AutoTutor: An evaluation of interface design , domain independence and dialogue modelling

Kalliopi Irini Malatesta

Master of Science
School of Cognitive Science
Division of Informatics
University of Edinburgh
2001

Abstract

This thesis is an analytical evaluation of AutoTutor an ITS that aims to simulate naturalistic tutorial dialogues. Features under consideration are the system's domain independence, the interface design, the support of deep reasoning in the tutorial dialogue and student modelling. Tutors with similar credentials are studied in order to specify the requirements of such a system. A pilot study is described for the purposes of evaluating the actual performance of the tutor on the isolated features. The architecture and design of the system are analysed in detail through a porting procedure to a new domain. Concrete and feasible suggestions are put forward regarding the usability and extensibility of the system.

Acknowledgements

I would like to say a big thank you to Judy Robertson, who although supervised me unofficially was better than any supervisor I could ever imagine. I would also like to thank my supervisor Peter Wiemer-Hastings.

I wish to thank Angelina for her lovely designs, Beata for her technical and moral support and all my coursemates and friends, especially the ones who were brave enough to participate in my pilot study. I would also like to thank Manolis for his help in latex and his support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Kalliopi Irini Malatesta*)

Table of Contents

Chapter 1

Introduction

Recent advances in Intelligent Tutoring System technology focus on developing tutors which act as conversational partners in learning. AutoTutor, one of the prevalent systems in this field, claims to simulate naturalistic tutoring sessions in the domain of computer literacy. One innovative characteristic of AutoTutor is the use of a talking head, realised by a computerised agent, as an interface with the user. The system is also claimed to be domain independent and to be capable of supporting deep reasoning in the tutorial dialogue.

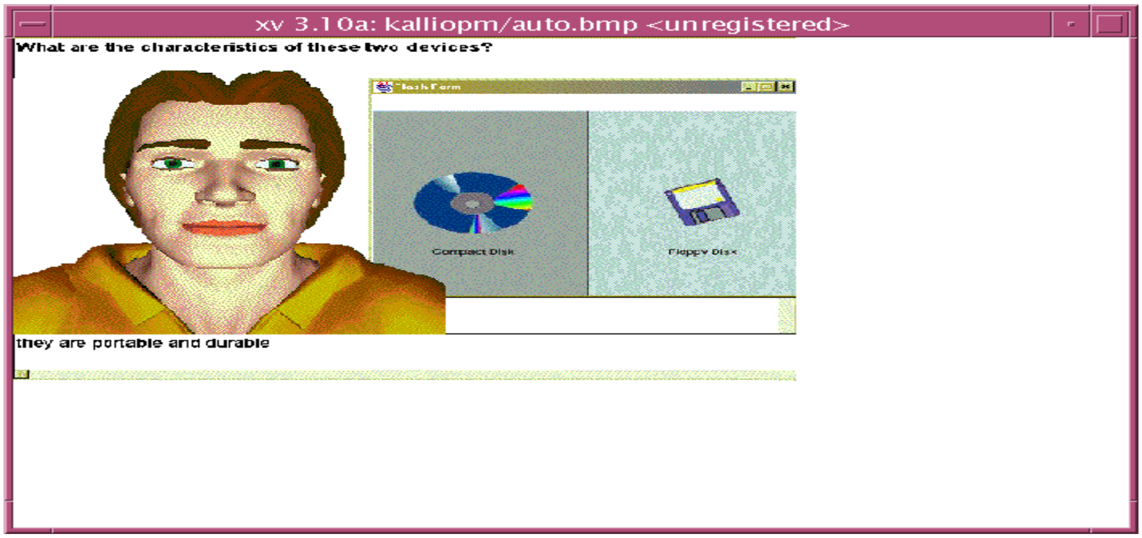
It is challenging to try and evaluate these claims. Strong motivation towards this goal, was the fact that the domain of AutoTutor does not require deep reasoning mechanisms. Porting the tutor to a new domain, a domain that could provide the grounds for more in depth qualitative reasoning, was identified as a step that could allow a hands-on evaluation of the system.

A thorough review of the literature and the progress in the field, made clear the need to re-evaluate some claims made regarding the system's performance and envisage feasible modifications that could be implemented. This process involved careful consideration of the available alternative approaches to design, and a detailed study of ways to integrate novel theories and strategies in the existing system. A pilot study on the running version of AutoTutor will shed light on the actual performance characteristics of the system.

Porting the tutor in a new domain aims to provide the data for a broader evaluation of AutoTutor as a domain-independent ITS.The procedure of porting sheds light to a wide variety of problematic areas that are not documented in the AutoTutor literature. Issues of usability and extensibility are addressed. Based on the results, concrete suggestions are made, on feasible modifications of the system.

1.1 AutoTutor Preview

AutoTutor attempts to simulate naturalistic tutorial dialogues in the domain of computer literacy. The system is based on simple concepts inspired by human-to-human dialogue structure. Careful consideration of one-to-one tutoring dialogue transcripts has provided AutoTutor with a teaching strategy that allows multi-turn dialogue patterns and co-construction of knowledge from tutor and student. AutoTutor asks open questions that require paragraph-long answers. This elicits answers from the student that portray her current knowledge of the topic. The tutor uses language delivered by an agent interface, along with other modes such as diagrams and animated agent gestures to communicate with the student. The following figure is a snapshot of a tutoring session.



xv 3.10a: kalliopm/auto.bmp <unregistered>

What are the characteristics of these two devices?

they are portable and durable

Figure 1.1: AutoTutor on Computer Literacy

1.2 Main Thesis Goals

The thesis describes an evaluation of claims put forward for the AutoTutor framework in terms of:

1. Domain Independence (How portable is the current software implementation?)
2. Interface Design (Does the use of an agent improve or hamper the interaction with the users?)
3. Support of Deep Reasoning (Does the system allow deep reasoning mechanisms to be activated in the tutorial dialogue?)
4. Dialogue Management (Are the claims made regarding the performance of the dialogue management consistent with the tutor's behaviour? Are they sufficient?)
5. User Modelling (How does the absence of a user model affect the system performance?)

The evaluation is performed on a qualitative level and is only concerned with verifying or disproving claims made regarding the framework's capabilities. The teaching effectiveness of the tutor is not raised as an issue and thus will not be addressed throughout this thesis.

1.3 The Structure of the Thesis

The second chapter looks at the related literature on one-to-one tutoring and grounding tutorials to examples in order to set the background theory for the analysis of the design features of three dialogue tutors. AutoTutor along with CIRCSIM-Tutor and Atlas-Andes are studied in depth, and conclusions are drawn regarding the specifications such a type of tutor must meet and the correspondence of those specifications to AutoTutor.

Chapter three explores the actual characteristics of the system and juxtaposes them to the claims made regarding its performance. The interface approach, the dialogue management, the student modelling techniques and the deep reasoning mechanisms are analysed through the observation of users interacting with the system and their valuable feedback. This analysis provides a more realistic estimate of the system's performance in terms of the features of interest.

Chapter four describes the process of porting AutoTutor in a new domain that supports deep reasoning. Motivations for such an approach are outlined, along with argumentation supporting the choices made. Through this attempt various software engineering issues are raised regarding the architecture and design of the framework.

Chapter five outlines some suggestions for improving the existing AutoTutor framework, both on a short and long term basis keeping in mind the characteristics and idiosyncrasies of the system, identified in the previous chapters.

Chapter six is the final chapter that puts the whole attempt on evaluation together and draws conclusions on current status and future perspectives.

Chapter 2

Literature Review

In order to evaluate the significance of the claims made regarding AutoTutors performance as a domain independent teaching partner that supports deep reasoning and simulates naturalistic tutorial dialogues, it is necessary to analyse recent progress in this field. By studying the features of three state of the art tutors that use collaborative dialogue in one-to-one tutoring sessions, we can set the grounds for an evaluation and a new approach in AutoTutor. A close look in the modules of the three chosen systems, CIRCSIM-Tutor, Andes and Autotutor will dictate the criteria to attain this goal. First, a general analysis of research on one-to-one tutoring is judged as essential, to justify the choice of the specific field of tutors, in the group of recent ITSs developed.

2.1 One to One Tutoring

In search of alternatives to classroom teaching, researchers have focused on various tutoring methods. One-to-one tutoring has received much of this attention, perhaps because it allegedly caters to the idiosyncratic needs, misconceptions, and knowledge deficits of a particular student.

In 1984, Benjamin Bloom defined the ”two-sigma problem,” which states that students who receive one-on-one instruction perform two standard deviations better than students who receive traditional classroom instruction. An improvement of two standard deviations means that the average tutored student performed as well as the top 2 percent of those receiving classroom instruction (Bloom, 1984).

According to Cohen et al.'s meta analysis of 52 tutoring studies, tutored students performed .4 standard deviations higher than conventional classroom controls. It is informative to note that these advantages were not related to the amount of training that the tutors received or to the age differences between tutor and student. Tutors in most school systems consist of students, paraprofessionals, and adult volunteers, yet even these minimally trained tutors are effective (Cohen et al., 1982).

One re-occurring finding in several studies (Graesser et al., 1995; Fox, 1993; Hume et al., 1996; Moore, 1995) is that human tutors rarely adhere to ideal tutoring models that are often integrated into intelligent tutoring systems. Instead, human tutors tend to rely on pedagogically effective strategies that are embedded within the conversational turns of the tutorial dialogue.

The question that arises is: What is it about human tutoring that facilitates this learning? Many researchers argue that it is the collaborative dialogue between student and tutor that promotes the learning (Fox, 1993; Graesser et al., 1995; Merrill et al., 1992). Through collaborative dialogue, tutors can intervene to ensure that errors are detected and repaired and that students can work around impasses. The consensus from these studies is that experienced human tutors maintain a delicate balance, allowing students to do as much of the work as possible and to maintain a feeling of control, while providing students with enough guidance to keep them from becoming too frustrated or confused (Core et al., 2000).

Keeping this in mind, one can imagine designing a personal training assistant for each learner in a classroom. An assistant could pay attention to the participant's learning needs, assesses and diagnoses problems, and provides assistance as needed. The assistant could perform many of the routine instructional interventions and alert the instructor of learning problems that are too difficult for it. By taking on basic assistance tasks, the instructor would be free to concentrate on training issues that require greater expertise.

Providing a personal training assistant for each learner is beyond the training budgets of most organisations. However, a virtual training assistant that captures the subject matter and teaching expertise of experienced trainers provides a captivating new option. The concept, known as intelligent tutoring systems (ITS) or intelligent computer-aided instruction (ICAI), has been pursued for more than three decades by researchers in education, psychology, and artificial intelligence. Today, prototype and operational ITS systems provide practice-based instruction to support corporate training, K-12 and college education, and military training.

The goal of ITS is to provide the benefits of one-on-one instruction automatically and cost effectively. Like training simulations, ITS enables participants to practice their skills by carrying out tasks within highly interactive learning environments. However, ITS goes beyond training simulations by answering user questions and providing individualised guidance. Unlike other computer-based training technologies, ITS systems assess each learner's actions within these interactive environments and develop a model of their knowledge, skills, and expertise. Based on the learner model, ITSs tailor instructional strategies, in terms of both the content and style, and provide explanations, hints, examples, demonstrations, and practice problems as needed.

Research on prototype systems indicate that ITS-taught students generally learn and translate the learning into improved performance in a competitive rate to that of classroom-trained participants. Focusing on ITSs that attempt teaching through collaborative dialogue the findings of the effectiveness of such systems is quite promising. This is discussed in section 2.3.

2.2 Learning from Examples

Examples are theoretically regarded as important components of learning and instruction. Especially in the case of one-to-one tutoring it has been reported that most questions asked by tutors were embedded in a particular example (Graesser, 1993). Sweller (1988) has suggested that worked examples have cognitive benefits over active problem-solving. Active problem-solving often leads to dead-ends, or lengthy, erroridden solution paths. Providing students with worked examples reduces the student's cognitive load by eliminating futile problem-solving efforts.

Others claim that examples are most beneficial when they are rich in context and anchored in real-world situations. Presumably, these anchored examples include challenging material, are motivational in nature, and ultimately facilitate transfer to new problems Person (1994, chap. 1).

Based on this research it is made apparent that grounding tutoring to examples is particularly important in one-to-one tutoring and thus in the design of Intelligent Tutoring Systems that aim to simulate naturalistic tutorial dialogues.

The next issue that comes to mind is which is the optimal selection mechanism for these examples in terms of learning outcomes. According to Person (1994)'s macro and micro analysis of examples that tutors generate during one-to-one tutoring sessions, tutors rely on curriculum scripts to control the flow of the tutoring sessions. Curriculum scripts allow tutors to follow organised gameplans rather than jumping from topic to topic for each student error. The results of this study indicate that most tutor examples are generated by curriculum scripts and the textbook. It is also reported that most tutor-generated examples are used to elaborate or explain difficult topics and are rarely prompted by student errors or misconceptions. These conclusions will be used to guide the specific choices that need to be made in designing the new domain knowledge and curriculum script.

2.3 Related Work

Core et al. (2000) have identified three events that a tutor must be able to deal with:

1. failure (students may answer a tutor question wrong or the whole tactic may not be working)
2. interruptions (students may interrupt with a question)
3. the need to revise their tactics (student behaviour may indicate that the tutor can skip steps in an explanation or directed line of reasoning)

Only then can tutor and student co-construct explanations and tutors can walk students through lines of reasoning. In this section we will mainly focus on AutoTutor and then on ANDES and CIRCSIM-Tutor, hoping to theoretically define the requirements such a system must meet from an architectural and designing point of view. The findings of this attempt will then be juxtaposed with the claims made in the literature regarding the AutoTutor framework characteristics and on the actual performance as it will be investigated through a pilot study.

2.3.1 AutoTutor

AutoTutor is an animated pedagogical agent that serves as a conversational partner with the student. It aims to be a simulation of a human tutor. It provides an introductory computer literacy course, aiming to assist students in learning the fundamentals of hardware, operational systems and the Internet. AutoTutor's tutoring interaction is based on in-depth studies of human tutors, both skilled and unskilled (Graesser, 2000). Thus, two versions of the ITS were planned to be developed: one that simulates the dialogue moves of normal, untrained human tutors, and one that would simulate dialogue moves that are motivated by mere sophisticated, ideal tutoring strategies such as Socratic dialogue, modelling-scaffolding-fading, and strategic hinting. Since the ideal tutoring strategy version has yet to reach the phase of a running version only the first version will be assessed in the remainder of the text.

AutoTutor is an amalgamation of classical symbolic architectures (e.g. those with propositional representations, conceptual structures, and production rules) an architectures that have multiple soft constraints (e.g. neural networks, fuzzy production systems). The tutors major modules include an animated agent, a curriculum script, language analysers, latent semantic analysis (LSA, (Landauer and Dumais, 1997)) and a dialog move generator.

AutoTutor's graphical interface is comprised of four features : a three-dimensional talking head , which is capable of speech-related gestures, a text box for typed student input, a text box that displays the problem/question being discussed, an a graphics box that displays pictures and animations that are related to the topic at hand. The talking head feature aims to set the grounds for a pedagogical and polite interaction. When the learner's contribution is incorrect or vague, for example, the speech generated is accompanied by agent animation that can be positive and polite whereas the face has a puzzled expression. It should be pointed out that the current animation set of the tutor does not always succeed in conveying the desired reaction. This is a finding that is further discussed in the evaluation chapter, where feedback from users is analysed.

