies in their knowledge or mental models, the way one would in a well-structured domain (e.g., VanLehn et al., 1992). For example, in mathematics or physics, if one does not see the relation between two equations, it may be harder to ignore the knowledge gap. In ill-structured domains, to the extent that there are such gaps, they may be "obscured" by the many inferences that can be made. Thus, in an ill-defined domain, specific prompts may provide just the right amount of focus: enough to give students a better idea of what inferences and interpretations are interesting, but not so much that they draw students' attention away from useful thoughts that they would otherwise have. In order to test this hypothesis, we conducted a small empirical study to compare the relative advantages of specific and general prompts for the study of US Supreme Court oral argument.

The paper is structured as follows: we first explain the framework for legal argumentation that we would like students to apply to the argumentation transcripts. We then describe the design and outcomes of the experiment, and discuss our results in light of literature on self explanation and ITSs for ill-structured domains.

A FRAMEWORK FOR INTERPRETING ARGUMENTATION TRANSCRIPTS

Our goal is to help students understand the normative and cognitive role of the Justices' hypotheticals in legal argument. As we noted above, advocates make their case by proposing a test or standard for deciding the issue at hand in this and future cases. These tests may be based on the relevant statutory or constitutional texts, if any, and interpretations in past cases involving the issue. The advocate asserts that (a) the proposed test or standard

[3] www.oyez.org/oyez/frontpage

is the right standard for the court to apply in deciding the issue, and (b) when applied to the facts of the case, the standard yields the outcome urged by the advocate. The Justices employ hypothetical cases to draw out the legal consequences of adopting the proposed standard and applying it to this and future cases. The hypotheticals explore the meaning of the proposed test, its consistency with relevant legal principles, policies, and past case decisions, its application to the case's facts, and its sensitivity to changes in the facts. In this work, we are trying to help students identify instantiations of a novel model of this kind of argumentation. In particular, we would like to help students identify (1) the proposed tests for deciding the current case and the reasons justifying it, (2) the hypotheticals that challenge the proposed test, the nature of the challenge, and the accompanying reasons, and (3) the advocate's response to the challenge in one of three forms: disputing the hypothetical's significance, modifying the proposed test, or abandoning the proposed test.

The targeted interpretative process is illustrated using the oral argument transcript in *Dennis LYNCH, etc., et al., Petitioners v. Daniel DONNELLY et al.* 465 U.S. 668 (1984), which was argued before the US Supreme Court on October 4, 1983. An excerpt appears in Table 1. The City of Pawtucket, Rhode Island erected an annual Christmas display in the heart of the shopping district. The display, which was owned by the city, comprised among other things, a Santa Claus house, reindeer pulling Santa's sleigh, candy-striped poles, a Christmas tree, carolers, cutout figures representing a clown, an elephant, and a teddy bear, hundreds of colored lights, and a large banner that reads "SEASONS GREETINGS." It also included a crèche consisting of the traditional figures, including the infant Jesus, Mary and Joseph, angels, shepherds, kings, and animals. Pawtucket residents and the American Civil Liberties Union filed suit, claiming that the city's inclusion of the crèche in the display was unconstitutional.

The Establishment Clause of the First Amendment to the U.S. Constitution states that "Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the Government for a redress of grievances." *U.S. Const. Amendment I*. The Supreme Court had explained that the purpose of the Establishment and Free Exercise Clauses of the First Amendment is "to prevent, as far as possible, the intrusion of either [the church or the state] into the precincts of the other." *Lemon v. Kurtzman,* 403 U.S. 602 (1971). At the same time, however, the Court had recognized that "total separation is not possible in an absolute sense. Some relationship between government and religious organizations is inevitable." *Ibid*. In every Establishment Clause case, the Court tries to reconcile the tension between the objective of preventing unnecessary intrusion of either the church or the state upon the other, and the reality that, as the Court has often noted, total separation of the two is not possible. Thus, the issue in the *Lynch v. Donnelly* case is whether Pawtucket, R.I.'s crèche display is a violation of the Establishment Clause. A small excerpt of the oral argument is shown in Table 1.

The attorney representing the ACLU argued that the Pawtucket Christmas display should be considered unconstitutional (line 139, Table 1, left column). As often happens in these transcripts, a test was implied although the advocate did not state one explicitly (e.g., there are no clearly marked "if" and "then" parts). One possible formulation of Mr. DeLuca's test is as follows: "if a city owns a fundamental religious symbol and displays it adjacent to its City Hall, it is in violation of the Establishment Clause." Other valid formulations may be possible, including formulations with more abstract or less abstract terms, and formulations with different numbers of pre-conditions, illustrating the ill-structured nature of the domain. One challenge that students face therefore is to recognize when advocates' statements imply a test and to arrive at a suitable and accurate formulation of that test. In response to the attorney's test, the Justices posed a hypothetical (in this case, a slight variation of the facts of the case – see line 141 in Table 1), in an apparent attempt to explore how low the attorney wanted to set the threshold for violations of the Establishment clause. The Justice's hypothetical was specifically aimed at exploring whether, under the attorney's proposed test, the display of a religious symbol adjacent to the City Hall is sufficient for the City to violate the Establishment Clause, even if the City does not own the symbol in question. The attorney's response (line 142) implies that ownership is not necessary and that mere sponsorship by the City of a display that contains a religious symbol, even one not owned by the City itself, is unconstitutional. This can be seen as a broadening of the test originally formulated. Once again the test is not stated in explicit "if-then" format, nor does the attorney indicate explicitly that he is changing his test, let alone how he is changing it. It is thus up to the student to provide an accurate formulation. As the example illustrates, relating a transcript to the argumentation model is an interpretative process that goes well beyond paraphrasing what occurs in the transcript. We are not claiming that this kind of detailed analysis of Supreme Court oral argument is necessary in order to fully understand the court's decision in the case. Rather, we mean to suggest that it is a viable and interesting way to learn about argumentation.

DESIGN OF THE EXPERIMENT

Materials

The materials we used in this study were transcripts of US Supreme Court oral arguments for two cases, including *Lynch v. Donnelley*, presented above. We edited the transcripts slightly, in order to reduce the time that students would need to work through them. However, as much as possible, we tried to retain their authentic

nature. We then inserted self-explanation prompts into the transcripts, with "generic prompts" for the control group, and "specific prompts" for the experimental group. The prompts were inserted at places where we identified key components of our argumentation framework: tests, hypotheticals, and responses to hypotheticals. As we mentioned above, there is no one correct way of applying the "test/hypothetical/response" model to a given transcript – but for purposes of adding self-explanation prompts, agreement in this regard is not necessary. The specific prompts asked students to interpret the transcript in terms of our argumentation framework (see Table 1, column labeled "Specific SE prompt"). The generic prompts, inserted at the same locations in the transcripts, merely said "Explain." Not all contributions in the transcripts had associated prompts (Table 1 has a greater density of prompts than the overall transcript). The materials were presented to students as Excel spreadsheets.

Table 1. Excerpt of an argument transcript with examples of generic and specific self explanation prompts

Transcript	Specific SE prompt	Generic SE prompt
137. ORAL ARGUMENT OF AMATO A. DE LUCA, ESQUIRE ON BEHALF OF THE RESPONDENTS	Which party does Mr. DeLuca represent?	Explain.
138. MR. DE LUCA: Mr. Chief Justice, and may it please the Court, with the possible exception of the cross, the nativity scene is one of the most powerful religious symbols in this country, and most certainly one of the most powerful Christian religious symbols in this country. It is, as all of the parties agree and acknowledge, the biblical account of the birth of Christ, the Christian Messiah, the Redeemer, according to the gospels of Matthew and Luke as contained in the New Testament.		
139. Pawtucket's purchase, the maintenance, and the erection of the fundamental Christian symbol involves government in religion to a profound and substantial degree. It has aligned itself with a universally recognized religious symbol and belief. I would like to bring to the Court's attention that although the religious symbol, the creche, is contained in a display that is on private property -- it is owned by the Slater Museum Historical Society -- it is adjacent to the City Hall. City Hall is approximately 100 feet away from this area.	What is Mr. DeLuca's test concerning the issue of whether a city's creche display violates the Establishment Clause? Write as clear and succinct a version of his proposed test as you can.	Explain.
140. Also, the creche and the display itself is -- there is a ceremony that is held by the mayor of the city of Pawtucket each year, a lighting ceremony, which announces the commencement of the display in the Hodgson Park area. The music that is played at the display is the same music that is also played inside of City Hall, and all of the festivities that take place at the display and at City Hall are paid for and sponsored by the city of Pawtucket.	What is the significance of the proximity of the creche to City Hall?	Explain.
141. QUESTION: Well, Mr. DeLuca, you say that although the property, the real property, I take it, on which the creche is located is private, it is only -- it adjacent to city property. Now, if the city did not own the creche itself, so that everything that was contributed to the display, including the creche, were privately owned, it wouldn't violate the First Amendment, the fact that it was right next door to the City Hall, would it?	What is the relationship of the Justice's hypothetical to Mr. DeLuca's test?	Explain.
142. MR. DE LUCA: Well, I think that in the -- I think that in understanding that the city owns all of the symbols and all of the artifacts that are contained in this display, and assuming that that -- the creche were purchased and paid for privately without any other explanation that it is private, then I think it would still violate the establishment clause for the First Amendment, because there is no indication to anyone looking at that that the display or the creche is not part of the broader display which is put up and sponsored by the city.	How does Mr. DeLuca respond to the Justice's hypothetical? What effect would the response have on his proposed test? Explain whether the response to the hypothetical leads to a change in the proposed test, and if so, what change.	Explain.

Subjects

The 17 participants in the study were recruited from a group of students enrolled in a 6-week summer program for newly-accepted law students prior to their first year in law school. Students were selected for this program on the basis of such factors as extended time out of school, disadvantaged economic background, etc. Participation in the study was voluntary. All subjects were paid to participate in the experiment. The students were divided into two conditions, balanced in terms of LSAT scores. The LSAT is the Law School Admissions Test. It is a moderately good predictor of success in law schools, and many law schools in the US use LSAT scores as a factor in deciding which students to admit.

Procedure

All students participated in three sessions, each of which lasted approximately 2.5 hours. During the first two sessions, they studied transcripts of two US Supreme Court cases. At the beginning of each session, they were given a short introduction to the case they were about to study, some background material about the legal issues it presented, and a brief summary of the argumentation model. They then studied the transcript, typing answers to the self-explanation prompts into the Excel spreadsheet. Students in the "Specific" condition were given the transcripts with the specific prompts, students in the 'Generic" condition those with generic prompts. At the end of the first two sessions, all participants took a survey, but since the results are not yet available, we will not mention them any further. The third session was a post-test session, which consisted of two parts: an Argumentation Transfer Test and a Domain Transfer Test, described next.

Tests

In the Domain Transfer Test, students were given a transcript for a third case dealing with a different area of the law, compared to the first two cases. This time, the transcript did not contain any self-explanation prompts. Apart from that, the transcript was presented in the same format as before (i.e., Excel spreadsheet). The students then took a survey about this transcript, similar to the kind of survey they had completed at the end of each of the first two sessions, in which they were asked which of the Justices' hypotheticals was the most problematic and to assess the quality of the advocate's response and to formulate a better one if possible. This task was very similar in structure and materials (except for the absence of prompts and the switch to a new area of the law) to the tasks carried out during the first two sessions.

In the Argumentation Transfer Task, students were given a description of the facts of a case that dealt with a very similar legal issue to that encountered during the second session. This time, however, the students were not asked to study a transcript, but were asked to help an attorney prepare to argue the case before the US Supreme Court. Thus they were asked to formulate a test for deciding the case that would give a favorable outcome (as opposed to interpreting what test is being used in a transcript) and to predict what hypotheticals the Justices would be likely to pose (as opposed to interpreting what hypotheticals are being used in a transcript and why). Thus, they were engaged in *making* the kinds of arguments of which so far they had only studied examples.

Two legal writing instructors independently graded the surveys. Since there is no standard way of grading these kinds of surveys, we designed a grading form, which asked the grader to rate the quality of the tests, hypotheticals and responses formulated by the subjects, and to rate how well the subjects had understood the legal issues of the cases. Most items were graded using a 5-point Likert scale (e.g., "How well did the student formulate a test for Ms. Stone to propose that would lead to a favorable result for the ACLU?"). In addition, the form asked the graders to summarize or characterize the student's answers in textual form. The graders were also asked to rate the hypotheticals formulated by the students with respect to 12 (positive and negative) characteristics (e.g. "concise with no irrelevant details", "very creative", or "irrelevant to the argument").

RESULTS

We first evaluated the inter-rater reliability of the two legal writing instructors who graded the post-test materials. For the Likert Scale questions, which as mentioned cover the majority of the test items, we adjusted the grades assigned by one rater, subtracting one from each grade to achieve a common mean grade between the two graders. Then, counting as agreement grades that differ by no more than 1, we computed Cohen's Kappa as $\kappa=0.75$. This level indicates a satisfactory reliability in relative quality estimation. For the hypothetical characteristic questions, Cohen's Kappa was less than 0.7, but the percentage agreement was 0.74, which can be considered a satisfactory level of inter-rater reliability for yes/no questions. Having established that the inter-rater reliability was satisfactory, we based all of our following data analyses on the average of the two graders' opinions.

We first computed a single overall score for each subject, which included all survey items with Likert Scale or yes/no answers. We also computed subscores by only considering items that were related to a specific test (Argumentation Transfer / Domain Transfer) or a specific aspect of the argumentation model (test / hypothetical / response). With respect to the overall scores, in the full sample there was no main effect of condition, either on the Argumentation Transfer Test ($F(1,15)=0$, $p>.9$) or in the Domain Transfer Test ($F(1,15)=.725$, $p>.4$). Nor was there any significant difference with respect to the specific item types (or model aspects).

We then divided up the sample by means of a median split based on the students' LSAT scores, creating a. "lower LSAT" group that contained 8 students, and a "higher LSAT" group with 9 students. The students in the lower LSAT group scored significantly lower than their counterparts in the higher LSAT group throughout both tests ($F(1,15)=4.774$, $p<.05$), consistent with the predictive value claimed for the LSAT scores. We then considered whether the specific and generic prompts may have affected the students differently, depending on their ability level (as measured by LSAT scores). We found an interaction effect between ability level and condition, as illustrated in Figure 1: while the lower ability subjects group benefit more from specific prompts, the higher ability subjects are supported better by generic prompts. For the overall survey data, this interaction is at the borderline of significance ($p=.05$, repeated measures analysis). For the Argumentation Transfer Test

considered separately, the interaction effect is statistically significant ($F(3,13)=9.096$, $p<.01$). (There was no statistically significant interaction on the Domain Transfer Test.) A more detailed analysis showed that this effect is largely due to test items in which students were asked to generate hypotheticals (interaction effect, $F(3,13)=7.010$, $p<.01$). For the test items that asked students to formulate a test, there is a marginally statistically interaction ($F(3,13)=3.354$, $p<.1$) indicating that the higher-ability subjects did better when trained with generic self-explanation prompts. In test items related to responses to hypotheticals, no significant interaction effect was found.

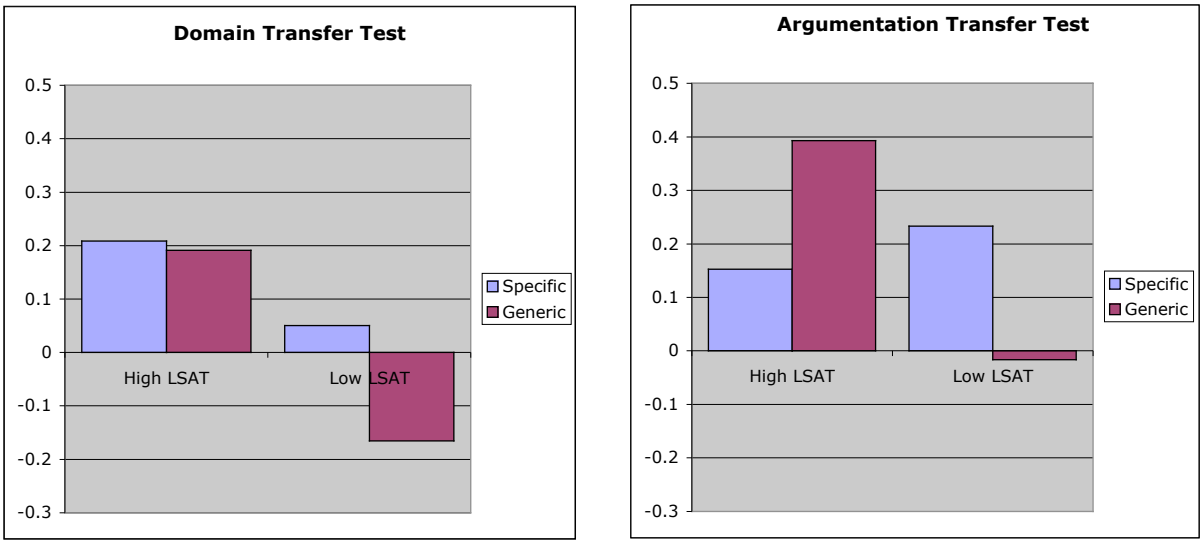

Figure 1. Results of the Domain / Argumentation Transfer Tests

We have begun to analyze the self-explanations in an attempt to better understand the interaction effect found at the post-test. The students' answers to one of the self-explanation prompts (i.e., the self-explanations typed during the first two sessions of the experiment – these self-explanations were part of the "training," not part of the post-test survey) illustrate that it is a challenging task to interpret a transcript and relate it to the argumentation framework described above. Table 2 shows the answers given by the 17 study subjects to the prompt shown in Table 1, line 139, where Mr. DeLuca, one of the attorneys, formulates a test, albeit implicitly so. The students' self-explanations are arranged according to their LSAT scores and the type of prompt that they received. The generic prompt, as always, merely asked students to "Explain." The specific prompt, shown in Table 1, asked students to provide a clear and succinct statement of Mr. DeLuca's test. As discussed above, one possible statement of the test is that "if a city owns a fundamental religious symbol and displays it adjacent to its City Hall, it is in violation of the Establishment Clause." No student provided a truly outstanding statement of the test. For example, it was rare to see an explicit if-then form. Some students did quite well (e.g., answers 3, 7, 9, and 14). Others provided statements that were not very "test-like," perhaps identifying some components of a test but not stating which way they cut (e.g., 1 and 5), or stating conditions that were rather abstract (e.g., 15). Some students did not really get to stating a test at all (e.g., 10 and 16, and obviously, 2). Thus, it is not an easy task to realize that a new test may be implied and to formulate the test, with a clear conclusion and a condition stated at an appropriate level of abstraction.

Based on the post-test results, one expects to see that the lower ability students provide better responses to the specific prompts than do they to the generic prompts, whereas for the higher ability students, one expects to see the opposite pattern. We will assess the explanations to see if this expectation is borne out. We will pay particular attention to whether students do "good things" in their responses (to either the specific or the generic prompts) that are not anticipated by the corresponding specific prompts. We will evaluate whether such "unprompted good things" are more likely to occur with generic prompts, and whether they relate to the relative advantages of specific and generic prompts mentioned in the introduction of the paper.

Table 2. Answers to specific and generic self explanation prompts.

	Answers to specific prompts	Answers to generic prompts
Lower LSAT	(1) Mr. DeLuca's test seems based upon whether the city pays for the display, what the meanings of the symbols are, where they are, and how they could be	(10) De Luca establishes that the creche has a religious legislative purpose that is excessive.
	(2) (none)	(11) Explains how the purchase of the creche by the city is definitely a government supporting a certain religious group by displaying the climax of their
	(3) DeLuca's test is that the RI city purchased, maintenance, and erected a Christian symbol. It clearly violates the Clause b/c governemnt is promotiong and subsidizing religion.	(12) Pawtucket's purchase, maintenance, and erection of the christian symbols involves the government in religion. The display is close to the the City Hall.
	(4) He sees the creche as a purely symbolic symbol and if the city displays and pays for it, it cannot say that it is not promoting religion. It cannot be seperated. Any government should not have religouis symbol on their property.	(13) De Luca is establishing his argument and trying to create a new context or framework for the hypotheticals to be drawn from. The basic argument is the religious symbol is universally recognized as a religious symbol and even though the creche is not on city property the creche is so close to city property it has the appearance of being part of the city's display.
Higher LSAT	(5) The amount of effort put out by the city in erecting the symbol. He also looks at where it is on display not just whether it is private or public property. It is really close to city hall.	(14) Council's test includes who purchased, maintained and displayed the creche. Furthermore, the test concludes that it is irrelevant that the creche was on private property. The display's close association with City Hall does not allow for patron's to distinguish what the city sponsors versus what is privately sponsored
	(6) It considers the degree of involvement of the government with religion. Is the government aligning itself with particular beliefs? How close is it to government property?	(15) Respondent begins argument by showing that the creche is known universally as a recognizable symbol, and that by displaying it, the city is promoting it. Respondent wants to point out that although the symbol has been placed on private land, it is close enough to the city hall to perhaps be confused as to being on city property
	(7) The city spent money on purchsing and maintaining the creche, a religious symbol, which satisfies the promotion of religion, which is a violation of the establishment clause. Also, though the creche is on private property it is in the backyeard of city hall. Through financial support the city has aligned itself	(16) He is trying to prove that the City has not fully seperated itself from the creche as it previously tried to convince the courts.
	(8) If the item is universally religious the government must not condone it. Here, Condoning means location of the nativity scene to the local government's Christmans celebrations.	(17) Making the point that the purchgase and matinance alone invlolves the govt. in the religion and helps to alighn them with that religion.
	(9) The involvement of the Government in religion requires the purchase, maintenance and facilitation of a fundamental religious symbol. The Government must have also aligned itself with that symbol and the belief. The symbol need not be on but near government	

DISCUSSION

Although the experiment did not confirm the hypothesis that specific prompts are more effective than generic prompts, it did produce an interesting result. There was no evidence in the full sample that students learn better when prompted with specific questions, rather than generic prompts that merely encourage them to explain. Instead, the experiment produced a statistically significant interaction, indicating that specific prompts are more helpful for students with lower ability, but generic prompts are more effective with better students. The interaction was seen on test items where students where asked to *make* arguments (as opposed to *studying* arguments, as they had done during the training phase) about a legal issue which by then had become somewhat familiar. There was no significant interaction on test items where students were asked to interpret a new transcript dealing with an unfamiliar legal issue. This interaction is consistent with the relative advantages and disadvantages of generic and specific prompts identified earlier in the paper. Specific prompts may be helpful because they have a scaffolding function: they lead students to useful inferences and perhaps lead them to identify gaps in their understanding (although the latter function is less certain in an ill-defined domain such as the current). However, with students who are inclined to make many inferences by themselves, without the help of a specific prompt, specific prompts may be harmful, in that they are likely to focus students' attention on a narrower set of inferences than they would otherwise have attended to. Generic prompts may be useful because they draw the students' attention to particular passages in the transcript, without restricting them to a small set of inferences.

At this point, it is not entirely clear to us what to make of the fact that an interaction effect was found with respect to the Argumentation Transfer Task but not with respect to the Domain Transfer Test. As mentioned, the latter involved a legal issue and area of the law that the students were not familiar with. It is possible that the new area was just too challenging for the students. It is possible also that having some basic grasp of the legal issues under study is a facilitating factor for learning argumentation skills. That interpretation is perhaps supported by the fact that during their summer program, outside of the study reported in this paper, the students had learned about the legal issues surrounding the First Amendment, which were targeted in the Argumentation Transfer Task but not the Domain Transfer Task. Further analysis of the data may shed more light on this issue.

In retrospect (although not *a priori),* what is surprising is not so much the fact that an interaction was found, but rather that it was found with a group of students very early on in their law school career – the study took place two months prior to the subjects' first year in law school. There was a significant range of student abilities in the sample, as measured by LSAT scores, although tilted somewhat towards the lower end of the LSAT spectrum. As mentioned, the participants in the study were recruited from the students enrolled in a summer school to help students prepare for law school. Participants in this program were selected based on factors such as extended time out of school and disadvantaged economic background. The fact that an interaction was found in this population suggests that the threshold ability level above which generic prompts are more effective is surprisingly low.

The findings from the current experiment are in line with findings by Conati and VanLehn (2000) who studied the effect of self-explanation support delivered by means of an intelligent tutoring system, and found that early on in students' development, more elaborate support is better, whereas later on, less elaborate support is better. The experiment dealt with worked-out physics problems (Newtonian mechanics), clearly a better-structured domain than argumentation. While the self-explanation support used in their experiment was more elaborate than in the current experiment, with the system dynamically selecting steps to explain based on a student model and providing feedback on students' self-explanations, their results could be interpreted (in tune with ours) as indicating that surprisingly early on in a student's development, support that is too elaborate becomes constraining. Coupling these results with literature on the expertise reversal effect, which states that in students' earlier developmental phases, examples are more effective than problem-solving practice, whereas in later phases the reverse is true (e.g., Kalyuga, Chandler, Tuovinen, & Sweller, 2001; Renkl & Atkinson, 2003) one gets an inkling then that the optimum level of support for any given student is continuously changing as the students develops. These changing needs present a challenge but also an opportunity for ITSs, suggesting that an ITS should be capable of varying its level of scaffolding even more so than previously thought (e.g., Collins, Brown, & Newman, 1989; VanLehn et al., 2000).

The current experiment seems to confirm some of the limitations of self-explanation prompts that were noted in previous experiments (Renkl et al., 1998) As illustrated, the responses that students typed to the self-explanation prompts leave room for improvement, consistent with Renkl's observations. Thus, another challenge for ITS research is to develop techniques for supporting self-explanation in an ill-defined domain beyond prompting, such as feedback on students' self-explanations (e.g., Aleven & Koedinger, 2002; Conati & VanLehn, 2000). The techniques developed in these earlier projects may not be applicable in ill-defined domain, since they depend on having an expert model that can produce a reasonably complete set of expert solutions. That assumption typically does not hold in an ill-defined domain, where often every (student or expert) solution is at least somewhat different (a point not mentioned in Herbert Simon's famous paper (1973) on ill-structuredness). In a companion paper to the current paper (Pinkwart, Aleven, Ashley, & Lynch, in press), we describe the next step in our project, the design of a system in which students self-explain argumentation transcripts by annotating them in a graphical language and receive feedback on their graphical annotations in the form of self-explanation prompts, as a form of adaptive prompting. The feedback is generated without the use of expert solutions.

