Schema Theory-based Computational Approach to Support Children's Conceptual Understanding

by

Zukeri Ibrahim

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computing

February 2006

The candidate confirms that the work submitted is his own and that the appropriate credit has been given where reference has been made to the work of others.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Abstract

Researchers acknowledge the difficulty faced by children in understanding new concepts. Explaining new concepts to children requires supporting their reasoning based on concrete objects and ideas. Human tutors normally use some dialogue to introduce new concepts and tailor the explanations to the prior knowledge of the children. There is a lack of interactive pedagogical agents that guide children's reasoning and adapt explanation to their cognitive state. The design of such agents can be based on learning theories that explain how children understand new concepts, as well as on studies of how human teachers support children's conceptual understanding.

The goal of this research is to develop a computational framework to inform the design of a pedagogical agent capable of engaging in a dialogue that supports children's conceptual understanding. The thesis proposes an approach for Schema Activation and Interpersonal Communication (SAIC) to support cognitive tasks that occur when a child is learning new concepts through one-to-one interaction with a computer agent. The approach is based on schema theory that explains how meaning-making occurs and stresses the importance of prior knowledge, and on the results of an experimental study that identifies strategies human teachers use to support schema-based cognitive tasks.

A novel architecture of a pedagogical agent whose behaviour is based on schema activation and modification is described. The architecture addresses three important issues: describing the process of activation and modification of relevant prior knowledge to be used in introducing new concepts; defining the reasoning and decision making of the agent to promote schema-based cognitive tasks; and providing adaptive explanations tailored to the child's relevant prior knowledge. The schematic knowledge of the SAIC agent is represented as frames, the dialogue is planned as a sequence of dialogue games, and the interaction language is implemented with linguistic templates extracted from a study with teachers. The applicability of the SAIC approach is demonstrated in a multimedia educational system '*Going to the Moon*', as an integrated component in a reading session. An experimental study with the multimedia system has validated the SAIC design approach and has examined the usefulness of the agent in supporting children's conceptual understanding in terms of improving their schematic knowledge.

The thesis makes original contributions to the fields of *Artificial Intelligence in Education* by defining reasoning and decision making based on the principles of schema theory, and by designing a schema-based pedagogical agent to support children's conceptual understanding; *Education* by demonstrating the application of learning theories to inform the design of intelligent tutoring systems; and *Knowledge-based systems* by demonstrating the feasibility of frames as the representation formalism in Intelligent Tutoring Systems, and by proposing some original mechanisms for using frames to design pedagogical agents.

To My Dear Parents

Acknowledgements

I would like to express my gratitude and respect to my supervisors Vania Dimitrova and Roger Boyle for their continuous support throughout the research. Their supervision ensures that this work becomes a success.

Many thanks to MARA who sponsors this research. The financial help I received in Malaysia and the UK is really appreciated.

Thanks to several people who directly and indirectly contributed to this work:

Malaysian teachers who participated in the WoZ study: Halim, Nadzim, Zakaria, Zulhanee, etc.

Debbie Elliot for her expertise in English that helped me to improve the template used by the SAIC agent.

Hardworking teachers at CHALCS: especially Anne Williamson, Eleanor Wood, Mr Bailey, Mr Jalloh, Sharon, and more than fifty wonderful children who took part in the evaluation study.

Karon McBride for her help with the proofreading that made this thesis much easier to understand.

Friendly people at the School of Computing for the stimulating discussions: Essam Kosba, Richardo Mazza, Christoper Buckley, Mohammed Haji, Bayan Sharwar, etc.

Malaysian people for their moral support: Kamariah Ibrahim, Wan Azam, Abdullah Mohd Zain, Shanudin Zakaria, etc.