It has yet to be empirically proven that the existence of the agent interface facilitates learning. The learner communicates with the tutor by typing answers or questions in a prompt line. This input is interpreted by the system using the technique of Latent Semantic Analysis (LSA) (Person et al., 2001).

AutoTutor's knowledge about computer literacy is represented by LSA (Wiemer-Hastings et al., 1999). It is a statistical method that calculates the similarity of words or phrases appearing in particular contexts, to pre-stored corpus of related texts. The 'truth' of a student's contribution is evaluated by computing its match with expected answers to a question, or expected solutions to a problem. This implementation of the interface is closely related to the teaching strategies the system follows. Depending on the calculated 'correctness' of the answers provided the system commits to various dialogue moves accordingly. It should be noted that LSA does not take into account any syntactic information. This unavoidably limits the tutors understanding of the contributions. A more detailed analysis on LSA will be made in the following chapter.

Some of the 12 dialogue moves available are: 'pumping' the student for more information, 'prompting' the student to fill in a missing word, phrase or sentence in a discourse context, providing 'immediate feedback' in the case that a misconception or erroneous answer is detected, 'splicing' correct information as soon as the student produces a contribution that is obviously error ridden, 'hinting' when the student is having problems in answering a question and finally 'summarising' the data taught so far. AutoTutor's delivery of dialogue moves is organised within a 5-step framework which has been identified as unique to normal human tutoring interactions in Graesser et al. (1995):

- Step 1: Tutor asks question (or presents problem)
- Step 2: Learner answers question (or begins to solve problem)
- Step 3: Tutor gives short immediate feedback on the quality of the answer.
- Step 4: Tutor and learner collaboratively improve the quality of the answer.
- Step 5: Tutor assesses learner's understanding of the answer.

Typically, Step 4 is a lengthy multi-turn dialogue in which the tutor and student collaboratively contribute to the explanation that answers the question or solves the problem. It is claimed that this is the Step that promotes active student learning. Empirical studies motivated the decision to eliminate Step 5 form AutoTutor's design (Person et al., 2001). During this step, tutors frequently ask global, comprehension-gauging questions (e.g. "Do you understand?"). Past research indicates that students' answers to these questions tend to be somewhat paradoxical. For example, good students are more likely to say, "No, I don't understand" than poor students.

One hundred hours of naturalistic tutoring sessions where analysed and dissected into discourse and pedagogical strategies. Fuzzy production rules select appropriate topics and tutor dialogue moves form a rich curriculum script. Autotutor is considered to use naturalistic teaching strategies. The curriculum scripts comprise of examples and deep reasoning questions of the type ”Why?”, ”What-if?”, ”How?”, ”What-if-not?”. Learning is accomplished by interacting didactic declarative knowledge with example problems and cases. The system claims to assist the students in actively constructing subjective explanations and elaborations of the material. The tutor's dialogue moves in a collaborative exchange aim in providing effective scaffolding for a student to build such self-explanations without the computer fully knowing what the student knows. Thus there is no analytical student model. Instead the system only keeps a tutorial history log file.

At every stage in the tutoring session, a set of production rules controls selection of a subtopic that is appropriate to the student's needs and the teacher's goals. AutoTutor's dialogue move generator is governed by 15 fuzzy production rules that primarily exploit data provided by the LSA module. Each fuzzy production rule specifies the parameter values in which a particular dialogue move should be initiated. The dialogue move production rules are tuned to the following four parameters:

1. the quality of the learner's Assertions in the preceding turn,

2. the learner's ability level for the topic,

3. the extent to which the topic has been covered,

4. student verbosity.

The first tree parameter values are computed by LSA, whereas the fourth is simply a measure of how much (rather than how well) the student is contributing to the tutoring topic.

It is important to point out that it is not specified how the system reasons about the needs of the student. The subtopics come from a set of instructional materials contained in the curriculum script, developed by experts in education and in the subject domain. The material in the curriculum script of the system covers one macrotopic, for example, the Internet in a computer literacy class. There are 12 topics within each of the 3 macrotopics in the computer literacy curriculum script (36 total). The set of curriculum scripts provides a rich set of responses from which AutoTutor can choose, based on the evaluation of the student's contribution and on the dialogue selection rules. In other words the variety of responses is not dynamic since it is a set of canned text utterances. The tutor cannot respond correctly to an unexpected student contribution and has limited functionalities in promoting student initiative.

In an attempt to improve AutoTutor's performance as a conversational partner, Preson et al. (2000) incorporated the Dialogue Advancer Network (DAN). This new module aimed to solve many turn-taking problems that where observed in the previous version of the system. A schematic representation of the DAN is available in the appendix. The dialogue is categorised in Advancer States that advance the conversation by clarifying who has the floor in the conversation. AutoTutor keeps the floor after an elaboration for example, by articulating a predetermined discourse marker (e.g. "Moving on") and selecting another dialogue move.

A major component of the language analysers is the speech act classifier. This module assigns the student's input into one of five speech act categories: Assertion, WH-question, YES/NO question, Directive, and Short Response. The Assertions are considered more relevant to the present implementation of LSA. Namely, LSA is used to assess the quality of a learner contribution after it has been classified as an Assertion. The developers of AutoTutor have given less focus on the strategies for dealing with the other speech categories. This approach limits the student initiative possibilities. A more detailed discussion on this is included in section 4.3.

In CIRCSIM, discussed in the next subsection, explicit encoding of teaching knowledge in dialogue management rules makes its teaching strategies more strongly linked to the reasoning skills required in learning. AutoTutor has a more shallow approach on teaching reasoning. In AutoTutor's domain knowledge is encoded in its curriculum script. As mentioned earlier there is a curriculum script for each macrotopic to be covered. The script includes didactic descriptions, tutor-posed questions, example problems, figures and diagrams. Each topic in the curriculum script is represented either as a structured set of propositions or as a free text format. It consists of an information delivery item that sets up the common ground between the student and Autotutor, a seed question and a set of subtopics. There are four types of subtopics: (1) simple question/answer, (2) question/answer with didactic content, (3) problem solution, and (4) graphic display question/answer. Notably the information delivery item rarely commits the dialogue to follow on a specific example. In most cases the dialogue is set up around declarative knowledge.

Each subtopic is ranked on sophistication (high, medium, low) and on chronological order (early, middle, late). Some topics may have specific ordering constraints, for example that subtopic A must be covered before subtopic B. Associated with each subtopic is: an ideal complete and correct answer; a set of additional good answers (which grows with experience), ranked good, better, best; a set of bad answers that embody student errors, bugs and misconceptions (which also grows with experience); a set of hints, ranked for low, medium and high-quality student contributions (i.e. a low-ranked hint provides more basic information than a high-ranked hint); a set of questions that the student would be likely to ask with appropriate answers; and a good succinct summary.

Regarding AutoTutor's limitations in the curriculum script approach it is evident that, because the information is pumped from a knowledge base and the utterances and explanations emerge from canned text, its performance is limited from the resources of its library. Although it is claimed that it can update its knowledge base through practice it is not explained how this is done and thus makes the claim less convincing. CIRCSIM (see next subsection) on the other hand is more effective in the sense that it is capable of reasoning on the expert knowledge stored. AutoTutor adopts a declarative representation of knowledge and has no representation of causality between aspects taught, whereas CIRCSIM represents knowledge in the form of variables and causal relationships. Another disadvantage of AutoTutor is that a large corpus of training material is required.

Autotutor's student modelling techniques are deliberately limited. It is assumed that it suffices to keep a log of the tutorial dialogue and calculate the learners ability based on the learner's assertions in all of the previous learner turns in the dialogue. An Assertion quality score is computed for the set of Assertions in any given learner turn. The quality score is computed by a comparison of the supplied answers to the ideal good answers. It is either fortified if the answer is correct or similar to the right answer, or penalised in the case of a wrong answer. Based on this score the good student receives difficult topics whereas the under-achieving student receives easy topics. This approach is closely linked with the structure of the domain knowledge. The curriculum script along with the dialogue history, the quality of the student's contribution and the production rules that define the teaching strategy of the tutor will determine the way the tutorial will progress.

A major disadvantage of the absence of student modelling is that the tutor has no student profile stored and hence each time the student commences a new tutorial session he is considered as complete novice because there is no record of what he has been taught so far. Also there is no functionality that would allow the system to adapt their teaching strategies to the student's particular needs and learning style.

2.3.2 CIRCSIM-Tutor

CIRCSIM-Tutor is an intelligent tutoring system using a natural language interface to tutor medical students on problem-solving in the domain of reflex control of blood pressure. While using the system the student is presented with a predefined procedure and then is asked to predict the qualitative changes in seven core variables at three different chronological stages of the reflex cycle. These predictions are then used as the basis of a tutoring dialogue to remediate any misconception that the student has revealed.

According to Evens et al. (2001), CIRCSIM-Tutor uses natural language for both input and output. It can handle a variety of syntactic constructions and lexical items, including sentence fragments and misspelled words. It aims to generate text that is both pedagogically and linguistically realistic. To that end, transcripts collected from experiments devised to capture tutoring discourse using human-to-human tutoring sessions, on a keyboard-to-keyboard format, were used as a source of information about lexical, syntactic requirements and pedagogical requirements for the machine tutor.

The main components of CIRCSIM-Tutor are the planner, the text generator, the input understander, the student model, the knowledge base, the problem solver and the screen manager. The main loop in CIRCSIM-Tutor asks the student to pick a problem to solve. For each of the three stages of the problem it first calls the problem solver to obtain a set of correct answers, then it collects the student's predictions and sets up a tutoring dialogue for that stage.

The planner is given the student's predictions, plus a student model showing errors that the student has made. It sets up a series of tutoring goals to remedy those errors in a logical sequence. The planner is a hierarchical one and so once it has set up a list o tutoring goals, it calls itself to make a discourse plan for tutoring the top goal on the list, by choosing one of a small number of tutoring strategies. The most common discourse plan for tutoring a single erroneous variable looks like this:

1. Elicit the determinants of the erroneous variable
2. Elicit the actual determinant which is operating now
3. Elicit the relationship between the actual determinant and the erroneous variable
4. Elicit the correct value.

The planner sequences through these steps, invoking the text generator to ask the question, invoking the input understander to fetch the answer, and invoking the student modeller to verify the answer and update the student model.

Spelling correction was judged as essential for the specific domain, after looking at the sentences the students typed in the transcripts of the tutoring sessions (Lee and W., 1998). This was due to the frequency of abbreviations that are used in the taught domain. These were added to the lexicon, along with error forms too short to recognise by standard correction algorithms, like “teh” and “hte” and “fo”. It was also observed that students invented spontaneous abbreviations quite often by stopping typing part of the way through a word. This was handled by reducing error cost for missing letters as the system got closer to the end of a word. All forms of a concept are included in the spelling correction process. The spelling correction algorithm picks the most likely candidate amongst transpositions, elisions and substitutions. A weighting routine that recognises that some errors are more likely than others, is used to increase the speed of of the process.

In Glass (2001) the new improved mechanism of processing language input is discussed in depth. It is made clear that natural language understanding, although advertised as unconstrained, is limited by the usage of closed questions that take short answers. This choice significantly simplifies the implementation of the natural language understanding component. It is claimed that this strategy does not restrict tutoring of the specific domain. This mechanism is capable of categorising the student contribution in a more detailed way (eight categories), instead of merely “right” and “everything else”. It can also recognise a “near miss”, where the student’s response is not what the tutor expected, but it is nevertheless close enough that the tutor can introduce extra dialogue steps to bring the student from the near-miss answer to the desired answer. Another new functionality added is the capability of recognising a true-but-irrelevant answer instead of simply classifying it as wrong.

The natural language understanding mechanism is based on a cascade of finite state machines. Each machine produces an output, which is usually some modification of the input. This design practically applies filters that transform the student contribution to a more “understaentdable” for the tutor form. The input understander can also produce representations for student initiatives that are asking for the definition of a term. Processing in the CIRCSIM-Tutor input understander consists of the following steps in sequence: lexicon lookup, spelling correction, processing by finite state transducers, lookup in concept ontologies, and finally matching to the question. Matching takes place in order to product a representation of an answer. It should be noted that this is accomplished through *ad hoc* code for each type of question. This hard-coded module has been identified as one of the limitations of the system that is planned to be dealt with.

The information in the knowledge base is organised around variables and causal relations. This information is used by the problem solver to produce a solution to the problem presented to the student, and it is used by the generation program to discuss the relevant domain knowledge and problem-solving algorithms with the student. The discourse planner uses the knowledge base to issue hints. It responds to student input such as "I don't understand" to a question, by identifying the variable immediately affecting the variable under discussion, then mentions this variable to the student as a hint, then reissues the question. The knowledge base is also accessed from the student modeller, which used it to evaluate the truth of student input.

It is worth pointing out that the causal links in the knowledge base do not join together to provide a complete solution for all the variables. This approach suffices for the needs of the system since it aims to be able to argue for a correct solution to a particular cardiovascular variable of interest (anatomical reasoning). Nevertheless it limits the systems performance because only one causal explanation is stored in the knowledge base for the change in each variable and thus only one explanation is available in each situation. The availability of more than one explanations would make the tutor's assertions more convincing to the student.

Student modelling is achieved in four different levels: the global assessment (an overall assessment of the student's performance), the procedure-level assessment (an assessment of how the student is performing on this procedure so far), the stage assessment (one for each of the three stages of the current problem), and the local assessment (measured for each variable that has been tutored in this stage). The student model does not store any measure of student unease.

CIRCSIM-Tutor contains a large decision table to determine the order for tutoring the selected concepts. After the sequence is determined, a plan is chosen for tutoring each concept. A plan consists of a single tutoring tactic or a series of tutoring tactics. There are four major types of tutoring tactics, one for each major category of plan:

1. Ask a series of questions

2. Explain (give a declarative sentence)

3. Hint Remind ("Remember that...")

4. Acknowledge (4 possible forms: the forms depend on the number of expected components as an answer to a question and on whether the student answer was correct, partially correct or wrong)

The discourse planner produces a discourse plan that specifies both the content and overall structure of a dialogue session. The basic planning cycle is implemented with an augmented finite state machine. The tutoring plans are kept on a stack. The main factor governing state transitions is which type of student input is received. Other factors used are the type of concept being tutored and whether this is the student's first or second try at answering a question. Hints are generated for the first or second try. After two failures the system will tell the student the right answer.

Based on the turn planning mechanism developed by Yang (2001), each tutoring tactic is first realised as a separate sentence based on a set of goals (e.g. negative feedback, give-answer, acknowledge input etc) determined by the discourse processor, and then a new level of discourse planning is introduced in order to generate the structure of the tutorial turn as an integral whole, and not just a sentence at a time. This makes the dialogue more natural, fluent and coherent and thus more similar to a human tutoring session. In order to keep the dialogue moving, the tutor adopts strategies that allow the system to respond in an appropriate way even if it does not understand the student input.

CIRCSIM-Tutor is the first ITS with a NL interface which has been formally evaluated, but the results of the evaluation are not available yet.