CONCLUSION

In a well-structured domain, generic self-explanation prompts may be more effective than specific prompts, presumably because they leave individual students more latitude in discovering deficits in their own knowledge, even if specific prompts might provide more help in *leading* them toward specific deficits and possible ways of addressing them. We hypothesized that when students study complex, authentic argument transcripts in an ill-structured domain, specific prompts may provide useful scaffolding without being too constraining. We focused on prompts that ask students to interpret a transcript with respect to a specific argumentation framework and hypothesized that these prompts would lead students to useful inferences in a way that generic prompts would not. The results of the experiment indicate that this hypothesis holds true for lower ability students but not for higher ability students, who did better with generic prompts. The interaction was seen with respect to students' overall test scores, but was confined to an Argumentation Transfer Task, in which students were presented with a fact situation that involved a familiar legal issue, and were asked to make arguments rather than interpret arguments, as they had done during the training phase. To our knowledge, this interaction is a novel result in the self-explanation literature. The result seems consistent with the hypothesized advantages of specific prompts

relative to generic prompts. The surprise is how low the threshold is in terms of the ability level at which the advantages of generic prompts outweigh those of specific prompts.

The aptitude-treatment interaction discovered in our experiment is relevant to the design of the next generation of adaptive ITS that engage students in self explanation. Self-explanation is an attractive educational approach for developing intelligent tutoring systems for ill-defined domains. Even without formal domain models, systems can prompt students to explain learning resources. However, if the interaction between ITS and learner is mediated through self-explanation prompts, the design of these prompts is essential. The current experiment suggests further that the prompts should be adapted to the student's ability level and that some amount of feedback on students' self-explanation is desirable.

An open issue is how the interaction effect that we found bears on current theories of self-explanation. We are currently coding the self-explanations given by the subjects in our study in order to analyze them for specific characteristics that might qualitatively explain the interaction.

ACKNOWLEDGEMENTS

This research is sponsored by NSF Award IIS-0412830. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

REFERENCES

Aleven, V. (in press). An intelligent learning environment for case-based argumentation. *Technology, Instruction, Cognition, and Learning.*

Aleven, V. (2003). Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. *Artificial Intelligence,* 150, 183-237.

Aleven, V., & Koedinger, K. R. (2002). An effective meta-cognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*(2), 147-179.

Atkinson, R. K., Derry, S. J.; Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research 70*(2) 181-214.

Chi, M. T. H., de Leeuw, N., Chiu, M., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18,* 439-477.

Chi, M. 2000. Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In *Advances in Instructional Psychology*, 161-238. Hillsdale NJ, Lawrence Erlbaum.

Collins, A., Brown, J.S. & Newman, S.E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing and matematics. In L.B. Resnick (Ed.), Knowing, learning and instruction: Essays in honor of Robert Glaser (pp. 453-494). Hillsdale, NJ: Erlbaum.

Conati C. & VanLehn K. (2000). Toward computer-based support of meta-cognitive skills: a computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education, 11,* 398-415.

Dworkin, R. (1986). *Law's Empire.* Cambridge, MA: Harvard University Press.

Frank, J. (1930). *Law and the Modern Mind.* New York: Brentano's.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology 93*(3), 579-588.

Llewellyn, K. (1951). *The Bramble Bush.* New York: Oceana.

Muntjewerff, A. J., & Breuker, J. A. (2001). Evaluating PROSA, a system to train solving legal cases. In J. D. Moore, C. L. Redfield, & W. L. (Eds.), *Johnson Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future, Proceedings of AI-ED 2001* (pp. 278-285). Amsterdam: IOS Press.

Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (in press). Toward legal argument instruction with graph grammars and collaborative filtering techniques. *Proceedings ITS 2006.*

Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skills acquisition: A cognitive load perspective. Educational Psychologist, 38, 15-22.

Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: the effects of example variability and elicited self-explanations. *Contemporary Educational Psychology, 23*, 90-108.

Renkl, A. (2002). Learning from worked-out examples: Instructional explanations supplement self-explanations. *Learning & Instruction, 12,* 529-556.

Schworm, S., & Renkl, A. (2002). Learning by solved example problems: Instructional explanations reduce selfexplanation activity. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society,* (pp. 816-821). Mahwah NJ, Lawrence Erlbaum.

Simon, H. A. (1973). The structure of ill-structured problems. *Artificial Intelligence, 4,* 181-201.

VanLehn, K., Freedman, R., Jordan, P., Murray, C., Rosé, et al. (2000). Fading and deepening: The next steps for Andes and other model-tracing tutors. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Intelligent Tutoring Systems: 5th International Conference* (pp. 474-483). Berlin: Springer-Verlag.

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

Individualizing Self-Explanation Support for Ill-Defined Tasks in Constraint-based Tutors

Amali Weerasinghe
Intelligent Computer Tutoring Group
Dept. of Computer Science & Software Engineering, University of Canterbury, New Zealand
acw51@cosc.canterbury.ac.nz

Antonija Mitrovic
Intelligent Computer Tutoring Group
Dept. of Computer Science & Software Engineering, University of Canterbury, New Zealand.
tanja@cosc.canterbury.ac.nz

Abstract. We present the first phase of a project with the goal of developing a general model of self-explanation support, which could be used in constraint-based tutors for both well- and ill-defined domains. We studied how human tutors provide additional support to students learning with an existing intelligent tutoring system designed to help students learn an ill-defined task (database modeling using the ER model). Although the tutors were not given specific instructions to facilitate self-explanation, there were instances when self-explanation support was provided. Analysis of these interactions indicates that they have helped the students to improve their understanding of database design. These findings will serve as the basis for defining the self-explanation model. We also discuss directions for future work.

Keywords: self-explanation, intelligent tutoring systems, student modeling, ill-defined tasks

INTRODUCTION

Studies indicate that some students acquire shallow knowledge even in the most effective Intelligent Tutoring Systems (ITS) (Aleven et al., 1999). Self-explanation (SE) is described as an "*activity of explaining to oneself in an attempt to make sense of new information, either presented in a text or in some other medium*" (Chi, 2000), and has been shown to facilitate the acquisition of deep knowledge (Chi et al., 1989). There are several ITSs that facilitate self-explanation, most of them teaching well-defined tasks. For example, SE-Coach (Conati, and VanLehn, 2000) prompts students to explain solved physics examples. In the PACT Geometry Tutor (Aleven et al., 1999), students explain solution steps by selecting definitions and theorems from a glossary. NORMIT (Mitrovic et al., 2004) is an ITS for data normalization, in which students are expected to self-explain while solving problems. All these domains are closed-ended, as problem solving is well structured, and therefore self-explanation expected from learners can be clearly defined. Database design is an open-ended task: the final result can be defined in abstract terms, but there is no algorithm to find it. Constraint-based tutors have demonstrated their effectiveness in teaching ill-defined tasks, such as database design and querying (Mitrovic et al., 2004), and software design (Baghaei et al., 2005). We have extended the database design tutor with a SE facility (Weerasinghe and Mitrovic 2006). Even though all the above mentioned ITSs facilitate self-explanation, only SE-Coach supports adaptive self-explanation customised to the student's knowledge and self-explanation skills.

Our long-term goal is to develop a model of self-explanation which will provide adaptive support to learners for both well- and ill-defined tasks. Since we previously implemented self-explanation support for the database design tutor, the initial work on this project started with the same tutor. We are currently developing an SE model, which will be incorporated into EER-Tutor (Zakharov et al., 2005). In order to develop this model, we need to consider three basic decisions: when to prompt for self-explanation, what to self-explain and how to obtain self-explanations from learners. As the first step, we conducted an observational study to investigate how students interacted with EER-Tutor, while getting additional help by a human tutor through a chat interface.

A brief discussion of database design is given in the following section. We then discuss the functionality of EER-Tutor, followed by a description of the observational study. Analysis of the student interactions are presented in the Observations Section. We then discuss how the findings from the study can be incorporated in the self-explanation model. Future work and conclusions are presented in the final section.

DATABASE DESIGN

Database design is a process of generating a description of a database using a specific data model. Most database courses teach conceptual database design using the Entity-Relationship (ER) model, a high-level data model originally proposed by Chen (1976). The ER model views the world as consisting of *entities*, and *relationships*

between them. The entities may be physical or abstract objects, roles played by people, events, or anything else data should be stored about. Entities are described in terms of their important features (*attributes),* while relationships represent various associations between entities, and also may have attributes. There is no algorithm to use to derive the ER schema from a given set of requirements. The learner needs to decide on the appropriate constructs to use, such as types of attributes/entities. For example, the learner might be given a problem illustrated in Figure 1 (note that this is a very simple problem). From the problem text, it is obvious that *students* and *groups* are of importance. Therefore, the learner might start by drawing the entities first. Each student has an id, and the learner needs to use his/her world knowledge to realize that ids are unique, and therefore represent that attribute as a key attribute (shown on the diagram as underlined). The number assigned to each group is unique, and therefore it should also be a key attribute. In Figure 1, the student has made a mistake by showing GROUP as a weak entity, and group number as a partial key. Next, the learner has to think about the relationships between identified entities. In the problem shown in Figure 1, students work in groups, and for each possible association between a student and a group, it is necessary to represent the role. The Role attribute describes the association, and therefore it should be an attribute of the relationship. The student also needs to specify other integrities, such as cardinality ratios (shown as N on the diagram) and participations (shown as single or double lines).

As can be seen from this simple case, there are many things that the student has to know and think about when designing databases. The student must understand the data model used, including both the basic building blocks available and the integrity constraints specified on them. In real situations, the text of the problem would be much longer, often ambiguous and incomplete. To identify the integrities, the student must be able to reason about the requirements and use his/her own world knowledge to make valid assumptions.

Database design, similar to other design tasks, is an ill-defined task, because the start/goal states and the problem-solving algorithm are underspecified (Reitman, 1964). The start state is usually described in terms of ambiguous and incomplete specifications. The problem spaces are typically huge, and operators for changing states do not exist. The goal state is also not clearly stated, but is rather described in abstract terms. There is no definite test to decide whether the goal has been attained, and consequently, there is no best solution, but rather a family of solutions. Design tasks typically involve huge domain expertise, and large, highly structured solutions.

Although design tasks are underspecified, Goel and Pirolli (1992) identify a set of 12 invariant features of design problem spaces, such as problem structuring, distinct problem-solving phases, modularity, incremental development, control structure, use of artificial symbol systems and others. Problem structuring is the necessary first phase in design, as the given specifications of a problem are incomplete. Therefore, the designer needs to use additional information that comes from external sources, the designer's experience and existing knowledge, or needs to be deduced from the given specifications. Only when the problem space has been constructed via problem structuring, problem solving can commence. The second feature specifies three problem-solving phases: preliminary design, refinement and detail design. Design problem spaces are modular, and designers typically decompose the solution into a large number of sparsely connected modules and develop solutions incrementally. When developing a solution, designers use the limited-commitment mode strategy, which allows one to put any module on hold while working on other modules, and return to them at a later time.

In previous work, we have shown that constraint-based tutors are highly effective in teaching ill-defined tasks such as database design (Suraweera & Mitrovic, 2004) and query definition (Mitrovic & Ohlsson, 1999; Mitrovic et al., 2004). Our tutors compare the student's solution to a pre-specified ideal solution, which captures the semantics of the problem, thus eliminating the need for a problem-solver, which is difficult (or even impossible) to develop for such instructional domains. The constraint-based tutors are capable of identifying alternate correct solutions as constraints check that the student's solution contains all the necessary elements, even though it might be different from the ideal solution specified by the teacher. Goel and Pirolli (1988) argue that design problems by their very nature are not amenable to rule-based solutions. On the other hand, constraints are extremely suitable for representing design solutions: they are declarative, non-directional, and can describe partial or incomplete solutions. A constraint set specifies all conditions that have to be simultaneously satisfied without restricting how they are satisfied. Each constraint tests a particular aspect of the solution, and therefore supports modularity. Incremental development is supported by being able to request feedback on a solution at any time. At the same time, CBM supports the control structure used by the designer (student), as it analyses the current solution looking at many of its aspects in parallel: if a particular part of the solution is incomplete, the student will get feedback about missing constructs. CBM can be used to support all problem-solving phases. Therefore, we believe that CBM can be applied to all design tasks.

EER-TUTOR: ENHANCED ENTITY RELATIONSHIP TUTOR

EER-Tutor is aimed at the university-level students learning conceptual database design. For a detailed discussion of the system, see (Zakharov et al., 2005); here we present some of its basic features. The system complements traditional instruction, and assumes that students are familiar with the ER model. The system consists of an interface, a pedagogical module, which determines the timing and content of pedagogical actions,

and a student modeller, which analyses student answers and generates student models. EER-Tutor contains a set of problems and the ideal solutions to them, but has no problem solver. In order to check the student's solution, EER-Tutor compares it to the correct solution, using domain knowledge represented in the form of more than 200 constraints. It uses Constraint-Based Modelling (Mitrovic, et al, 2004) to model the domain and student's knowledge. The interface (illustrated in Figure 1) is composed of three windows tiled horizontally. The top window displays the current problem and provides controls for stepping between problems, submitting a solution and selecting feedback level. The middle window is the main working area, in which students draw ER diagrams.

Feedback from the system is grouped into six levels according to the amount of detail: *Correct*, *Error Flag*, *Hint*, *Detailed Hint*, *All Errors* and *Solution*. The first level of feedback, *Correct*, simply indicates whether the submitted solution is correct or incorrect. The *Error Flag* indicates the type of construct (e.g. entity, relationship) that contains the error. For example, when the solution in Figure 1 is submitted, *Error Flag* provides the message *Check your entities, that's where you have some problems*. This is associated with the error GROUP being modelled as a weak entity instead of a regular entity. *Hint* and *Detailed Hint* offer a feedback message generated from the first violated constraint. For the solution in Figure 1, the hint message is *Check whether all the weak entities are necessary. Check whether some of your weak entities should be represented using some other type of construct*. On the other hand, the corresponding detailed hint is more specific: GROUP *should not be an entity. It may be extra or you may want to represent it using some other type of construct*, where the details of the erroneous object are given. Not all detailed hint messages give the details of the construct in question, since giving details on missing constructs would give away solutions. A list of feedback messages on all violated constraints is displayed at the all errors level (as indicated in the right-hand pane in Figure 1). The ER schema of the complete solution is displayed at the final level (solution level).

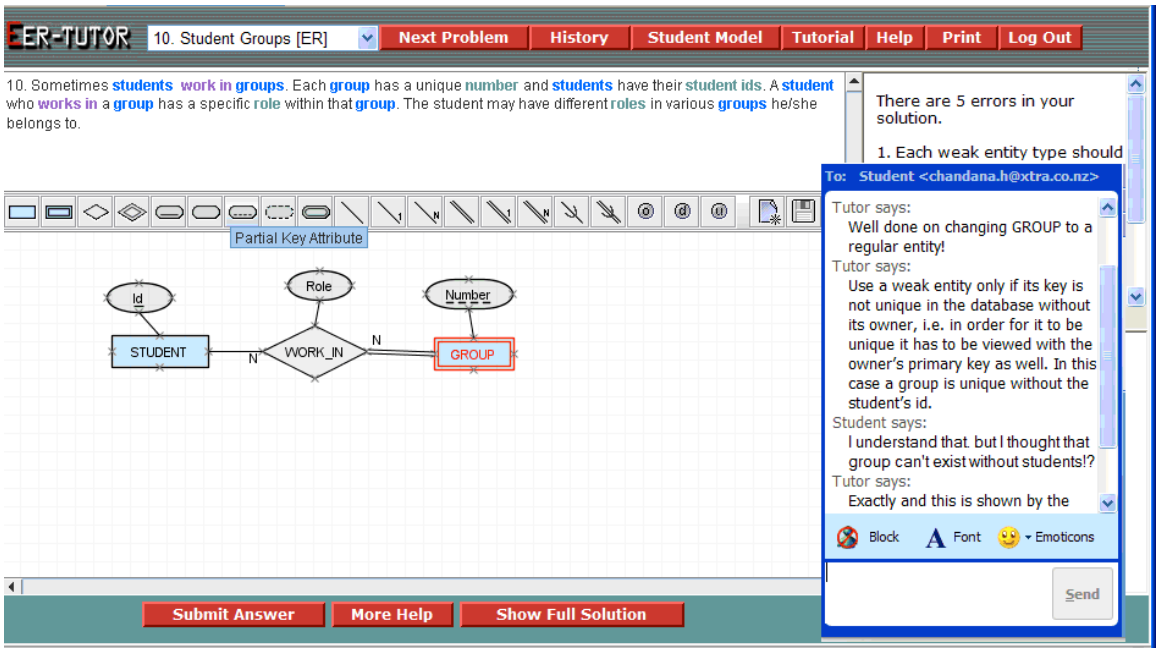

Fig. 1. Interface of the enhanced EER-Tutor

When the student submits the first attempt at a problem, a simple message indicating whether or not the solution is correct is given. The level of feedback is incremented with each submission until the feedback level reaches the detailed hint level. In other words, if the student submits the solutions four times, the feedback level would reach the detailed hint level, thus incrementally providing more detailed messages. Automatically incrementing the levels of feedback is terminated at the detailed hint level to encourage the student to concentrate on one error at a time rather than all the errors in the solution. The system also gives the student the freedom to manually select any level of feedback according to their needs.

PRELIMINARY STUDY

The study was conducted in August 2005 at the University of Canterbury, and involved students enrolled in an introductory database course and experienced tutors. These experienced tutors will be referred to as tutors, while EER-Tutor as the system or the ITS hereinafter. All the tutors had several years of experience providing assistance to students in labs and/or teaching small groups. The study was scheduled only after the relevant material was taught in the classroom. The version of EER-Tutor used was enhanced with a chat interface (Figure 1), so that the tutors could provide one-to-one feedback to students. We wanted to make the bandwidth between

the student and the tutor to be similar to that between the student and the ITS. As a result, tutors could observe only the students' interactions with the ITS. Participants and tutors were located in separate rooms.

The tutors were expected to guide the students towards solutions using appropriate methods like asking questions etc. However, they were not given any specific instructions on providing assistance. Student participants were not told that a human was involved in the study. They also had the opportunity to initiate intervention through the chat interface or the *More Help* button in the interface.

At the beginning of the study, the students were asked to sit a pre-test online. All learner interactions were recorded. Students were expected to use the system for at least an hour. However, the students themselves decided when to end the session. Although initially we wanted the participants to sit a post-test immediately after the study, it was not possible due to another evaluation study which was conducted simultaneously. Therefore, the post-test was administered later on. All participants were asked to fill out a questionnaire at the end of the session to understand their perceptions about the system and interventions through the chat interface. At the end of each session, the tutors were also interviewed to understand their views on the tutoring experience.

We initially analysed the recordings without the tutors, to investigate how students were prompted by different tutors. As the second step, whenever possible, the recordings were analysed with the tutors to clarify how they decided on the timing and the level of feedback provided through the chat interface.

The experimental set up of this study varies from previous studies of tutorial dialogue in a number of ways. First, the tutors were expected to provide additional support to the feedback given by the system. The tutors were also expected to respond to learners' questions. This contrasts with those studies of Chi. et al. (Chi and Siler, 2001) and Graesser, Person et al. (1995) in which the tutor was expected to lead the dialogue through a series of questions. Second, the learner interacts both with the system and the tutor. Although Merill et al. (1992) have studied tutorial dialogues in the context of problem-solving, the tutor was the only source of feedback for the student as s/he solved problems on paper. Finally, the tutors in our study needed to decide not only how to guide the student but also when. This differs significantly from the study in which the tutors analysed recorded interactions of students to perform motivation diagnosis (De Vicente and Pain, 2002).

OBSERVATIONS

Seven students and four professional tutors participated in the study, with at most two students per tutor. The mean on the pretest was 75.5% (sd=17.9), which was higher than the performance of the whole class (mean=58.1, sd=23.5). We expected this, as the participants were self-selected. Still the range of background knowledge was sufficiently large (ranging from 57% to 100%). Only two students have completed the post-test, hence it is not possible to compare the effect the learner interactions had on performance. The average duration of the sessions was 85 minutes (sd=20). The average number of attempted problems was 11 (sd = 5), and all participants completed all attempted problems. Average number of attempts per problem is 2.8 (i.e. received feedback from EER-Tutor that many times). We discuss observations in two different categories: (i) type of feedback provided in the interventions and (ii) timing of interventions.

Type of Feedback Provided

The interactions between the tutors and the students were analysed to identify different episodes, each pertaining to a single topic (Chi, 2001). There were a total of 69 episodes. In addition to discussing the current problem state, some episodes focused on helping with the interface (such as labeling constructs), motivate and praise the student, suggest to try a more challenging problem, complete the session or help with technical problems (e.g. web browser suddenly closing). The maximum and the minimum number of episodes initiated by a tutor during a session was 20 and 4 respectively. Surprisingly, these 20 episodes occurred in a session of 1.5 hour duration which is not the longest session (the longest session lasted approximately 2 hours). In the session which consisted of 4 episodes, the first intervention occurred only in the 19th problem (the student completed 22 problems).

We are mainly interested in 37 episodes which discussed the current problem state or the relevant domain concepts. The following statistics were calculated using these 37 episodes. The average number of such episodes per tutor was 9.25. Five episodes contained a single utterance each, which was initiated by the tutor. For instance, a tutor utterance that occurred just after the completion of a problem was "Remember that the participation for weak entity is always total". The longest episode consisted of 9 utterances of which 4 were by the tutor. The student made more utterances than the tutor in only 2 episodes. Furthermore, only 2 episodes were student-initiated. This indicates that the tutor is more likely to be active in the interventions.

Only 20 (54%) episodes were considered to facilitate self-explanation. The criterion to label an episode as facilitating self-explanation is whether it discussed concepts that went beyond the current error, or facilitated justifications for the correct modelling decision. An example is presented in Figure 2, which occurred while the student was working on the problem shown in Figure 1. This dialogue contains two episodes, because it covers two different concepts (one related to weak entities, and the other one about total participation). Even though the tutor is leading the dialogue, the student has justified his decision for modeling GROUP as a weak entity

(*Student1*). Therefore, the dialogue provided an opportunity for the student to explicitly self-explain his modeling decision. Also, the student was able to identify and repair the misconception he had with weak entities and total participation after the tutor's explanation (*Tutor3*). Student's utterance about learning material during this session (*Student2*) can be considered as evidence of implicit self-explanation.

Tutor1:	Well done on changing GROUP to a regular entity!
Tutor2:	Use a weak entity only if its key is not unique in the database without its owner, i.e. in order for it to be unique, it has to be viewed with the owner's key as well. In this case, a group is unique without the student's id.
Student1:	I understand that, but I thought that group can't exist without students!?
Tutor3:	Exactly, and this is shown by the total participation in the relationship.
Student2:	I have learned something today

Fig.2. A dialogue from the study

The highest number of self-explanation dialogues in a session was 7, while the lowest was 2. As can be expected, the highest number of self-explanation dialogues occurred in the longest session. Even though all tutors initiated interaction episodes, two of them did not have any self-explanation episodes. This may be because the tutors were not explicitly asked to facilitate self-explanation, but to assist the students with problem solving.

When data was analysed to identify different strategies used by tutors, three strategies were prominent. Tutors were rephrasing feedback, providing problem-independent explanations and stating their observations before starting to discuss the problem state. The tutors who did not have any self-explanation eposides in their sessions mainly rephrased feedback to enable the student to understand their own mistakes. For example, the tutor prompted "Does AUTHOR need to be an entity?" or "The cardinalities of BORROWED_FROM needed fixing". Rephrasing feedback may have been effective because most students realised that the additional feedback was provided by a human observing their problem-solving process. If the self-explanation model is to repeat the same kind of prompting, it is difficult to ascertain whether it will have the same effect (Lepper, et al., 1993). The second strategy was to discuss the current problem state and then provide a problem-independent explanation. Figure 1 represents an example. These explanations provided an opportunity for the student to repair his/her mental model of the domain and generated further conversation. The third strategy used was to state the tutor's observations before starting to discuss the problem state. For example, tutor started the dialogue by saying "You seem to be having a few problems with relationships. Think about this. Can a student be enrolled in a course without involving a department?"

As the knowledge base in EER-Tutor is represented as a set of constraints, the errors were recorded as constraint violations. We analysed how frequently constraints were violated after related errors were discussed in self-explanation episodes, to see whether tutor interventions helped students to improve their knowledge. If these constraints represent psychologically appropriate units of knowledge, then learning should follow a smooth curve when plotted in terms of constraints (Anderson, 1993). To evaluate this expectation, the participants' logs were analysed, and each problem-state after a tutor invention in which a constraint was relevant was identified. These identified occasions are referred to as *occasions of application* (Mitrovic, et al, 2004). Each constraint relevance occasion was ranked 1 to *n*. For each occasion we recorded whether a relevant constraint was satisfied or violated. We then calculated the probability of violating a constraint on the first occasion of application, the second occasion and so on, for each participant. The probabilities were then averaged across all participants and plotted as a function of the number of occasions when a constraint was relevant (Figure 3).

As can be seen from Fig. 3.a, there is an outlier, increasing the probability of violating a constraint in the 4^{th} and the 5^{th} occasions. This is due to a single student violating the constraint dealing with total participation. For this student, the tutor provided a problem-independent explanation to help him identify and repair the misconception he had with weak entities and total participation (Figure 1). The explanation was not related to a problem state later on, as the discussion was a follow-up from another error related to weak entities. This may have been a reason for the subsequent violations of this constraint.

Figure 3.b shows the learning curve with the outlier removed. The probability of 0.22 for violating a constraint at its first occasion of application decreased to 0.02 at its eighth occasion of application, displaying a 90.9% decrease in the probability. The results of the mastery of constraints reveal that students seem to learn ER modelling concepts which were discussed by the tutors.