Abbreviations

AI	Artificial Intelligence
AIED	Artificial Intelligence in Education
CHALCS	Chapeltown and Harehills Assisted Learning Computer School
ITSs	Intelligent Tutoring Systems
LTM	Long-Term Memory
SAIC	Schema Activation and Interpersonal Communication
STM	Short-Term Memory
WoZ	Wizard of Oz

Conventions

In this thesis, the terms *student* and *learner* are assumed to have the same meaning; referring to a child learning new concepts in a guided reading exercise. The terms *teacher* and *tutor* are also assumed equal; referring to a person explaining new concepts to children in a one-to-one interaction.

For the convenience of thesis writing, male gender for the learner and the teacher will be used. The pronoun *we* refers to the author and *our* refers to author’s.

Publications

Some of the work in this thesis has been presented prior to thesis submission:

Ibrahim, Z., Dimitrova, V., Boyle, R. (2005). Evaluation of a Schema-Based Pedagogical Agent to Support Children's Conceptual Understanding, **12th International Conference on Artificial Intelligence in Education AIED05**, Amsterdam, Netherlands, 18-22 July 2005 (poster).

Ibrahim, Z., Dimitrova, V., Boyle, R. (2005). A Schema-based Pedagogical Agent to Support Children's Conceptual Understanding, **Workshop on Mixed Language Explanations in Learning Environments (XLANG)**, in conjunction with **AIED05**, Amsterdam, Netherlands, 19 July 2005.

Ibrahim, Z., Dimitrova, V., & Boyle, R. (2004). Capturing human teachers strategies for supporting schema-based cognitive tasks to inform the design of an intelligent pedagogical agent, **Workshop on Modelling Human Teaching Tactics and Strategies**, held in conjunction with the **7th International Conference on Intelligent Tutoring Systems ITS04**, pp. 5-14. 2004.

Ibrahim, Z., Dimitrova, V., & Boyle, R. (2003). SAIC: A Computational Approach for Supporting Children's Conceptual Understanding. In Young Researchers Track of the **11th International Conference on Artificial Intelligence in Education**. Sydney, Australia, IOS press.

Ibrahim, Z., Dimitrova, V. (2002). Dynamic Adaptive Ontological Modelling to Support Children's Conceptual Understanding. Proceedings of the **6th Human Centred Technology Postgraduate Workshop "Tools for Thought: Communication and Learning Through Digital Technology"**, ISSN 1350-3162, University of Sussex.

Contents

Abstract i

Acknowledgements iv

Abbreviations v

Conventions vi

Publications vii

List of Appendices xiv

List of Figures xv

List of Tables xvi

Chapter 1 Introduction 1

1.1. The Methodology of This Research 2

1.2. Possible Contributions of the Work 3

1.3. The Structure of This Thesis 4

Chapter 2 Schema Activation and Modification to Support Children's Conceptual Understanding 6