2.3.3 Andes

Andes (Gertner and VanLehn, 2000) is an intelligent tutoring system, currently under development, for classical, newtonian physics (Freedman, 2000). Although still in a not fully integrated phase, it is especially informative to study the goals set for the system to attain and the theory supporting them. Andes belongs in the category of model tracing tutors (MTT). Its major objectives are: to increase the learning of the participating physics students by making more effective use of the time they spend studying examples and solving problems, to advance the understanding of how students learn difficult and sometimes counterintuitive subjects such as physics by studying how different styles of tutoring change students' learning processes and outcomes, and to teach qualitative reasoning in the domain and not a mere algebraic and symbol-pushing approach.

One of Andes primary goals is to be sensitive to student intentions and current status. The system wishes to try and infer the student's plans, goals and intentions in real time so that planning of the tutorial dialogue can take these factors into account. Another challenge, set for the system to meet, is accurate assessment of student knowledge in order to make good pedagogical decisions. Bayesian reasoning is recruited in order to design a student model that not only tracks student's performance, but also has a record of prior probabilities of student behaviour.

The version of Andes which is considered related to our analysis is that of Atlas-Andes. Freedman et al. (2000) state that Atlas is a system that provides a general purpose planning engine and robust input understanding component that can be used to augment any tutoring system with dialogue capabilities. Atlas-Andes is an ITS that stems from the use of Andes as a host to the Atlas system. Motivation for this merger was the fact that Andes was mainly focusing on teaching quantitative problem solving in the physics domain, and a qualitative approach through a natural language interface was considered essential to increase deep learning gains. This merger is implemented through the module of the Conceptual Helper.

In the plain Andes version, the feedback provided was immediate and in the form of red-green feedback in the equations the student typed in. A shallow learning problem was identified, because the student often continued guessing until she performed an action that received positive feedback. In that way the student learnt to perform the right actions for the wrong reasons. In the Atlas-Andes version, Atlas makes use of Andes solution graph corresponding to the current problem, for diagnosing student errors. Each time the student performs a GUI action that is determined to be incorrect, the conceptual helper attempts to match student actions to the closest action in the solution graph that has not been successfully accomplished yet. Each node of the solution graph is associated with one or more Knowledge Construction Dialogues (KDE). This design aims to promote learning through directed lines of reasoning that are encapsulated in the KDEs. If the Conceptual Helper identifies one or more KCDs that the student has not yet seen, it passes the control over to Atlas.

Atlas is comprised of the Atlas Planning Engine (APE), and CARMEL. APE is an integrated planning and execution system at the heart of the Atlas dialogue management system. APE controls a mixed-initiative dialogue between a human user and a host system, where turns in the "conversation" may include graphical actions and/or written text. APE according to Freedman (2000) has full unification and can handle arbitrarily nested discourse constructs, making it more powerful than dialogue managers based on finite-state machines. It is a "just in time" planner specialised for easy construction and rapid generation of hierarchically organised dialogues.

CARMEL is Atlas's input understander. It extracts relevant information from student answers and passes it back to the planner. It is composed of an English syntactic parsing grammar and lexicon; parsing algorithms, semantic interpretation and repair.It also contains a formalism for entering idiomatic and domain specific semantic knowledge. The goal behind CARMEL approach is to achieve the most complete deep analysis possible within practical limits by relaxing constraints only as needed. A spelling corrector has been integrated in the lexical look-up mechanism.

Freedman (2000) puts forward the assertion that classical planning is inappropriate for dialogue generation precisely because it assumes an unchanging world. A more appropriate approach is considered to be that of "reactive planning".Using Andes as a host system for Atlas the benefits acquired include: (1) allowing for reactive planning since it is impossible to account for all possible student responses in a conversation between the ITS and the user; (2)multiple tutoring protocols because human tutors will sometimes change their method of tutoring based on the student answer; (3) multi-turn planning; (4) plan modification and retry; (5) lexical variety which makes the conversation less repetitive.

A comparative evaluation between Atlas-Andes and Andes has been performed in Rose et al. (2001). In this study, students using the dialogue enhanced version performed significantly better on a conceptual post-test than students using the standard version. A full scale evaluation of the system in terms of learning outcomes has not been performed as yet.

ITS	**Dialogue Planner**	**Input Understander**	**Text Generator**	**Domain Kn/dge**	**Feedback**	**Student Model**
CIRCSIM	F.S.M turn planning	keyword matching, near miss, closed Qs	YES	causal rel.	on 2 failures	Yes
ANDES	ATLAS (reactive planning)	CARMEL	Directed lines of reasoning	Solution Graph	Conceptual Helper	Bayesian network of concepts
AUTO TUTOR	LSA	Open Qs	canned text	LSA	depending on LSA score	No

Table 2.1: Comparison of Tutors

It is important to note that the latest version of CIRCSIM-Tutor (version 3), also adopts the use of the APE application. This choice verifies the need for a more detailed discourse planner, one that has separate modules for planning the curriculum, the turns, and the discourse respectively.

AutoTutor does not support a student model, as it will be made clear in the next section, and thus cannot be considered as a Model Tracing Tutor. This means that the analysis of reactive planning must be limited by the restrictions imposed by AutoTutor's implementation. Nevertheless, the arguments put forward in the Andes-Atlas tutor for the need of a mixed initiative dialogue and a deeper comprehension of the students intentions are indicative of the changes that could be made on the current AutoTutor version. It remains to be seen, in the analysis in the third chapter of the thesis, whether AutoTutor can truly address either deeper reasoning or mixed initiative dialogue issues.

2.3.4 Grounds for Comparison

Having analysed the basic features of the three tutors, it is now time to set the grounds for a comparison. A comparison that will dictate the specifications such a system must meet. Table 2.1 summarises the previous discussion and helps spot the major differences in the three ITS approaches.

Based on the events, put forward by Core et al. (2000), that a dialogue tutor must effectively handle, the literature indicates that AutoTutor's design accounts for two out of three. The system can account for failure of the student to respond to a question by providing immediate feedback and hinting or prompting her towards the right answer. The design also allows handling of student interruptions in the dialogue. Regarding the third event, being capable of revising his tactics depending on the student needs, the system overall fails to do so since his teaching strategy is hard coded in the curriculum script. The extent to which he succeeds in dealing with the other two events will be empirically tested in the following chapter.

AutoTutor seems to lack in certain areas in comparison to the two other tutors. The system's dialogue management is restricted by the simplicity of its dialogue moves and the inflexibility of the canned text utterances. An obvious proposal in architecture modification would be to integrate Atlas in a new version of AutoTutor. If it is indeed as domain- and task- dependent as advertised then it would be the optimal solution to use it to replace the existing dialogue management module of AutoTutor.

A student model is also a functionality that appears to be necessary when looking at the approach of the other two tutors. Modelling the students performance, as shown in the case of CIRCIM-Tutor and Atlas-Andes, can help tailor the dialogue in the students particular needs and set the requirements for a multi-teaching strategy approach. Mixed initiative in the discourse process is also proven as essential. Thus one point that demands further research is the flexibility AutoTutor allows in the dialogue and ways of improving it.

The idea of letting the student know what type of answer the tutor is expecting (an idea implemented in CIRCSIM-Tutor) is one that facilitates the discourse and renders dialogue more comprehensive for the student. Also, providing the student with the choice of actively intervening in the dialogue and taking the initiative to skip elaborations that she feels are not needed, can speed up the process of the session and thus keep the student's interest at a high level.

Nevertheless, as it will become apparent in Chapter 4, while taking a closer look at the characteristics of the current implementation of the AutoTutor framework, the underlying architecture is much more restrictive than advertised in the relevant literature review. Through the procedure of porting AutoTutor in a new domain, we aim to reveal the actual behaviour and potentials of the system and identify feasible modifications on the current architecture. This analysis will be guided from the goals set in the introduction and particular focus will be given on the framework's domain independence, on its deep reasoning capabilities, on the quality of the interface design, on the dialogue management and finally on issues of student modelling.

2.4 Summary

This chapter described tree dialogue based intelligent tutors on the level of their design and architectural approach. Theoretical conclusions were drawn regarding the specifications such a type of tutor must meet and the correspondence of those specifications to the system at hand, AutoTutor. This theoretical analysis needs to be grounded into actual facts. In the next chapter we will investigate the validity of the claims made regarding AutoTutor's characteristics.

Chapter 3

Investigation of AutoTutor in Action

A pilot study was designed in order to collect feedback on the *actual* characteristics of the tutor from the user's end. It is to be made clear that a formal evaluation of the system is beyond the goals of this thesis. We are not interested in the learning gains of the tutoring dialogue. What concerns us is to investigate and evaluate the validity of the claims made regarding the systems performance in terms of:

- Quality of Interface Design
- Dialogue Management
- Student Model
- Deep Reasoning

Eight subjects were recruited in this study, amongst which four were PhD students in various disciplines (psychology, interface design, agent interface, cognitive science) and four were MSc students in cognitive science. Their feedback proved valuable both from an expert's and a user's perspective.

3.1 Pilot Study

The subjects where asked to interact with the tutor for about 15 to 20 minutes, until they have successfully covered three topics of the Hardware macro-topic and one topic of the Internet macro-topic. Then they were asked to fill in a questionnaire that inquired their overall opinion about the tutor and specific information regarding the aspects of interest listed above. The questionnaire used can be found in the appendix.

The overall impression of the tutor was recorded mostly as “not very good” (one subject found it good, six found it not very good and one put it down as bad). This is a finding that justifies our interest in further investigating the tutors actual performance. The detailed results of the pilot study will be portrayed throughout the remainder of this section according to the issue they relate to [1].

3.2 Interface

Some of the basic claims of AutoTutor’s founders were strongly contradicted by the results of this pilot study. It is speculated that the agent interface yields better impressions than a conventional text to text application, without any empirical data to back this intuitive claim. The only evaluation performed on the interface so far, is that in Link et al. (2001).In this study, the factors that influence the perception of feedback delivered by an agent are investigated. The evaluation was derived by user ratings on how positive or negative the agent’s feedback seems to be, in conditions where the speech parameters and facial expressions of the talking head were manipulated. Their results support the claim that verbal and nonverbal cues are additive. Specifically, participants relied on both linguistic expressions and the mouth curve.

There are several weaknesses in this evaluation approach that are worth mentioning. Firstly, only a small number of features of each of the two modalities was tested. The values of these features were chosen arbitrarily. Moreover the assumption that the agent should be designed to mimic a human tutor in his facial expressions and gestures is not supported by empirical evidence.In a analysis by Dehn (2000), on the impact of animated interface agents it is made clear that such an assumption has yet to be validated.

In the same study, a systematic review of empirical evidence on agent interfaces is conducted. The conclusions report that although an agent character is largely perceived

[1] The results of the study concerning learning outcomes where omited because they are beyond the purpose of this evaluation.

to be more entertaining, other dimensions, such as utility, likeability and comfortability are moderated by the kind of animation used and the domain in which the interaction is set. It is clearly stated that present studies do not suffice to enable us to make clear predictions as to what type of animations employed in what type of domain will result in positive attitudes towards the system. Another important point of this study is that all attempts on evaluation so far in this field, concerned limited short-term interactions with the systems. It is unknown if longer exposure to an animated agent might sustain or wear of the positive effect of entertainment ratings. It is interesting to see how these conclusions are verified from the feedback collected in the pilot study.

On the interface related questions, the feedback collected was mostly negative. The prevalent complaint in all user's suggestions was the poor quality of the agent's voice articulation.

Having to listen so hard made me lose parts of the information, reported one of the users. This feature had immediate impact in the focus of attention of the user. User's were distracted from the actual content of the tutorial dialogue, by the extra effort they had to put, in understanding the tutor's utterances. This is the reason why native English speakers where preferred as subjects, so that there would be no confounding effects by their level of mastery in English. Figure 3.1 depicts the overall impression of the tutor as recorded in the pilot study. Figure 3.2 shows that it is not obvious to the users that indeed an agent interface is better than a conventional one. This finding is consistent to the arguments put forward earlier on in this section.

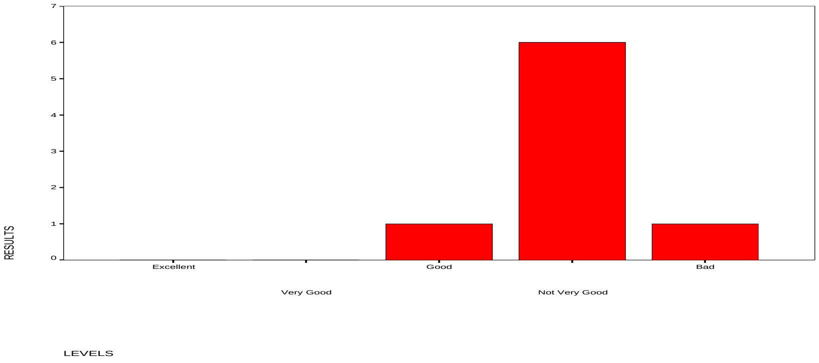


Figure 3.1: How would you characterise your overall impression of the tutor?

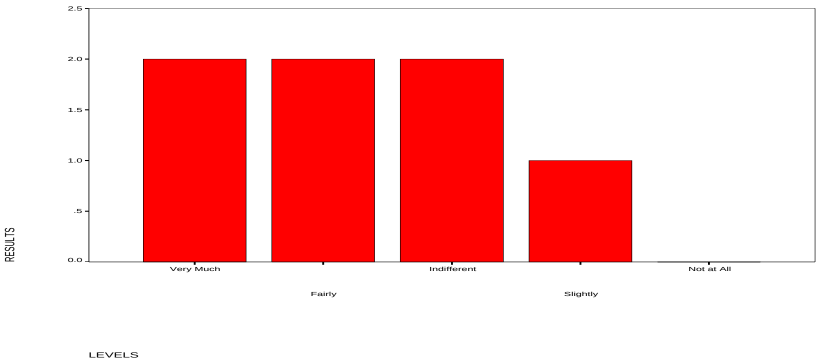


Figure 3.2: Would you say the talking head made a difference over a conventional user interface?

Figure 3.3 shows the users' discontent regarding the agents voice articulation. Evidence of their difficulty in understanding the tutors utterances is the fact that they very often requested the tutor to repeat an utterance multiple times. The log files of the tutoring session reveal that on average ten requests for repetition where made in sessions of approximately fifteen minutes duration. This number is considerably high.

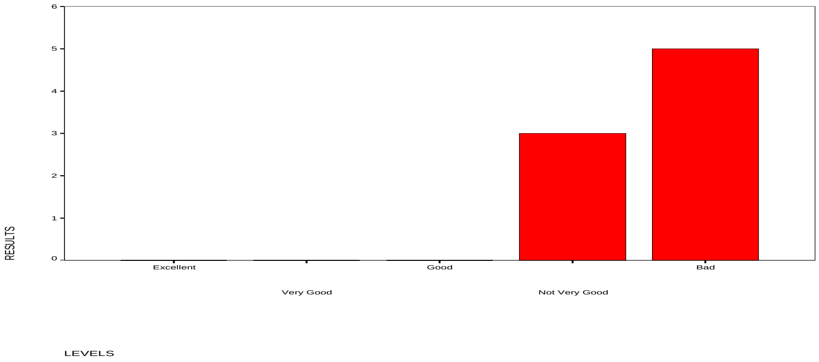


Figure 3.3: Did you find the tutor's voice articulation:

The pilot study showed that the users' perception of the agent's feedback was not clear. Most of them reported that they could not figure out if the agent was pleased with their contribution or if he thought it was incorrect. This increased the levels of their unease in the interaction with the system, since they had no knowledge of the progress of the tutorial. This dissatisfaction can be clearly observed both in the remarks the users put down at designated free comments area of the questionnaire. It was also pointed out that the prosody of the speech was unnatural, making it difficult to distinguish when the tutor was actually posing a question.