Twenty-eight different constraints were discussed in the self-explanation episodes. Three students did not violate any constraints in subsequent occasions after the tutor interventions. These students were tutored by three different tutors who followed strategies like rephrasing feedback, providing problem-independent explanations and stating tutor's observations at the beginning of the discussion. This suggests that all these strategies have been effective in helping the students learn domain concepts.

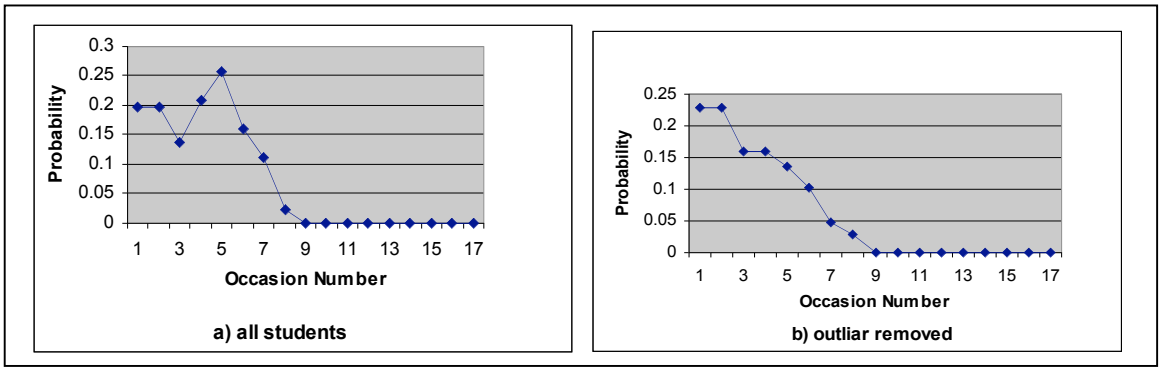

Fig. 3. Probability of violating a constraint as a function of number of occasions when that constraint was relevant

Timing of Interventions

The original version of EER-TUTOR provides feedback on demand, i.e. only when the student submits the solution. The tutors in this study also provided delayed feedback, which was well-received by the participants. Delayed feedback also provided an opportunity for students to correct the mistakes themselves. There were few instances where the student made a mistake and corrected it after referring the problem text again. For example, one of the problems required students to model CAR as an entity and *Colour* as a multi-valued attribute of CAR. The student modelled *Colour* as a simple attribute and then changed it to multi-valued as the last sentence in the problem text indicated that a car can have many colures. In such a situation, immediate feedback would not have been welcomed by the student, as he may have felt the intervention being intrusive.

The important issue with delayed feedback is how the tutors decided that the students needed help. In our study tutors provided help when the student (i) made the same type of mistake repeatedly (ii) asked for more help using the *More Help* button (iii) was inactive for some time, (iv) reacted to feedback, or (v) asked a problem-specific question through the chat interface. These scenarios will be discussed in detail in the next section.

Prototype of the Self-Explanation Model

The self-explanation model will be used to decide when to prompt to self-explain, what to self-explain and how to obtain self-explanations from learners. The model consists of three parts: error hierarchy, self-explanation dialogues and meta-constraints. Each component is now described in turn.

A hierarchy of errors was developed after analyzing different students' errors (Weerasinghe, 2003). A high-level view of the hierarchy is given in Figure 4. Nodes in this hierarchy are ordered from basic domain principles to more complicated ones. Violated constraints for each type of error are represented as leaves of the hierarchy. Some error types are further divided into sub errors. Constraints for some nodes are given in separate lines to indicate that each constraint corresponds to a specific error type. For instance, each constraint assigned to the node *Syntax Errors* specifies a different type of error. The error hierarchy enables the system to locate the appropriate dialogue to be used in case of incorrect student submission. If there are multiple errors, depth-first search identifies the dialogue to initiate with the student.

Self-explanation is facilitated through tutorial dialogues. We designed a tutorial dialogue for each type of error. Each dialogue consists of 4 stages. In the first stage, the model informs the student about the concept that s/he is having difficulty with, and then asks for a justification of the student's action. The purpose of the second stage is to assist the student in understanding why his action is incorrect. The third stage prompts the student to specify how to correct the mistake. A further opportunity is provided in the fourth stage to review the domain concept learned. An example of a tutorial dialogue is given in Figure 5. Initially, the system identifies the domain concept the student has problems with, and asks the student to explain it (*EERTutor1*). We have yet to decide how to obtain self-explanation from learners i.e. whether to provide a list of possible answers from which the correct one could be selected or to incorporate natural language processing module enabling learns to use free-form questions. If the student fails to provide the correct answer (*Student1*), s/he will be asked a more specific question that provides a further opportunity to understand the fundamental principle that is violated (*EERTutor2*). However, if s/he fails to correct the mistake even after going through a series of detailed questions, as the last resort the tutor will provide an explanation on how to the correct the mistake together with a brief description about the fundamental principle that needs to be learnt (*EERTutor5-7*). The dialogues use various types of interactions such as simple questions (*EERtutor1*), fill-in-a-blank (*EERTutor7*), or true-false questions, to motivate the student to self-explain. When a certain mistake is repeated, the model informs the student of its observations (*EERTutor1*), thereby providing an opportunity to reflect on his/her domain knowledge. As all

dialogues facilitate self-explanation by pointing out errors (*EERtutor3*), students are given opportunities to reflect on their problem solving procedure, which is another important meta-cognitive skill.

ALL ERRORS
- Syntax errors
  - 1 – More than one construct is used to model a word of the problem
  - 2 – Database does not have any entities
  - 3 – A connector is not connected to two entities
  - 4 – A connector is used to connect attributes is not a single line with no cardinality
  - 5 – An attribute is connected to more than one other attribute
  - 6 – An attribute is connected to an entity and another construct
  - 101 – An attribute is connected to a relationship and another construct
  - 7 – An entity is directly connected to another entity
  - 8 – An attribute is not connected to any other construct
  - 9 – A relationship is connected to another relationship
  - 10 – Names of entities and relationships are not unique
  - 21 – A relationship is not connected to two entities
- Semantic errors
  - Using an incorrect construct type
  - Extra constructs
  - Missing constructs
  - Connecting an attribute to an incorrect construct
  - Errors dealing with cardinalities and participation

Fig. 4. Overall view of the error hierarchy

Ideal SE behaviour will be represented as a set of meta-constraints, and will enable individualization of the self-explanation dialogues. As we discussed previously, the SE dialogues are pre-specified sequences of questions; however, for each individual student, the SE model will decide on the entry point to the dialogue, or the timing of the dialogue. The observations from the study will be used to develop meta-constraints.

As delayed feedback provided by the tutors in this study was well-received by the student participants, the self-explanation model will also provide delayed feedback. The critical issue then is to decide when it will be beneficial to intervene. As noted earlier, tutors intervened when the student (i) made the same type of mistake repeatedly, (ii) asked for more help using the *More Help* button, (iii) was inactive for some time, (iv) reacted to feedback, or (v) asked a problem-specific question through the chat interface.

When a student made the same error repeatedly, tutors provided a problem-independent explanation of the domain concept they have difficulty with. Some tutors initiated the dialogue by stating their observations. For instance, if it is difficult for the student to identify a weak entity, the tutor's initial response was "*You seem to be having some difficulty with regular entities. Let's look at regular entities in detail*." followed by an opportunity to discuss the corresponding domain concept. One meta-constraint will check for these situations, and will be violated when the same error is made in the last *n* attempts. In that case, a dialogue corresponding to the mistake will be initiated, but the dialogue would start from the problem-independent question (*EERTutor1* in Figure 5).

As noted in (ii) (asking for more help using the *More Help* button), the student will be given an opportunity to receive more help for each feedback message provided by the system. If more help is requested, then the corresponding self-explanation dialogue will be initiated. For instance, if the student requested more help on CHAPTER being modeled as a regular entity and he has not made the mistake repeatedly, then dialogue will discuss the error within the current context (*EERTutor3* in Figure 5). Hence the self-explanation dialogues will be adaptive based on the student's domain knowledge and the self-explanation skills.

Even though all tutors intervened when a student has been inactive for about a minute, they tended to wait longer when the student is in the initial problem-solving phase (i.e. student has not submitted his solution and has not received any feedback from the system so far). One of the meta-constraint will identify that situation, by checking whether the student has made any attempts at the current problem, and has been inactive for a specified period of time. (such as 1.5 minutes, the time period we observed in the study). Violation of this constraint will initiate an evaluation of the student's solution even though it has not been submitted yet, and also prompt the student to identify which concept he is having difficulty with. For instance, if the student is having difficulty with the regular entity CHAPTER, and the evaluation of his solution identifies that it needs to be modeled as a weak entity, then the self-explanation model will help the student to understand the correct modeling decision through a series of questions. Figure 5 presents a sample dialogue that occurs between the student and the tutor in this case.

A student seems to be just reacting to feedback if he makes a single change without reflecting on the other changes that need to be carried out as a result. In ER modeling, certain changes trigger a number of other changes

in order to develop a syntactically correct model. For instance, if a regular entity with a key attribute is changed to a weak entity, then a partial key should be specified instead of the key attribute. Also one of the relationships of the regular entity needs to be changed to an identifying relationship etc. Sometimes students tend to make a single change suggested by the system and submit again for more feedback. This may lead to frustration and a feeling of exhaustion due to the number of attempts that the student has to go through to arrive at the correct solution. In such a situation, the student will be prompted to reflect on other related changes before submitting the solution. An example meta-constraint which identifies such situations is given below; when this constraint is violated, the system will remind the student to check whether other changes are necessary.

Relevance Condition: If the student has changed the solution to correct one mistake (i.e. violated constraint),
Satisfaction Condition: Then he should also make related changes.

For scenario (v) (asking problem-specific questions through the chat interface), the self-explanation model needs to enable learners to ask free-form questions while interacting with the system. We are currently investigating different alternatives. One alternative is to use TFIDF (Term Frequency Inverse Document Frequency) vector weighting scheme (Salton and Buckley, 1987) to retrieve information from a pre-defined question database. TFIDF is currently being used in another project by our group developing a question asking module while solving database mapping problems with the ERM-Tutor (Milik, et al., 2006). Another alternative is to incorporate a natural language processing module to the EER-Tutor.

EERTutor1: You seem to be having some difficulty with regular entities. Let's look at regular entities in detail. Can you tell me the general rule to decide whether something is a regular entity?
Student1*: I don't know*
EERTutor2: Ok! Let's try a simple thing. What is used to uniquely identify instances of a regular entity?
Student2*: The key attribute*
EERTutor3: Now tell me what is unique about *CHAPTER* regular entity?
Student3*: I'm not sure*
EERTutor4: Let me help you a bit. Check carefully whether any of the attributes of *Chapter* can uniquely identify the instances of *Chapter?*
{After sometime} Please tell me whether you managed to find any attributes
Student4: *No*
EERTutor5: Fine. It doesn't have any attributes that can uniquely identify instances of *Chapter*. So now can you tell me why modeling *Chapter* as a regular entity is incorrect?
Student5 : *Because it doesn't have a key attribute.*
EERTutor6: Great! Now tell me the best way to model *CHAPTER*?
Student6*: A weak entity*
EERTutor7: Well Done! Now select the correct pair of words to fill in the blanks so that you'll remember what you learnt.
A ….. is used to uniquely identify instances of a …….. entity.
Student7*: key attribute ,regular*
EERTutor8: Excellent, Go ahead and do the changes. Submit the solution when you're done.

Fig. 5. A sample tutorial dialogue

The short-term student model in EER-Tutor consists of a list of satisfied and a list of violated constraints for the student's solution, while the long-term model records the history of each constraint (Zakharov et al., 2005). In the enhanced version of EER-Tutor, the long-term model will additionally record details of the self-explanation skill of a student in terms of types of errors made (in terms of meta-constraints) and the level of prompting the student needed to correct his mistake for every constraint.

Conclusions and Future Work

Self-explanation is an effective learning strategy to facilitate deep learning. This research focuses on developing a self-explanation model for both ill- and well-defined tasks. As the first step, we conducted a preliminary study to observe how tutors prompt students to guide them towards solutions while using EER-Tutor, a constraint-based tutoring system for learning Entity-Relationship modelling. In addition to the feedback received by the system, the students were prompted by the tutors through a chat interface. Students also had the opportunity to initiate a dialogue with the tutor either through the chat interface or using the *More Help* button.

The interactions between the tutors and the students were analysed to identify the different episodes, each pertaining to a single topic. Only 20 (54%) of these episodes that discussed the current problem state or the relevant domain concepts were considered to facilitate self-explanation. These episodes either facilitated the discussion of domain concepts that went beyond the current error, or prompted justifications for the correct

modelling decision. The user logs were analysed to investigate how frequently a certain error occurred after each self-explanation episode. The analysis indicated that in spite of different tutoring strategies, the tutor interventions helped the learners to improve their understanding of ER modelling concepts.

The findings from the reported study are being used to develop the self-explanation model for the EER-Tutor. The next step is to incorporate the model into the EER-Tutor, and evaluate it in an authentic classroom environment. We will then implement the same SE model in an ITS for a well-defined task.

References

Aleven, V., Koedinger, K. R., Cross, K. Tutoring Answer Explanation Fosters Learning with Understanding. In: Artificial Intelligence in Education, Lajoie, S.P. and Vivet, M.(eds.), IOS Press (1999) 199-206.

Anderson, J. R., Rules of the mind. Erlbaum, Hillsdale, NJ, 1993.

Baghaei, N., Mitrovic, A., Irwin, W. A Constraint-Based Tutor for Learning Object-Oriented Analysis and Design using UML. In: C.K. Looi, D. Jonassen, M. Ikeda (eds), ICCE 2005, 11-18.

Chen, P. (1976). The ER Model - Toward a Unified View of Data. ACM Transactions Database Systems, 1(1), 9-36.

Chi, M. T. H. Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, (2000) 161-238.

Chi, M.T.H., Bassok, M., Lewis, W., Reimann, P., Glaser, R., Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. Cognitive Science, 13 (1989), 145-182.

Chi, M. T. H., S. A. Siler, et al. Learning from human tutoring. Cognitive Science 25 (2001), 471-533.

Conati, C., VanLehn, K. Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. Int. J. Artificial Intelligence in Education, 11 (2000) 389-415.

Goel, V., Pirolli, P. Motivating the Notion of Generic Design with Information Processing Theory: the Design Problem Space. AI Magazine, 10 (1988) 19-36.

Goel, V., Pirolli, P. (1992) The Structure of Design Problem Spaces. Cognitive Science, 16, 395-429.

Graesser, A. C., Person, N. et al. Colloborative dialog patterns in naturalistic one-on-one tutoring. Applied Cognitive Psychology, **9**, (1995) 359-3

Lepper, M.R., Woolverton, M. Mumme, D. L., and Gurtner, J.L. Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In Lajoie, S.P. and Derry, S. J. (eds.) Computer as Cognitive Tools, Lawrence Erlbaum, Hillsdale, New Jersey, (1993)75 -105.

Milik, N., Marshall, M., Mitrovic, A. Teaching Logical Database Design in ERM-Tutor. In: M. Ikeda & K. Ashley (eds) Proc. ITS 2006.

Mitrovic, A., Ohlsson, S., Evaluation of a Constraint-Based Tutor for a Database Language. Int. J. Artificial Intelligence in Education, 10 (1999), 238-256.

Merrill, D. C., Reiser, B. et al. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. Journal of the Learning Sciences 2(3) (1992) 277-305.

Mitrovic, A., Suraweera, P., Martin, B., Weerasinghe, A. DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Journal of Interactive Learning Research, 15(4), (2004) 409-432.

Reitman, W.R. (1964) Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-defined Problems. In: M.W. Shelly, G.L. Bryan (eds) Human Judgements and Optimality. New York, Wiley.

Salton, G., Buckley, C., Term Weighting Approaches in Automatic Text Retrieval. Technical Report #87-881 Computer Science Dept, Cornell University, Ithaca, NY, 1987.

Suraweera, P. and Mitrovic, A., An Intelligent Tutoring System for Entity Relationship Modelling. *Int. J. Artificial Intelligence in Education*, v14n3-4, 375-417, 2004.

Weerasinghe, A. Exploring the effects of Self-Explanation in the Context of a Database Design Tutor. MSc Thesis, University of Canterbury, 2003.

Weerasinghe, A., Mitrovic, A. Facilitating Deep Learning through Self-Explanation in an Open-ended Domain. Int. J. of Knowledge-based and Intelligent Engineering Systems, 10(1),(2006) 3-19.

Zakharov, K., Mitrovic, A., Ohlsson, S., Feedback Micro-engineering in EER-Tutor. In: Looi, C-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. Artificial Intelligence in Education AIED 2005, IOS Press, (2005) 718-725.

De Vicente, A., Pain, H. Informing the detection of the students' motivational state: An empirical study. In: Cerri, S.A., Gouarderes, G., Paraguacu, F. (eds.), Proc. 6th Int. Conf. Intelligent Tutoring Systems (2002) 933-943.

Guidance and Collaboration Strategies in Ill-defined Domains

Toby Dragon
Department of Computer Science
University of Massachusetts -Amherst
dragon@cs.umass.edu

Beverly Park Woolf
Department of Computer Science
University of Massachusetts -Amherst
bev@cs.umass.edu

Abstract. We developed tutors to support critical thinking in four ill-defined domains. Students work with science or art history tutors and receive feedback about their arguments as they sort, filter and categorize data. A coach provides guidance using a set of expert rules and an expert knowledge base, which creates the basis for the content-specific analysis of the student's argument. This paper describes these tutors, their current functionality and our future research to improve both guidance for an individual student and collaboration tools for multiple students.

Keywords: Ill-defined domains, problem-based learning, hypothesis generation, case-based coaching, collaboration, argumentation.

INQUIRY TUTORS FOR ILL-DEFINED DOMAINS

Introduction

A frequently mentioned weakness of education is that, in an effort to get through as much material as possible, many topics are covered briefly and stripped of their contextualizing complexity (Eylon & Linn, 1988; Koschmann et al., 1994). Especially in science education, students are left with a dizzying array of new terms and concepts that have no relationship to their prior knowledge or their everyday experience. Much of the learning that does occur provides the student with only inert knowledge (Eylon & Linn, 1988). For these reasons, teaching these ill-defined domains provides both a major challenge and a major opportunity for intelligent tutoring systems. We propose a set of tools and a working environment that allows students to successfully approach complex problems while still in context. The system supports the students while they structure and relate the knowledge that they gain, and provides feedback about both the domain knowledge and the learning process.

Our system, Rashi, takes an inquiry learning approach to these ill-structured domains. Inquiry learning has been shown to be more successful than typical classroom instruction: it stimulates interest and engages students (White & Frederiksen, 1995; Shute & Glaser, 1990). When students manipulate artifacts themselves and think freely about problems, they become more actively involved and generally become more systematic and scientific in their discovery of laws (White & Frederiksen, 1995). The increased interactivity alone has been shown to increase learning (Shute & Glaser, 1990).

The Rashi tutor currently supports reasoning in four domains; geology, biology, art history and forestry; and has been tested with hundreds of higher education students. Our inquiry tutors for science and art history contain open questions and problems with unfocused and ambiguous solutions. The tutor presents authentic cases in their contextual complexity and invites students to generate questions to investigate the case, gather and analyze data, and generate hypotheses based on inferences made from evidence. We now describe the set of tools provided by Rashi, and how these tools support a student's critical thinking.

Cognitive Tools

Rashi provides both an expansive method of presenting information and a set of tools to help students access and organize information.[1] It is domain independent and multi-disciplinary in that several disciplines share the same infrastructure and generic tools, Figure 1.

Data collection tools include:

[1] These tutors are located at http://ccbit.cs.umass.edu/Rashihome/projects/

- *The Image Explorer*: students click "hotspots" to move to new images, view video clips, and to collect data.
- *The Interview Tool*: students ask questions and receive audio, video, and text answers.
- *The Concept Library*: a hyper-text tool in which students read about content knowledge and collect snippets of text for their notes.
- *The Source Editor*: students access references to information outside of the Rashi system, such as websites and textbooks the instructor finds important. Students can also cite resources they find on their own.

Once data is collected, *critical thinking tools* help students formulate hypotheses, organize evidence and construct arguments, within a central data repository. These tools include:

- *The Inquiry Notebook*: facts collected from data gathering tools are automatically entered into the notebook and students enter facts directly as well.
- *The Argument Editor*: students create hypotheses and inferences and drag data from the notebook to support or refute their claims.

The data collection tools have the ability to present complex and ill-structured topics of study. The critical thinking tools help students to organize knowledge and evaluate real-life questions within these topics without simplifying them or disconnecting them from their context. This helps students not only learn the domain knowledge within the system, but helps prepare students for further study by providing them with a system and set of concepts which allow them to operate in real-world scenarios. If a student needs to develop her understanding beyond the internal representation the tutor provides, she can use outside resources, cited either by herself or the instructor in the Source Editor, and still use the same organizational tools to facilitate learning. This is especially important in these ill-defined domains because one can very seldom assume that all necessary or useful knowledge will be included in the internal representation of the domain.

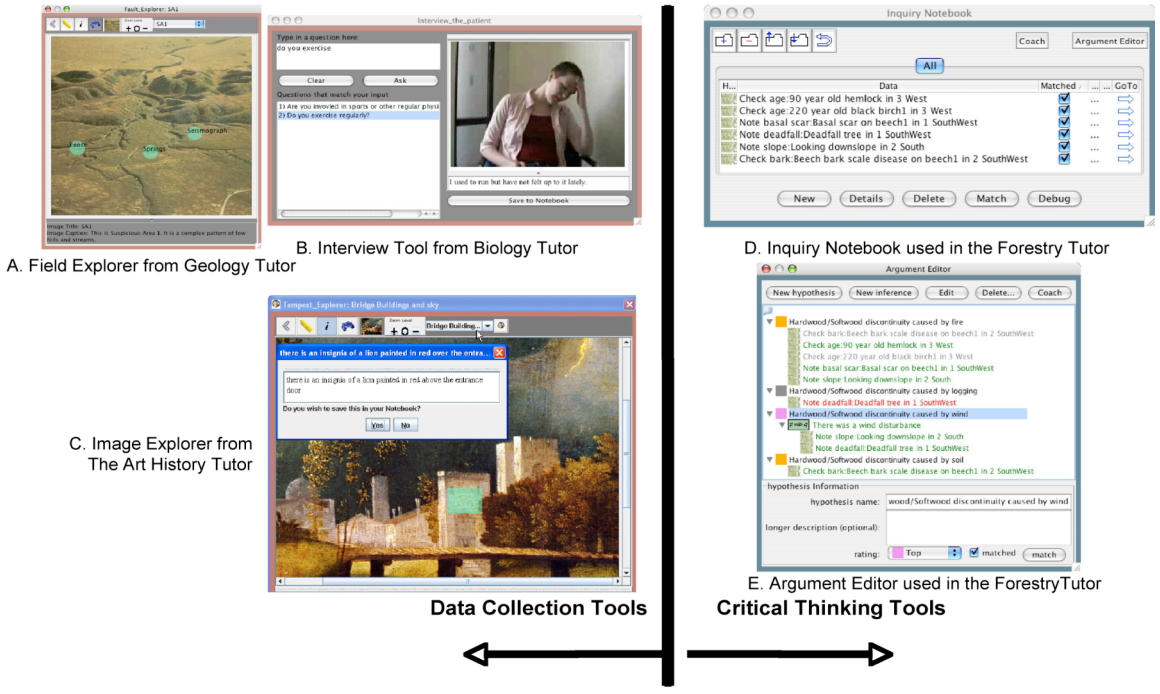

Figure 1. Data Collection and critical thinking tools are available in all the inquiry tutors.

The tools prepare students for further study in ill defined spaces by providing a structure that can be used in real-world scenarios, e.g., the same tools or conceptions can be used with resources outside of the tutoring environment. Different data collection methods (interactive text, still images, video and dynamic maps) create a broad and open-ended space for student exploration and acquaint them with methods commonly used by scientists in the field. This infrastructure has been used in three science domains, biology, geology and forestry (Woolf et al., 2002; 2003; 2005; Dragon et al., 2006).

Another reason the Rashi System is beneficial in these types of domains is that the tutor does not enforce a particular order of student activity; rather it allows students to move opportunistically from one phase to another. For example, in the Biology Tutor, students read a case description and use tools such as the Physical Examination and Laboratory to identify the patient's signs and symptoms. They might use the Interview Tool to

ask the patient about her complaints and organize physiological signs, medical history or patient examinations in the Inquiry Notebook. This can be considered a learner-centered approach, because it enables students to explore and query in a variety of ways. "Learner-centered" software increases the educational opportunity for all students by accommodating each learner's individual differences (Bransford et al., 1999). Rashi allows students to discover their own approach to solving cases, and also prepares them for real-life scenarios where one has no instruction as to what the next step should be.

When providing a system that teaches complex domains, one must provide both an expansive method of presenting information as well as useful and understandable tools to help students access and organize the information and their own thoughts. We have seen the set of tools Rashi provides to accomplish this task. Now let us look to how Rashi attempts to provide guidance to the student in these ill-defined domains.

GUIDING STUDENT LEARNING

Providing an appropriate environment to support learning is essential, yet the Rashi environment alone is not enough; students still become confused, lost, and acquire misconceptions while exploring the domains. For these reasons, we have developed and tested a strong coaching component that guides student exploration. The coach uses an expert system, detailing the domain knowledge that the teacher would like to impart. As a student moves through the inquiry cycle, the tutor matches the student's reasoning with the expert knowledge (Dragon et al., 2006) and provides five feedback types:

1. *Hypothesis Feedback*: promotes consideration of multiple hypotheses and offers a list of hypotheses from the expert knowledge base.
2. *Support and Refutation Feedback*: encourages top-down argument construction by urging students to supply supporting or refuting data for their arguments.
3. *Argument Feedback:* encourages bottom-up argument construction by urging students to consider higher-level arguments, possibly offering an argument the student might have missed.
4. *Relationship Feedback*: helps students identify relationships between propositions used in their arguments.
5. *Wrong Relationship Feedback*: identifies contradictions between relationships in the student argument and the relationships in the expert knowledge base, Figure 2.