2.1. Introduction 6

2.2. Difficulties Faced by Children When Reading to Understand New Concepts 7

2.2.1. Designing a pedagogical agent to help children learn new concepts 8

2.3. Cognitive Foundations: Interpretation of New Information Using Schemas 9

2.3.1. Schema theory 10

2.3.2. Schema activation 13

2.3.3. Schema modification 14

2.3.4. The characteristics of children thinking 16

2.3.5. Deriving design principles from schema theory 17

2.4. Educational Foundations: Viewing Teaching as an Interaction to Support Schema-based Cognitive Tasks 18

2.4.1. Reading comprehension 18

2.4.2. Supporting student's cognitive tasks 19

2.5. Computational Foundations: Relevant Computational Approaches 21

2.5.1. Theoretically-inspired systems 21

2.5.2. Dialogue-based pedagogical agents 24

2.6. Issues to Examine 27

2.6.1. Schema-based strategies used by human tutors to explain new concepts in one-to-one interaction with children 27

2.6.2. Representation of the children's concrete prior knowledge 27

2.6.3. Modelling the schema-based cognitive tasks 28

2.6.4. Maintaining schema-based explanatory dialogue 28

2.6.5. Validation of the design in studies with users 29

2.7. Conclusions 29

Chapter 3 Deriving Schema-based Explanation Strategies: a WoZ Study 31

3.1. Introduction 31

3.2. Justification of Study Design Decisions 32

3.3. Wizard of Oz Experimental Study 34

3.3.1. Objectives 34

3.3.2. Experimental settings 34

3.3.3. Data analysis 37

3.4. Results 40

3.4.1. The use of prior knowledge to help interpret a new concept 40

3.4.2. Speech act of human tutors 41

3.4.3. Dialogue episodes 43

3.4.4. Tutoring strategies for the dialogue episodes 45

3.5. Conclusions 47

Chapter 4 The SAIC Architecture and Student Modelling Mechanism 48

4.1. Introduction 48

4.2. The SAIC Architecture 49

4.2.1. Issues illustrated by the WoZ study 49

4.2.2. The proposed architecture 50

4.3. Knowledge Representation in SAIC 54

4.3.1. Knowledge representation in ITSs 54

4.3.2. Representation of schematic knowledge in SAIC 55

4.3.3. SAIC reasoning to support the explanation strategies derived from the WoZ study 57
4.4. Student Modelling in SAIC 62
4.4.1. Student modelling in ITSs 62
4.4.2. Overlay student model 63
4.4.3. Representation of student schematic knowledge in SAIC 65
4.4.4. Initialising student schematic knowledge using stereotype 68
4.4.5. Opening student models to teachers 69
4.5. Updating Schematic Student Model 70
4.5.1. Approach to update SAIC student model 70
4.6. Conclusions 73
Chapter 5 Dialogue Planning and Adaptive Explanations 74
5.1. Introduction 74
5.2. Information Processing Models of Schema Activation and Modification 75
5.2.1. Supporting schema activation 76
5.2.2. Supporting schema modification 77
5.3. Defining the Schema Activation Process 79
5.3.1. Notations 79
5.3.2. Desription of the information processing model for schema activation 79
5.4. Defining the Schema Modification Process 82
5.4.1. Defining accretion 82
5.4.2. Defining tuning 84
5.4.3. Defining restructuring 86
5.4.4. Defining creation 87
5.5. Schema-based Dialogue 88
5.5.1. Characteristics of the SAIC dialogue 88
5.6. Using Dialogue Games to Plan the Dialogue in SAIC 90
5.6.1. Mechanisms for planning tutoring dialogue 90
5.6.2. Definitions of the main concepts in the SAIC dialogue model 93
5.6.3. Schema-based dialogue games 96