A novice user in computer literacy stated that

I wasn't sure if answers were sarcastic, does "right" mean "you know nothing"? It made me feel, Oh my god I'm completely stupid!. This point is particularly important since we would expect the tutor to have a more possitive and motivation impact on novices in the domain taught.

It is interesting to report some observations made of the users interacting with the system. The users took the agents reactions seriously and were pleased when his feedback was evidently positive. One of the users said

Rather do this than read a book. They often made gestures back at him. According to Dehn (2000) the extent to which the users perceive the agent as believable has immediate influence on their expectations from him. They often felt frustrated when correct contributions where dismissed with an ambiguous feedback gesture or even a negative one.

3.3 Dialogue Management

We aim to investigate the quality of the dialogue in terms of:

1. Coherence,

2. Clarity of the tutor's questions,

3. Interactivity,

4. Impression that tutor understood the contributions.

The coherence of the dialogue was rated intermediately (Figure 3.4). A problem identified was that many questions were too open-ended, resulting to a sense of confusion from the user's side.

Users were puzzled by some of the tutors questions, especially the prompts, were it was not clear to them what the tutor was expecting. Their understanding of the tutor

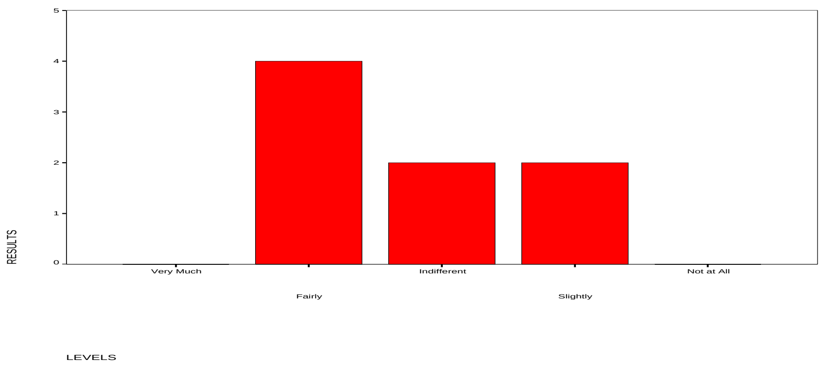


Figure 3.4: Did you find the dialogue coherent?

was hampered by the poor voice articulation mentioned earlier and thus the intermediate ratings observed in Figure 3.5.

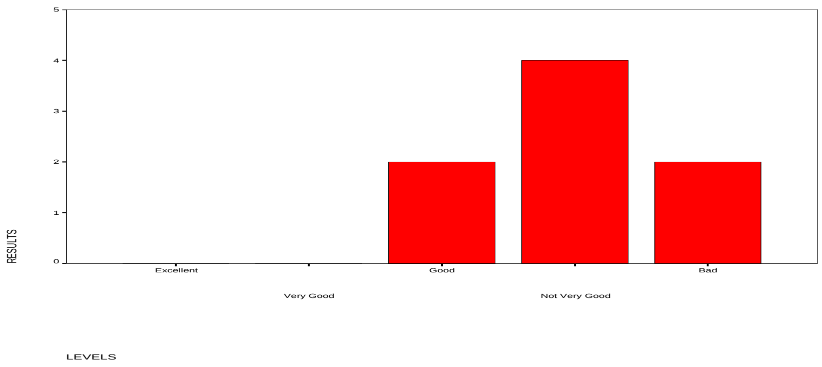


Figure 3.5: Did you find the clarity of the questions:

Moreover the dialogue is structured in such a way that student initiative is very limited. [2] This leads to one sided dialogues, in the sense that the tutor maintains the control of the tutorial flow and thus the student is not free to postulate her speculations about the topic at hand. The student is restricted to answering questions and has no possibility of posing her own, apart from mere definition inquiries.

Regarding the limited sharing of control over the tutoring process, the users often felt frustrated towards the tutor because they felt there was no actual interaction. They felt limited to the tutor's instruction plan. Figures 3.6 and 3.7 along with the additional

[2] As student initiative we mean any student contribution to the dialogue that is not an answer to a question asked by the tutor.

comments, indicate that the session was not perceived to be interactive and that more user initiative was expected in order to conduct a truly collaborative dialogue.

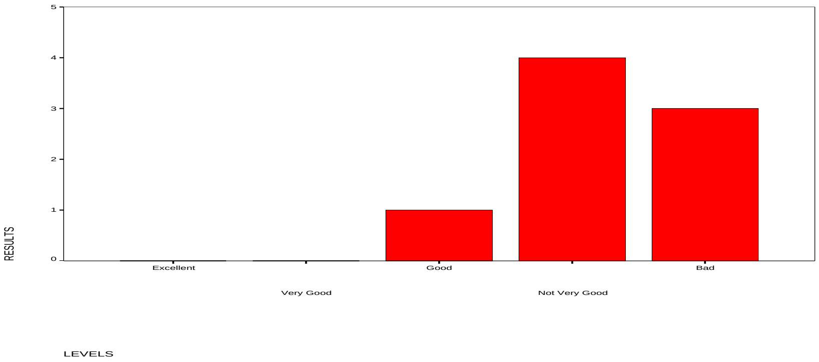


Figure 3.6: How would you characterise tour interaction with the tutor?

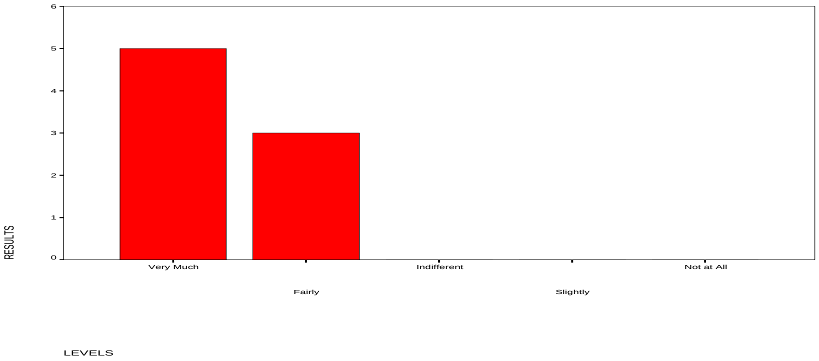


Figure 3.7: Do you think that more student initiative functionalities could improve the dialogue?

Users reacted positively to the tutor's capability for answering to WH-questions. A problem observed in this event is that the question posed by the tutor is never replied to when a WH-question is handled. This is a weakness that is apparent in the whole of the dialogue management. As the tutor formulates a question, he then expects the student to respond to his specific request. If the student types in a contribution long before the tutor finishes posing the question the tutor will still believe that the student input was a response to his question, unless it was a WH-question (Figure 3.8).

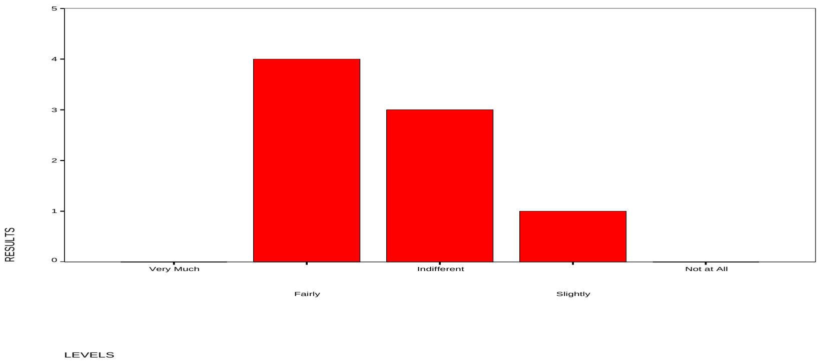


Figure 3.8: How much did you feel the tutor understood your contributions?

3.4 Student Model

It has been stated earlier that AutoTutor's approach to student modelling is deliberately very limited. This choice has serious implications in the flow of the tutorial dialogue. The tutor is often repetitive because he has no idea of the knowledge of the student. In the pilot study it was observed very often that the student knew the correct answer to a question, LSA did not manage to match the correct answer with its pre-stored data and thus the tutor fell into repetition in a way frustrating the user. This can be seen in the following bar-gram (Figure 3.9):

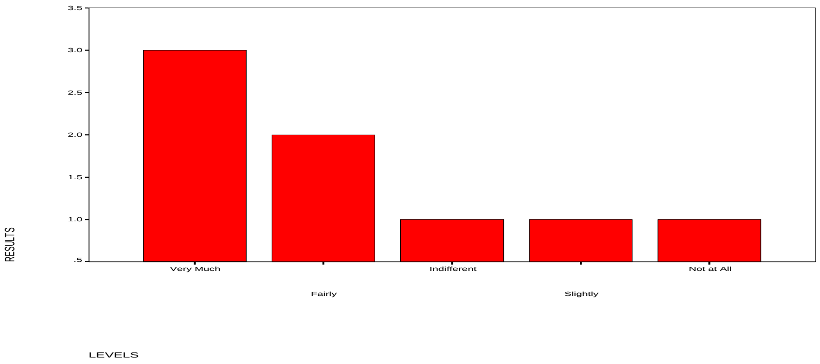


Figure 3.9: Did you at any time feel frustrated towards the tutor?

The absence of a user model also has serious implications on the usability of the system. The tutor stores no profile of the user. This makes it pointless for someone to attempt to use the system more than once because he will be forced to go through the same topics covered in his initial session with the tutor.

3.5 Deep Reasoning

AutoTutor literature postulates that the tutor supports deep reasoning in the tutorial dialogue. This claim is difficult to investigate since the domain taught and the instruction approach adopted do not provide the ground for deep reasoning dialogue patterns. The domain of computer literacy does not permit reasoning mechanisms to be activated since it constrains the dialogue on a descriptive and definition oriented level. Concepts taught are mostly entities such as hardware components and there are no causal relationships between them that could trigger in-depth conversations.

It should be noted that, as expected from the fact that LSA does not take any syntactic information into account, that there exist erroneous contributions that are evaluated as correct. Although it is highly unlikely that a user would type in that “a computer is in RAM”, in the case that she did, the system would respond with positive feedback. This is an issue that needs further investigation in a domain where such a change in word order would be a likely misconception.

3.6 Summary

After the theoretical evaluation of AutoTutor’s design and architecture in comparison with two ITS of the same family, in Chapter 2, this chapter explored the actual characteristics of the system and juxtaposed them to the claims made regarding its performance. The interface approach, the dialogue management, the student modelling techniques and the deep reasoning mechanisms were analysed through the observation of users interacting with the system and their valuable feedback. This analysis provided a more realistic estimate of the system’s performance in terms of the features of interest. The following chapter will take a software engineering approach of analysing these features, through an attempt of porting AutoTutor in a new domain.

Chapter 4

Porting as a Means of Architecture Evaluation

4.1 Motivations for Porting

Porting the AutoTutor framework has been identified as a means of evaluation, since it will require deep investigation and understanding of the code. Only through an holistic approach on the implementation's features and a hands-on attempt on verifying claims on the framework's performance can such an evaluation be realised.

Through the porting procedure, issues of domain independence, portability and extensibility will be addressed on both a software engineering and an ITS designing perspective.

As pointed out earlier, the depth of AutoTutor's conversations is limited by its subject. Computer Literacy attempts only to familiarise students with the basic concepts of computers, and does not get into any deep issues. Thus, many of AutoTutor's questions have a short-answer feel; the ideal answers can be summed up in one or two words. A more complicated domain would set the grounds for testing if indeed the system can support deeper reasoning in the discourse, as claimed by its developers.

4.2 Domain

Although our reasons for porting are not dealing with issues on learning outcomes, a careful selection of the new domain was considered essential in terms of future extensibility of the system developed.

Motivations for a Research Methods oriented domain are explained in the following section. Our main concern was to focus on the procedure of porting the tutor in a new domain for purposes of evaluating the framework, as outlined in the thesis goals. In order to do so, a domain that supports deeper and qualitative reasoning had to be identified. The new tutor will be called RMT, which stands for Research Methods Tutor.

The domain chosen to be taught is that of Experimental Design in Behavioural Research Methods. A possible future full scale implementation of the tutor would aim to teach first year college students in either psychology or cognitive science, the fundamental concepts of True Experimental Design through a tutorial dialogue on specific experimental design examples. Prior preliminary knowledge of the domain by the students will be assumed.

4.2.1 Topic Selection

Motivated by the finding that most examples in naturalistic one-to-one tutoring dialogues originate from text books (Person, 1994) and having already decided on a research methods related domain, a specific topic selection from the Cozby (1989) text book was decided.

The topic selection was based on the existing studies of human-to-human tutoring in research methods conducted by Person (1994). The Tutoring Research Corpus of the Institute of Intelligent Systems at the University of Memphis, was collected from upper-division college students who were enrolled in a course on research methods in psychology. According to Cohen and Manion (1989) topics which involve quantitative skills lead to more positive outcomes than topics which focus on nonquantitiative skills (e.g. creative writing).

The corpus is considered representative of the tutors and students normal tutoring environment. Tutors are normally older students, paraprofessionals, and adult volunteers who have not been extensively trained in tutoring techniques. Students participating in the scheme are of college-level with all levels of achievement included, rather than being restricted to students who are having difficulty in the course. The course instructor selected six topics that are normally troublesome for the students. Each topic had related subtopics that were to be covered in the tutoring session. The topics and subtopics are specified below:

- Variables: operational definitions, types of scales, values of variables.
- Graphics: frequency distributions, plotting means, histograms.
- Statistics: decision matrix, Type I and II errors, t-tests, probabilities.
- Hypothesis to Design: formulating a hypothesis, practical constraints, control groups, design, statistical analyses
- Factorial Designs: independent variables, dependent variables, statistics, main effects, cells, interactions.
- Interactions: independent variables, main effects, types of interactions, statistical significance.

The students were exposed to the material on two occasions prior to their participation in the tutoring session. First, each topic was covered in a lecture by the instructor before that topic was covered in the tutoring session. Second, each student was required to read specific pages in a research methods text (Cozby, 1989). The students therefore had multiple chances to learn the material.

After a detailed study of the topics covered in the transcripts, an keeping in mind the remarks made regarding the value of grounding one-to-one instruction to examples, the topic selection for the new domain was derived from the fifth chapter of Cozby (1989) on True Experimental Design. Some of the examples discussed in this chapter where used as material in the “Hypothesis to Design” and “Variables” topics. Using the transcripts of the related examples, in conjunction with the actual text form the Cozby (1989) book, four example based topics where selected as teaching material for the

Topic	Terms Introduced	Terms Previously Taught
Crowding	Independent Variable Dependent Variable Confounding Cause-Effect Relationships	–
Motivation	Random Allocation Control Group Pretest/Post-test Equivalent Groups	Independent Variable Dependent Variable Confounding Cause-Effect Relationships
Schedule	Threats to Validity Non-equivalent Control groups Demand Characteristics	Independent Variable Dependent Variable Confounding Cause-Effect Relationships Random Allocation Control Group Pretest/Post-test Equivalent Groups
Smoking	Mortality Sample Size Internal Validity	Independent Variable Dependent Variable Confounding Cause-Effect Relationships Random Allocation Control Group Pretest/Post-test Equivalent Groups Validity Non-equivalent Control Groups Demand Characteristics

Table 4.1: Topics Table

new domain. Each topic has a title inspired by the example situation to be discussed, and a list of fundamental concepts that will be focused on during the teaching dialogue (Table 4.1).