Feedback includes contextual information about the case being solved. For example, when the tutor informs a student she needs more support for a certain argument, it can also bring her to a location where this support can be found. When correcting a student's relationships, Figure 2 bottom, the tutor can directly address and correct a student's misconceptions. The coach also promotes good inquiry behavior by encouraging students to engage in sound reasoning. For example, if a student randomly collects data and does not explicitly make arguments about the case, the coach will ask her to propose a hypothesis. Once the student makes a hypothesis, the coach urges the student to support or refute it with evidence.

Teachers can adjust the coach in a number of ways: specifying the order in which competing domain knowledge is presented, and promoting a certain order to the inquiry process (Dragon et al., 2006). For example, an instructor might want students to see a crucial piece of evidence before anything else when the student is considering a certain hypothesis. The teacher assigns nodes in the expert knowledge base with an *importance* rating that guides the coach in how to select among nodes that are otherwise equivalent. This allows the instructor to include large amounts of data in the expert knowledge base and still rest assured the student will be pointed toward the most important data first. The instructor can also specify the order in which the different types of feedback are presented. This allows them some control over the ordering of steps in the inquiry process that the coach will present. For example, if the instructor wants the students to collect data before creating relationships, she can make the support and refutation feedback come before relationship feedback to promote this type of behavior. These features allow the instructor who authors the case some ability to fine-tune the behavior of the coach, guiding students through ill-structured spaces in a specified way.

Future Work On Guidance

While the coach does currently engage in these activities, there is still much room for improvement and many open questions as to how best to approach this task. Some of these areas that we will discuss briefly are: When should the coach provide help, should it intervene in student work? How best can the coach traverse the expert knowledge base in order to support the student? What information can be used from the student argument, and how can one balance contradicting information?

The subject of when to offer help is widely debated in the tutoring field. On one hand, students who are engaged should not be interrupted unnecessarily because it can be disruptive to their thought process and seen as an annoyance. On the other hand, research has indicated that students are not incredibly successful at identifying when they need help (Aleven & Koedinger, 2000). This indicates that a system should monitor student behavior

in order to intervene at key moments to support learning. Thus far, the Rashi system has only offered help on demand, but we are considering methods of monitoring the student to immediately notice and react to both statements of misconception (such as creating an incorrect relationship) as well as bad inquiry behavior (such as collecting large quantities of random data without a noticeable reason for doing so).

The order in which to traverse the knowledge base when coaching students is a difficult question. Good inquiry requires the consideration of many hypotheses, yet the coach should not distract the student by changing subject while they are engaged. So the real question at hand is how to balance a breadth-first vs. a depth-first approach to traversing the expert knowledge base. Along the same lines, one must decide whether the coach should help the student work down from hypotheses to data, or help them work from data collected up towards hypotheses (top-down vs. bottom-up approach). Research has indicated that a mixture of top-down and bottom-up is most conducive to learning (Krajcik et al., 1998). One possibility we are considering for these questions

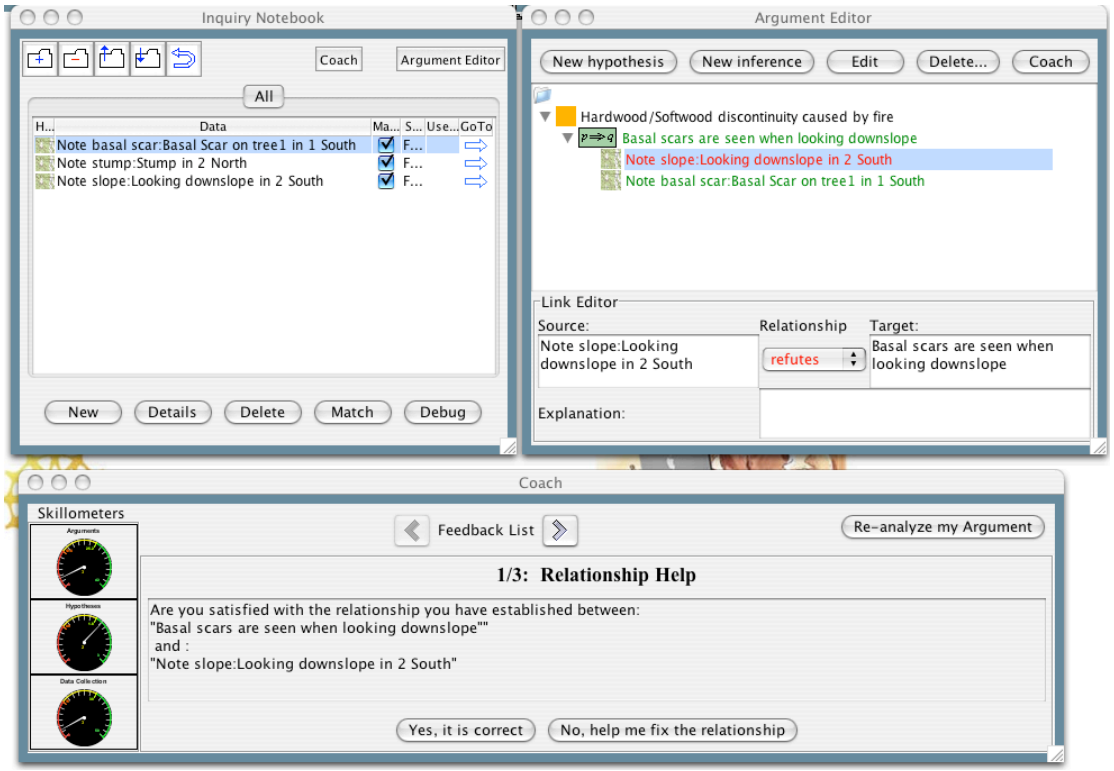

Figure 2. The student receives help from the Forestry Tutor.
Student observations, data and hypotheses are automatically recorded and analyzed in the Inquiry Notebook (top left). This student posits hypotheses in the Argument Editor (top right) and uses a drag and drop feature to move data from the Inquiry Notebook to link supporting or refuting evidence to hypotheses. Here the coach has identified a relationship which the student has set up incorrectly, and is encouraging the student to reconsider.

involves weighting which step is more crucial (e.g. if a hypothesis is almost completely supported, then it should be less important to investigate than a hypothesis that has no support). Another involves tracking the recency with which a student has operated upon relevant material (e.g. if they have recently collected some data, help them work from that data towards a hypothesis).

Finally, we consider the different methods of analyzing student work in these ill-defined inquiry domains, and how to deal with contradictions posed by different analyses. Earlier versions of Rashi used a syntactic coach that analyzed not the content of a student argument, but the structure. For example, if the coach found that a student had more support for an argument that was ruled out than for an argument that was considered the student's top hypothesis, it would ask the student to consider the argument again. This idea has obvious flaws, because one piece of supporting evidence can outweigh any number of other pieces of evidence depending on the meaning of that evidence. For this reason, we moved to an expert knowledge base approach. Yet the expert knowledge base approach requires that every domain have one of these expert knowledge bases, which is time-consuming and difficult to define. So there are obvious advantages and disadvantages to both approaches. We are developing a hybrid approach which attempts to utilize the best of both methods, but this does give the coach competing ideas about how to give advice, and it is not always clear which advice is most useful. The best approach so far seems to be using the expert knowledge base advice when available, and the syntactic advice when the other is missing.

So we can see that Rashi does offer some compelling methods of guidance through ill-defined domains, providing students with both domain knowledge and direction in the inquiry process itself. As we discover better solutions to the problems discussed, the coaching will become more useful and productive. Until then, there is another, equally interesting approach to helping students in ill-defined domains: collaboration.

PROPOSED COLLABORATION TOOLS

There are many difficulties with instructing a student in ill-defined domains. We presented some of these issues and proposed solutions to them, but many other issues remain. So far in our discussion, we have focused on creating an environment that supports students and intelligent help to guide students through their work. This guided help approach makes a major assumption, that the system can understand student work. Since we are still far from having a solid natural language recognition system, and we allow students to enter knowledge from outside the tutor, as well as allowing them to phrase their arguments and hypotheses in their own terms, there will always be information provided by the student that the system does not understand. This is one of many clear limitations of an artificial intelligence approach to teaching in these domains.

On the other hand, perhaps new pedagogy can help us understand student work. For example, collaborative learning is one of the most valuable educational approaches in terms of improved learning outcome (Johnson & Johnson, 2005). Often groups will outperform even the best individuals in the group, producing knowledge that none of its members would have produced by themselves and leading to the generation of new ideas (Ellis et al., 1994). Collaboration encourages students to question processes, ask for advice and monitor each other's reasoning (Slavin, 1990). It is effective especially for students who tend to be left behind in science classrooms, including women and under-represented populations (Ellis et al., 1994). Nearly 600 experimental studies and over 100 corollary studies clearly indicate that cooperation results in higher achievement and greater productivity, more caring, supportive and committed relationships, and greater psychological health, social competence, and self-esteem (Johnson & Johnson, 1989). Other research points to collaboration resulting in a boost of efficiency and accuracy, and problem solving ability (Okada & Simon, 1997).

We intend to develop collaboration tools to support synchronous, symmetric cooperation through the web. A chat system will support students to discuss cases and propose actions as if they were seated in front of the same computer. The tutor will keep track of student actions and discussions and will analyze computer logs to investigate, for example, how collaboration influences learning styles.

We intend to investigate a number of research questions. For example, do group collaborative activities improve individual domain and/or inquiry learning? Does efficient (in the sense of group success) and effective (in the sense of positive group dynamics) collaboration reflect improved individual learning as measured in post-tests? Does collaborative learning result in more effective learning or more positive confidence among women and minorities as compared with individual learning?

Not all collaborative groups make progress in similar settings: it is therefore important to understand groups and to help decide which groups are productive and should remain together, and which groups are not making progress (Tedesco & Rosatelli, 2004). We will develop both a performance model that documents how cognitive events influence completion of a task and a model of the knowledge sharing contributions of the students in the group. Research involves comparing the performance of individuals in different contexts: when students work alone and when working in groups. For this reason, Collaborative Rashi, (C-Rashi), will track all actions performed by students and will align chat exchanges with the logs of the actions taken on the Rashi system, so we can reconstruct the interaction sequence (Soller et al., 2004).

The interface shown in Figure 3 divides the student window into four areas. Area 1 shows the original Human Biology Tutor (which in turn can be divided into frames, as decided by the Rashi server). This is what users see when connected to the original, individual (not collaborative) tutor. Areas 2 and 4 are used for communication among participants; area 2 is student input to the chat while area 4 contains a log of the chat. Area 3 is used to manage the control of the interaction with C-Rashi, and the engine behind it can implement different policies. Verbal exchanges are classified into different phases, Propose, Discuss, Review, incorporating fifteen sentence openers that help analyze effective peer dialogue (Soller & Lesgold, 2003). In addition, three 'quick' buttons ('OK', 'Yes', and 'No') are provided (Area 2) to indicate agreement. Each sentence opener represents a different cognitive process related to the problem solving phase and are simple enough for students to find and select. Students start their conversation using the openers (which also saves them some typing) and then complete the phrase in their own text. The tutor will manage the conversation through the use of topic threads (based on context) that attempt to structure the discussion to reflect the structure of the team's decision processes. In this way, C-Rashi will support learners through the various phases of problem solving, facilitating an extended, in-depth, on-topic discussion and providing a coherent view of the argument. The synchronous, symmetric environment will use a dedicated applet to manage 'mouse control' by introducing the concept of token. A student's clicks on Area 1 in Figure 4 are effective only when she has the token. To prevent the user from clicking on links when she does not have the mouse, a JavaScript function will check the ownership of the token.

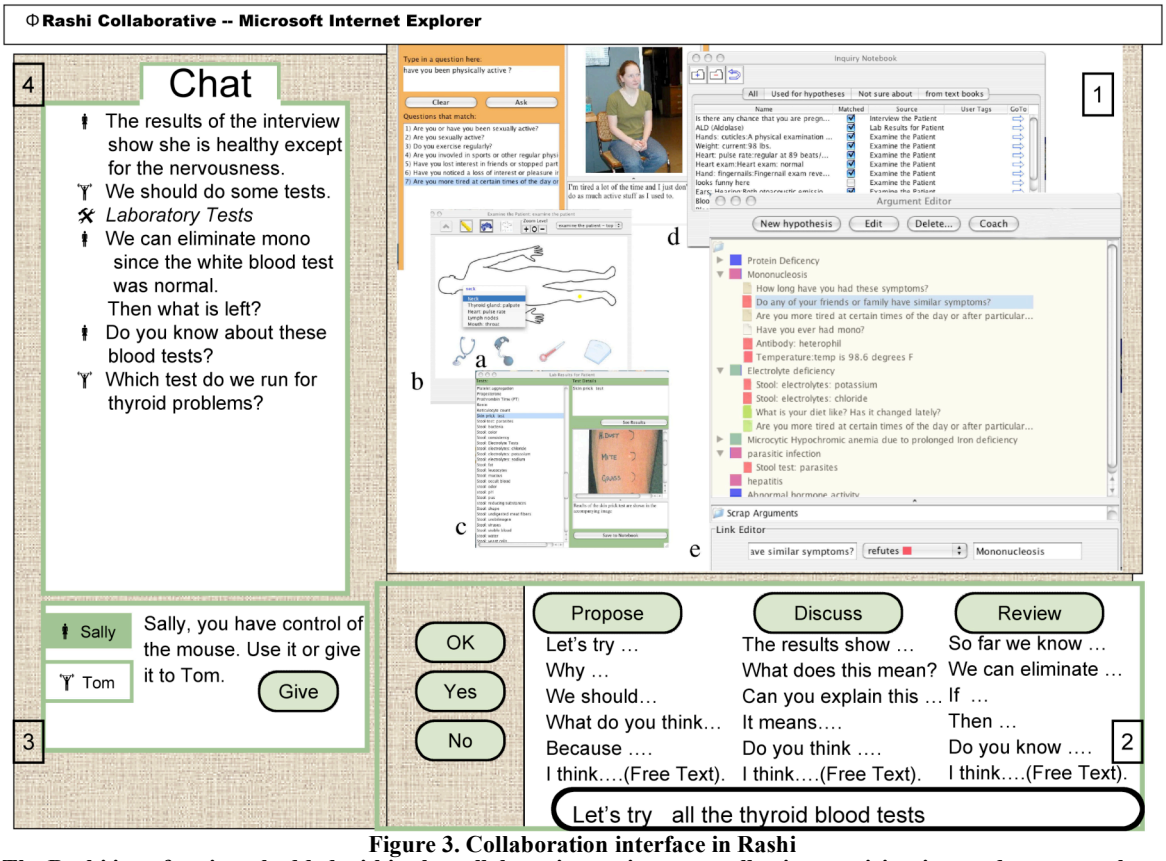

Figure 3. Collaboration interface in Rashi
The Rashi interface is embedded within the collaborative environment allowing participating students to work on an inquiry case (1), communicate with teammates (2), use the shared mouse (3) and review the conversation (4).

Collaboration Theory and New Software Functionality

We base our collaboration environment in part on Karl Smith's widespread and successful 'cooperative learning' model (Smith et al., 2005). This model emphasizes structured or formal collaborative work, and has many elements that are ideally suited to computer-supported learning: positive interdependence (of task, identity, resources, role, reward, and goals), individual and group accountability (challenging and motivating work for every student in a group) and authentic interaction (brainstorming, planning, social and team building skills, and solidarity). Collaborative Rashi will be modified in the following ways to reflect this theory:

- *Forming the team:* We will experiment with various ways to form student teams. Placing students in groups and telling them to work together does not always result in cooperation. We will have teams predefined by the teacher and groups dynamically formed 'on the fly'. We will also experiment with teams selected based on using student models. After collaboration, student polling will ask each participant whether each team member was helpful or not, and this information will be used to improve the student model approach.
- *Monitoring tools and instructor control panel:* The teacher will monitor general trends in student work at individual, class, and community levels, such as the number of hypotheses logged, or whether certain information was recorded. Monitoring tools facilitate accountability within groups and within classes making visible the general level of work a student has done without revealing the solution in progress.
- *Roles (duties) support*: Assigning roles, including task manager, skeptic, accuracy checker, social facilitator and record keeper, has been an essential part of many successful case-based learning methods. Students should have the experience of each different role in a semester. The system will send reminders (for students to discharge their duties, triggered by intelligent rules) and will indicate which tools should be used to accomplish that role (e.g., the skeptic could use the critique feature mentioned above). All students are expected to work collaboratively to solve the problem; roles are extra duties or 'hats' that they put on periodically.

PRIOR EVALUATION OF RASHI

The proposed changes to the Rashi system listed in this report constitute a large investment of time and energy. Yet the project has met with much success, and there is good reason to believe these changes will be fruitful.

The Rashi cases have all been evaluated at Hampshire College and the Universities of Massachusetts and Rhode Island with undergraduates as well as middle school science teachers. The Biology Tutor was evaluated several times in large (500 students) university lecture-based classroom. However, as there was only time to use a few short cases, we consider this evaluation to be a pilot study to test the evaluation instruments. Nevertheless, the results were encouraging: students quickly learned the software and posed open-ended and authentic questions, planned queries and engaged in on-line research. We have also noted significant correlations between a student's inquiry skill level and some of the Rashi use metrics. In particular, there were significant positive correlations between a student's measured inquiry skill level and the number of *hypotheses* posed by that student, the number of *arguments*, the number of *items in the notebook*, the number of *explanations* entered by students, the use of notebook organizing tools and the overall use of Rashi tools. As this is what one would expect, this adds some credence to the validity of the pre-post test instrument. We interpret these results as supporting the usability of the software and its perceived usefulness. Interviews, surveys, essay questions, group discussions, and pre-post essay activities have shown that participants were enthusiastic and impressed with the potential of Rashi as an educational tool. Interactivity was seen as a very positive attribute, with the *Patient Examination* feature in Biology cited as one of the better components. Students' perception of learning the inquiry process was favorable.

Students found the Rashi cognitive tools helpful to organize data and create good arguments. Students were highly positive toward the software, especially toward the tools that allowed them to gather data. They were challenged and very engaged (as anecdotal evidence, one said that if the software were available, she would ask her parents to buy it for her as a Christmas present). At first, students treated the system like a "video game." They were drawn to the *Image Explorer* and other highly visual elements. In conversation with each other, students generated interesting hypotheses and entered data into the Inquiry Notebook, which helped them conduct their analyses and helped faculty analyze their approaches. We observed students going to their texts often and posing multiple hypotheses. Students discussed their reactions with an evaluator and completed surveys. Their overall reactions were very positive, despite certain individual issues. Students did not have difficulty navigating through the tools after the initial explanation and thought it would be fairly easy for a naive user to catch on. Most aspects of the system were rated positively by roughly half of the users.

RELATED WORK AND CONCLUSIONS

Related work in inquiry and argumentation tutors has led to case-based learning environments and tools for gathering, organizing, visualizing, and analyzing information during inquiry (Aleven & Ashley, 1997; Suthers et al., 1997). Some systems support authentic inquiry and knowledge sharing, and several track and analyze student data selections, providing students with space to explore subject matter from microeconomics to medical diagnosis. However, most of these systems do not evaluate a student's hypotheses and have more of a data collecting than an experimentation feel; others are narrowly applicable and do not allow for interactive tutoring. Most are restricted to a single domain since limiting the domain allows the designer to facilitate specific types of interactions (Aleven & Ashley, 1997). Other systems are built with the primary goal of teaching the inquiry process and not the domain.

Rashi maintains a student model that compares student arguments and collected data with the knowledge in expert system, and a domain model that provides a structure with which to model the student. The tutor i) provides a free exploratory space while tracking student arguments, ii) provides intelligent contextual advice and critiques a student's evidence; iii) and remains domain-independent and flexible enough to quickly encode new cases and new domains. In this way, the system provides means for teaching a broad range of ill-defined domains, including science and liberal arts. The current system provides a full set of tools that allow students to engage in inquiry learning with complex, real-world cases. These tools provide structure and organization while still allowing for freedom of exploration and a learner-centered approach. The coaching system guides the student in both domain knowledge and the inquiry process to help those who are lost or confused.

We are currently developing many improvements to the system. Coach improvements include addressing the issues of when to offer help, how to best traverse the knowledge base, and how to handle contradictory goals when analyzing the student argument in different ways. We plan to make the system a collaborative environment that supports both group work and peer and instructor review.

ACKNOWLEDGEMENTS

Research on Rashi was funded in part by 1) the U.S. Department of Education, "Expanding a General Model of Inquiry Learning", Fund for the Improvement of Post Secondary Education, Comprehensive Program, #P116B010483, B. Woolf, P.I., by 2) the National Science Foundation under grant DUE-0127183, "Inquiry Tools for Case-based Courses in Human Biology," M. Bruno, PI and Woolf, Co-PI, by 3) National Science Foundation, CCLI #0340864, "On-line Inquiry Learning in Geology," D. Murray, P.I., B. Woolf co-PI, 4) and by 4) National Science Foundation DUE-0341521 "Reading the Forest Floor: Online Inquiry Learning in Forestry." B. Woolf, P.I., and L. Winship, Co-PI, and 5) National Science Foundation DUE-0341197, "Engaging Undergraduates in On-line Inquiry Learning: A Case-based Cyber Library in Human Biology," M. Bruno PI.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

Albanese, M. A., & Mitchell, S. (2003). Problem-based learning: a review of literature on its outcomes and implementation issues. *Acad Med, 68*(1), 52-81.

Aleven, V., & Ashley, K. D. (1997). Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment. In B. d. Boulay & R. Mizoguchi (Eds.), *Artificial Intelligence in Education, Proceedings of AI-ED 97 World Conference* (pp. 87-94). Amsterdam: IOS Press.

Aleven, V., & Koedinger, K. R. (2000). *Limitations of student control: Do students know when they need help?* . Paper presented at the 5th International Conference on Intelligent Tutoring Systems.

Alloway, G., Bos, N., Hamel, K., Hammerman, T., Klann, E., Krajcik, J., et al. (1996). Creating an inquiry-learning environment using the World Wide Web, *International Conference of Learning Sciences*.

Arroyo, I., Beal, C. R., Murray, T., Walles, R., & Woolf, B. P. (2004). Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. In *Intelligent Tutoring Systems, 7th International Conference* (pp. 468-477). Macei0, Alagoas, Brazil: Springer.

Aspy, D. N., Aspy, C. B., & Quimby, P. M. (1993). What doctors can teach teachers about problem-based learning. *Educational Leadership, 50*(7), 22-24.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (1999). *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Academy Press.

deVries, E. (2002). Computer-mediated epistemic dialogue: Explanation and argumentatin as vehicles for understanding scientific notions. *Journal of the Learning Sciences, 11*(1), 63-103.

Dragon, T., Woolf, B. P., Marshall, D., & Murray, T. (2006). Coaching within a domain independent inquiry environment, *8th International Conference on Intelligent Tutoring Systems*. Jhongli, Taiwan.

Ellis, S., Klahr, D., & Sieglar, R., (1994, April). The birth, life, and sometimes death of good ideas in collaborative problem solving. Paper presented at the annual meeting of the American Educational Research Association, New Orleans.

Eylon, B.-S., & Linn, M. C. (1988). Learning and instruction: An examination of four research perspectives in science education. *Review of Educational Research, 58*(3), 251-301.

Hakkarainen, K. (2003). Progressive Inquiry in a Computer-Supported Biology Class. *Journal of Research in Science Teaching, 40*(10), 1072-1088.

Johnson, D., and Johnson R., (2005) The Cooperative Learning Center, The University of Minnesota, http://www.co-operation.org/index.html. April 2005.

Koschmann, T. D., Myers, A. C., Feltovich, P. J., & Barrows, H. S. (1994). Using technology to assist in realizing effective learning and instruction: A principled approach to the use of computers in collaborative learning. *Journal of the Learning Sciences, 3*(3), 225-262.

Krajcik, J., Blumenfeld, P. C., Marx, R. W., Bass, K. M., Fredricks, J., & Soloway, E. (1998). Inquiry in project-based science classrooms: initial attempts by middle school students. *The Journal of the Learning Sciences, 7*(3&4), 313-350.

Mennin, S. P., Friedmann, M., Skipper, B., Kalishman, S., & Snyder, J. (1993). Performances on the NBME I, II, and III by medical students in the problem-based learning and conventional tracks at the University of New Mexico. *Academic Medicine, 68*(8), 616-624.

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

Pea, R. (1994). Seeing what we build together: Distributed multimedia learning environments for transformative communications. *Journal of the Learning Sciences, 3*(3), 285-299.

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

Shute, V. J., & Glaser, R. (1990). A Large-scale Evaluation of an Intelligent Discovery World: Smithtown. *Interactive Learning Environments, 1*, 51-57.

Slavin, R. E. (1990). Research on cooperative learning: Consensus and controversy. *Educational Leadership 46*, 12: 52-54.

Smith, K.A., Sheppard, S.D., Johnson, D.W., and Johnson, R.T. (2005). "Pedagogies of Engagement: Classroom-based Practices," Journal of Engineering Education, Vol. 94, No. 1, 2005, pp. 87–101.

Soller, A. (2004). Computational modeling and analysis of knowledge sharing in collaborative distance learning. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 14 (4), 351-381.

Soller, A. and Lesgold, A. (2000). 'Knowledge acquisition for adaptive collaborative learning environments' American Association for Artificial Intelligence Fall Symposium, Cape Cod, MA, USA

Suebnukarn, S., & Haddawy, P. (2004). A collaborative intelligent tutoring system for medical problem-based learning, *International Conference on Intelligent User Interfaces*.

Suthers, D. D., Toth, E., & Weiner, A. (1997). An Integrated Approach to Implementing Collaborative Inquiry in the Classroom, *Computer Supported Collaborative Learning (CSCL'97)* (pp. 272-279). Toronto.

Tedesco, P.A., & Rosatelli, M.C. (2004). Helping Groups Become Teams: Techniques for Acquiring and Maintaining Group Models. J. Mostow, & Tedesco, P. (Eds.). *Designing Computational Models of Collaborative Learning Interaction*. ITS 2004 Workshop, Maceio-Alagoas, Brazil.

Vernon, D. T., & Blake, R. L. (1993). Does problem-based learning work?: A meta-analysis of evaluative research. *Academic Medicine, 68*(7), 550-563.