5.6.4. Dialogue games to manage the dialogue in SAIC 99

5.6.5. Switching between dialogue games 100

5.7. Dialogue Utterances Templates 101

5.7.1. Choosing a language to interact 101

5.7.2. Generating dialogue utterances in SAIC 102

5.7.3. Generating SAIC utterances 104

5.8. Conclusions 106

Chapter 6 Implementation of SAIC in '*Going to the Moon*' 107

6.1. Introduction 107

6.2. The Domain: Introductory Astronomy 107

6.2.1. The suitability of the domain for a reading session 108

6.2.2. Preparation of the lesson 109

6.3. Authoring a Multimedia System '*Going to the Moon*' 110

6.3.1. Issues to consider when authoing multimedia educational system 110

6.3.2. Authoring process of the multimedia system 111

6.3.3. Textual and graphical media elements in '*Going to the Moon*' 113

6.4. Developing a SAIC Agent 114

6.4.1. Student cognitive module 115

6.4.2. Dialogue modeller 117

6.4.3. Interface of the SAIC agent 119

6.5. Implementation of SAIC in an Educational Multimedia System 120

6.5.1. Integrating the SAIC agent into Going to the Moon 120

6.5.2. Calling and interacting with the SAIC agent 121

6.6. Examples of SAIC Adaptive Explanations 121

6.6.1. Example of a diagnose game 122

6.6.2. Example of an activation game 123

6.6.3. Example of a tuning game 126

6.7. Conclusion 126

Chapter 7 Evaluation of the SAIC Pedagogical Agent 128

7.1. Introduction 128

7.2. An Overview of Tutoring System Evaluations......129
7.3. Aims of the Evaluation......130
    7.3.1. Validating the SAIC design principles......130
7.4. Formative Evaluation of the SAIC Agent Prototype......131
    7.4.1. Goals of the formative evaluation......131
    7.4.2. Experimental design of the formative evaluation......131
    7.4.3. Results of the formative evaluation......132
7.5. Summative Evaluation of the SAIC Agent......133
    7.5.1. Summative evaluation method......134
    7.5.2. Data analysis......138
7.6. Results of the Analysis......141
    7.6.1. The effectiveness of the SAIC agent......141
    7.6.2. The interactions with the SAIC agent and the explanation of domain concepts......143
    7.6.3. Types of improvement made by the participants......146
    7.6.4. The usefulness of the SAIC agent: teacher and student interview......150
7.7. Discussion: Validating the SAIC Design Principles......155
    7.7.1. Design Principle 1: During the interaction with the child, the agent has to activate prior knowledge and use it to introduce new information......155
    7.7.2. Design Principle 2: The agent should promote the cognitive tasks proposed by schema theory......156
    7.7.3. Design Principle 3: The agent should provide concrete examples and should avoid the use of abstract concepts......158
7.8. Summary: Analyses, Findings and Limitations of the Study......159
    7.8.1. Analyses......159
    7.8.2. Findings......161
    7.8.3. Limitations of the study......162
7.9. Conclusions......163
Chapter 8 Conclusion......164
8.1. Summary......164
    8.1.1. Adequacy of the SAIC approach......165
    8.1.2. Generality of the SAIC approach......166

8.2. Contributions of This Work 168

8.2.1. Contributions to AIED research 168

8.2.2. Contributions to Education 169

8.2.3. Contributions to knowledge-based systems 169

8.3. Future work 169

8.3.1. Possible applications of the SAIC approach 169

8.3.2. Potential improvements of the SAIC approach 173

References 178

Appendices

Appendix A	**List of all new concepts in the lesson and examples of their schematic knowledge.**
Appendix B	**The usual format of a CHALCS reading session.**
Appendix C	**Guidelines for teachers.**
Appendix D	**Student allocation into control and experimental groups.**
Appendix E	**Samples of pre-test transcripts.**
Appendix F	**Post-test briefing script.**
Appendix G	**Samples of post-test transcripts.**
Appendix H	**Samples of log files.**
Appendix I	**Questions for student group interview.**
Appendix J	**Transcripts for student group interview.**
Appendix K	**Questions for teacher interview.**
Appendix L	**Transcript for teacher interview.**
Appendix M	**Analysis of the student explanations.**

List of Figures

Figure 3.1 The components of the Woz-based system architecture 36

Figure 3.2 A screen shot of the prototype used in the study 36

Figure 4.1 The proposed architecture of a SAIC pedagogical agent integrated in an educational multimedia system. The components of the SAIC agent are shown in the shaded area 51

Figure 5.1 Activating relevant prior knowledge from the LTM into the STM – an example 77

Figure 5.2 Modifying the activated schema – an example 78

Figure 6.1 The front page of the multimedia system..113

Figure 6.2 A screenshot of the multimedia system and the components of the multimedia presentation...113

Figure 6.3 The components of the SAIC agent's interface 119

Figure 6.4 This screenshot shows the SAIC agent is activated and integrated into the multimedia presentation 120

Figure 7.1 The number of improvements made by the students in the control and experimental groups. Each improvement indicates better schematic knowledge of one domain concept........................142

List of Tables

Table 3.1 Atomic speech acts used by the tutors in the WoZ study. The acts are adapted from DISCOUNT (Pilkington, 1999)........................41