The four topics will be covered by the tutor in a serial manner and thus the structure of the taught terms is aggregates as each example topic is covered. It is assumed that the users of the system will already be familiar with the basic concepts of experimental design.

A detailed version of the examples and a prototype tutorial dialogues on them can be found in the Appendix C. It should be noted that these dialogues had to be significantly modified in order to correspond to AutoTutor's dialogue moves.

4.3 LSA and Corpus

Latent semantic analysis (LSA) is a major component of the mechanism that evaluates the quality of student contributions in the tutorial dialogue. In a study by Wiemer-Hastings et al. (1999) LSA's evaluations of college students' answers to deep reasoning questions are found to be equivalent to the evaluations provided by intermediate experts of computer literacy, but not as high as more accomplished experts in computer science. LSA is capable of discriminating different classes of student ability (good, vague, erroneous, versus mute students) and in tracking the quality of contributions in the tutorial dialogue.

LSA is a corpus-based, statistical mechanism. It was originally developed for the task of information retrieval: searching a large database of texts for a small number of texts which satisfy a query (Wiemer-Hastings et al., 1999). The training of LSA starts with a corpus separated into units which are called documents or texts. For the AutoTutor corpus, the curriculum script was used, with each item as a separate text for training purposes. The corpus also included a large amount of additional information from textbooks and articles about computer literacy. Each paragraph of this additional information constituted a text. The paragraph is said to be in general , a good level of granularity for LSA analysis because a paragraph tends to hold a well-developed, coherent idea.

LSA computes a co-occurrence matrix of terms and texts. A “term” for LSA is any word that occurs in more than one text. It compresses a large corpus of texts into a space of 100 to 500 dimensions. The K-dimensional space is used when evaluating the relevance or similarity between any two bags of words, X and Y. The relevance or similarity value varies from 0 to 1; a geometric cosine is used to evaluate the match between the k-dimensional vector for one bag of words and the vector for the other bag of words. In AutoTutor one bag of words is the set of Assertions within a turn. The other bag of words is the content of the curriculum script associated with a particular topic, i.e., good answer aspects and bad answers (Graesser, 2000). AutoTutor calculates a general goodness and badness rating by comparing the student contribution with the set of good and bad answers in the curriculum script for the current topic. More importantly, it compares the student response to the particular good answers that cover the aspects of the ideal answer. Two measures are calculated for this comparison (Wiemer-Hastings et al., 1999):

- **Completeness:** the percentage of the aspects of the ideal answer for the current topic which "match" the student response (A "match" is defined as a cosine between the response vector and the text vector above a critical threshold).
- **Compatibility:** the percentage of the student response (broken down into speech acts) that "match" some aspect of the ideal answer.

In the same study, it was shown that a threshold of .55 with a 200 dimensional space were the settings that correlated the highest with the average ratings of four human raters (two accomplished and two intermediate experts). Thus these were the settings chosen for AutoTutor. It was concluded that the LSA space of AutoTutor exhibits the performance of an intermediate expert, but not an accomplished expert. This was noted as satisfactory, since AutoTutor aims to simulate an untrained human tutor.

In the development of the Research Methods Tutor, the same values for dimension and threshold were adopted. The documents originated from seven text books on research methods and from articles and tutorials published on the Internet. As explained earlier, the domain for the new tutor was restricted from General Research Methods in Behavioural Sciences, to the subset of True Experimental Design. Thus only the relevant chapters form each book were scanned. This choice was supported by Wiemer-Hastings et al. (1999) finding that more of the right kind of text is better for LSA. it also made the collection of the corpus a very time-consuming and tedious procedure since only one or two chapters in each text book were deemed relevant to the desired domain.

4.3.1 Corpus Size

The corpus collected for the computer literacy domain was 2.3 MB of documents. A series of tests were performed on the amount of corpus and the balance between specific and general text (Wiemer-Hastings et al., 1999). As expected, LSA's performance with the entire corpus was best, both in terms of the maximum correlation with the human raters and in terms of the width of the threshold value range in which it performs well. One surprising result was the negligible difference between the 1/3 and 2/3 corpora. It was observed that there is not a linear relation between the amount of text and the performance of LSA. Another surprising finding was the relatively high performance of the corpus without any of the supplemental items, that is, with the curriculum script items alone.

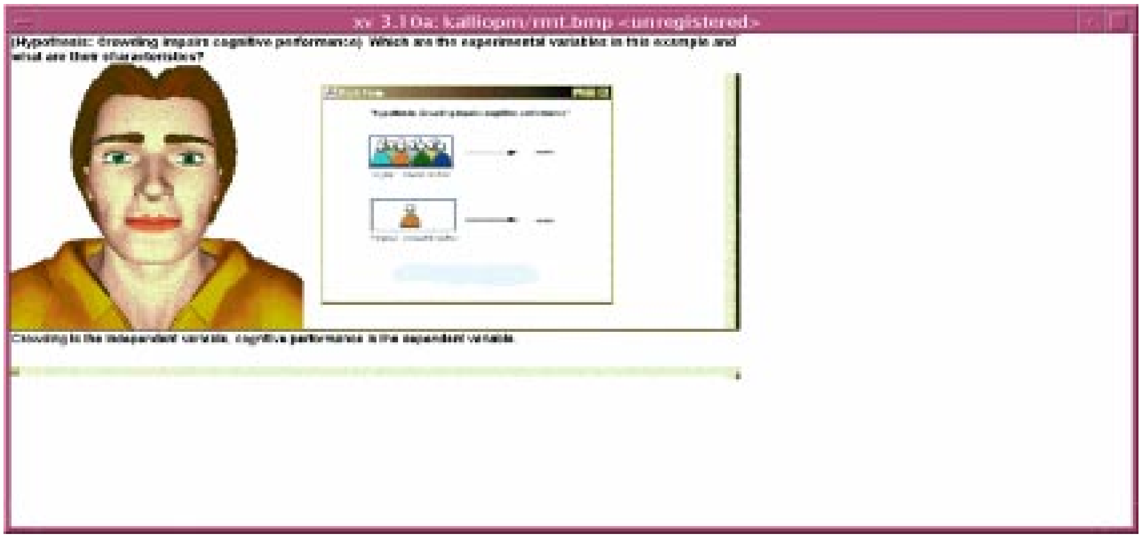
The fact that there is very little difference in the performance of LSA between the 1/3 and 2/3 of the corpus, is a finding that will be used as a supportive argument for the corpus size collected for the Research Methods Tutor. The size of the corpus that was finally obtained on true experimental design is 750Kb. This renders it close to a third of the AutoTutor corpus. Since the whole procedure was extremely time consuming and the performance of the system was not expected to improve in the case the corpus size was doubled, that size was accepted as optimal for the purpose of this thesis.

4.4 Development of Curriculum Script

The curriculum script is the module that organises the topics and the content of the tutorial dialogue. In the case of RMT, since the overall implementation did not aim to be considered as a full version of the tutor, the four topics in Table 5.1 are members of the macrotopic of Experimental Design. AutoTutor provides three levels of difficulty (easy, medium, difficult). In RMT the curriculum script developed included only the easy level since its short term goal was to test the overall behaviour of the framework in the new domain.

The format of the curriculum script is based on tokens. Different tokens precede each type of dialogue move or topic heading. The authoring of the script is quite a time consuming procedure since appart from the tokens the commands for the agent gestures have to be enbedded as well. The topics have a Graphic Display + Question + Answer format. The dialogue is initialised by the presentation of the example of a design on which the tutor and student are going to collaboratively work on improving. An image accompanies each subtopic of the topic covered in order to clarify the conceptual relationships that are being discussed. To make this more clear we will take a closer look at the curriculum script developed for the crowding example. A full scale version of the script is available in the Appendix D.

Initially the student is presented with the hypothesis that “Crowding impairs cognitive performance” and a preliminary experimental design on testing it. A multi-turn dialogue will follow, aiming in putting across the fundamental concepts of the topic taught. An image depicting the current experimental design is used to facilitate understanding. This is shown in Figure 4.1.



(Hypothesis: Crowding impairs cognitive performance) Which are the experimental variables in this example and what are their characteristics?

Crowding is the independent variable, cognitive performance is the dependent variable.

Figure 4.1: AutoTutor on Experimental Design

Particular attention was given in constructing dialogue moves that promote deeper reasoning in the extent possible due to the limited available dialogue moves. The formulation of the questions was done by adopting the dialogue structure of the tutoring transcripts available, to the form of hints, prompts, elaborations and splices. A major restriction in this approach was the fact that word order is not taken into account from the language understanding component. So, the design of the questions had to be done in such a way that word order was not of significance. This proved quite limitating since deep reasoning is achieved only through the discussion of causal relations were the direction of causality, and thus the word order in the phrase, does matter.

4.5 Lessons Learnt and Problems Identified

During the procedure of software porting, unexpected difficulties were encountered that are not mentioned in the claims of the AutoTutor's framework capabilities. Overall the task's execution was particularly hindered by the fact that the code contains minimal documentation. The seven modules of the framework are implemented in Java in a package form of twenty two directories. Due to the approach adopted, that uses diverse programming technologies, the code demonstrates few of the advantages of object oriented programming, such as object abstraction. The notion of interfaces and event handlers is prevalent in the code structure, which in a sense makes it modular. Nevertheless the approach becomes problematic when there is a large interdependency between the modules and the relationships between classes become intricate and opaque.

The domain independence of the framework lies in the idea of using a separate *data* directory in which all the domain dependent parameters are stored, along with parameters of general purpose for the system's settings. Unfortunately this approach was not carried out consistently, leaving domain specific information hard coded in the core of the program. This fact, along with the spartan documentation drives us to the conclusion that issues of portability and domain independence are not realistically documented in the claims of the systems developers.

Also, the approach for storing the parameters of the systems in file names was found to be cumbersome. There is a significant dependency on file names making the modification of parameters and the management of the settings very difficult.

Collecting sufficient corpus alone is an extremely time consuming procedure that has to be followed for any new domain. Along with the problem of authoring a curriculum script in a token-format that seems obsolete in comparison to the new technologies adopted in the overall implementation, it is clear that the system is not as portable as "advertised".

A primary limitation of the system which is unavoidably reflected in all its modules is the knowledge representation approach. The use of LSA shows satisfactory effects in "understanding" student contributions. Nevertheless, the fact that this understanding is local and limited in the sense that no more information can be extracted from LSA apart from a similarity metric to pre-stored information is problematic. While concepts and causal relationships have no means of being represented, it is impossible to escape the inflexibility of canned text, of poor user modelling and of cumbersome dialogue management.

During the porting process it became apparent that the ignorance of word order in a domain that supports deeper reasoning is severely constraining. In a discussion on experimental design a phrase like “the dependent variable has an effect on the independent variable” is incorrect. Unfortunately although the system will have the correct phrase pre-stored, i.e. “the independent variable has an effect on the independent variable” in the case that the inverted order phrase above is typed it will fail to recognise the misconception and moreover will positively acknowledge an erroneous input.

4.6 Summary

This chapter describes the process of porting AutoTutor in a new domain that supports deep reasoning. Motivations for such an approach are outlined, along with argumentation supporting the choices made. Through this attempt various software engineering issues are raised regarding the architecture and design of the framework. A mapping of the weaknesses identified in the previous chapter with their corresponding software implementation helps identify the actual potentials of the system. The following chapter will address the weaknesses identified and propose feasible solutions.

Chapter 5

Feasible Suggestions for Improvement

The next step in this constructive evaluation of the AutoTutor framework is to propose feasible solutions and modifications. It is important to emphasise on the finding that the current architecture of AutoTutor does not support the modifications we would ideally propose. The current approach is heavily restricted and although its performance is judged as satisfactory up to a point, it is majorly impaired by the fact that serious modifications, such as adding a user model or improving knowledge representation demand re-engineering in the core of the design. Nevertheless there are some improvements that are feasible and could significantly alter the overall performance of the system.

5.1 Structured LSA

LSA's is based solely on the "semantic similarity" of texts, taking no account of syntax. A new approach to this statistical method, put forward by Wiemer-Hastings (2000), introduces the idea of incorporating some syntactic information in LSA, instead of the "bag-of-words" approach that simply adds together term vectors to make a vector for a text. Student answers can be pre-processed by first performing a syntactic segmentation of the sentences. This allows grouping of words which belong together into "components", and attributes a pseudo-semantic role of the components as derived from syntactic argument structure. Then anaphora resolution algorithms are applied. In cases of conjunction phrases, the arguments of the phrases are distributed into to as many complete phrases as their adjuncts. Alternatives on ways to calculate the overall similarity between propositions based on the similarities of the components are still under investigation.

This new approach is still in a pioneering phase. More data needs to be obtained in order to estimate its effectiveness. The addition of syntactic information in the language understanding component could significantly improve AutoTutor's capabilities. The representation of causal relationships will be possible. The knowledge of direction of causality between two concepts could amend AutoTutor's current deficit of acknowledging false contributions as true when words match a correct phrase but are in inverted order and thus have a different meaning.

5.2 Student Model

The incorporation of Structured LSA could also improve the student modelling capabilities of the system. If instead of mere terms there is a representation of concepts extracted from the nouns in the contributions, then an implementation of a Bayesian network of concepts as in the case of the CIRCSIM-Tutor could:

- solve the problems of repetitiveness in the dialogue,
- provide an analytical mechanism of assessing weaknesses and misconceptions of the student,
- allow for multiple teaching strategies.

It is important to point out that such an approach would need serious changes in terms of software engineering. Incorporating Structured LSA is feasible in the sense that the LSA module is independent. Taking advantage of all the possibilities it can offer is a sensitive issue, since it is a trade-off. If changes make the code even more intricate it is a good idea to return to the drawing board and redesign the system from scratch, keeping in mind the lessons learnt.

5.3 Dialogue Management

On a more short term basis, a concrete solution on improving the dialogue management can be put forward. Inspired by the higher level dialogue planning implemented by Yang (2001) in the CIRCSIM-Tutor, a new module can be implemented on top of the existing dialogue planning mechanism. The number of dialogue moves can be kept the same, a choice that would facilitate the software programming required. The dialogue moves can be enhanced with “linking” information. Specific hints can be associated with specific prompts or elaborations that deal with similar concepts. The new module can take advantage of this linking information and be able to dynamically plan the tutorial dialogue.

The new module can also provide more student initiative opportunities. The student should be able to intervene in the dialogue when she feels a concept has been adequately covered, in order to avoid user frustration that was observed in the pilot study. This would easily be implemented by adding a menu option in the interface that would fire a request to skip to the next topic.

The addition of a spell checking module is also an attainable goal in the current implementation. Since LSA works better with lengthier contributions the likelihood of spelling mistakes is increased. A spell check applied on the input before it is processed by the language understanding component could increase its effectiveness. Jargon terms of the domain taught in can also be added in the lexicon to improve understanding. The spell checking module will then be capable to calculate the closest match of the student’s misspelled input to its pre-stored information.