White, B., & Frederiksen, J. R. (1995). *Developing Metacognitive Knowledge and Processes: The key to making scientific inquiry and modeling accessible to all students* (No. CM-95-04). Berkeley, CA: School of Education, University of California, Berkeley.

Woolf, B. P., Marshall, D., Mattingly, M., Lewis, J., Wright, S., Jellison, M & Murray, T. (2003). Tracking student propositions in an inquiry system. In U. Hoppe, F. Berdeho & J. Kay, (Eds.) Artificial Intelligence in Education, Proceedings of AIED 2003, IOS Press, pp. 21-28.

Woolf, B. P., Reid, J., Stillings, N., Bruno, M., Murray, D., Reese, P., Peterfreund, A. & Rath, K. (2002) A General Platform for Inquiry Learning, Proceedings of the 6th Int'l Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science 2363, 681-697, France.

Woolf, B., Murray, T., Marshall, D., Dragon, T., Kohler, K., Mattingly, M., Bruno, M., Murray, D. & Sammons, J. (2005). Critical Thinking Environments for Science Education, International Conference on AI and Education, 2005, Amsterdam.

Providing Support for Creative Group Brainstorming: Taxonomy and Technologies*

Hao-Chuan Wang[1], Carolyn P. Rosé[1], Tsai-Yen Li[2], Chun-Yen Chang[3]

[1]Carnegie Mellon University
Pittsburgh PA, USA
{haochuan, cprose}
@cs.cmu.edu

[2]National Chengchi University
Taipei, Taiwan
li@nccu.edu.tw

[3]National Taiwan Normal University
Taipei, Taiwan
changcy@ntnu.edu.tw

Abstract. This paper describes our plans and current work towards developing a user modeling-based learning environment for creative group brainstorming in qualitative domains. This research framework aims to guide the investigation and integration of different brainstorming activities, theoretical foundations and supportive technologies in order to benefit learning most. Intervention studies and system development are proposed to take place in parallel to ensure that the design of system behavior is informed by research results.

Keywords: Group Brainstorming, Creative Problem Solving, Collaborative Learning

INTRODUCTION

Research in the area of Intelligent Tutoring Systems (ITS) has achieved impressive results in improving student learning especially in well defined problem solving domains, such as basic algebra (Koedinger et al., 1997) and quantitative physics (VanLehn et al., 2005), just to name a couple. In this paper we begin to consider how to expand the frontiers of this success into areas less touched by ITS research. The research objectives of this work are: (1) to provide more open ended qualitative scientific problem solving opportunities for students, also known as creative problem solving (CPS), and (2) supporting productive small group dynamics for collaborative idea generation, joint problem solving, and learning, respectively. Recent work begins to approach this area, such as exploratory problem solving (Kumar et al., 2006), natural language-based tutorial dialogue systems (Grasser et al., 2005; Kumar et al., 2006), and adaptive support for collaborative learning (Gweon et al., 2006). In this paper, we present a tutoring system that brainstorms with students, called VIBRANT (Virtual Brainstorming), which we propose as an instructional tool for science education.

Supporting CPS poses different challenges than the types of problem solving domains more frequently studied in the ITS community. Consider the following sample CPS question: *"What are the possible factors that might cause a debris-flow hazard to happen?"*, and subsequently, *"How could we prevent it from happening?"* One salient characteristic of this task is the unlikelihood of modeling the process of answering these questions in a procedural and goal-directed manner as in more traditional ITSs. Notice that the goal for students here is not to select and then apply a known procedure for solving this problem. In contrast, the main purpose here is to let students actively generate the candidate problem solving steps/options by themselves. Also, because problems such as these do not have a single right answer, another important feature of this type of task could be the necessity of recruiting human judges in evaluating the goodness of students' problem solving behaviors. In fact, beyond offering students the opportunity to generate possible solutions to problems, these tasks offer students the opportunity to weigh and balance trade-offs between alternative solutions. In our previous work, students were required to answer CPS questions independently without accessing external resources or peer support. Human graders were recruited to score student answers quantitatively using a rubric devised by domain experts (Chang & Weng 2002). However, this operationalization of CPS has previously been used primarily in assessing students' problem solving abilities, and the focus was not specifically on how to scaffold students' idea-generation and creative problem solving, and/or to investigate the embedded instructional opportunities with an additional factor of the presence of peers for collaboration. In contrast, the purpose of VIBRANT is to support students in the process of CPS, with the goal of improving their problem solving skills.

* A short version introducing the technologies discussed in this workshop paper is included as a poster in the main conference of ITS 2006 (Wang et al., 2006).

Thus, our current work focuses on data-driven design and implementation of system behavior for supporting individual/collaborative brainstorming activities. Three types of brainstorming activities are considered, including (1) brainstorming for idea generation, (2) brainstorming for creative problem-solving, and (3) brainstorming for inquiry learning, which are all related. VIBRANT offers user modeling-based supportive technologies for activities including *cognitively oriented support providing brainstorming feedback* and *socially oriented support for discussion group formation*. The combination of *activities*, *the degrees of socialization (e.g., group vs. solitary brainstorming)*, and *technologies* results in a very rich design space for researchers to investigate. Our research goal is to explore how to combine findings from the social psychology community on how to maximize productivity in brainstorming with findings from the learning sciences community on how to maximize the learning benefit from exploration. Rather than arguing that we have the answer, instead we present VIBRANT as a framework in which researchers may begin to address the many research questions that stand in between our current understanding and the ultimate solution.

In the remainder of the paper, we describe our theoretical foundation, drawn from both communities of research. We then discuss our hypotheses and the research questions we must address in order to support brainstorming activities, primarily creative problem solving and active inquiry, for learning. We then describe VIBRANT, a technological infrastructure for supporting this work. We evaluate VIBRANT's ability to offer coherent feedback to students in response to their brainstorming behavior in a CPS task. We conclude by discussing how we will use the VIBRANT system to address our research questions.

THEORETICAL FRAMEWORK

In this section, we identify three types of brainstorming activities connected with learning, which together form a theoretical foundation for guiding the design of VIBRANT and our planned future empirical studies. Table 1 (see Appendix) offers an overview of the types of empirical investigations we either draw from or plan to conduct. We build on previously published theories, interventions, key concepts such as the distinction between performance and learning outcomes, and methodologies for operationalization and measurement.

A conceptual continuum identifying *performance-oriented* and *learning-oriented* activities is delineated in Table 1. The activity of idea-generation, as frequently characterized in the literature of social psychology and organizational studies, focuses more on *performance* rather than *learning*. By contrast, the activity of inquiry learning, which emphasizes students' active knowledge acquisition/construction, is clearly at the other side of the performance-learning continuum. In our view, CPS combines concerns from both ends of this spectrum. On the one hand, when working on CPS tasks, it is the quality/quantity of solutions (i.e., performance) that the organization or individual tends to optimize, and which is a more obvious measure of success. On the other hand, in learning environments, a more open-ended creative problem solving task may encourage problem solvers to explore the domain, to establish links among learned concepts actively, to interact with peers with argumentative discourses, and to detect what knowledge components are to be further acquired. The act of idea generation may lead a student to a new area of the exploration space that offers a valuable opportunity for learning. Thus, one may argue that it is not the exploration per se that is important, but the opportunities for learning that may result from that exploration.

Brainstorming for Idea Generation

The cognitive task of idea generation in brainstorming groups has been extensively studied in social psychology through controlled experiments (Diehl & Stroebe, 1987). In the study of group brainstorming for idea-generation, empirical work has repeatedly revealed phenomena related to process losses, in which a group with mutually-interacting members may not always perform better than a collection of non-interacting individuals whose contributions are pooled statistically (i.e., nominal group) (Hill, 1982; Diehl & Storebe, 1987). Theoretical explanations for process losses have been proposed and tested empirically, including social pressure (e.g., evaluation apprehension), social loafing (e.g., "free riding"), and production blocking (Connolly, 1993; Diehl & Storebe, 1987 & Kraut, 2003). Early electronic brainstorming systems (EBS) designed to avoid process losses resulting from these causes have shown great promise (Connolly, 1993). Thus, we believe it is within the realm of what state-of-the-art technology can support to address concerns in the realm of learning in CPS tasks.

It is considered that idea generation activities can be used as the foundation for stimulating students' active thinking for performing other higher level activities in the taxonomy, including creative problem solving and inquiry learning, although it appears that only doing idea-generation may *not* help students acquire domain knowledge effectively. A series of interventions is proposed in Table 1 in order to test several social psychological hypotheses (intervention ig1-ig4), and to further verify the relation between students' idea generation and domain learning. For example, in a CPS task, whether more and better ideas generated would lead to better problem solving and learning.

Brainstorming for Creative Problem Solving

Based on the literature of creativity research and science education, the Creative Problem Solving (CPS) model generally describes problem solving as a process consisting of two qualitatively different phases: divergent and convergent thinking (Basadur, 1995; Chang & Weng, 2002; Osborn, 1963). The literature proposes that a problem solving process can be decomposed into several stages, typically including fact-finding, problem-solving, idea-finding, and solution-finding. The stages of fact-finding and idea-finding are more divergent thinking oriented, while the stages of problem-solving and solution-finding are considered more convergent thinking oriented. In each stage, some work proposed that two sub-phases of ideation and evaluation can be further identified (Basadur, 1995).

We forsee opportunities for students to *learn*, hypothetically, partly subject matter and partly meta-cognitive skills, during CPS tasks. Obviously, idea-generation appears to play an evident role in the part of divergent thinking in CPS tasks, which may help stimulate students' active thinking and engagement with other memebers within the group who may help them see the problem from a different perspective. Furthermore, in convergent thinking sub-phases, students are required to evaluate, explain and negotiate ideas they have generated, which appears to offer valuable *knowledge events* similar to the activity of self-explanation that can be employed in shaping effective tutoring (Aleven & Koedinger, 2002). From a different angle, by tracing and coding students' argumentative process in CPS, the analytic framework of argumentative knowledge construction may be applied towards providing process-oriented instructional support (Gweon et al., 2006; Weingerger & Fischer, 2006). In a broader sense, CPS-based tutoring appears to be situated at the intersection of ITS, Computer Supported Cooperative Work (CSCW) and Computer Supported Collaborative Learning (CSCL), or correspondingly, cognitive psychology, social psychology and educational psychology. While engaging in cooperative work in the context of CPS, it is reasonable to expect that valuable opportunities for tutoring and collaborative learning are interwoven within the processes of work, in which workers may need to recall or even acquire skills not used before, and peer workers' opinions may provoke cognitive conflict that may lead to active learning and conceptual changes.

Brainstorming for Inquiry Learning

Inquiry as an approach to learning typically consists of processes of exploring the targeted problems or phenomena, asking questions, and making discoveries, achieving new understanding and fulfilling personal curiosity (NSF, 2002). The underlying educational ideology states that students will be able to develop durable and transferable science process skills and to continue pursuing lifelong learning in a self-directed means based on the meta-cognitive skills of inquiry and discovery that they derived from inquiry learning.

However, as recent review articles and empirical studies have argued, the approach of *pure* discovery learning that strictly prohibits intervention from teachers did not show any evidence of performing better than other approaches, e.g., direct instruction, in terms of subject matter learning, transfer to new situations, and the acquisition of basic scientific skills (Anderson et al., 1998; Klahr & Nigam, 2004; Mayer, 2004). Nevertheless, this is not to say just filling the curriculum with all lecture-style instructions will be fine. Mayer (2004) proposed that guidance, structure, and focused objectives should be incorporated into activities of inquiry and discovery. Instead of making an oversimplified dichotomous choice between "to discover" or "not to discover", what is more important should be an attempt to balance the two extremes of teachers' instructions and students' constructions with the goal of actually promoting cognitive activities (Rosé et al., 2005). The questions researchers are confronted with now are: how much guidance, in what form, and under what condition can serve as scaffolding and make inquiry learning an effective approach (de Jong, 2005).

Virtually every inquiry activity begins with "asking questions", and then students may be motivated to move on to "finding answers", and subsequently, "asking *better* questions" that incrementally leads students to really learn from inquiry and discovery. Idea-generation and CPS activities are potentially useful to be cast as inquiry activities. Instead of asking students to conceive ideas or solutions that can be used in problem solving, if designed properly, we may ask students to conceive some questions at a meta-level against a CPS task, which may lead to better observation of the task structure for performing better idea-generation at a later round. The "question-generation" activity appears to introduce new opportunities and challenges to the design of system behavior and instruction, but should be a worthy goal for our project to pursue in the long term.

RESEARCH QUESTIONS AND HYPOTHESES

Group versus Individual

One set of research questions we must address relate to what aspects of creative problem solving are best addressed with individual learners and which with groups. We expect that the ultimate answer will involve a

balance of these two. Dillenbourg (1999) indicated that a variety of collaborative activities may contribute to learning, which could be course assignment sharing, joint problem solving, or even performing cooperative work. A driving force behind learning in groups could be that group members will typically bring unique resources, perspectives and backgrounds into collaborative activities (Kraut, 2003) However, as argued above, the phenomena of process losses cast doubts on the oversimplified assertion that group must be better than individual in terms of work performance, but also revealed the needs of sophisticated analyses on the micro features and interaction dynamics of a group.

Performance versus Learning

In the taxonomy presented in Table 1, the distinction between performance-oriented and learning-oriented brainstorming activities has been made. However, this is not to say that the activity of idea-generation has nothing to do with learning. As emphasized previously, we consider idea-generation as the basis for other higher-order brainstorming activities that would lead to valuable instructional and learning opportunities. An issue of interest to educational practitioners and organizations is about how do we enable the transfer of what students learn to improve their work performance in solving new problems? It appears that brainstorming activities and the taxonomy may fit well to the investigation of relations among learning process, transfer, and work performance.

The distinction between work performance and learning may be confounded with the comparison of group versus individual performance. It should be clear that the evaluation criteria or desirable results for work performance and learning are rather different. For example, for the debris-flow problem solving task introduced in the Introduction, if the purpose is to evaluate the work performance, the number of good ideas generated or viability of the solutions, either individually or collaboratively, for preventing the damage of debris-flows may serve as the criteria. However, if the purpose is to evaluate how well do *individual* students learn or acquire knowledge, measures addressing individuals' knowledge status or meta-cognitive skills are required. It is unlikely to be valid in measuring learning at the group level. For the measure of learning, beyond traditional educational achievement testing, it is considered that the technology of user profiling may serve the later purpose well. It is argued that different measures should be constituted and applied in research questions regarding performance and learning respectively.

Learning and Transfer

Through the lens of cognitive science, it is essential to ask questions of what do we expect students to learn and transfer by doing these brainstorming activities, either within a group or in a solitary manner, either supported by other agents (human or computer) or not. As prior work in this area have posited, one of the central issues in education is about *transfer*, which refers to how knowledge acquired in one situation or task can be applied in another, perhaps novel and unfamiliar, situation (Anderson, 1993, Anderson et al., 1995).

Two targets of transfer are possible for the research to pursue. *First*, more conventionally, it is intended to enable the learning and transfer of domain knowledge through brainstorming activities. For example, for the debris-flow question, although students can be taught declaratively about the physics, geology, and ecology related to the phenomena of debris flows, an open ended CPS task would give students the opportunity to actively summarize and interconnect what they have acquired via natural language, detect what they have not mastered, and seek instruction accordingly. What is different from what is typical of traditional ITSs is the absence of a prescribed ideal solution path as well as the presence of peer collaboration. Therefore we do allow more exploration and variability of students' behaviors. Empirical studies in comparing CPS-tutoring and typical ITS approach as planned in Table 1 may verify the value of exploration and collaboration in terms of domain learning. *Second*, perhaps debatably, we may go on to investigate the possibility of improving students' meta-cognitive skills such as information foraging, self-explanation, or more generally, creative problem solving ability. In other words, the question is, after students practicing one CPS task, can we expect students to perform better or equally well in another previously unfamiliar CPS task? Will students become more capable or strategic in interacting with peers and seeking for information? Our current focus is not on this aspect, but it could be interesting, also challenging, to touch these questions in the future.

System Tuning and Optimization

In order to optimize VIBRANT's instructional effects, we may incorporate research evidences resulting from intervention studies as proposed in Table 1. Thus, our plan is to use VIBRANT as a research platform in which to address our research questions and then to fold our findings back into the design of new versions of VIBRANT in order to offer students higher quality educational opportunities.

For example, in the brainstorming activity of idea-generation, we are interested in knowing what would be better, either the system instantiates a single agent or a group of agents to interact with students, where agents

may be computer agents, human participants, or some combination of the two. We may go on to ask if a group of agents should be better, whether the group members should be homogeneous or heterogeneous in opinions, and also, whether the brainstorming feedbacks provided by agents should sound evaluative (i.e., more critical) or supportive.

An experimental paradigmused in our prior work (Gweon et al., 2006), which we may adopt is to employ "confederate peer agents", or humans behaving in a highly prescriptive manner, in order to investigate effects of group dynamics on individuals. We believe that the confederate peer agent experimental paradigm provides an appropriate level of experimental control while allowing us to evaluate the impact of agent capabilities beyond the current state-of-the-art for handling open-ended inputs. However, for some studies we may take a different approach. For example, when the intention is to study the effects of these interventions in groups with larger size, say 50, it may be more practical to program VIBRANT properly to enable such an experiment. It is foreseen that VIBRANT will play dual roles in our future work, at the one hand, it is a tutoring system that can be used in educational activities, while at the other hand, it serves as a research tool to enable simulations of group dynamics for shaping future computer supported collaborative learning.

SUPPORTIVE TECHNOLOGIES

Based on the taxonomy of brainstorming activities, at an abstract level, VIBRANT provides two types of support for each specific activity, which are (1) cognitively oriented support providing brainstorming feedback and (2) socially oriented support for discussion group formation. As an abbreviation, the former is called as the *intelligent support*, and the later is called the *social support*. Certainly, the actual system behavior for each specific brainstorming activity will need further adaptation and tuning, such as a dialogue script capable of differentiating evaluative tone from supportive tone in providing intelligent support, and a matching function for recommending either the most similar peer or the most different peer in providing social support. The design decisions of system behavior can be made in an evidence-based manner by making references to interventions studies proposed in our research framework. VIBRANT, as a web-based system, is designed to be versatile and flexible to mimic particular behaviors suggested by our studies easily. Figure 1 depicts the architecture of VIBRANT, which mainly consists of three functional modules, including the brainstorming agent, the user modeling (UM) agent, and the user interface (UI) at the client side. In this section, we describe the system characteristics at an abstract level that aims to generalize across various brainstorming activities.

User Modeling

In order to provide adaptive support in response to students' exploration of the task when performing idea-generation, CPS, and active inquiry, a learning environment would require corresponding user-modeling technologies integrated for tracing students' knowledge status and performing instructional decisions accordingly.

Based on our prior work (Wang et al., 2005a, 2005b), knowledge of solving a CPS task is modeled as a bipartite graph-based formal user profile (fUP), in which the connections between a student's ideation (i.e., section of divergent thinking) and explanation (i.e., section of convergent thinking) are explicitly represented. For a particular CPS task, student U's ideation in solving the problem is represented as a set of ideas $A_U=\{a_{U1}, a_{U2}, \ldots, a_{Un}\}$, and explanation is represented as a set of reasons $B_U=\{b_{U1}, b_{U2}, \ldots, b_{Um}\}$. A fUP is denoted as an undirected bipartite graph $G_U=(V_U, E_U)$ where $V_U=A_U \cup B_U$ and $A_U \cap B_U=\phi$. The mapping between A_U and B_U, modeled as $E_U=\{e_{ij}\}$ representing a linkage between an idea a_i and a reason b_j, is of a many-to-many nature, in which one idea may have several reasons, and several ideas may connect to an identical reason. If the focus of study is only on the activity of idea-generation, then a simple fUP that contains only A_U would suffice.

A prescriptive Domain fUP as shown in Figure 1 is created by the domain expert by using appropriate authoring tools. The domain fUP is employed by the system in building user profiles. A collection of historic fUPs is managed by the system and can be retrieved later for fulfilling specific decision making as well as offline collaboration among peers.

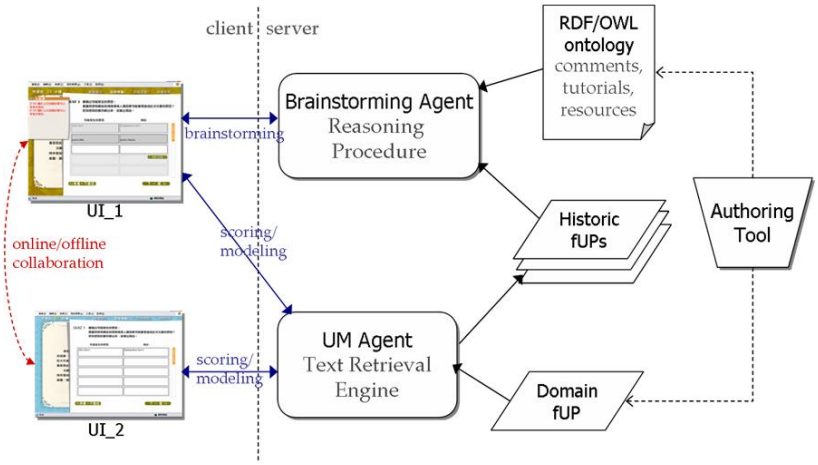

Figure 1. System Architecture for VIBRANT

Intelligent and Social Support

A formal ontology is constructed serving as the core device for organizing experts' CPS ideas and learning resources, including feedback texts and learning materials. The ontology consists of an `is-a` hierarchy organizing experts' ideas into several levels of abstraction. Prescriptive feedback messages are attached to specific idea nodes at lower levels and categorical nodes at higher levels in the ontology.

Finite state machines (FSMs) are designed to retrieve learning resources such as feedback. In FSMs, the finite set of *states*, Q, represents the range of the system's functional behavior, including actions such as `check_coverage` or `move_upward_in_hierarchy`, while the finite *alphabet*, Σ, represents the set of events that can trigger a transition from one state to another, such as `all_sub_nodes_covered` which in this case triggers a transition to a state called `get_new_cateogry`. Transition functions $\delta : Q \times \Sigma \rightarrow Q$ represent designers' instructional decisions of what behavior should be triggered when particular events are observed, which are particularly useful to adapt system behaviors to accommodate specific brainstorming activities.

The *feedback* prepared by the system consists of two parts, a *comment* and a *tutorial*. Two separate FSMs are designed for the generation of each part. *Comments* are evaluative texts responding directly to the most recent idea submitted by the student to the system, while the *tutorial* is the instruction that directs the student to the next logical focus node, which may either be an idea node or a categorical node, selected by the system adaptively based on its model of the student. In the current design, a *comment* is a function of the current idea entry, while a *tutorial* is a function of the current idea entry and the student's fUP built incrementally during the brainstorming process. In other words, a context of students' previous responses in the same session is incorporated into the mechanism for retrieving *tutorials*.

The use of the `is-a` hierarchy is considered beneficial for the FSM-based feedback generation. *First*, the hierarchy of topics provides a basis for supporting a more organized and coherent brainstorming process. The system may select a next focus for tutorial to maximize the students' local coverage of categorical nodes that have been partially addressed by the students' idea entries. The system may then proceed to provide feedback that may lead students to cover all related categorical nodes at a higher level. *Second*, *comments* can be fetched strategically at a more generalized level in the ontological hierarchy when a particular idea proposed by the student is semantically ambiguous and thus results in low similarity scores as computed using vector-based information retrieval methods. The strategy may help remedy the insufficiency of IR-based methods for computing semantic similarity and to improve the relevance of system-prepared comments against students' ideas. Along with a student's producing more ideas, her/his profile, fUP, also evolves incrementally. A later instructional decision, including feedback generation or social recommendation, will further make use of these fUPs.

Given a collection of historic fUPs done by previously visited students, we may re-model the system of fUPs as a tripartite graph with hyperedges. The condition of student p having an idea q and explaining it by reason r can be represented as a hyperedge e_{pqr} , which results in a tripartite graph $H=(V, E)$ where $V=S \cup A \cup B$ and $E=\{e_{pqr}\}$. A variety of analyses can then be computed over the tripartite graph for social structure discovery. *First*, one may conceive local heuristics to extract particular (hyper-)edges as cues for social recommendation. *Second*, we may apply Social Network Analysis (SNA) methods (Wasserman & Faust, 1994) such as co-occurrence analysis against the tripartite graph for clustering users and then forming discussion groups

according to the structural and global information. With this tripartite graph, the same techniques can also be used to recommend a next brainstorming focus of an idea in a data-driven manner.

PRELIMINARY EVALUATION

We conducted an evaluation of VIBRANT feedback generation using a corpus containing 163 entries of self-generated ideas from 25 Taiwanese high school students on the debris-flow question as inputs for simulation. For each input entry, two comment/tutorial generation methods, *with-hierarchy* (H) or *without-hierarchy* (NH), were invoked to generate two versions of simulated comments and tutorials. The H method generates feedback using the aforementioned approach, while the NH method does not make use of a category-based brainstorming plan, so that the brainstorming focus motivating the tutorial is selected randomly from the pool of non-covered concept nodes in the domain model. In the NH method, the comment offered is the comment attached to the most similar node in the domain model, and *no* remedial device was used against potential low relevance of comments retrieved at the instance-level. Note that two types of message were evaluated separately, comment or tutorial texts, and therefore, results of two parts of message generation, comment generation (CG) and tutorial generation (TG), are reported.

We recruited two independent judges to rate the quality of the comment and tutorial offered by the two different methods. The two judges were asked to evaluate the quality of each comment/tutorial text against the idea entry it targets. Each coder assigned a binary score of *acceptance* (Acceptable/Not_Acceptable) to the comment/tutorial message generated by the two methods against the same idea entry. The judges then assigned a nominal score of *subjective preference* (H: With-hierarchy/ NH: Without-hierarchy/ S: Same/ N: Neither) to indicate which version among the pair of messages they preferred or considered better against this idea entry. The coders were blind to the method used in generating the messages. Although the two versions of message were presented in pair for rating, the order of which version presented in the left side and which one in the right was randomized. After data cleaning which excluded pairs containing empty messages generated by either method, totally 138 pairs of comments (85% comment pairs) and 153 pairs of tutorials (94% tutorial pairs) were used in data analyses.