Table 3.2 Complex speech acts used by the tutors in the WoZ study.................42

Table 3.3 Schema-based dialogue episodes identified in the WoZ study 44

Table 3.4 Dialogue episodes to manage dialogue and their definitions 45

Table 3.5 Tutoring strategies used by the tutors to perform dialogue episodes 46

Table 5.1 Agent dialogue rules 95

Table 5.2 Student dialogue rules 96

Table 6.1 List of 18 new concepts in the lesson 'Going to the Moon' 109

Table 7.1 The description of the data collected in the experimental study 138

Table 7.2 The number of improvements for each ability of the experimental group. The numbers indicate how many students have demonstrated improvement for 0, 1, 3, 4, 5, 6 and 7 concepts, respectively...143

Table 7.3 The domain concepts, number of triggers and number of improvements 145

Table 7.4 The number of improvement type made by the control and experimental groups 147

Table 7.5 The number of improvement type made by students in each ability group 148

Table 7.6 Why the students enjoyed interacting with the SAIC agent.............153

Table 7.7 What the students enjoyed most when interacting with the SAIC agent...154

Table 7.8 Reasons why some students were unhappy when interacting with the SAIC agent...154

Table 7.9 Dialogue episodes used by the SAIC agent and their frequency in the interaction 157

Table 7.10 A summary of tutoring strategies used by the SAIC agent to perform the dialogue episodes and the frequency of invocation for each strategy 158

Chapter 1

Introduction

Children's conceptual understanding can be developed using guided reading exercises based on non-fiction texts. Although 'reading for understanding' is a skill that children are encouraged to acquire at an early age, most researchers agree that students find it difficult to learn using this method and continue to need support from their teachers as they acquire cognitive skills and learn new concepts. There is a need for research into new and better ways for providing support for children's learning during text-based activities.

It is generally acknowledged that successful teachers are skilled at promoting the sort of reasoning processes that develop children's conceptual understanding. Experienced tutors have comprehensive domain knowledge, know their students personally and are able to adapt their explanations to individual students who, as a consequence, gain deeper understanding of new concepts. As a result of receiving this kind of help, children are able to understand new concepts more effectively and become autonomous readers. Without such support, the students are less likely to be able to understand the new concepts presented in their guided reading lessons. Therefore, it is important to find ways of optimising the support available to help students and guide their reasoning processes.

One way forward might be to employ more human tutors. Unfortunately, most educational institutions find it difficult to provide human tutors in sufficient numbers to support students adequately and this is especially true in the case of guided reading because it is an activity that demands a series of one-to-one interactions between the tutor and the student. It often proves too costly and impractical to provide such intensive support.

An alternative solution would be to use computers in the classroom to replace some or all of a human tutor's role and support students as they learn new concepts. This would enable institutions to provide the one-to-one support necessary for effective learning from guided reading at a much lower cost than at present. A pedagogical agent based on recent advances in research into multimedia and artificial intelligence and capable of explaining new concepts to students could simulate some or all of the functions of an experienced human tutor. The domain and student knowledge available to such an agent would be represented in knowledge bases, and artificial intelligence methods used to infer explanation strategies that best matched a student's learning needs. If based on an appropriate learning theory, this new pedagogical agent would generate adaptive explanations similar to those supplied to students by human tutors. Research in the field of Artificial Intelligence in Education (AIED) has shown that existing tutoring systems do not in general support cognitive tasks in a way that leads to conceptual understanding (Luckin & du Boulay, 1999; Aist, 2001). Thus, the goal of our research is to build and test a pedagogical agent that is capable of supporting children's conceptual understanding.

The design of a pedagogical agent would require two lines of enquiry: theory and experiment. A theoretical consideration of the interpretation process would define the tasks to be supported by the pedagogical agent and experimental work would identify the explanation strategies to be employed by the computer tutor. This combination of theoretical and practical work would create a framework of teaching principles and strategies based on an appropriate learning theory.