On a long term basis, if a more detailed student model is constructed, then it is possible to use the ATLAS dialogue planning engine to handle discourse management. This would allow a variety of teaching strategies and a dialogue tailored to a students specific needs. It would also allow the system to effectively account for unexpected user input. By using AutoTutor as a host system for Atlas, more structure will be added in the tutorial dialogue and the planning will be executed dynamically, rendering the system more capable in reacting to various student behaviours.

5.4 Interface

An obvious improvement that is strongly put forward by the results of the pilot study, is the incorporation of a new speech synthesiser, one in which the voice articulation will be clearer. A solution would be the integration of the the Festival Speech Synthesis System, developed at the Centre for Speech Technology Research at the University of Edinburgh. Issues of compatibility would arise since the use of either the Microsoft Agent or the Sizzle Agent in the AutoTutor framework commit the system to the use of their own speech synthesis software. Still it is an issue that calls for immediate action since the poor voice articulations significantly hampers the dialogue. The student is distracted from the tutoring session in his attempt to understand what the agent is saying. A more short term solution to the problem would be the addition of text bubbles. This feature could be activated every time the user requests for the tutor to repeat an utterance. In that way the user will be able to focus more on the actual tutorial context and not in trying to decode the tutors speech.

As pointed out in the corresponding section in Chapter 3, there remains a lot of research to be conducted regarding the interface module. The choice of an animated agent has yet to be proven effective in the domain of ITSs. Furthermore, a systematic review and evaluation of the features of the agent is necessary. This would comprise of a test of the agent interface against a text-to-text implementation of the system. If the agent interface is indeed proven as a more effective then the next step would be to improve the agent's gestures, and facial/vocal expressions based on analytical studies of their appropriateness. A test on the extent to which agent gestures should be deployed in favour of user comfortability is important. As reported by Dehn (2000) it often is the case that the more realistic the agent character the less comfortable the user feels. After amending the agent's current weaknesses, an investigation on long term exposure to the system will produce tangible evidence for its appropriateness as an interface for a tutoring system.

5.5 Portability Issues

It is necessary to address portability issues in the current framework. The development of an Authoring Toolkit would facilitate the currently time consuming construction of the curriculum script. It would also allow the author of the script to focus on the actual content of the script and not in getting the syntax correct for the script to compile with no errors. This is very important because a curriculum script should be put together by experts of the taught domain and not by software developers.

Also, a careful re-examination of the code is required in order to remove hard coded domain information form the framework, and document the software in a clear manner that would make the implementation friendlier to developers who wish to extend it.

On the same software engineering perspective, it is important to adopt a new approach in reading the parameter information of the system. The current approach was found cumbersome. A data base approach for entering, modifying and retrieving data would solve the current file-name dependency observed.

5.6 Summary

This chapter has outlined some suggestions for improving the existing AutoTutor framework, both on a short and long term basis. After having identified the characteristics and idiosyncrasies of the implementation during the procedure of porting (Chapter 4), we have listed feasible solutions for the weaknesses identified in Chapter 3.

Chapter 6

Conclusions

This final chapter summarises the primary contributions of the thesis, and relates them back to the main thesis goals put forward in the Introduction. The evaluation of AutoTutor's five features of interest, domain independence, interface design, dialogue management, student modelling and deep reasoning, was conducted on three separate levels:

1. Theoretical
2. Empirical
3. Architectural

The strengths and weaknesses of the framework are identified and discussed on all levels and feasible solutions are suggested for extending the system.

6.1 AutoTutor's Strengths and Weaknesses

6.1.1 Theoretical Level

The literature review and the comparison of AutoTutor with CIRCSIM-Tutor and Atlas-Andes delimited the strong and weak points of the system with respect to specifications set by theory on ITS technology and the approaches adopted by the other two tutors. AutoTutor attempts to simulate naturalistic tutorial dialogues in the domain of computer literacy. The system is based on simple concepts inspired by human-to-human dialogue structure. Careful consideration of one-to-one tutoring dialogue transcripts has provided AutoTutor with a teaching strategy that allows multi-turn dialogue patterns and co-construction of knowledge from tutor and student. AutoTutor asks open questions that admit of paragraph-long answers. This elicits answers from the student that portray her current knowledge of the topic. The tutor uses language delivered by the agent, along with other modes such as diagrams and animated agent gestures to communicate with the student.

The fact that knowledge is represented in the system by LSA allows the language understanding component to demonstrate a performance which has been evaluated as equivalent to human tutors of intermediate expertise on the domain taught. With a rich corpus on the domain and a well designed curriculum script, the tutor can assess the student's contributions quite effectively.The problem resides in the cases where either the students contribution is unexpected, or it is not contained in the pre-stored information of the system. It is evident that, because the information is pumped from a knowledge base and the utterances and explanations emerge from canned text, its performance is limited from the resources of its library. The system's dialogue management is restricted by the simplicity of its dialogue moves and the inflexibility of the canned text utterances.

Suggestions for improvements in the current version emerged from the study of the design of the two other tutors. The idea of letting the student know what type of answer the tutor is expecting (an idea implemented in CIRCSIM-Tutor) is one that facilitates the discourse and renders dialogue more comprehensive for the student. Also, providing the student with the choice of actively intervening in the dialogue and taking the initiative to skip elaborations that she feels are not needed, can speed up the process of the session and thus keep the student's interest at a high level.

Enhancing the system with multiple teaching strategies tailored to the students needs was an appealing suggestion for improvement. For this to be done a detailed student model is necessary. The idea the integration of a student modelling mechanism was prevalent after the comparison but it was not possible to estimate its feasibility on this theoretical basis of analysis. A closer look in the limitations of the system's architecture made the alternatives available more clear.

As a subsequent step to the theoretical analysis of the system's characteristics, a hands on investigation of the systems actual performance in terms of dialogue management, quality of interface, deep reasoning and student modelling, was decided as the next logical step.

6.1.2 Empirical Level

As stated clearly through the thesis, the purpose of the evaluation attempted in not at all concerned with issues of effectiveness in learning. The investigation of the system's actual performance was triggered by the desire to validate the claims made regarding the four features mentioned. A pilot study was put together in order to observe users' interaction with the tutor and collect valuable feedback on their impressions. For this reason the half the subjects selected for the study were experts in domains related to the features under evaluation.

The overall impression of the tutor was not flattering. This fact raised a lot of questions regarding the choice of the interface and its features. Literature in the field of agent interface design suggested that further investigation in the effects of such an interface is necessary in order to justify its adoption. A prevalent deficit in AutoTutor's interface is the voice articulation that is very difficult to understand. This fact, along with the feedback gestures that were perceived as confusing lead to the conclusion that if indeed it is empirically proven that such an interface is better than a conventional one, a lot remains to be done in order to identify the optimal attributes for the agent's features.

The implications of the absence of a user model became apparent in the study of the system's actual performance. The dialogue was often recorded as repetitive due to the fact of the poor representation of the user's knowledge. The re-usability of the system was assessed as very low since there is no user profile stored and each time the user wishes to interact with the system it has no knowledge of the topics covered in previous sessions and the user's performance in them.

Dialogue management depicted a behaviour that was not documented in the AutoTutor literature. Although according to the Dialogue Advancer Network (see appendix) the dialogue is supposed to be with mixed initiative, it proved to be significantly one sided with all the control on the tutor's side. The system is able to handle unexpected answers only in the case that they are a WH-question and that the definition of the term in question exists in the tutors glossary. Even then the flow of the dialogue is abnormally interrupted since the tutor is unable to resume the conversation where it was left and arbitrarily skips to the next dialogue move.

LSA's performance was recorded as good in the case of long answers. The problem is that usually users are hesitant in typing in long answers since they assumed that the tutor is capable of understanding them. Deep reasoning questions and dialogues were not recorded since the domain of computer literacy does not support this type of reasoning.

The familiarisation with the tutor's actual characteristics was thought to require a software engineering approach in order to map the performance of the system's modules with their specific implementations and to elicit what solutions are feasible. Porting the framework in a new domain was the defined as the step to meet this requirement.

6.1.3 Architectural Level

Porting the system to a new domain that supports deep reasoning in the dialogue was identified as the approach to a more in depth analysis of the system's architecture from a software engineering perspective. The driving force for this attempt was the desire to reveal the actual behaviour and potentials of the system and come up with feasible modifications on the current architecture. Although our reasons for porting are not dealing with issues on learning outcomes, a careful selection of the new domain was considered essential in terms of future extensibility of the system developed.

A close encounter with the characteristics of the current implementation of the AutoTutor framework revealed that the underlying architecture is much more restrictive than advertised in the relevant literature review. The porting procedure was seriously hindered by the intricate form of the code and the lack of sufficient documentation.

A significant limitation of the system which is unavoidably reflected in all its modules is one that derives from the knowledge representation approach. The use of LSA shows satisfactory behaviour in “understanding” student contributions. Nevertheless, problems lie in the fact that this understanding is local and limited in the sense that no more information can be extracted from LSA apart from a similarity metric to pre-stored information. While concepts and causal relationships have no means of being represented, it is impossible to surpass the inflexibility of canned text, of poor user modelling and of cumbersome dialogue management. This finding imposes serious restrictions in the extensibility of the system.

Solutions alleviating the reported weaknesses and boosting the systems performance towards the goals theory dictates have been put forward both on the basis of building on top of the current implementation, as well as on re-designing some of the major modules of the framework.This analytical evaluation of AutoTutor shed light on issues that were obscurely documented in the literature. This thesis aspires to serve as a guide to developers who wish to extend the existing implementation of AutoTutor.

Appendix A

Dialogue Advancer Network

Appendix B

AutoTutor Questionnaire

Appendix C

Dialogues

C.1 Crowding Example

In the experimental method all extraneous variables are controlled. Suppose you want to test the hypothesis that crowding impairs cognitive performance. To do this, you might put one group of subjects in a crowded room and another group in an uncrowded room. The subjects in each of the groups would then complete the same cognitive tasks.

Tutor:What would you measure?

Student: cognitive performance.

Tutor:That’s right. The variable you will be measuring is called the ... (prompt)

Student: Dependent Variable.

Tutor:Correct! So the Dependent Variable is cognitive performance. And what is the independent variable?

Student: I dont know.

Tutor:The independent variable is the variable the researcher manipulates to create groups that differ in the levels of the variable. The researcher then compares the groups in terms of their scores on the dependent variable.

So in this case the independent variable is ...?

Student: crowding?

Tutor:Yes.

Now suppose that the subjects in the crowded group do not perform as well on the cognitive tests as those in the uncrowded condition. Where would you attribute the differences?

Student: in crowding.

Tutor:Yes, if there is no other difference between the groups. But what if the room in which the crowded group was placed had no windows and the room with the uncrowdedgroup did have windows-for example, they were in two different rooms in a high school? Would you be able to draw conclusions regarding the causality relationship of crowding and poor performance?

Student: I don't know.

Tutor: In that case it would be impossible to know whether the poor scores of the subjects in the crowded group were due to the crowding or to the lack of windows. The variables of crowding and window presence are confounded. How would you avoid this confounding factor?

Student: By keeping the window variable constant in the two groups.

Tutor: Yes! By holding the window variable constant, the presence or absence of windows might have affected subject performance, but the effect of the windows would be identical in both conditions.

TO SUMMARIZE : Confounding occurs when the researcher fails to control some extraneous variable. When a variable other than the manipulated variable has been allowed to exert a differential effect in the two conditions then the two variables are confounded. Good experimental design involves eliminating possible confounds that result in alternative explanations. A researcher can claim that the independent variable caused the results only when there are no competing explanations. When the results of an experiment can confidently be attributed to the effect of the independent variable, the experiment is said to have internal validity. To achieve internal validity, the researcher must design and conduct the experiment so that only the independent variable can be the cause of the results.

C.2 Motivation Example

Tutor: In true experiments the researcher has complete control over the research situation.

Today we will look at designs that only involve a posttest.

First we have the one-group posttest-only design. This is a simple design in which a group of subjects is given a treatment and then tested on some dependent variable. Suppose you wanted to test the effectiveness of one of the popular motivational programs in which people attend a series of lectures and group activities.

In this case the dependent variable measured will be.... ?

Student: the effectiveness of the training.

Tutor: That's right. Suppose you decided to ask the people who took the training, whether it was effective. Imagine that the subjects state that the treatment was effective and that they feel better about themselves.

Is this sufficient to conclude that the treatment was truly effective?

Student: No.

Tutor: Why not?

Student: ....

Tutor: (hint) A good experimental design must be able to prove causality between the treatment and the effect. In this example, is it guaranteed that the change in the subjects'attitude can be solely attributed to the effect of the training program?

Student: .....

Tutor: Can you think of an improvement in the design?

Tutor: So, one improvement would be to introduce a control group that doesn't take the treatment. You might try to find a group of people who matched the treatment group on as many variables as possible: age, income, education, and so forth. Although the design is an improvement over the one-group posttest-only design, it still has a serious flaw. Can you spot the flaw?

Student: The two groups are not equivalent?

Tutor: The flaw is that the control group is not equivalent in every way to those who took the training. Do you have any ideas why?

Student: ...

Tutor: The most important difference is that the test group members selected themselves for the training and the control group members did not. Thus, we have a nonequivalent control group because the two groups were not randomly constituted from the same population.

Tutor: A nonequivalent control group is better than no control group, but you would have to consider this study a quasi experiment at best because the subjects were not randomly assigned to groups. Can you think of a way to improve this design?

Student: by randomly assigning subjects to condition.

Tutor: Random allocation of subjects is the answer. The only way you could construct a control group that was equivalent to the training group would be to ask the program organization to provide a list of all people who applied for the training. Then you would randomly place half of them into a control group that would not be allowed to take the training.

Summarizing: Lets take it form the top. Both designs we discussed, one-group posttest only and posttest only with non-equivalent control group, are not considered true experiments because of their weaknesses mentioned. A true experimental design requires random allocation of subjects in the control and the treatment conditions in order to be able to make claims regarding the causality of the effect observed.

C.3 Schedule Example

Tutor: Even though there can be no perfect experiment, two particular elements of design provide control over so many different threats to validity that they are basic to all good experimental designs: (1) the existence of a control group or a control condition and (2) the random allocation of subjects to groups. Keeping this in mind, lets take a close look at an example.

Tutor: Suppose a company introduced a new work schedule whereby its employees put in four 10-hour days a week instead of five 8-hour days. If output increased, management would probably credit the new schedule. This conclusion represents an improvement over one that might have resulted from a one-group posttest-only design because in this case, you know that a behavior change did follow the treatment.

So which is the independent variable in this case?

Student: the work schedule.

Tutor: Correct. And the dependent variable is ...?

Student: the work output.

Tutor: That's right.Can you think of any other threats to the validity of the experiment?

Student: ...

Tutor: (hint) Threats to validity are other potential causes of the increase in output. Can you think of any?

Student: ...

Tutor: Workers may have responded to the attention paid them by management when the change was initiated. This is called demand characteristic.

Or any number of events may have led to increased productivity: favorable weather conditions that allowed the workers to get work on time, a change in seasons that made the plant more comfortable, or a favorable response to a new supervisor. What are these occurrences of for the validity of the experiment?

Student: ...

Tutor: These occurrences represent threats to internal validity: The change was caused by a variable other than the one management thought to be responsible. Threats to external validity could arise from the possibility that these workers are young and like long hours, whereas older workers might have preferred shorter days.