The dependent variable of *acceptance* indicating the proportions of messages accepted for the H and NH methods was first analyzed. Independent-samples chi-square tests were conducted to examine the difference between H and NH. For the CG part, no statistically significant differences in comparing H vs. NH were found in either coder's ratings. For the TG part, tutorials generated by both methods also appear to be equally well with no significance, no matter evaluated by which coder. We did find significantly different distribution on the variable of *subjective preference* for TG over CG. Statistically significance were detected by using chi-square analyses, specifically for coder 1's preference votes: $\chi^2(3, N=291) = 26.621$, $p < .001$, and for coder 2's: $\chi^2(3, N=291) = 34.843$, $p < .001$. In a post-hoc inspection of the data, we found that H was preferred more in the TG condition rather than in the CG condition.

In summary, we found that in the TG part, the with-hierarchy method was rated higher than the without-hierarchy method, which is very different from the trend revealed in CG. The results may imply that the with-hierarchy method can help produce better tutorial texts, but not comments, as supports in response to students' ideas generated. This makes senses that the selection of brainstorming foci in TG for guiding students' exploration would benefit from the context (i.e., the ontological hierarchy and students' fUPs), while local information is sufficient for CG. The preliminary evaluation shows evidences of usefulness of our current design on the core component in VIBRANT. Note that what was evaluated here was the quality of feedback but not its instructional effects against real students. Evaluations on instructional effects are planned in the near future. In addition, in this evaluation, the inter-rater reliability between the two coders was found to be low. We also aim to refine the operationalization of "acceptance" and "preference" to enhance the inter-rater reliability in replicating a similar evaluation design.

CURRENT DIRECTIONS

In this paper, we describe our taxonomy and technologies for creative group brainstorming. Three kinds of brainstorming activities identified in the context of science education are idea-generation, creative problem solving, and inquiry learning. The user modeling-based learning environment, VIBRANT, is proposed to incorporate feedback generation and social recommendation technologies to support various brainstorming activities by properly authoring the system behavior. Interventions and corresponding measures are proposed to better understand group dynamics in brainstorming groups in various tasks, which may inform the design of future intelligent tutoring systems for ill-defined domains.

REFERENCES

Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive tutor. *Cognitive Science*, Vol. 26, pp. 147-179.

Anderson, J. R. (1993). *Rules of the Mind*. NJ: Lawrence Erlbaum Associates.

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

Anderson, J. R., Reder, L. M. & Simon, H. (1998). Radical constructivism and cognitive psychology. In D. Ravitch (Ed.) *Brookings papers on education policy 1998*. Washington, DC: Brookings Institute Press.

Basadur, M. (1995). Optimal Ideation-Evaluation Ratios. *Creativity Research Journal*, Vol. 8, No. 1, pp.63-75.

Chang, C-Y., Weng, Y-H. (2002). An Exploratory Study on Students' Problem-Solving Ability in Earth Sciences. *International Journal of Science Education*, 24(5), pp. 441-451.

Connolly, Terry. (1993). Behavioral decision theory and group support systems. (pp.270-280). In L. Jessup & J. Valacich (Eds.). *Group support systems*. NY: Macmillan.

Diehl, M., & Storebe, W. (1987). Productivity loss in brainstorming groups: toward the solution of a riddle. *Journal of Personality and Social Psychology*, 53(3), 497-509.

Dillenbourg P. (1999). What do you mean by collaborative learning?. In P. Dillenbourg (Ed) *Collaborative-learning: Cognitive and Computational Approaches* (pp.1-19). Oxford: Elsevier

de Jong, Ton. (2005). Scaffolding inquiry learning: How much intelligence is needed and by whom? Invited speech in 12th International Conference on Artificial Intelligence in Education*, Frontiers in Artificial Intelligence and Applications*, 125, 4-4.

Gweon, G., Rose, C. P., Carey, R., & Zaiss, Z. S. (2006). Providing support for adaptive scripting in an on-line collaborative learning environment. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2006)*.

Hill, G. W. (1982). Group versus individual performance: are *N*+1 heads better than one? *Psychological Bulletin*, 91(3), 517-539.

Klahr, D., & Nigam, M. (2004). The equivalence of learning paths in early science instruction: effects of direct instruction and discovery learning, *Psychological Science*, 15(10), 661-667.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1), 30-43.

Kraut, R. E. (2003). Applying Social Psychological Theory to the Problems of Group Work. In J. Carroll (Ed.), *HCI Models, Theories, and Frameworks* (pp. 325-356). NY: Morgan-Kaufmann Publishers.

Kumar, R., Rose, C., Aleven, V., Iglesias, A., & Robinson, A. (2006) Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context. *Proceedings of International Conference on Intelligent Tutoring Systems (ITS 2006)*.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? the case for guided methods of instruction, *American Psychologist*, 59(1), 14-19.

National Science Foundation (2002). Inquiry: thoughts, views and strategies for the K-5 classroom, *Foundation*, Vol. 2.

Osborn, A. (1963) *Applied Imagination: Principles and Procedures of Creative Problem Solving*. New York: Charles Scribner's Sons.

Rose, C. P., Aleven, V., Carey, R., & Robinson, A. (2005). A first evaluation of the instructional value of negotiable problem solving goals on the exploratory learning continuum. *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, 563-570.

Wang, H-C., Chang, C-Y., & Li, T-Y. (2005a). Automated scoring for creative problem-solving ability with ideation-explanation modeling. *Proceedings of 13th International Conference on Computers in Education (ICCE 2005)*, 522-529.

Wang, H.-C., Li, T-Y., & Chang, C-Y. (2005b). A user modeling framework for exploring creative problem-solving ability. *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, 941-943.

Wang, H-C., Li, T-Y., Rose, C. P., Huang, C-C., & Chang, C-Y. (2006). VIBRANT: A Virtual Brainstorming Agent for Computer Supported Creative Problem Solving. *Proceedings of 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*.

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. NY: Cambridge University Press.

Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer supported collaborative learning. *Computers & Education*, 46, 71-95.

VanLehn, K., Lynch. C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: five years of evaluation. *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, 678-685.

APPENDIX

Table 1. A research framework for guiding future works

	Idea generation	*Creative problem solving*	*Inquiry learning*
Theory	social psychology	creativity research, science education	science education, learning sciences
Continuum (Evaluation)	*Performance (ideas/solutions produced)* ⟵		⟶ *Learning (knowledge acquired)*
What to be learned?	*N/A*	*Subject matters*	*Meta-cognitive skills*
Intervention	ig1. group sizes (including single vs. group)* ig2. homogeneity of group* members ig3. anonymity ig4. types of feedback (e.g., evaluative vs. supportive feedback)	ig1-ig4 cps1. different problem solving tasks for the same topic (e.g., less vs. more open ended problem solving tasks) cps2. orientation of tutorial dialogue (e.g., divergent thinking-oriented prompts vs. convergent thinking-oriented prompts)	ig1-ig4 cps1 iqr1. degree of guidance available (e.g., unstructured vs. structured information sources) iqr2. amount of error information
Measurement	♦ number/quality of generated ideas ♦ member attitude (e.g., satisfaction)	♦ number/quality of generated ideas ♦ completeness/novelty of CPS entries ♦ likelihood of having the problem solved ♦ domain achievement test ♦ member attitude	♦ number/originality of proposed questions ♦ (retentive) domain achievement test ♦ member attitude

* group members can be either real students or simulated agents, so a group can be a mix-up of human peers and simulated agents

Teaching Case Analysis through Framing: Prospects for an ITS in an ill-defined domain

Ilya M. Goldin
Intelligent Systems Program
Learning Research & Development Center
University of Pittsburgh
goldin@pitt.edu

Kevin D. Ashley
Intelligent Systems Program
Learning Research & Development Center
University of Pittsburgh
ashley@pitt.edu

Rosa L. Pinkus
Neurosurgery/Medicine
School of Medicine
University of Pittsburgh
pinkus@pitt.edu

Abstract. Intelligent Tutoring Systems research has made assumptions that may be violated in ill-defined tasks. We describe ethics case analysis, an important educational yet ill-defined task in the domain of bioengineering ethics. We discuss an ITS approach for this task that involves structuring the learning experience and using AI to help guide student peer reviewers.

Keywords: intelligent tutoring system, ill-defined domain, bioengineering ethics, case analysis, framing

INTRODUCTION

A basic assumption in the design of Intelligent Tutoring Systems is that an ITS needs to interpret student output. This is necessary so that the ITS can provide appropriate feedback, or to assess knowledge for student modeling, or to measure performance. In domains such as geometry or algebra, it is often possible to design the problems that students solve so that the result is numeric or at least symbolic and constrained in a way that an ITS can interpret. In other words, these problems are well-defined.

An accepted approach to teaching bioengineering ethics is to use textual cases to illustrate important domain concepts and methods of moral reasoning. The role that cases play in the classroom is to permit students to immerse themselves—to situate their problem solving—in the complexity and variety of authentic ethical dilemmas. This methodology, a kind of problem-based learning (PBL), is distinct from and often complements alternative teaching practices, which may emphasize philosophical moral theories. The PBL setting requires that students learn to analyze cases. For example, the case analysis method taught in the popular textbook (Harris et al., 2000) asks students to consider the morally relevant facts of the case, both known and unknown; to structure the analysis via the conceptual issues that can relate the facts to each other; to use their moral imagination to propose and compare alternative resolutions to the dilemma; and finally to justify a particular resolution.

The fact that students may produce a wide range of acceptable responses marks the case analysis task as ill-defined. An ethical dilemma may have no good resolutions at all, or it may have multiple ones. In fact, usually only "paradigm" cases will have clear-cut definitive answers. Realistic problems will be more complex and more equivocal as one considers alternative frameworks for resolving the problem and justifying the resolutions. Not only do ethical problems rarely have definitive answers, the answers depend in part on how the problem is conceived or "framed", as well as on the justifications. (Pinkus et al., in preparation) A large part of problem solving in this ill-defined domain involves constructing (i.e., framing) a representation of the problem, which may include additional constraints and possible actions. As a result, a given problem as posed can become a different problem depending on the constraints and conditions that a solver adds in order to better define the problem or to suggest alternative actions. In addition, for many ethics problems, the issue is not only identifying relevant moral principles to justify an answer, but also mapping the concepts in the principles to the situation at hand, which may not be clear-cut and may require a good deal of subjective interpretation.

One consequence of the ill-defined nature of ethics case analysis is that the most appropriate representation for student case analyses is free-form text. Only natural language enables describing the problem scenarios in sufficient detail and for considering the implications of particular details on alternative resolutions. Another consequence is that even if an ITS could understand the text, the possibility of a wide range of acceptable answers makes providing feedback, modeling student knowledge, or measuring performance correspondingly difficult for a human tutor, much less a machine.

We have studied how one educator addresses the ill-defined nature of ethics case analysis by encouraging students to frame the cases they analyze and by gauging their skills with a manually administered specially-

designed Assessment Instrument. We have demonstrated the Instrument's validity and reliability in assessing some important moral reasoning skills. (Goldin et al., 2006; Goldin et al., in preparation) Our objective here is to propose an ITS that helps to teach analysis of textual ethics cases. Our design adapts a computer-supported collaborative learning program (SWoRD or Scaffolded Writing and Rewriting in the Discipline) to support peer review of the case analyses and leverages a database of student-authored analyses manually annotated with the Assessment Instrument. First, we describe the pedagogical technique for dealing with this ill-defined task and the Instrument, and summarize our evaluation of the Instrument's validity and reliability. We then outline our proposed system, suggest how it will incorporate AI techniques in helping students learn to analyze ethics cases, and how to evaluate it. To conclude, we review related work, and consider links to other ill-defined domains.

ETHICS CASE ANALYSIS THROUGH FRAMING: ASSESSMENT CHALLENGES

Educators have developed some ingenious pedagogical strategies using PBL to deal with the ill-defined nature of ethics case analysis. Author Pinkus, a professional ethicist, assigns students a capstone exercise in a required Bioengineering Ethics class: the task is to write a one or two page case study based on a student's area of engineering expertise. In these cases, the protagonist, an engineer, is faced with a dilemma, which is caused by, or will have ramifications for, the engineer's professional duties. The course of action is unclear, and requires analysis. Each student presents the case to the class for comment and then writes a paper that analyzes the case using the methods taught in the class. This approach, where students create cases close to their professional expertise and interests, has been shown to be a positive factor in student learning. (Pinkus et al., in preparation)

From the viewpoint of intelligent tutoring systems, this approach poses challenges. An ITS needs not only to be able to model a student's solution where many alternative solutions are acceptable, but also a case that the student has designed herself. Indeed, even if we simplify the task by asking students to analyze an assigned case (and forego the pedagogical benefits of student-authoring of cases), students trained to apply the Harris method will do so differently because of how they *frame* the case.

Framing Dilemmas through Labeling, Defining, Applying Concepts

Framing is one strategy for dealing with ill-defined domains. As noted, ethics problems, except in paradigm cases, rarely have a definitive answer or even a definitive description. Students must add constraints to the facts of the case and thus articulate what the ethical dilemma is. This in turn, affects how a moral principle or a professional ethics code applies to the problem. Mapping the principles to the situation and weighing the effects of alternative actions or using one's moral imagination to create compromises in order to resolve conflicts are among the skills that students must learn. The methods of moral reasoning described in the Harris text and elsewhere provide a conceptual framework for viewing the cases. The framework can be derived from moral theories, principles, the moral imagination, or from tacit rules embedded in engineering practice. Once a case is framed, the moral dilemmas that characterize it can be articulated. Then, professional knowledge and key concepts can serve to "filter" morally relevant facts and alternative resolutions can be proposed.

Thus, one approach to ill-defined aspects of ethics case analysis requires the problem-solver to define the problem better through framing. The latitude one has in framing a case, however, is not intuitive to an engineering student. Typically in engineering, one is given a problem already framed and asked to solve it, such as in textbook standardized ethics cases. Given the importance of framing in ethics problem-solving, asking students to create their own cases is an ideal pedagogical exercise; it requires them to frame the problems.

Although the facts of a case are the most obvious properties of a dilemma, their relative importance becomes apparent only after they are framed within the conceptual issues implicit in the case. Thus, even if an analysis ought to begin with identification of facts, this is insufficient for purposes of student modeling or assessment. Consider, for instance, the conceptual issue of informed consent. A typical dilemma that involves this issue is whether a person has been properly informed about the risks inherent in a medical procedure, and has granted consent to be exposed to these risks, probably with the hope of deriving some benefit. For example, a participant in evaluating a new medical treatment must be informed and grant consent before being subjected to the treatment. To analyze a case in terms of informed consent, it is important to recognize at least two protagonists: one with expert knowledge and skills; the other in need of the expert's service, but whose permission is a prerequisite for accepting those services. The expert will have to inform the person in need, especially about risks and benefits of accepting the service and of any alternatives that could be used. The person accepting the service can only grant consent if this "informing" is non-coercive, and if he demonstrates understanding.

The example of the informed consent frame comes from an awareness of professional responsibilities, which is an outcome of "role morality," i.e., the obligations inherent in one's role as a professional, as well as from the related concepts of autonomy and respect for persons. One may also frame a concept from personal or common morality. Personal morality means that one's personal values can be a legitimate factor in how one views a case, even if these values are clearly not shared by others. In the informed consent case, the physician may personally

not agree with a patient's decision to forego life-sustaining treatment for her pancreatic cancer yet legal considerations and medical ethics guide the physician to respect the patient's informed decision. Common morality means that concepts like honesty and respect for persons are universally shared, and can guide framing. Experienced ethical reasoners are "careful to identify issues and to specify conditions under which specific professional role obligations recommend particular actions, [to elaborate] conditions which would affect the moral analysis of a problem, in part through posing hypothetical variations of the problem, and [to justify] resolutions in terms of those conditions which they conclude apply in the problem." (Keefer & Ashley, 2001)

Thus, one way a student can frame a case is to claim that particular concepts, such as informed consent, constitute the frame through which the case ought to be viewed. The next step is to define the issue in a general, abstract way, like informed consent in the above paragraph; this shows an understanding of the properties of the issue removed from the details of a given case. Finally, the student must explain how the definition maps to the case at hand. We call these three steps *labeling* the concept as such; *defining* it; and *applying* it.

"Sometimes apparent moral disagreement turns out to rest on conceptual differences where no one's motives are in question. These are issues about the general definitions, or meanings, of concepts." (Harris et al., 2000, p. 46) Definitions are particularly important when open-ended terms are in play, such as "acceptable risk" posed by an engineer's creation, or "the public" whose safety the engineer ought to hold paramount. Such open-ended language figures prominently in abstract ethics principles, and even in the more detailed codes of ethics. Yet "attempts to specify the meanings of terms ahead of time can never anticipate all of the cases to which they do and do not apply. No matter how precisely one attempts to define a concept, it will always remain open-ended; that is, it will always remain insufficiently specified, so that some of its applications to particular circumstances will remain problematic." (Harris et al., 2000, p. 50) This requires a problem-solver to apply the concept to the specifics of the case, i.e., to say whether a given fact situation constitutes an occurrence of a concept. When a problem-solver labels, defines and applies an open-ended, i.e., ill-defined, term, she frames the conceptual issue.

Consider this excerpt from a good case analysis (Figure 1) showing defined or applied concepts. (Coders annotate on a computer, and the terms that are concept labels can be automatically identified; thus, we omit the labels.) Using her definition of the concept of "responsibility of a bioengineer" as a hub, the author applies the concepts of confidentiality, autonomy, and safety to consider whether to disclose the pilot's condition.

> **Example:** <*concept-applied="safety"*> Jeff - the bioengineering student - is concerned for the safety of the pilot and of his future passengers since he plans to continue flying despite the doctor's advice to stop. </*concept-applied*> <*concept-applied="responsibility-of-bioengineer"*> Jeff wonders whether he is responsible for telling the airline of Joe's condition since the neurosurgeon and his advisor will not. </*concept-applied*> <*concept-applied="confidentiality"*> Would Jeff be breaching the confidentiality constraints set by the IRB form if he informed the airline? </*concept-applied*> Is there a solution to this issue that would serve the best interests of all parties? ….
>
> <*concept-defined="responsibility-of-bioengineer"*> Researchers have various obligations and prerogatives associated with their profession, and these responsibilities can be referred to as their role morality. </*concept-defined*> [1] For example, <*concept-applied="responsibility-of-bioengineer"*> researchers have responsibilities to their experimental subjects mandating that the subjects' safety be of utmost importance. Furthermore, <*concept-defined="autonomy"*> researchers should respect their subjects' autonomy, allowing them to decide if they want to participate and allowing them to discontinue the experiment at any point. </*concept-defined="autonomy"*> <*concept-applied="confidentiality"*> Furthermore, a subject's identity should be kept as confidential as possible. </*concept-applied="confidentiality"*></*concept-applied="responsibility-of-bioengineer"*> Except under unusual circumstances, <*concept-defined="confidentiality"*> the only people who should have access to the subject's records are the investigators and staff who run the experiments. </*concept-defined*> According to the Bioethics Advisory Commission (of August 2001), "Protecting the rights and welfare of those who volunteer to participate in research is a fundamental tenet of ethical research". [2] <*concept-applied="responsibility-of-bioengineer"*> When deciding whether or not to inform the airline of Joe's condition, Jeff needs to be cognizant of the responsibilities he has toward his subjects, particularly his responsibility to respect their confidentiality. However, he also has to consider <*concept-applied="safety"*> his responsibility to protect Joe's safety, which may be in danger if he continues to fly despite his medical condition. </*concept-applied="safety"*> In addition to researchers' obligations to their subjects, they also have obligations to society. </*concept-applied="responsibility-of-bioengineer"*>

Figure 1: An excerpt from a case analysis, annotated for defined or applied concepts.

Assessment Instrument for Labeling, Defining, and Applying

In this way, labeling, defining, and applying (LDA) serve as an operationalization of framing of concepts, similar to how the NSPE Board of Ethical Review fleshes out abstract ethics code provisions when they analyze exemplar cases for posterity. (Ashley & McLaren, 2001) The LDA operationalization is embedded in an Assessment Instrument for bioengineering ethics case analysis. (Pinkus et al., in preparation) The Instrument is a set of questions that invites coders to assess whether students acquire Higher-Level Moral Reasoning Skills (HLMRS). LDA operationalize one of these skills with questions about labeling, defining and applying of over 40 concepts. The list of concepts has been derived from the Harris text and from student essays, and includes consideration of moral theories, principles, codes of ethics, and common, personal, and role moralities. (Other

HLMRS recognize other ways to frame a case, e.g., from professional knowledge rather than from conceptual issues, but that requires a different operationalization of framing; for example, see (Martin et al., 2005).)

We evaluated the Assessment Instrument's validity by measuring whether it reflects student learning. We also conducted a study to evaluate how reliably the instrument can be applied by comparing how well independent human coders agreed on their coding assignments. (Goldin et al., in preparation) The sensitivity study showed that the LDA operationalization is valid, because it is sensitive to student learning gains during a semester-long class. We compared student skills at analysis of short assigned ethics cases, as measured by the Assessment Instrument, at pre- and posttest times (n=13, Figure 2). Students labeled, defined, or applied very few concepts at pretest, and significantly more at posttest. At pretest, students do not label, define or apply (code "none") 94.5% of the concepts, and they never do all three (code "LDA"). At post-test, they invoke significantly more concepts, including when they comprehensively label, define and apply the concept.

Coder Pair	Label	Define	Apply	Other 4 HLMRS
Trained vs. trained (N=12)	0.893	0.892	0.868	0.324
Naïve vs. trained (N=29)	0.626	0.472	0.577	0.158

Table 1: Agreement between coders (Cohen's Kappa) on Assessment Instrument annotations

The reliability study showed that the Assessment Instrument can be applied reliably by trained independent coders, and fairly reliably even by untrained coders. Three pairs of coders annotated 41 student-authored term papers using the Assessment Instrument. Agreement for the LDA (Table 1) was much higher than that for the more abstract HLMRS, reflecting the value of operationalizing one of the HLMRS. Note that for measuring IRR, one usually trains coders on a range of possible answers to some particular question. In our study, each term paper contained a new, student-authored case, meaning that we could never train our coders on the particular "question" they would annotate. Consequently, the high level of agreement on LDA is especially noteworthy. The promising results of evaluating the Instrument lead us to design the system described below.

PROPOSED SYSTEM ARCHITECTURE

Given the challenges posed by the ill-defined nature of case analysis in bioengineering ethics to the task of designing an ITS, it is clear that traditional ITS technology is insufficient. The system needs to accommodate the student-authored cases and analyses. While Natural Language Understanding is an active area of research, we are a long way from computer comprehension of student essays. That means that humans need to do the bulk of understanding the texts. At the same time, we hold out the hope that gradual advances in NLU technology will some day permit the system to bear a greater load. Thus, our goal is to produce a system that

a) organizes the process of case creation, analysis, and gathering feedback so that it may
b) enhance this process to the extent that technology allows today, and
c) collect data on this process that can be used to improve the state of the art.

Case analysis is a writing task. Traditional classroom writing instruction suffers from two basic flaws: the teacher can be overburdened by the obligation to provide feedback as class size grows, and even in small classes students may lack the opportunity to respond to feedback by submitting multiple essay drafts. One way to address this is to ask student peer reviewers to provide feedback to each other. This fits with the requirement that humans need to do the bulk of understanding the texts. By staging peer review online with the help of a system like SWoRD (Cho & Schunn, 2005), we tackle goals (a) and (c). Ideally, peer feedback helps drive home the significance of having framed the case in one way rather than another, and the act of peer reviewing constitutes a learning opportunity in itself. Furthermore, peer review assures the students that their classroom work is not an abstract exercise, but has real consequences—lessons professional ethics courses seek to teach. We intend to compare the learning effects of SWoRD-aided peer review versus traditional classroom practice.

Finally, we aim to enhance the peer review process by asking how the system could encourage students to

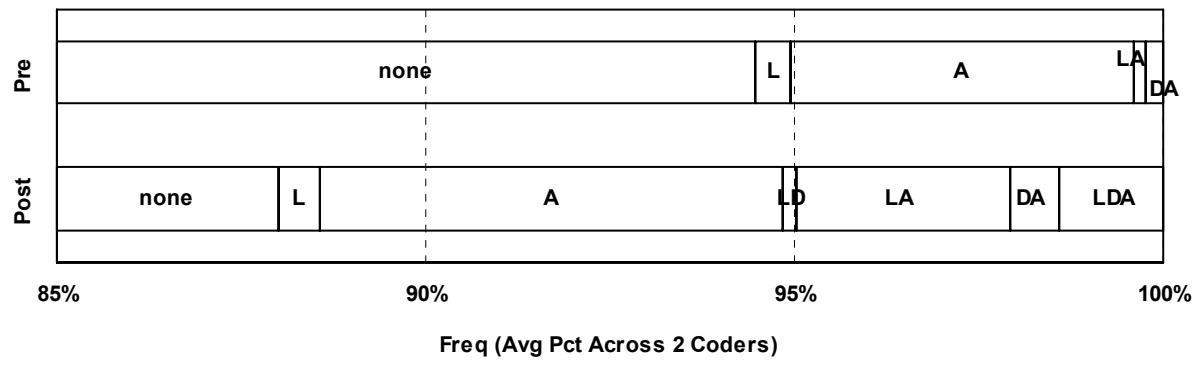

Figure 2: Change in LDA scores from pre- to posttest; LA = Labeled and Applied, not Defined, etc.

4. Evidence of moral reasoning skills

This dimension is about the evidence of moral reasoning skills in the author's case analysis. Did the author appear to: (1) employ professional engineering knowledge to frame the issues, (2) view the problem from multiple levels (e.g., that of the engineer, the employer, the client, the patient, the public, regulators, etc.), (3) flexibly move among the multiple levels in his/her analysis, (4) identify analogous cases and explain the analogies, and (5) employ a method of moral reasoning in conducting the analysis? In connection with (5), did the author identify moral reasoning concepts relevant to analyzing the case? Did the author label, define, and apply these concepts?