This thesis makes the case for the use of a pedagogical agent capable of supporting children as they learn new concepts through guided reading activities. A pedagogical agent designed in such a way that it simulates the explanation strategies applied by human tutors should support children's learning effectively. Our hypothesis is that an interaction between a child and a pedagogical agent that supports the child's interpretation process will help the child *better understand new concepts*, and that educational systems incorporating the pedagogical agent will be *more effective at explaining new concepts* than educational systems without it.

Following the discussion above, our *research questions* are:

- How do children understand new concepts in guided reading exercise?
- How do human teachers support children's conceptual understanding in one-to-one interactions?
- How do we design a pedagogical agent capable of supporting children's conceptual understanding?
- Is the pedagogical agent effective and useful at supporting children's conceptual understanding?

1.1 The Methodology of This Research

The objective of this study is to design and test an architecture for a pedagogical agent capable of supporting children's conceptual understanding. Following Self's (1999) ITS research methodology that combines theoretical and empirical investigations, we have conducted this research undertaking the following steps:

- Investigation of a learning theory that explains how humans understand new concepts. We refer to the learning theory called schema theory (Bartlett, 1958) that explains how humans understand a new concept; the teaching principles of this learning theory will be analysed in order to derive design principles for our pedagogical agent.
- Examination of tutoring strategies used by human teachers to identify how the agent will explain new concepts to children. These strategies will define the teaching knowledge of the agent and will be used as a key source for dialogue planning.
- Precise description of the behaviour of the agent including a mechanism for utilizing the domain expertise, reasoning about the student's conceptual understanding, and conducting explanatory dialogue to promote schema-based cognitive tasks.
- Implementation of the pedagogical agent to illustrate the validity of the formal description. It is critical that the agent will be implemented in an educational system to demonstrate its role in learning environments.
- Evaluation of the pedagogical agent integrated into an educational system in a real setting with students to validate the design principles and examine the potential benefits of the approach.

1.2 Possible Contributions of the Work

This work is expected to contribute to the fields of AIED, education and knowledge-based systems.

Artificial Intelligence in Education (AIED). The design of an interactive pedagogical agent capable of explaining new concepts to primary schoolchildren and to support their conceptual understanding will be a new application of artificial intelligence techniques to multimedia educational systems. In particular, a design architecture for a pedagogical agent based on the principles of schema theory will be established and the new agent will build on an examination of dialogue strategies to support schema-based cognitive tasks and interact with the students to support cognitive tasks. Using schema-based reasoning, the agent will be capable of simulating the help offered by human teachers by generating adaptive explanations tailored to the needs of an individual student.

Education. An analysis of schema theory will provide new teaching principles that could be considered by the designers of pedagogical agents. An examination of dialogue strategies based on the principles of schema theory will provide insight into how cognitive tasks intended to develop conceptual understanding might be supported. The agent will adapt its explanation according to the thinking characteristics of the primary schoolchildren, and this will result in guidelines that human tutors can employ when teaching reasoning skills to children in this age group.

Knowledge-based systems. Our new pedagogical agent will make a contribution to the field of knowledge-based systems by utilising schematic knowledge. The reasoning and decision-making processes of the pedagogical agent will be based on schematic domain knowledge and will draw on information contained in a model of the child's schematic knowledge. The pedagogical agent will include a computational model of natural language interaction between the agent and a child in a one-to-one interaction that is novel because it demonstrates an application of the generation of explanations through the use of reasoning based on schema theory.

1.3 The Structure of This Thesis

This thesis is organised into four parts:

Part 1: Motivation and methodology

Our motivation and methodology are presented in chapters 2 and 3. In Chapter 2 we set out our motivation for our research through a review of relevant existing work and thereby establish the theoretical basis of the research and identify the main issues requiring further investigation. In this chapter we consider the difficulties faced by children when learning new concepts and view the problem from cognitive, educational and computational perspectives. In Chapter 2 we also discuss in detail the potential benefits of activating prior knowledge and performing the modification of existing schemas when devising personalised explanations. We summarise the issues that have not yet been addressed in previous research and justify the need for our study.