Tutor: Can you think of better ways in designing the experiment?

Student: ...Introduce a control group?

Student: ...Randomly allocate subjects in a control and a target group.

Tutor: Why do you think a control group is essential?

Tutor: This study would have been better designed by forming two groups through random allocation of workers to different schedules so that one group would remain on the old schedule as a control group. This control would have eliminated the threats to internal validity.

Tutor: Now lets suppose that random allocation of workers to two groups is not possible. Also lets suppose that the company has two plants, one could be switched to the 4-day week while the other was kept on the 5-day week. Productivity in the two plants could then be compared. Would the two groups be equivalent?

Student: ...

Tutor: What type of design would we have in this case?

Student: ...

Tutor: This example is a nonequivalent-control-group design. Differences between the workers at the two plants or in the plants themselves may account for the results instead of the work schedule. The addition of a nonequivalent control group to a pretest-posttest design improves the control sufficiently that the design may he considered a quasi experiment.

C.4 Smoking Example

Tutor: Another type of experimental design, appart from the posttest-only design, is the pretest-posttest design.

The only difference between the posttest-only design and the pretest-posttest design is that in the latter a pretest is given before the experimental manipulation is introduced.

Can you think of advantages of such a design regarding the equivelancy of the two groups?

Student: ...

Tutor: This design makes it possible to ascertain that the groups were, in fact, equivalent at the beginning of the experiment. However, this precaution is usually not necessary if subjects have been randomly assigned to the two groups.

Can you think of circumstances where random assignment wouldn't suffice?

Student: ...small sample size.

Tutor: The sample must be sufficiently large. With a sufficiently large sample of subjects, random assignment will produce groups that are virtually identical in all respects. Although there are no clear-cut rules for specifying a ”sufficiently large” sample, a minimum of 10 subjects per group is a good rule of thumb. The larger the sample, the less likelihood there is that the groups will differ in any systematic way.

Lets look at an example. A researcher wishes to study the effect of a method for reducing smoking. In this case the dependent variable is ...?

Student: smoking

Tutor: Correct. So, the group that doesn't receive the treatment will be called ...?

Student: control group.

Tutor: And the group that does receive the treatment will be called ...?

Student: Experimental group.

Tutor: That's right. So the two levels of the independent variable are ....?

Student: treatment and no treatment.

Tutor: In order to sustain internal validity in the design how must the researcher allocate the subjects in the two groups?

Student: Randomly.

Tutor: Correct, so random allocation will guarantee that the two groups are equivalent.

What would happen in the case that some subjects drop out of the experiment because of its long duration?

Student: the two groups will no longer be equivalent.

Tutor: Correct. The drop-out factor in experiments is called mortality.

People may drop out for reasons unrelated to the experimental manipulation, such as illness; but sometimes mortality is related to the experimental manipulation. Even if the groups are equivalent to begin with, different mortality rates can make them nonequivalent. How might mortality affect a program designed to reduce smoking?

Student: ...

Tutor: The heaviest smokers in the experimental group might wind up leaving the program; when the posttest is given, only the light smokers would remain, so that a comparison of the experimental and control groups would show less smoking in the experimental group even if the program had no effect.

How can a pretest help deal with problems of mortality?

Student: ...

Tutor: Use of a pretest makes it possible to assess the effects of mortality; you can look at the pretest scores of the dropouts and know whether mortality affected the final results. Mortality is likely to be a problem when the experimental manipulation extends over a long period of time. In such a situation, a pretest is a very good idea.

Appendix D

Curriculum Script

curriculum script Experimental Design curriculum script normal time 5400

topic Crowding topic abbrev CR topic phase Early topic normal time 1800

picture question answer-1 easy

picture-1 crowding3.gif

info-1 Here is how the tutoring session is going to go. First, I'll ask you a question or present you with a problem. Second, you'll type in your answer. Then, we'll have a conversation by taking turns improving on the answer. It's very important that you try to improve your answer by adding to it each time it's your turn. When we finish answering the question together, we will move on to the next question or problem. There are a few simple rules you need to follow to make this conversation work. Please try to avoid typos, and always try to use appropriate punctuation. For example, end a question with a question mark. When you finish your turn, you have to hit the return key so I'll know you are ready for me to respond to your contribution. After I finish speaking it will be your turn again. <pause 100> When you finish your turn I may give you some type of feedback on your answer, and then I'll follow my response either with a question or some additional information for you to consider. At times, my feedback may seem a little grumpy, but don't take it too seriously, it is just my computerized personality showing through. <pause 100> The idea is for me to try to guide you to help you improve on your initial answers. Sometimes you may not be able to understand me. After all, I'm just a computer tutor and my voice sounds funny because I'm not quite human. Also, sometimes I'll say something that is hard to understand because the information I'm delivering might be a little complicated. You can always ask me to repeat anything I've said. <pause 50> So! We are ready to start! <speakStyle PosNeutral>In the experimental method <pause 50>all extraneous variables are controlled. <pause 50> Suppose you want to test the hypothesis that crowding impairs cognitive performance. <pause 50> To do this, you might put one group of subjects in a crowded room,lets assume this is called the crowded condition, <pause 50> and another group in an uncrowded room, which will respectively be called the uncrowded condition. The subjects in each of the groups would then complete the same cognitive tasks in the form of a test.<pause 50> So here's your <clip gaze*> question. <pause 100>

question-1 <picture crowding3.gif> (Hypothesis: Crowding impairs cognitive performance)<pause 100> <clip Point mid> Which are the experimental variables in this example and what are their characteristics?

ideal-1 The independent variable is existence of crowding and the dependent variable is cognitive performance. The independent variable has two levels, crowding and no crowding. For each condition we measure the cognitive performance of the subjects.

concept-1 dependent variable concept-1 Dependent variable concept-1 measured variable concept-1 variable measured concept-1 dependent concept-1 cognitive performance

concept-2 independent variable concept-2 Independent Variable concept-2 manipulated variable concept-2 the variable manipulated concept-2 crowding concept-2 existence of crowding concept-2 number of people concept-2 number of people in the room concept-2 crowdedness

concept-3 cause-effect concept-3 cause effect concept-3 cause-effect relationship concept-3 causality

concept-4 lower concept-4 poorer concept-4 worse concept-4 not as good

pgood-1-1 The manipulated variable is crowding and the variable measured is cognitive performance.

pelab-1-1 Crowding and cognitive performance.

phint-1-1-1 <speakStyle PosNeutral

>

What kind of variable is <clip gaze*

>

the existence or not of crowding? phintc-1-1-1 Crowding is the independent variable.

phint-1-1-2 <speakStyle PosNeutral

>

What does the manipulated variable express in<clip gaze*

>

this experimental design and <pause 100> what are its two levels used? phintc-1-1-2 Crowding is the manipulated variable and crowded and uncrowded are the two levels.

phint-1-1-3 <speakStyle PosNeutral> What kind of variable is <clip gaze*> cognitive performance? phintc-1-1-3 Cognitive performance is the dependent variable.

phint-1-1-4 <speakStyle PosNeutral> Which is the variable of which the values are measured in this experimental design, and what is its formal name? phintc-1-1-4 Cognitive performance is the dependent variable.

pprompt-1-1-1 <speakStyle PosNeutral>Crowding in this case is a variable called<clip Proclivity Slient><clip gaze*> ppromptc-1-1-1 independent variable. ppromptk-1-1-1 independent

pprompt-1-1-2 <speakStyle PosNeutral>The variable manipulated in order to get two discrete experimental conditions is <clip Proclivity Slient><clip gaze*> ppromptc-1-1-2 the amount of crowding ppromptk-1-1-2 crowding

pprompt-1-1-3 <speakStyle PosNeutral>The dependent variable in this design is <clip Proclivity Slient><clip gaze*> ppromptc-1-1-3 the existence of crowding ppromptk-1-1-3 crowding

pprompt-1-1-4 <speakStyle PosNeutral> The variable measured <pause 100>, the dependent variable, in this case is <clip Proclivity Slient><clip gaze*> ppromptc-1-1-4 the dependent variable in this case is cognitive performance. ppromptk-1-1-4 cognitive performance pgood-1-2 The independent variable is the variable the researcher manipulates to create groups that differ in the levels of that variable. The researcher then compares the groups in terms of their scores on the dependent variable. In this case the independent variable is crowding.

pelab-1-2 The manipulated variable is crowding.

phint-1-2-1 <speakStyle PosNeutral> In this design, the researcher manipulates a variable in order to create two groups that differ in the levels of that variable. What would be the two appropriate levels for this example? phintc-1-2-1 crowding and no crowding

phint-1-2-2 <speakStyle PosNeutral> The independent variable is one that is believed to cause some change in the value of the dependent variable.<pause 100> In this example the independent variable is crowding<pause 100>. Which is the dependent variable that we expect will be affected by the two different levels of the independent variable, crowding and no crowding? phintc-1-2-2 cognitive performance

pprompt-1-2-1 <speakStyle PosNeutral> The manipulated variable, in this case crowding, is also called <clip Proclivity Slient><clip gaze*> ppromptc-1-2-1 the independent variable ppromptk-1-2-1 independent variable

pprompt-1-2-2 <speakStyle PosNeutral> The variable that we expect the independent variable to have effects on <pause 100> is formally called<clip Proclivity Slient><clip gaze*> ppromptc-1-2-2 the dependent variable ppromptk-1-2-2 dependent variable

pgood-1-3 The dependent variable is the variable whose scores the researcher measures, for each experimental group. In this case the dependent variable is cognitive performance.

pelab-1-3 Cognitive performance is the dependent variable whose scores we measure.

phint-1-3-1 <speakStyle PosNeutral>Which is the variable that we monitor in this example of two groups, one in a crowded room and one in an uncrowded room? phintc-1-3-1 cognitive performance

phint-1-3-2 <speakStyle PosNeutral>The experiment aims to prove a cause-effect relationship between the independent and which other variable? phintc-1-3-2 dependent variable.

pprompt-1-3-1 The variable that we measure, otherwise mentioned as the independent variable, <pause 100> in this example measures <clip Proclivity Slient><clip gaze*> ppromptc-1-3-1 <clip Proclivity Slient> cognitive performance <clip gaze*> ppromptk-1-3-1 cognitive performance

pprompt-1-3-2 <speakStyle PosNeutral> The relationship between the independent and the dependent variable is formally called <clip Proclivity Slient><clip gaze*> ppromptc-1-3-2 <clip Proclivity Slient> cause-effect <clip gaze*> ppromptk-1-3-2 cause-effect relationship

pgood-1-4 The dependent variable, in this case cognitive performance, is the variable that the researcher measures for each different level of the independent variable.

pelab-1-4 Cognitive performance is the dependent variable that the researcher measures for each different group of the experiment.

phint-1-4-1 <speakStyle PosNeutral> What does the researcher measure for both the crowded group condition and the uncrowded room condition? phintc-1-4-1 cognitive performance

phint-1-4-2 <speakStyle PosNeutral> Do we expect to find a difference in the measures of cognitive performance <pause 100> amonst the two conditions? phintc-1-4-2 yes

pprompt-1-4-1 <speakStyle PosNeutral> Cognitive performance, measured through equivalent tests is called the <clip Proclivity Slient><clip gaze*> ppromptc-1-4-1 dependent variable ppromptk-1-4-1 dependent

pprompt-1-4-2 <speakStyle PosNeutral> In the crowded condition we would expect the results on the cognitive test <pause 100> in comparison to the uncrowded condition results, <pause 100> to be <clip Proclivity Slient><clip gaze*> ppromptc-1-4-2 lower ppromptk-1-4-2 poorer

good-1 The independent variable is the amount of crowding and the dependent variable is cognitive performance.

good-1 The amount of crowding and cognitive performance.

good-1 Crowding and no crowding are the levels of the manipulated variable and cognitive performance is the variable measured.

good-1 Crowding and no crowding are the levels of the independent variable and cognitive performance is the dependent variable.

good-1 Existence or not of crowding is the independent variable and cognitive performance is the dependent variable.

good-1 The crowded condition and the uncrowded condition are the levels of the independent variable in the example and cognitive performance is the dependent variable.

good-1 Crowding and test scores on cognitive tasks.

good-1 Crowdedness and cognitive performance.

good-1 The existence of crowding and cognitive performance.

good-1 Crowdedness and scores on cognitive tasks.

good-1 The independent variable is crowdedness and the dependent variable is cognitive performance.

good-1 The independent variable is crowding and the dependent variable is the score on cognitive tasks.

hint-1 Remember, <pause 100> the independent variable is the variable the researcher manipulates in order to create groups that differ in the levels of that variable.

hint-1 Keep in mind that the independent variable has two levels in this example <pause 100>, the crowded condition and the uncrowded condition.

bad-1-1 Crowding is the dependent variable.

bbad-1-1 Existence of crowding is the measured variable.

splice-1-1 <speakStyle PosNeutral>Existence of crowding is the independent variable because it is the variable that is manipulated, <pause 100> cognitive performance is the dependent variable for which we measure scores. <clip gaze*>

bad-1-2 Cognitive performance is the manipulated variable.

bbad-1-2 Cognitive performance is the independent variable.

splice-1-2 <speakStyle PosNeutral>The manipulated variable is crowding. <clip gaze*>

summary-1 <speakStyle PosNeutral> The independent variable is the variable the researcher manipulates to create groups that differ in the levels of the variable. In this case, crowding and no crowding are the two levels of the independent variable.