Your Comments: Provide specific comments about the paper's evidence of moral reasoning skills. If the author *employed relevant professional knowledge to frame the issues,* or *viewed the problem from multiple levels* or *moved flexibly among those levels*, or *identified relevant analogous cases* and *adequately explained the analogies*, or *used a moral reasoning method in conducting the analysis,* point that out and congratulate them! If the author did not do so, try to suggest potential fixes to these problems**.** In looking for evidence of a method of moral reasoning, look to see if the author identified relevant moral reasoning concepts in analyzing the case, or labeled, defined, and applied such concepts. Suggest relevant concepts and how to label, define, and apply them in the case. (The **GLOSSARY** defines and provides examples of many concepts from this course.)

<Peer reviewer writes free-form comments here>

Your Rating: Based on your comments above, how would you rate the evidence of moral reasoning skills in the author's case analysis?

☐	7. Excellent	The paper shows strong evidence of all five moral reasoning skills. The author labels, defines, and applies relevant concepts in his/her case analysis.
☐	6. Very good	The paper shows strong evidence of all but one of the first four moral reasoning skills. Regarding the fifth, the author labels, defines, and applies most of the relevant concepts in his/her case analysis.
☐	5. Good	The paper shows some evidence of two of the first four moral reasoning skills. Regarding the fifth, the author labels, defines, and applies some of the relevant concepts in his/her case analysis.
☐	4. Average	The paper shows some evidence of only one of the first four moral reasoning skills. Regarding the fifth, the author either labels or applies some of the relevant concepts in his/her case analysis but does not always label, define and apply each concept.
☐	3. Poor	The paper shows almost no evidence of any of the first four moral reasoning skills. Regarding the fifth, the author alludes to some of the relevant concepts in his/her case analysis but does not always label, define, and apply each concept.
☐	2. Very poor	The paper shows no evidence of any of the first four moral reasoning skills. Regarding the fifth, the author alludes to some of the relevant concepts in his/her case analysis but does not label, define, and apply each concept.
☐	1. Disastrous	The paper shows no evidence of any of the first four moral reasoning skills. Regarding the fifth, the author does not label, define, or apply any relevant concepts in his/her case analysis.

Figure 3: Moral reasoning skills criterion for peer reviewers

frame cases. We examine two strategies: first, adapt SWoRD to the domain of case analysis; second, provide reviewers domain-specific feedback that builds on existing case analyses already annotated for framing.

Adapting SWoRD to Peer Reviewing of Ethics Case Analyses

SWoRD is a web-based instructional system that supports reciprocal student authoring and student peer reviewing. Its aim so far has been to improve writing by focusing reviewers on prose flow, logical argument, and insight. Of course, these are aspects of writing that an ethics case analysis should also include. Prose flow concerns how well the author identifies the main points and transitions from one point to the next. Logical argument is "the extent to which each paper is logically coherent in terms of text structure that organizes various facts and arguments," and "how well the main arguments are supported with evidence." Insight involves "the extent to which each paper contributes new knowledge and insight to the reader. In classes, this is operationally defined as new knowledge or insight beyond required class texts and materials." (Cho & Schunn, 2005).

Our goal for SWoRD is to help students learn not only writing skills, but domain reasoning skills: how to analyze ethics cases by framing. We hope that if we reach the peer reviewers, not only will they learn, but they will in turn reach the student authors. SWoRD has been deployed and evaluated in many classrooms, including domains as varied as psychology and physics, but with the focus on writing quality described above. It has not been applied to improve domain reasoning skills, nor in the context of engineering ethics.

We will adapt SWoRD to engineering ethics by defining a criterion that focuses student peer reviewers on the higher-level moral reasoning skills, and, in particular, on the skill "using a method of moral reasoning," operationalized in terms of whether or not authors label, define, and apply ethical concepts. In essence, the criterion teaches the reviewers to apply the Assessment Instrument, a valid and reliable method of assessment. The goal is to teach the reviewers to critique the case analyses in terms of the HLMRS, and relevant ethical concepts in particular. This, in turn, will encourage student authors to do the same. A new page of the form that

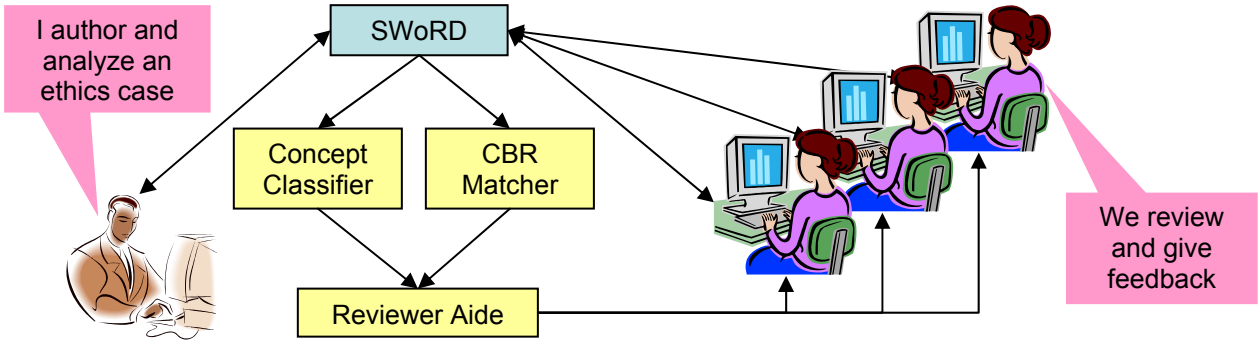

Figure 4: System architecture: SWoRD plus Reviewer Aide with Concept Classifier and CBR Matcher

reviewers fill out (Figure 3) comprises a detailed description of the new criterion, followed by instructions, a space for comments, and a seven point scale along which the reviewer rates the paper. In addition to focusing the reviewers and authors on the HLMRS, the system will facilitate access to an already developed Glossary of the concepts, their definitions and application examples, like the entry for confidentiality:

> Confidentiality requires that all professionals hold information about a client/patient/subject "safe" i.e. undisclosed because it was given to the professional with the idea that it would not be disclosed to others. This may include information given by the client or information gained by the professional in work paid for by the client. (Harris et al., 2000, p. 132)
>
> **Example:** Jeff is an airplane pilot who has been discovered to have vertigo in a research study in which he is enrolled. When the researchers involved report this finding to Jeff's medical physician without first asking Jeff, Jeff becomes furious because his confidentiality has been breached and because he could lose his pilot's license.

The question remains whether the peers can provide good feedback. As the SWoRD creators note, "One of the fundamental challenges is that peer reviewers are novices in their disciplines." (Cho & Schunn, 2005) Presumably, novice reviewers face most difficulty with domain-specific reasoning skills. Novices will lack the expertise to make informed judgments about higher level moral reasoning skills in the domain of engineering ethics. Apart from the new domain-specific criterion we will introduce, SWoRD itself addresses this challenge with the distributed expertise of multiple reviewers, statistical checks on review accuracy, and authors' back-reviews. Evaluations of SWoRD have shown that student authors improve their writing more from feedback of multiple peers than from single-peer or single-expert feedback (Cho & Schunn, 2005), even if the expert's feedback is of higher quality, according to a second, blinded expert.

Using AI to Enhance Peer Reviewing

Aside from adapting SWoRD, we are exploring how we can enhance peer review through system-generated feedback. The goal here is to ease the job of the reviewers who are faced not only with providing feedback to their peers, but also with learning difficult new skills themselves. While the new domain-specific criterion asks the reviewers to note what concepts are relevant to the case at hand, we hope to be able to show the reviewers what concepts have already been defined or applied by the author, and what concepts were relevant in similar cases. We will test out two new approaches: a *Concept Classifier* to detect what concepts have been defined or applied in a new case analysis; and a *CBR Matcher* to locate existing case analyses similar to the new one, and to report what concepts were defined or applied in them. The results of the processing performed by the *Concept Classifier* and *CBR Matcher* will serve as input to a new *Reviewer Aide* component. The *Reviewer Aide* will use a model of peer review to determine what feedback to display to the peer reviewers and in what form.

Both the *Concept Classifier* and the *CBR Matcher* tailor their feedback to the case analysis under review, and focus the feedback on framing through reporting on concept definitions and applications. We will train the *Concept Classifier* and the *CBR Matcher* on an already collected corpus of case analyses. It contains approximately 150 term papers written by graduate and undergraduate students as capstone exercises in Pinkus's Bioengineering Ethics course. We are in the process of completing manual annotation of the corpus using the Assessment Instrument to indicate where students define and apply any moral reasoning concepts, as in Figure 1. The annotation is being performed at the sentence level with the GATE natural language engineering software (Cunningham et al., 2002); approximately two thirds of the corpus has been annotated already. The papers represent the "multi-disciplinary" knowledge domain that comprises bioengineering ethics.

The database can be considered a "casuistry" of cases that provides examples of how a set number of ethical concepts and principles are used to frame and define issues that students have identified. The sample excerpt from a case analysis illustrates the coders' annotations. All these case analyses have been authored by students taught with the Harris text as part of their final class projects. They cover many of the topics addressed in the

Your Comments: Provide specific comments about the paper's evidence of moral reasoning skills. Suggest relevant concepts and how they apply. (The **GLOSSARY** has definitions and examples of many concepts from this course.)

Notes from the Reviewer Aide:

- This paper refers to 'safety' on lines 220 and 225, but there doesn't seem to be a definition. Here is a definition and example of safety from the **GLOSSARY**.
- This paper defines 'responsibility of a bioengineer'. Here are some sample definitions of responsibility of a bioengineer from similar case analyses. Does your author get it right?

Figure 5: Reviewer Aide prompt to peer reviewers

Harris text, such as honesty and the obligation to disclose information, in a bioengineering context. Students created and analyzed their own bioengineering ethics fact situations dealing with such questions as, "Should a graduate researcher report a defect in the cusp of a tissue-engineered heart valve that he is evaluating for another purpose?" and "How can an informed consent be written to enable a study of unexpected slips?" In connection with our experiments testing the sensitivity of the Assessment Instrument, we also collected 28 short student analyses of standardized bioengineering ethics cases such as one adapted from the first artificial heart transplant, which have also been annotated using the Assessment Instrument

Our hope is that the *Concept Classifier* can learn to detect concept definitions and applications in new case analyses. We will try to train a Naïve Bayes bag-of-words classifier (McCallum & Nigam, 1998) on definitions and applications in our corpus, starting with the most frequently occurring concepts (the full Instrument has over 40 concepts). We will then augment the term vectors with positional and natural language features like distance to beginning of document and part-of-speech tags. If this machine learning experiment is successful, then the *Concept Classifier* could make a probabilistic judgment whether a student author has:

1. Labeled a moral reasoning concept using proper terminology, and defined or applied it.
2. Labeled a moral reasoning concept, but failed to define or apply it.
3. Defined or applied a moral reasoning concept, but failed to label it as such.
4. Failed to label, define or apply a moral reasoning concept that is salient for an assigned case.

Another way to enhance peer reviewing with the help of AI is to show reviewers examples of how cases similar to the one at hand have been analyzed. We will attempt to create a *CBR Matcher*, which will use a Nearest Neighbor algorithm to make a probabilistic judgment whether a new essay is similar to existing analyses in the corpus. If so, then moral reasoning concepts discussed in existing essays may also be relevant to the new one; it would be easy to report what concepts were discussed in existing essays thanks to manual annotation by human coders. Similarity between case analyses can be determined by any shared concepts, or by whether the essays belong to the same curricular bioengineering 'track'. We will categorize our corpus according to six "specialty tracks" defined by the Bioengineering Department at the University of Pittsburgh within its graduate degree program: Cellular and Organ Engineering; Biomechanics of Organs, Tissues, and Cells; Biosignals and Imaging; Physiology and Biophysics; Neural Engineering; Rehabilitation Engineering and Human Movement. The results from this search for similar cases can be used to aid the peer reviewers in providing feedback to authors. Of course, the *CBR Matcher* presupposes that similar cases will have concepts framed in similar ways, and that reviewers and authors will benefit from such information.

The *Reviewer Aide* determines what feedback to display to the reviewers (Figure 5) based on input from the *Concept Classifier* and the *CBR Matcher*. It will make decisions about relevance, timeliness, informativeness, and accuracy of the feedback using criteria like "use only the search results from *Concept Classifier* that exceed a relevance threshold," "do not overwhelm reviewer with too many search results," "always refer to specific textual passages in feedback to reviewer," and "do not draw reviewer's attention to concepts that she already discussed in her comments." (We will refine criteria and thresholds after usability evaluations with reviewers.)

The feedback from the *Reviewer Aide* depends on the input from the *Concept Classifier* and *CBR Matcher*. While the criteria outlined above will moderate the quality of the feedback, the feedback might still be inaccurate, untimely, or irrelevant. Ultimately, the reviewer has to assess the *Reviewer Aide's* feedback, and choose to incorporate it (or not) in her own feedback to the author. Thus, the quality of the feedback the author sees should improve even if the reviewer disagrees with the *Reviewer Aide*: first, the reviewer will filter low-quality feedback, and second, the reviewer will modify mediocre feedback for the author's benefit. Since even in rejecting the *Reviewer Aide's* comments, the reviewer has been prompted to consider their relevance, we can use rejected or reviewer-modified feedback to evaluate and improve the system's performance. We can maximize this effect by having the *Reviewer Aide* present different feedback to different reviewers.

While we have high hopes for the system proposed here, we will also measure its contributions empirically. We will compare three conditions: traditional classroom teaching vs. SWoRD augmented with the domain-specific moral reasoning criterion vs. SWoRD augmented with the new criterion as well as with the *Reviewer Aide*, *Concept Classifier*, and *CBR Matcher*. To measure student learning, we will compare the capstone exercises described above across the conditions, i.e., the authentic task in this context. The measures for this task must include both traditional holistic grading and the Assessment Instrument, so as not to favor any condition

with a tailored learning measure. Our use of SWoRD with and without the *Reviewer Aide* facilitates additional comparisons across the two conditions: the value of the *Reviewer Aide* will be apparent first, if student authors give higher ratings to those reviewers who use the *Aide*, and second, if the student essays improve more between drafts when reviewers use the *Aide*. The development of the *Concept Classifier* and *CBR Matcher* will prompt another set of evaluations. As both are machine learning techniques, the appropriate comparisons will be on measures like precision and recall. Furthermore, data on the accuracy of these components will be necessary in tuning the *Reviewer Aide's* internal thresholds for presenting feedback to the reviewers.

RELATED WORK

Our focus on objectively measuring whether students learn moral reasoning skills provides ethics pedagogy with new empirical methods. Significantly, our system does not require a special case representation, it can handle never-before-seen cases within the domain of bioengineering ethics, and it orchestrates a useful collaboration through a reciprocal student authoring and reviewing. Thus, the system is likely to engage students more actively in ethical reasoning over a wider range of cases than "textbook-on-computer" resources like (Madsen; , "Online Ethics Center for Engineering and Science", 2006). Software like (Andersen et al., 1996; Searing, 2000) and web-based systems like (Goldin et al., 2001; Keefer & Ashley, 2001; McLaren, 2003; McLaren & Ashley, 1999; Robbins, 2005) support interactive case analysis, but unlike our approach, they lack instructional feedback and opportunity for collaborative learning.

Our focus on detecting LDA of key concepts complements research in automated essay scoring (AES) on higher-level features that indirectly relate to the quality of a written analysis and that improve perceived validity of the scoring model. In detecting instances of defining and applying, our *Concept Classifier* would deal directly with student texts as in (Landauer et al., 2003a, 2003b; Larkey, 1998), but does so by finding "proxes" or stand-ins for good case analyses as in (Burstein et al., 2001; Burstein et al., 2003; Page, 1966, 2003).

We seek to advance ITSs for writing by applying an ITS in a domain where it is the norm to analyze ill-defined problems in natural language. ITS that work with essays include Select-a-Kibitzer (Wiemer-Hastings & Graesser, 2000), Summary Street (Steinhart, 2001), AutoTutor (Graesser et al., 2000), Apex (Lemaire & Dessus, 2001), and Criterion (Higgins et al., 2004). These systems not only evaluate student essays on the fly, but they also provide feedback and encourage students to correct and rewrite their essays and resubmit them for new feedback. So far, however, ITS for writing can detect only fairly general features. For instance, Criterion and e-rater learn to detect 'discourse segments' like thesis, main idea, supporting idea, and conclusion, but they work on essays that have a rigid structure (an introduction, three supporting paragraphs, and a conclusion, about 300 words long). Alternatively, systems like the Intelligent Essay Assessor (Landauer et al., 2003a) that can address term-paper length essays detect general features, like coverage or absence of broad topics based on comparisons to past graded papers. An ability to detect finer-grained features such as examples of defined and applied concepts by the *Concept Classifier* would enable a writing ITS to give more detailed feedback.

CONCLUSION

In designing an ITS for engineering ethics, our objective is to extend the SWoRD approach to a technical domain where ill-defined problem solving and conceptual framing of cases are important. Our system will direct the feedback on HLMRS and on labeling, defining, and applying concepts through the student peer reviewers, who will decide whether to pass it along. This may help filter out any inapplicable feedback. Deciding which feedback to pass along is also a learning opportunity for the peer reviewers, who are students in the same class. Our approach offers a way to leverage the domain expertise embodied in a corpus of student papers from past offerings of the Bioengineering Ethics class.

The ill-defined properties of bioengineering ethics case analysis discussed here are not uncharacteristic of case analysis in other domains. The underlying task of case analysis requires the kind of argumentation that one finds in rhetoric, and, by extension, other humanities disciplines, and especially other problem-based learning scenarios. We believe that the Assessment Instrument can help engineering educators in engineering ethics domains beyond bioengineering (Goldin et al., 2006), and we hope that the combination of the Instrument with SWoRD and AI techniques will also generalize to those settings. While intelligent tutoring technology has been successful at well-defined tasks, its role in ill-defined tasks is less clear, and may require a more cautious approach, such as using AI in a support of human peer reviewing.

ACKNOWLEDGMENTS

This work has been supported in part by National Science Foundation Engineering and Computing Education grant #0203307. We thank our collaborators Christian Schunn and Janyce Wiebe for help writing an NSF proposal to continue this work.

REFERENCES

Andersen, D., Cavalier, R., & Covey, P. (1996). *A Right to Die? The Dax Cowart Case*: Routledge.

Ashley, K. D., & McLaren, B. M. (2001). *An AI Investigation of Citation's Epistemological Role.* Proceedings of Eighth International Conference on Artificial Intelligence & Law (ICAIL-01).

Burstein, J., Marcu, D., Andreyev, S., et al. (2001). *Towards Automatic Classification of Discourse Elements in Essays.* Proceedings of Meeting of the Association for Computational Linguistics.

Burstein, J., Marcu, D., & Knight, K. (2003). Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. In *IEEE Intelligent Systems*.

Cho, K., & Schunn, C. D. (2005). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers and Education, in press*.

Cunningham, H., Maynard, D., Bontcheva, K., et al. (2002, July, 2002). *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications.* Proceedings of 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia.

Goldin, I. M., Ashley, K. D., & Pinkus, R. L. (2001). *Introducing PETE: Computer Support for Teaching Ethics.* Proceedings of International Conference on Artificial Intelligence & Law (ICAIL-2001), St. Louis, MO.

Goldin, I. M., Ashley, K. D., & Pinkus, R. L. (2006). *Assessing Case Analyses in Bioengineering Ethics Education: Reliability and Training.* Proceedings of International Conference on Engineering Education, San Juan, Puerto Rico.

Goldin, I. M., Pinkus, R. L., & Ashley, K. D. (in preparation). Sensitivity and Reliability of an Instrument for Assessing Case Analyses in Bioengineering Ethics Education.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., et al. (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments, 8*, 149-69.

Harris, C. E., Jr., Pritchard, M. S., & Rabins, M. J. (2000). *Engineering Ethics: Concepts and Cases* (2nd ed.). Belmont, CA: Wadsworth.

Higgins, D., Burstein, J. C., Marcu, D., et al. (2004). Evaluating Multiple Aspects of Coherence in Student Essays. In *Proceedings of the Annual Meeting of HLT/NAACL*.

Keefer, M. W., & Ashley, K. D. (2001). Case-based Approaches to Professional Ethics: a systematic comparison of students' and ethicists' moral reasoning. *Journal of Moral Education, 30*(4), 377-98.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003a). Automated scoring and annotation of essays with the Intelligent Essay Assessor. In M. D. a. B. Shermis, Jill (Ed.), *Automated Essay Scoring* (pp. 87-112).

Landauer, T. K., Laham, D., & Foltz, P. W. (2003b). Automatic essay assessment. *Assessment in Education, 10*(3).

Larkey, L. S. (1998). *Automatic Essay Grading Using Text Categorization Techniques.* Proceedings of 21st Int'l Conference on Research and Development in Information Retrieval (SIGIR-1998), Melbourne.

Lemaire, B., & Dessus, P. (2001). A System to Asess the Semantic Content of Student Essays. *Journal of Educational Computing Research, 24*(3), 305-20.

Madsen, P. Ethical Judgments in Professional Life. Retrieved April 8, 2006, from http://www.andrew.cmu.edu/course/80-241/, login: guest, password: guest

Martin, T., Rayne, K., Kemp, N. J., et al. (2005). Teaching Adaptive Expertise in Biomedical Engineering Ethics. *Science and Engineering Ethics, 11*, 257 –76.

McCallum, A. K., & Nigam, K. (1998). *A comparison of event models for naive Bayes text classification.* Proceedings of 1st AAAI workshop on learning for text categorization, Madison, WI.

McLaren, B. M. (2003). Extensionally Defining Principles and Cases in Ethics: an AI Model. *Artificial Intelligence Journal, 150*, 145-81.

McLaren, B. M., & Ashley, K. D. (1999). *Case Representation, Acquisition, and Retrieval in SIROCCO* Proceedings of Third International Conference on Case-Based Reasoning, Munich, Germany.

Online Ethics Center for Engineering and Science. (2006). Retrieved April 8, 2006, from http://onlineethics.org/

Page, E. B. (1966). *The imminence of grading essays by computer.* Proceedings of Phi Delta Kappan.

Page, E. B. (2003). Computer grading of student prose, using modern concepts and software. *Journal of Experimental Education, 62*(2), 243-54.

Pinkus, R. L., Gloeckner, C., & Fortunato, A. (in preparation). Cognitive Science Meets Applied Ethics: Lessons Learned for Teaching.

Robbins, R. (2005). The Ethical Assistant.

Searing, D. R. (2000). Ethos System: Taknosys Software Corporation.

Steinhart, D. J. (2001). *Summary Street: An Intelligent Tutoring System for Improving Student Writing through the use of Latent Semantic Analysis.* University of Colorado, Boulder, Colorado.

Wiemer-Hastings, P., & Graesser, A. (2000). Select-a-Kibitzer: A Computer Tool that Gives Meaningful Feedback on Student Compositions. *Interactive Learning Environments, 8*(2), 149-69.

Culture in the Classroom: Challenges for Assessment in Ill-Defined Domains

Amy Ogan
Human Computer Interaction Institute, Carnegie Mellon University
aeo@andrew.cmu.edu

Vincent Aleven
Human Computer Interaction Institute, Carnegie Mellon University
aleven@cs.cmu.edu

Christopher Jones
Modern Languages, Carnegie Mellon University
cjones@andrew.cmu.edu

Abstract. While cognitive tutoring has been successful in well-defined domains, much less work has been completed in cognitive tutoring in ill-defined domains. Skill assessment is one of the difficult challenges that that must be solved before cognitive tutoring can become a reality in these domains. Assessment formats commonly used by instructors in ill-defined domains do not always have a correct answer or follow a finite set of solution paths. This necessitates clever ways of evaluating student responses to open-ended questions in which variability in both approach and final solution is a given. Additionally, there may be difficulty in covering the full problem space with a single assessment. In this paper we describe complementary assessments developed to evaluate a tutoring system in a classroom study. Critical analysis questions which are easily machine-interpretable and writing samples that address more nuanced understanding assess different skills that will be incorporated into a set of intelligent tutors that cover the cultural learning domain.

Keywords: ill-defined domains, assessment, intercultural competence

ASSESSMENT IN ILL-DEFINED DOMAINS

Cognitive tutors, a type of computer-based instruction that compares student actions to a model of correct and incorrect behavior and provides context-sensitive feedback and problem selection, have been effective at increasing student learning in real-world settings in domains with well-structured problems such as math and science domains. For example, use of the Algebra-1 Cognitive Tutor has been shown to increase student learning by at least one standard deviation over traditional classroom instruction [Koedinger, 1997]. The methodology which makes cognitive tutoring so successful in these domains, however, has made cognitive tutoring less practical in ill-defined domains. An ill-defined knowledge domain is one in which the following two properties hold: 1) each case of knowledge application involves the concurrent interaction of multiple conceptual structures such as schemas or organizational principles, each of which is individually complex, and 2) the interaction of these structures varies substantially across cases nominally of the same type, producing across-case irregularity [Spiro, 1995]. These properties push the limits of traditional cognitive tutoring methodologies, such as representing knowledge in the form of production rule systems. Although they present unique problems, ill-defined domains stand to benefit from intelligent tutoring as much as well-defined domains, because in these areas students profit greatly from tutoring strengths such as working at their own pace, receiving immediate feedback, and having privacy to practice skills before performing in front of the class.

In particular, skill assessment is one of the difficult challenges that that must be solved before cognitive tutoring can become a reality in ill-defined domains. Assessment is hard in these domains for several reasons. There may be difficulty in covering the full problem space. Ill-defined domains cannot be described in a finite set of production rules [Scandura, 2003]. This set of production rules enables assessment in well-defined domains to cover the complete set of skills specified as learning objectives. Ill-defined problems may have so many sub-problems that different people could focus on wholly different issues and still come up with viable solutions. A variety of problem types must be used to ensure that all skills are covered. Therefore we must choose the appropriate response types to elicit

useful student answers for each skill. Traditional tutors use standard response mechanisms such as multiple choice, or require numeric responses in quantitative domains. Much mathematical knowledge, for example, is well developed and linked to performance, and it is relatively simple to identify the correct answer for a range of questions requiring the understanding of basic concepts [Legree, 2005]. It is possible to leverage these techniques in ill-defined domains if we separate the skills of the domain which can be described as discrete or rule-based knowledge components from those which require an open, unconstrained environment to demonstrate in an ecologically valid manner.