Chapter 3 describes the experimental basis of the computational framework proposed in this thesis. We discuss the strategies used by experienced human tutors to diagnose students' knowledge, activate relevant prior knowledge and explain new concepts. In this chapter, we also discuss how the pedagogical agent will interact with the student. The findings of the experimental study presented in Chapter 3 identify the agent's speech acts, dialogue episodes and dialogue strategies.

Part 2: Computational Framework

The second part consists of chapters 4 and 5 that form the core of this thesis. In Chapter 4, we propose a novel design architecture: this is the computational framework we will use to design a pedagogical agent based on the results of the theoretical and experimental work discussed in chapters 2 and 3. We describe the functionality and operation of the main components of the agent and present in detail the knowledge representation and the student modeling mechanism.

Chapter 5 describes the support to be provided to the children by the agent and its dialogue mechanism. In this chapter we define how the adaptive explanation of new concepts will be generated and communicated to the student using a template-based natural language. Chapter 5 is an in-depth presentation of our computational framework for the personalised interactive explanation.

Part 3: Prototype and validation of the framework

The prototype phase of our research, which includes implementation and evaluation, is presented in chapters 6 and 7. In Chapter 6, we present the pedagogical agent integrated in an educational system to demonstrate how it works in educational settings. This chapter demonstrates an application of the ITS design approach proposed in this thesis; in it we demonstrate how to develop the pedagogical agent following the computational framework defined in chapters 4 and 5, and how to integrate the agent into existing multimedia educational systems. Chapter 7 presents an evaluation of the pedagogical agent carried out to study the agent’s possible effectiveness and its usefulness in supporting children’s conceptual understanding. The study provides a validation of the design principles adopted during the development of the agent.

Part 4: Conclusion and future extension

In Chapter 8, we present the conclusion of this thesis. This chapter describes our achievements, identifies the work’s limitations and makes suggestions for future work that would lead to improvements to the computational framework presented in this thesis. The thesis provides a design architecture for the interactive explanation of new concepts that is open to further refinement, as pointed out in Chapter 8.

Chapter 2

Schema Activation and Modification to Support Children's Conceptual Understanding

2.1 Introduction

The goal of this work is to design a pedagogical agent capable of supporting children's conceptual understanding. This chapter provides a theoretical framework for the examination of how an understanding of the principles of schema theory would inform the design of the pedagogical agent.

One of the main issues in Artificial Intelligence in Education (AIED) research is how to inform the design of Intelligent Tutoring Systems (ITSs) capable of supporting students' learning. The design of ITSs can be informed by observation of how human teachers teach or how students learn or by examination of a learning theory that explains how learning occurs. In contrast to the observation approach, the examination approach provides design principles derived from an established learning theory. The design based on learning theory can be formally described following research on the learning theory and the application of the theory in education. Design based on sound learning theories may help students become autonomous learners. Moreover, the design may help students realise the process required to understand new concepts.

The aim of this chapter is to present an overview of supporting conceptual understanding from cognitive, educational and computational perspectives. The overview is intended to motivate the research questions addressed in Chapter 1. We will outline the potential of designing a pedagogical agent that is informed based on schema theory, and will refer to some problems that need further investigation. We will also argue that interaction is important in supporting children's conceptual understanding as it provides means to support the explanation process. We will draw out research issues that interaction brings to light and will relate them to questions in informing the design of pedagogical agents based on learning theory.

Section 2.2 describes the difficulties faced by children when reading to understand new concepts. In sections 2.3, 2.4 and 2.5, we provide cognitive, educational and computational foundations for understanding new concepts, respectively. Then, further issues to examine are outlined in Section 2.6.