The dependent variable is the one whose scores are measured, in this case cognitive performance. The researcher compares the two groups on their scores in a cognitive performance test <unload> .

picture question answer-2 easy

picture-2 two.gif

question-2 <picture two.gif> Now, <clip Point mid> <pause 100> suppose that the subjects in the crowded group do not perform as well on the cognitive test as those in the uncrowded condition. Also suppose, <pause 100> that all other factors that could possibly influence cognitive performance are the same for both groups<pause 100>. In this case,<pause 100> where would you attribute the difference in performance?

ideal-2 The differences can be attributed to the effect of crowding on cognitive performance. They prove a cause-effect relationship between the independent and the dependent variable.

concept-1 dependent variable concept-1 Dependent variable concept-1 measured variable concept-1 variable measured concept-1 dependent concept-1 cognitive performance

concept-2 independent variable concept-2 Independent Variable concept-2 manipulated variable concept-2 the variable manipulated concept-2 crowding concept-2 existence of crowding concept-2 number of people concept-2 number of people in the room concept-2 crowdedness

concept-3 cause-effect concept-3 cause effect concept-3 cause-effect relationship concept-3 causality

pgood-2-1 The differences can be attributed to the effect of crowding on cognitive performance.

pelab-2-1 The differences can be attributed to the effect of the independent variable on the dependent variable.

phint-2-1-1 <speakStyle PosNeutral>The aim such an experiment is to show cause and effect relationships between the independent and the dependent variable. In this case, <pause 100> what do the differences in cognitive performance tell us? phintc-2-1-1 The differences in cognitive performance between the two conditions show that there is a cause-effect relationship between crowding and cognitive performance.

phint-2-1-2 <speakStyle PosNeutral>Why is it important that the two groups are affected by the same factors, apart from crowding? phintc-2-1-2 It is important so that we can make predictions about cause-effect relationships between crowding and cognitive performance.

pprompt-2-1-1 <speakStyle PosNeutral>The two groups have to differ only in respect to <clip Proclivity Slient><clip gaze*> ppromptc-2-1-1 the independent variable ppromptk-2-1-1 crowding

pprompt-2-1-2 <speakStyle PosNeutral> If the groups have identical factors affecting their performance then the researcher is able to make predictions regarding<clip Proclivity Slient><clip gaze*> ppromptc-2-1-2 causality ppromptk-2-1-2 the effect of crowding on cognitive performance

good-2 The differences can be attributed to the fact that crowding impairs cognitive performance.

good-2 We can draw conclusions regarding the effect of crowding on cognitive performance.

good-2 These differences could be attributed to the effect of the independent variable on the dependent variable.

good-2 The differences in cognitive performance between the two groups can be attributed to the existence of crowding.

good-2 The differences between the two groups can be attributed to crowding.

good-2 Cognitive performance is impaired by crowding.

good-2 The subjects did not perform as well in the crowded group because of the crowding condition.

good-2 Crowding impairs cognitive performance.

good-2 Cognitive performance is lower in the crowded condition.

good-2 On the effect of crowding on cognitive performance.

good-2 The difference in performance is due to crowding.

good-2 The difference in cognitive performance is due to the effect of crowding.

good-2 The difference in the results can be attributed to the effect of crowding.

good-2 The difference is due to the fact that crowding impairs cognitive performance.

good-2 The difference can be attributed to the fact that crowding has an effect on cognitive performance.

good-2 The difference can be attributed to the fact that the there is an effect on cognitive performance due to crowding.

good-2 The differences in cognitive performance between the two conditions show that there is a cause-effect relationship between crowding and cognitive performance.

good-2 We can conclude that there is an effect of crowding on cognitive performance.

good-2 We can conclude that that crowding impairs cognitive performance.

good-2 It is important so that we can make predictions about cause-effect relationships between crowding and cognitive performance.

good-2 We can claim that crowding impairs cognitive performance.

good-2 We can make predictions about causality of crowding on lower cognitive performance.

hint-2 Remember!, <pause 100> that the point of such an experiment is to prove a cause-effect relationship between the independent and the dependent variable.

hint-2 Keep in mind, <pause 100> that the experimenter is interested in proving causality between crowding and cognitive performance.

bad-2-1 To the effect of the dependent variable on the independent variable.

bad-2-1 The differences can be attributed to the effect of the dependent variable on the independent variable.

bad-2-1 The differences can be attributed to the effect of cognitive performance on crowding.

bad-2-1 To the effect of cognitive performance on crowding.

bbad-2-1 Crowding increases cognitive performance.

splice-2-1 <speakStyle PosNeutral>The differences are due to the effects of !crowding on cognitive performance <clip gaze*>

bad-2-2 To chance.

bad-2-2 to chance

bbad-2-2 to circumstances

splice-2-2 <speakStyle PosNeutral>Since the experimental conditions are adequately controlled for, <pause 100> it is safe to claim that these differences are not due to chance! <clip gaze*>

summary-2 <pause 100>The aim of the experiment is to allow the researcher to make valid claims on the existence of a causal relationship between crowding and cognitive performance. <pause 100> In the case that all extraneous variables are controlled and the only difference in conditions between the two groups is the dependent variable <pause 100> then it is safe to make claims that indeed <pause 100> there is a cause-effect relationship between the independent and the dependent variable.<pause 100> In this case, <pause 100> that crowding impairs cognitive performance. <clip gaze*>

qapair-2-1 causal relationship

qapair-2-1 causal relationship is a relationship where variation in one variable causes variation in another.

qapair-2-2 idependent variable

qapair-2-2 An independent variable is a qualitative or quantitative entry that can be measured and represent a concept studied. It is manipulated by the researcher in order generate various experimental conditions that correspond to different levels of the variable. The researcher studies the effect of the independent variable on a dependent variable.

qapair-2-3 dependent variable

qapair-2-3 A dependent variable is a qualitative or quantitative entry that can be measured and represent a concept studied. Its the variable that is observed and measured for the various levels of the independent variable.

question answer-3 easy

question-3 Now, <pause 100> consider what would happen if the factors affecting the two groups, the one in the crowded condition and the one in the uncrowded condition, where different. For example, what would happen if the crowded group was placed in a room with no windows, and the ungcrowded group was in a room with windows. Suppose that once more, the subjects in the crowded condition preformed poorly, and the subjects in the uncrowded condition preformed well. Would you be able to conclude a causality relationship between crowding and poor cognitive performance? Please justify your answer.

ideal-3 No. A causality relationship could not be concluded because the independent variable of crowding is confounded by the windows factor. It will not be clear if the poor cognitive performance observed in the crowded condition was due to crowding or to the absence of windows.

concept-1 dependent variable concept-1 Dependent variable concept-1 measured variable concept-1 variable measured concept-1 dependent concept-1 cognitive performance

concept-2 independent variable concept-2 Independent Variable concept-2 manipulated variable concept-2 the variable manipulated concept-2 crowding concept-2 existence of crowding concept-2 number of people concept-2 number of people in the room concept-2 crowdedness

concept-3 cause-effect concept-3 cause effect concept-3 cause-effect relationship concept-3 causality

concept-4 windows concept-4 confounding variable concept-4 existence of windows concept-4 absence of windows concept-4 existence and absence of windows concept-4 windows and no windows concept-4 windows factor

concept-5 confounded concept-5 poor concept-5 wrong

pgood-3-1 No, because the design is confounded.

pelab-3-1 The design is !confounded!

phint-3-1-1 <speakStyle PosNeutral> In the case that an extraneous variable, <pause 100> that can affect the dependent variable, is not controlled for <pause100>, where would you attribute the lower cognitive performance scores of the crowded condition? <clip gaze*> phintc-3-1-1 The lower scores can then be attributed either to windows, or to crowding or to both windows and crowding.

phint-3-1-2 <speakStyle PosNeutral>If extraneous variables are not controlled in the experimental design <pause 100>, there is no way of proving a cause-effect relationship between the independent and the dependent variable, because it is !confounded! <pause 100> In this case which is the confounding variable? <clip gaze*> phintc-3-1-2 windows pprompt-3-1-1 <speakStyle PosNeutral> When extraneous variables are not controlled for,<pause 100> and affect the two groups differently,<pause 100> then we state that the experimental design is <clip Proclivity Slient><clip gaze*> ppromptc-3-1-1 confounded ppromptk-3-1-1 poor

pprompt-3-1-2 <speakStyle PosNeutral> The existence of windows in one condition and not in the other is an extraneous variable that is also called <clip Proclivity Slient><clip gaze*> ppromptc-3-1-2 confounding variable ppromptk-3-1-2 a confounding variable

pgood-3-2 If there are other differences between the two groups then the differences observed in cognitive performance cannot be attributed to the effect of the independent variable.

pelab-3-2 If the conditions of the two groups are not equivalent then the differences cannot be attributed to the effect of the independent variable.

phint-3-2-1 <speakStyle PosNeutral>Why can't you make predictions about causality, <pause 100> when the conditions of the two groups differ in more aspects,<pause 100> than the independent variable? phintc-3-2-1 Because, the design is confounded.

phint-3-2-2 <speakStyle PosNeutral> In order to prove a cause-effect relationship between crowding and cognitive performance <pause 100> what would you change in the current design? phintc-3-2-2 I would make sure that the windows factor is the same in both conditions.

pprompt-3-2-1 <speakStyle PosNeutral> If the factors affecting the two groups are different, <pause 100> it would be impossible to make predictions regarding <clip Proclivity Slient><clip gaze*> ppromptc-3-2-1 causality ppromptk-3-2-1 the effect of crowding on cognitive performance

pprompt-3-2-2 <speakStyle PosNeutral> To prove a cause-effect relationship <pause 100> we need to control for all possible extraneous <clip Proclivity Slient><clip gaze*> ppromptc-3-2-2 variables ppromptk-3-2-2 factors

good-3 If the conditions the two groups are in are different, then we can't draw conclusions about the effect of crowding on cognitive performance because other factors may have been the ones causing this difference.

good-3 If the conditions the two groups are in are different, then the experimenter can’t draw conclusions about the effect of crowding on cognitive performance because other factors may have been the ones causing this difference.

good-3 If the conditions the two groups are in are different, then the researcher can’t draw conclusions about the effect of crowding on cognitive performance because other factors may have been the ones causing this difference.

good-3 No.

good-3 no

good-3 A causality relationship cannot be concluded in this case.

good-3 Causality cannot be concluded between the independent and the dependent variable.

good-3 We cannot conclude a causality relationship.

good-3 We cannot prove causality between crowding and cognitive performance.

good-3 No causality relationship can be concluded.

good-3 No causality relationship can be concluded because the design is confounded.

good-3 The lower scores can then be attributed either to windows, or to crowding or to both windows and crowding.

good-3 Because the design is confounded.

good-3 Because the existence of windows is a factor that affects cognitive performance.

good-3 We can’t make predictions about causality because the design is confounded.

good-3 We can’t make predictions about causality because crowding and windows are confounded variables.

good-3 Crowding and windows are confounded variables.

good-3 The existence of crowding and the existence of windows are confounded variables.

good-3 Make sure that the windows factor is the same in both conditions.

good-3 Make sure that there are no windows in both conditions.

good-3 Make sure that there are windows in both conditions.

good-3 Make sure that the windows variable is the same in both conditions.

hint-3 Keep in mind that <pause 100> the existence of extraneous variables <pause

100> that are not controlled in the design can cause confounding <pause 100>, and do not allow the conclusion of causality relationships between the dependent and the independent variable.

hint-3 Remember that the experimenter must control all extraneous variables in order to make valid claims about causality.

bad-3-1 Yes

bad-3-1 yes

bad-3-1 Yes we can draw conclusions about causality.

bad-3-1 Yes the researcher can draw conclusions about causality.

bbad-3-1 Yes there is a causality relationship between the independent and the dependent variable.

splice-3-1 <speakStyle PosNeutral>!Careful!<pause 100>Note that we cannot draw conclusions about causality if all extraneous variables are not controlled <pause 100> because we cannot be sure if the effect observed is caused by the independent variable or by some uncontrolled factor like the existence of windows. <clip gaze*>

summary-3 In this example we are studying the existence of a causal relationship between crowding,and poor cognitive performance. <pause 100> In order to prove such a cause-effect relationship, in other words, that crowding impairs cognitive performance, <pause 100> we need to make sure that all extraneous variables that could affect cognitive performance are controlled.<pause 100> Only when the conditions of the two groups differ solely in the independent variable <pause 100> can we make claims about causality relationships.<pause 100> If there are other factors uncontrolled for,<pause 100> for example the existence of windows in the uncrowded condition <pause 100> but not in the crowded one, then we cannot be sure that the difference in cognitive performance is due to crowding. <pause 100> It could be due to the windows factor <pause 100> or even both.

qapair-3-1 confounding

qapair-3-1 Confounding is an error that occurs when the effects of two variables in an experiment cannot be separated, resulting in a confused interpretation of the results.

Bibliography

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:4–16.

Cohen, L. and Manion, R. (1989). *Research Methods in Education*. London: Routledge.

Cohen, P. A., Kulik, J. A., and Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, 19:237–248.

Core, M. G., Moore, J. D., Zinn, C., and Wiemer-Hastings, P. (2000). Modelling human teaching tactics in a computer tutor. In *In Proceedings of the ITS '00 Workshop on Modelling Human Teaching Tactics and Strategies*. under consideration.

Cozby, P. C. (1989). *Methods in Behavioural Research*. Mayfield Publishing Company.

Dehn, D. M. (2000). The impact of animated interface agents: a review of empirical research. *Human-Computer Studies*, 52:1–22.

Evens, M. W., Brandle, S., Chang, R. C., Freedman, R., Glass, M., Lee, Y. H., Shim, L. S., Woo, C. W., Zhang, Y., Yujian, Z., Joel A., M., and Allen A., R. (2001). Circsim-tutor: An intelligent tutoring system using natural language dialogue. In OH, editor, *Twelfth Midwest AI and Cognitive Science Conference, MAICS 2001*, pages 16–23, Oxford.

Fox, B. (1993). *The human tutorial dialog project*. Hillsdale, NJ: Erlbaum.

Freedman, R. (2000). Plan-based dialogue management in a physics tutor. In *Proceedings of the Sixth Applied Natural Language Processing Conference*. Seattle.

Freedman, R., Rose, C. P., Ringenberg, M. A., and VanLehn, K. (2000). Its tools for natural language dialogue: A domain-independent parser and planner. In *In proceedings of the Intelligent Tutoring Systems Conference.*

Gertner, A. S. and VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In *Proceedings of ITS 2000.*

Glass, M. (2001). Processing language input in the circsim-tutor intelligent tutoring system. *Artificial Intelligence in Education*, pages 210–221. J. D. Moore et al. eds.

Graesser, A. C. (1993). Questioning mechanisms during tutoring, conversation, and human-computer interaction. In *Memphis State University, Memphis, TN.* ERIC Document Reproduction Service No. TM 020 505.

Graesser, A. C., Person, N., and Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9:495–522.

Graesser, A., W.-H. K. W.-H. P. K. R. . t. T. R. G. (2000). Autotutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, 1:35–51.

Hume, G. D., Michael, J. A., Rovick, A., and Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, 5(1):23–47.

Landauer, T. and Dumais, S. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation. *Psychological Review*, 104:211–240.

Lee, Y. H. and W., E. M. (1998). Natural language interface for an expert system. *Expert Systems (International Journal of Knowledge Engineering*, 15(4):233–239.

Link, K. E., Kreuz, R. J., Graesser, A. C., and the Tutoring Research Group (2001). Factors that influence the perception of feedback delivered by a pedagogical agent. *International Journal of Speech Technology*, 4:145–153.

Merrill, D. C., J., R. B., and Landes, S. (1992). Human tutoring: Pedagogical strategies and learning outcomes. In *Paper presented at the annual meeting of the American Educational Research Association*.

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

Person, N. (1994). *An analysis of the examples that tutors generate during naturalistic one-to-one tutoring sessions*. PhD thesis, University of Memphis.

Person, N., Graesser, A. C., Kreuz, R. J., Pomeroy, V., and the Tutoring Research Group (2001). Simulating human tutor dialogue moves in autotutor. *International Journal of Artificial Intelligence in Education*, 12. to appear.

Preson, N. K., Graesser, A. C., and the Tutoring Research Group (2000). Designing autottutor to be an effective conversational partner. In Fishman, I. and (Eds.), S. O.-D., editors, *Fourth International Conference of the Learning Sciences*, pages 246–253.

Rose, C. P., Jordan, P.and Ringenberg, M., Siler, S., VanLehn, K., and Weinstein, A. (2001). Interactive conceptual tutoring in atlas-andes. In *Proceedings of AI in Education*.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12:257–285.

Wiemer-Hastings, P. (2000). Adding syntactic information to lsa. In *Proceedings of the Twenty-second Annual Conference of the Cognitive Science Society*, pages 989–993.

Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. C. (1999). Improving an intelligent tutor's comprehension of students with latent semantic analysis. In *AI in Education*, pages 545–542, Les Mans, France. Amsterdam: IOS Press.

Yang, F. J. (2001). *Turn planning for a dialogue-based intelligent tutoring system*. PhD thesis, Chicago, Illinois.