The most appropriate way to identify the extent of student knowledge for some skills in an ill-defined domain, however, may be through open-ended responses. In similar situations medical professionals do not collect exactly the same data and do not follow the same paths of thought [Charlin, 2004]. The multiple choice test format requires a unique right solution to problems, a correct conclusive answer to give to data specified in the problem. Assessment formats commonly used by instructors in ill-defined domains do not require a single correct answer nor follow a solution path with a limited number of alternatives. For example, a student asked to synthesize study of immigration issues in France could draw on a wide variety of historical, journalistic or even experiential data in formulating a valid response. This issue of format is a problem because machine interpretation of correctness, which is a strength of cognitive tutoring, is difficult to execute on open-ended or verbal responses. It is simple to programmatically determine whether student responses to closed question types such as multiple choice formats are correct. Even brief short answers with a limited range of possible correct answers may be reviewed with simple natural language processing. Item construction in formal, well-defined knowledge domains can easily incorporate general knowledge and expertise, and item revision based on the use of item statistics can maximize characteristics such as reliability and validity [Legree, 2005]. Limitations in these techniques, however, necessitate clever ways of evaluating student responses to open-ended questions in which variability in both approach and final solution is a given. Even when this is successfully accomplished, the grading of open-ended responses may still be construed as subjective and also sets limits on standardization across students.

In this paper we describe an approach to the difficulties in assessment associated with ill-defined domains. We discuss two distinct types of assessment, critical analysis questions that are easily machine-interpretable and writing samples from a discussion board that demonstrate deeper knowledge, that are utilized to evaluate a tutoring system while conducting a study in a real classroom setting. We then describe how these assessment formats will be incorporated into a system of complementary intelligent tutors.

INTERCULTURAL COMPETENCE

The domain in which we focus our work, intercultural competence, is particularly ill-defined and difficult to assess in an objective manner [Kramsch, 1993]. Although there is no current consensus on the exact definition of this relatively new term, intercultural competence in general refers to the abilities to "reflect and gain insight on native perspectives, opinions, and values; reflect critically and engage with *otherness"* [Scarino, 2000]. In an attempt to include these higher-order skills in every language classroom, the American Council on the Teaching of Foreign Languages (ACTFL) has set forth a number of content standards regarding what students should know and be able to do in the document *Standards for Foreign Language Learning in the 21st Century* [ACTFL, 1996]. A significant number of these standards focus on cultural understanding, e.g. Standard 3.2: "Students acquire information and recognize the distinctive viewpoints that are only available through the foreign language and its cultures". While less familiar than the four skills of reading, writing, listening and speaking traditionally associated with language acquisition, cultural learning is a critical part of the second language classroom and is the foundation upon which all other language skills are based [Kramsch, 1993].

SKILLS

Student assessment refers to the documentation of whether the learning objectives of a domain were achieved. To define the learning objectives in a discipline, first we must specify the skills that are involved in domain mastery and then we can determine how best to assess them. While experts are not in complete agreement, the literature on intercultural competence suggests a culture-general skill set (e.g. Kramsch, 1999) which involves several distinct proficiencies that students are expected to master:

1) Critical analysis of culture: This refers to the ability to relate products, practices, and behaviors to cultural attitudes and values. Also to the ability to notice elements of a foreign culture

2) Cultural perspective-taking skills: The ability to look at and express cultural elements from outside one's own cultural space
3) Taking an ethnorelative stance toward culture: Demonstrating tolerance for cultural differences

For our tutoring system and assessment, we will focus on the first two skills, i.e. a critical analysis of culture and cultural perspective-taking, which rely less on stable personality traits such as openness than does the third, long-term skill of developing tolerance for cultural difference. These skill definitions are abstract and it is questionable whether they can be described with formalizable rules that always produce correct answers. Demonstration of these skills is open-ended and the knowledge may even be co-constructed, with consensus not always being reached. These skills are distinct from one another, and they necessitate different methodologies to assess whether students have acquired them. In particular, the second skill is difficult to assess with current cognitive tutor methodology which describes domains as a complete set of production rules covering the problem space that students do or do not know at any stage in their learning trajectory.

ASSESSMENT APPROACHES

In a typical classroom, teachers may assess these skills through various methods such as observation of behavior, writing samples, developing portfolios of written and oral work, or simply through classroom discussion, all of which result in "an analyzed profile" of the student [Tyler, 1949]. It is suggested that it may be helpful for instructors to employ triangulation, the use of a set of these assessment techniques to "cross check" competence to provide evidence of the greater validity of the various measures used [Deardorff, 2004]. These techniques are difficult to reproduce with traditional cognitive tutoring methods. Prior assessment in computer-based instruction in the area of cultural competence has developed along two distinct trajectories, neither of which covers the full scope of skills involved in cultural learning. Several quantitative studies have focused solely on knowledge that can be objectively assessed, such as facts learned through instructional video. In a controlled study presented by Herron [2000], students watched French video as an 'advance organizational tool' to gather information that was then used in answering cultural assessment questions. This type of evaluation measures whether the student has learned discrete knowledge components about the culture. While these questions target practices and products, such as the question "What do French people typically eat for a midday snack?", they are too narrow to address more controversial ideas such as the core values and attitudes of the target culture, and ignore the skills involved in more interactive tasks such as cultural perspective taking. It is tempting to reduce the complexity of a domain by narrowing the instructional scope to objectively assessable facts. However, if real complexities exist and their mastery is important, this reduction is an inappropriate oversimplification and can lead to conceptual misunderstanding [Spiro, 1995].

On the other hand, those who take an extreme constructivist approach believe that learning is a personal interpretation of the world and therefore objective measurement is nearly impossible [Merril, 1991]. Some work on computer-based pedagogical approaches has been done following these theories, such as large-scale, resource intensive projects like "Cultura" or "A la rencontre de Philippe" [Furstenberg, 2001]. In a different take on cultural instruction, these projects invite students to construct their own knowledge of core values and attitudes in a different culture. This is accomplished by students, largely without instructor intervention, answering questionnaires about their own culture and communicating with a French classroom to evaluate the authentic cultural descriptions provided by the questionnaires from the other class. This is an extremely motivating and deeply informative method of cross-cultural learning, but presents difficulties in scaling up in implementation due to the great deal of overhead involved with linking classes across continents. To avoid this overhead, a comparable type of constructive learning is done using similar existing material in a UC Berkeley study, *Teaching Text and Context Through Multimedia* [Kramsch, 1999]. More importantly, both of these curriculum approaches lack assessment of student learning beyond surface-level self-report by students of perceived quality of learning and motivation, e.g. a scale-based response to "Do you feel like you learned a lot?" Few empirical or descriptive classroom studies have been completed on work that subscribes to these theories.

SYSTEM DESIGN

We incorporated these skills into a tutor we developed to teach this domain. The Cognitive Tutor Authoring Tools (CTAT) are a set of tools built by researchers at Carnegie Mellon University and Worcester Polytechnic Institute that facilitate rapid development of intelligent tutors (for a full description of the CTAT system, see [Aleven, in press]). Using these tools, developers can build tutors

that use model tracing to compare student action to a model of correct and incorrect steps and provide individualized hints and feedback. One advantage of CTAT is the ability to build "example-tracing tutors", which allow authors to visualize and employ the processes found in full intelligent tutors without complex AI programming. Building an example-tracing tutor is accomplished first by designing a Flash interface with specialized interface widgets that communicate with the example-tracing system. Next, the author uses the interface to demonstrate the correct and incorrect steps students may take when completing a problem, and the provided Behavior Recorder tool automatically creates a graph of the demonstrated behaviors. After the author generalizes the problem-solving steps recorded in the graph and attaches hint and feedback messages, the graph is ready to be used for tutoring. These tutors can then be easily deployed to the Internet. One of the new Flash widgets developed especially for the tutor described here is a video component that logs all actions performed on the video (e.g., play, pause, "rewind").

To find the cultural content we present in the tutor, we requested suggestions of feature films demonstrating cultural attitudes or behaviors from French instructors who utilize such authentic material in their classroom. Instructors were asked to provide brief descriptions of appropriate scenes that fall under several chosen themes of the French culture. The films were documented to find one- to two-minute video clips that present cultural information and afford a natural moment to pause and ask students to make a prediction about the events of the second half of the clip. Clips with these "teachable moments" were chosen such that the prediction is dependent on cultural knowledge and not the narrative content of the film.

Using the CTAT tools, we created a tutor to display the cultural film clips and prompt students with questions to assist them in noticing cultural features of the film. First, students can review details about the film they are about to see, including film credits, a brief plot summary of the movie, and a paragraph of context for the clip they will be viewing. As an illustration, one video used was *Monsieur Ibrahim*, a film from 2003 that deals with issues of cultures clashing among immigrants in Paris. The next screen presents the video where students view the first half of the clip. In the scene from *Monsieur Ibrahim*, a boy Moses walks into a neighborhood convenience store and continues a conversation with the elderly proprietor about the etymology of their names. The proprietor gently explains to Moses that he is Muslim, and not Arab. The boy asks, "Then why does my father say 'Go to the Arab's?". At this point the video pauses and students respond to a set of questions (see Fig.1) in which they: 1) predict the next event that will occur in the clip from a drop-down menu, 2) provide a more extensive natural language explanation of their choice, and 3) state what they believe an appropriate response to the situation might be in their own culture. The first question is presented in a menu format to suggest several appropriate responses and lightly constrain students to actual cultural possibilities. For this video clip, two of the possible responses include 'The neighbors don't take the time to get to know me' and 'Anyone in this profession is labeled an Arab'. Unlike a traditional cognitive tutor, any reply is accepted in the drop-down menu. In the second question, students are given space to explore their hypothesis and provide evidence from the clip or their knowledge of the French culture to support their reasoning. They may rewind and review the video up to this point as often as they like. The tutor does not provide feedback.

When students have provided answers to all three questions, the succeeding portion of the video clip plays and the ensuing cultural event is revealed. At the end of the video clip, a second set of questions appears that asks students to make an assessment of whether they were correct in their prediction or not. If they were not correct, they are asked to revise their prediction. Finally, students are given a set of characterization questions about the clip that may be answered with 'true', 'false', or 'maybe'. These questions are tutored with hints and error messages, as well as success messages that provide a summarization of the evidence for a correct answer. For example, one question states, "Monsieur Ibrahim lives in an isolated immigrant community", which would be correctly answered with 'false'. At this point, students may rewind and review the full clip as often as they like.

Figure 1: Screenshot of the initial set of tutor questions

Following the video, students participate in an asynchronous online class discussion where they use what they have seen to reflect on cultural differences, similarities or assumptions the class has about the French culture, and ask questions about the meaning of behaviors they have seen in the clips. The prompt that students see for *Monsieur Ibrahim* is, "Post at least one original post and one reply with questions, analysis, or other thoughts about the immigration issues in France you've seen! Think about what are the racial and ethnic stereotypes in France that you have seen depicted in this film to get started." The responses students provide regarding their personal experiences spark interesting comparisons with valuable information from multiple cultures. The writing samples posted to the discussion board form one part of the assessment for evaluating student improvement using the tutor, along with a post test that measures the cultural knowledge that students have acquired from the components in the film clip.

STUDY DESIGN

A full study was recently completed that used the assessments we developed to test the tutor. Data collection was done automatically through the Pittsburgh Science of Learning Center (PSLC) and Open Learning Initiative infrastructures. The PSLC (http://www/learnlab.org) is a "national resource for learning research" that includes authoring tools for online courses and support for running "in-vivo" learning experiments, that is, learning experiments is real educational settings, as opposed to the lab. Elementary French I Online is a PSLC-targeted course that is being developed as part of the Open Learning Initiative at Carnegie Mellon University, which is a program that provides a collection of "openly available and free online courses and course materials." This combination of web-based distribution and experimental rigor allows a number of innovative instructional components to be designed, evaluated, and deployed as part of the French Online Course.

The study was conducted in the Carnegie Mellon and University of Pittsburgh Elementary I French Online classes to test the hypothesis that attention-focusing techniques increase cultural learning in an online environment. Thirty-two students were randomly assigned within each class to either an experimental group that used the system described above or a control group that viewed the same video clips without intervention. The control group watched the clips as often as they liked, but without the attention-focusing caused by pausing the video at a critical moment and responding to the three sets of questions about the content. In each of three assignments spaced throughout the semester, groups were assigned two film clips to watch and discuss in place of a typical cultural reading and writing assignment for the class. Film clips corresponded to cultural themes that were explored in the classroom, such as immigration, employment, and education in the French culture. The materials for each assignment were linked from a single course webpage that provided students with background information about the theme. According to the format of the rest of the online course material, students worked through each assignment at their own pace. This information was written in English and was adapted from the cultural course materials that the tutor replaced. Students in the control condition performed exactly the same sequence of instruction, except that the "tutors" linked off their course materials simply presented the video clips without the attention-focusing techniques.

ASSESSMENT

Prior to the first assignment, demographic data was collected for each student. In each assignment, students first took a pre-test that explored their knowledge of basic information about the theme. For example, one question related to the theme of education asked 'The baccalauréat is…' (the baccalauréat is a French examination similar in some respects to the American SAT). The response was open-ended and allowed students to demonstrate all of their knowledge about each topic, while avoiding the testing threat of influencing what they notice in the film clip. Students completed tutor sessions with video clips from two different movies relating to the lesson theme. All actions that students performed in the tutor were recorded. After each tutor, students accessed the discussion board for that video clip and were required to post at least twice and encouraged to return to the discussion board to read other replies and to post again. All discussion posts were stored in a database, along with a link to the original post if it was a reply. Finally, students completed a post-test for the assignment that included the analytical questions described in the next section along with self-report on scales that measure items related to intercultural competence such as world-mindedness and attributional confidence [Sampson, 1957]. Our work on assessment is currently focused on the analytical post-test questions and the discussion board writing samples. This mixed assessment methodology addresses a set of concerns about ill-defined domains.

The analytical post-test questions are designed to be able to leverage traditional cognitive tutor techniques of assessment. The questions were developed with the help of a French citizen and language instructor through a component analysis of the cultural elements in the film clips. Cultural elements in the film were decomposed into a hierarchy that covered the main concepts in the theme. Each element was then formed into a question requiring cultural analysis that was situated within the context of the film. Each post-test covered three questions in true/false format relating to each film in the theme as well as two questions in multiple choice format that asked students to compare and contrast across films in the theme. In addition, several of the questions were followed by a short answer component which asked students to explain why they made the choice that they did. See Figure 2 for examples of the two types of questions. This type of assessment allows us to cover more completely the intercultural competence problem space by ensuring that students are assessed on their knowledge of the key cultural elements relating to each theme, specifically relating to values and attitudes that are deeper than the behavioral elements assessed in such work as the Herron study.

A multiple choice question on the theme of education:	Based on the clips you just saw, *Être et avoir* and *Le Péril jeune* are similar in EVERYTHING BUT: a) Students in both schools are expected to take responsibility for their learning b) Students in both schools are expected to obey authority c) Students in both schools are expected to have a positive attitude
A true/false question from the theme of employment:	From the factory scene in *Ressources humaines*, we know that social mobility is possible in French society.

Figure 2: Examples of post-test analytical questions

The writing samples from the discussion boards comprise the second main type of assessment. On each discussion board students were asked to post in general with cultural observations and were given one specific stimulus about the theme to help them begin, such as:

> "Post at least one original post and one reply with questions, analysis, or other thoughts about the immigration issues in France you've seen! Think about what racial and ethnic stereotypes in France you have seen depicted in this film to get started."

The writing on the discussion board addresses the limitations of closed format questions by allowing students to demonstrate intercultural competence skills in a less constrained, and therefore more ecologically valid, manner. To evaluate the quality of writing in the discussion posts, we hand code the responses into three major categories of good and bad cultural writing that were developed for a similar intercultural competence writing task [Steglitz, 1993]. In a general sense, a category 1 response gives no indication that the author recognizes the role of culture in describing behaviors or values of a culturally different other and may take the form of judgments or advice. Category 2 responses acknowledge that there are general cultural influences that may be the cause of what they notice, while category 3 responses relate behaviors and values to specific cultural phenomena, such as power

relationships or individualism. These student writing samples can be used to both measure learning outcomes directly and monitor behaviors that may lead to future learning. For example, students may demonstrate knowledge of a correct French perspective as well as demonstrate the ability to take a different perspective, even if the particular one that they take is not valid from the French point of view. This writing also serves as a record or portfolio of fine-grained learning that may assist in determining progress on intercultural competence skills over the course of the semester.

RESULTS AND DISCUSSION

ASSESSMENT VALUE

The analytical post-test questions have the ability to confirm or disprove an experimental hypothesis. In a preliminary analysis of the post-test question data from the French Online study, we found that 9 of the top 10 scoring students are from the experimental condition. While students in the control condition scored an average of only 64% on the post-tests, students in the experimental condition scored an average of 72%. These results partially confirm the hypothesis of the experiment, that attention-focusing techniques increase cultural learning in an online environment. To fully validate this result across cultural skills, we must also look at the other assessment type. In an average taken for each student of scores on Steglitz's scale for all discussion posts, the experimental condition once again outperforms the control condition with an average of 1.61 out of 3 to an average of 1.36 out of 3.

Although both assessment types show an advantage for the experimental condition, we argue that the analytical questions and the writing samples each add a unique value to the assessment of skills in this domain. The analytical post-test questions are designed to assess whether students have understood the knowledge components picked up as cultural elements in the video. They determine if the student can accurately characterize the French perspective. Because students may not express objective knowledge while writing in the discussion, the analytical questions confirm that students have learned these key cultural elements. For example, the following discussion post is written as an opinion without evidence, taken from a personal perspective. It does not demonstrate that the student has achieved deep cultural understanding or is able to approach the events in the film from a perspective that culture might have an influence on behavior:

> "I agree with Eiben's point. Who are we to judge people? I believe that we should not judge people from what we hear or what they look like. Plus it is the parent's duty to raise the child to better understand the world around them. If the boy's father had not mentioned to him that the store clerk was Arabe he would not have mention it to the clerk."

This same student, however, received an 87% average on the analytical questions. While he did not demonstrate perspective taking skills in his writing, he has shown that he was able to extract cultural information from the film clips.

The discussion writing samples can give us insight into the other skills that make up intercultural competence. They allow a deeper investigation of whether students can actually do perspective-taking. With this method of assessment as well, we see differences in the skills that students demonstrate. One of the five lowest scoring students on the analytical questions, who received a 47% average, was able to articulate various French perspectives in immigrant communities and present evidence from the film in his writing:

> "The two individuals walking through the neighborhood seemed to represent a dichotomy in French culture: the ignorant one who was afraid of the black people and the accepting one who lived amongst a scene of racially and ethnically diverse people. This shows that people in the French culture are not all on the same "wavelength." Hopefully, accepted diverse communities are the future of France and if they are, maybe the United States can take a hint from France in this particular sense."

This student may not have noticed all of the specific cultural elements in the film clip to answer analytical questions, but was able to provide an overview of the main cultural point of the clip from outside of his own personal beliefs.

In a preliminary analysis of the data, it does appear that these assessment techniques measure different skills. In a first round of data coding, each post on the discussion boards was given a score according to the categories developed by Steglitz and then an average score was calculated for each student across all of their posts. Additionally, each student was given an average score across all of

their post test responses. No significant correlation was found between the two average scores (adjusted r squared = .066, $p > .15$). Separately, each type of assessment could provide insight into different student misconceptions. When analyzed together, the discussion board could be used as a process measure to determine how and when students are accurately able to answer the post-test questions.

APPLICATION TO TUTORING

Beyond being an effective measure of student learning, these assessment methodologies also have great potential for incorporation into cognitive tutors. The analytical questions are more than simply a post-test, but can be used within a tutor as a problem solving exercise following the presentation of cultural material. While evaluating writing samples is not common in existing tutoring methodology, we are currently training machine learning algorithms to identify critical features of the discussion writing that can predict whether it is a good example of cultural writing, using the categories developed by Steglitz as criteria. This scale has been validated using human raters, but no work has been done on machine interpretation of the writing samples. Our current work focuses on identifying and predicting intermediate features that are relatively easy to detect through machine learning that may predict student performance levels on the higher-level cultural proficiency categories. For example, one feature with significant predicting power seems to be the extent of argumentation. Students may write inferences, which include evidence from the film and a conclusion; opinions, which are conclusions with no evidence; or recall of information, which is evidence without conclusions about what they have seen. Using the kappa statistic, a measure from 0 to 1 that gauges reliability between raters (with 0 being no agreement and 1 complete agreement), we are currently able to achieve kappa values between .5 and .9 for predicting the intermediate features we have identified, using a set of machine learning algorithms such as Support Vector Machines (SVM) that are known to work well with text. Employing these intermediate features, we have improved from a low baseline kappa value of .01 using just the words in the post as features to a kappa value of .4 using the current framework with the intermediate features as predictors. Work continues in this area to improve prediction of both the intermediate features and to identify features that better predict the high-level categories. With this machine interpretation, we have the eventual goal of developing a tutor for the discussion board that will give on-line, non-deterministic feedback to students on how they might improve their cultural writing, based on the intermediate features that more conducive to guided feedback.

Cognitive tutors in this domain can benefit from using both types of assessment in conjunction with each other. It may be relatively easy to determine whether the discussion writing is of good general quality by using features that are less complicated to extract from writing, such as being on-topic or attempting to make inferences from evidence. It is a much more difficult natural language problem to know whether the inferences expressed are correct French perspectives, because there are so many ways of communication in free response. In this case, a correlation with student responses to the analytical questions may give a complete picture of the current state of student understanding. Together, these assessments speak to the initial concerns we describe for assessment in ill-defined domains. The writing samples allow students to explore the problem space with an open-ended, unconstrained format that allow for a variety of solution paths that address any number of sub-goals. Machine interpretation of correctness is easily accomplished with the analytical post-test questions, and we are working towards a procedure for evaluating the writing samples automatically in a way that is conducive to the cognitive tutoring goals of tracking student knowledge and providing context-sensitive feedback. We have made steps toward covering the full problem space of the domain by exploring multiple types of assessment that identify different types of student misconceptions and assess disparate skills which cannot be assessed in the same way. In addition, this assessment methodology has the ability to be applied to other ill-defined domains with similar properties. For example, in art or literary criticism, students also must have a strong foundation in the key concepts of the domain. Additionally, in such domains in which dialogue is a critical component, they must be able to express opinions and support them with evidence. We believe that this combined assessment methodology will be effective for intelligent tutoring in the domain of intercultural competence, and has the potential to extend to tutors in other ill-defined domains.

ACKNOWLEDGEMENTS

We would like to thank the Pittsburgh Science of Learning Center for funding this work, as well as the Cognitive Tutor Authoring Tools team at Carnegie Mellon University for providing incomparable assistance with developing specialized tools for the tutor. Michael West and Cary Campbell continue to accommodate us as instructors for the French Online courses.

REFERENCES

ACTFL (American Council on the Teaching of Foreign Languages) (1996). Standards for Foreign Language Learning: Preparing for the 21st Century. New York: ACTFL.

Aleven, V., Sewall, J., McLaren, B. M., & Koedinger, K. R. (in press). Rapid authoring of intelligent tutors for real-world and experimental use. Accepted for presentation at the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006).

Charlin, B. & van der Vleuten, C. (2004). *Standardized Assessment of Reasoning in Contexts of Uncertainty: The Script Concordance Approach.* Evaluation of the Health Professions. 28 (3), pp.304-319.

Deardorff, D. (2004). The Identification and Assessment of Intercultural Competence as a Student Outcome of Internationalization at Institutions of Higher Education in the United States. Unpublished dissertation available at http://www.lib.ncsu.edu/theses/available/etd-06162004-000223/unrestricted/etd.pdf

Furstenberg, G., et al. (2001). Giving a Virtual Voice to the Silent Language of Culture: The Cultura Project. *Language Learning & Technology, 5*(1), 55-102.

Herron, C., & Dubreil S. (2000). Using Instructional Video to Teach Culture to Beginning Foreign Language Students. *CALICO*, 17(3), 395-429.

Koedinger, K. R.; Anderson, J. R.; Hadley, W. H.; and Mark, M. A. (1997). Intelligent Tutoring Goes to School in the Big City. *Journal of Artificial Intelligence in Education* 8(1): 30–43.

Kramsch, C., & Anderson, R. (1999). Teaching Text and Context through Multimedia. *Language Learning & Technology, 2*(2), 31-42.

Kramsch, C. (1993) Context and Culture in Language Teaching. Hong Kong: Oxford University Press.

Legree, P. J., Psotka, J. & Tremble, T. (2005). Applying Consensus Based Measurement to the Assessment of Emerging Domains (ARI Technical Report No. 1153). Alexandria, VA: U. S. Army Research Institute for the Behavioral and Social Sciences.

Merrill, M. D. (1991). Constructivism and instructional design. *Educational Technology*, May, 45-53.

Sampson, D.L., & Smith, H.P. (1957). A scale to measure world-minded attitudes. Journal of Social Psychology, 45, 99-106.

Scandura, J. (2003). Domain Specific Structural Analysis for Intelligent Tutoring Systems: Automatable Representation of Declarative, Procedural and Model-Based Knowledge with Relationships to Software Engineering. *Tech, Inst, Cognition and Learning*, Vol. 1, pp 7-57.

Scarino, A. (2000). 'The Neglected Goals of Language Learning', Babel, 3 (34), Summer, pp 4-11.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1995). Cognitive Flexibility, constructivism, and hypertext: Random access instruction for advance knowledge acquisition in ill-structured domains. In P. Stele, & J. Gale, *Constructivism in education* (pp. 85-108). Hillsdale, NJ: Erlbaum.

Steglitz, I. (1993). The Identification and Assessment of Intercultural Competence as a Student Outcome of Internationalization at Institutions of Higher Education in the United States. Unpublished dissertation available at http://www.lib.ncsu.edu/theses/available/etd-06162004-000223/unrestricted/etd.pdf

Tyler, R.W. (1949). Basic principles of curriculum and instruction. Chicago: University of Chicago Press.