2.2 Difficulties Faced by Children When Reading to Understand New Concepts

In classroom situations, reading is one of the main ways of learning for children. Aist (2000) argues that children need sufficient reading skills in order to learn and understand new concepts. The understanding of a written text requires knowledge, such as pragmatics and semantics, to enable the reader to understand the author's message in the text (Braunger & Lewis, 1998). Without the required skills and knowledge, children may encounter a variety of difficulties to read and understand new concepts presented in a lesson. In this thesis, the child is assumed to have some knowledge about the syntactic part of the text, e.g. noun, verb and adjective, and know how to pronounce the words in the lesson. These assumptions specify the focus and scope of this thesis, i.e. to help children alleviate the difficulties by addressing the skills involved in reading to understand.

Biddulph (2002) defines reading as "a construction of meaning from written text." Following the definition, this thesis focuses on one specific difficulty: *to make sense of new concepts* the students read in a new lesson. Thus, the difficulty is defined as a situation in which the student is unable to effectively understand a new concept due to the lack of the required meaning-construction skill, which can be explained in terms of the student's inability to effectively relate new concepts to his prior knowledge. The definition of the problem points to the need of a more able learning partner (Luckin, 1998), e.g. a human or a computer tutor, for assistance during the reading process. The mental process involved in the meaning construction will be discussed further from *Cognitive perspective* in Section 2.3.

The complexity of learning processes has been described by several researchers, e.g. Hunter (1964) and Rumelhart and Norman (1978). They stress the relation between a student's current and previous learning, and a variety of activities involved in the process. In learning new concepts through guided reading, students do not only read aloud the lesson, but also make an effort to understand the lesson by interpreting the messages of the author (Anderson & Pearson, 1988). This implies that to understand adequately new concepts presented in a lesson, a reader needs sufficient skills both to turn the print into sound, and at the same time to comprehend the meaning of the words in the lesson. Reading comprehension will be discussed further from the *Education perspective* in Section 2.4.

In traditional classroom situations (without computers), especially reading sessions, human tutors help children to alleviate the difficulty by explaining the complex new concepts (National Institute of Child Health and Human Development, 2000). Without help, the children may never become fluent readers and are more likely to drop out of school (National Center to Improve the Tools for Educators, 1996). Thus, it is important for the children to have a helper (Fomichov & Fomichova, 2000; Luckin, 1998) when reading lessons that have new concepts to ensure that they can understand the lessons. However, as highlighted by Aist (2000), it will take a lot of time for a teacher to interact with the children, understand the individual problems of each child, and explain the new words in the lesson tailoring the explanation to each child. Hence, it is difficult for human tutors to provide sufficient support to every child in guided reading lessons. Several works address the difficulties faced by the children in reading process, for example augmenting text with facts (Aist, 2001), selecting the reading materials (Aist & Mostow, 2000), modelling students' reading proficiency (Beck & Sison, 2004) and scaffolding children's reading comprehension using automated questions (Beck, Mostow & Bey, 2004).

A computer-based learning environment may be able to help the children by simulating the support provided by human tutors in guided reading exercises. The idea of having a computer tutor or a *pedagogical agent* in this learning environment is promising because of the assumed adaptive nature of the human support, which can be simulated if the agent has the required teaching knowledge and communication capability. Johnson (1998) defines pedagogical agents as "autonomous agents that support human learning, by interacting with students in the context of interactive learning environments." Following this definition, a student's conceptual understanding can be supported through interaction with a pedagogical agent (as discussed in Section 2.4). Therefore, we argue that reading with the help of a pedagogical agent that is always ready to explain new concepts may increase the children's conceptual understanding.

2.2.1 Designing a pedagogical agent to help children learn new concepts

Self (1999) proposes a methodology for designing ITSs that emphasises the need for precise design principles to develop ITSs or, in our case, a pedagogical agent that will help the children learn new concepts. Following the methodologies for capturing pedagogical expertise in intelligent tutoring systems outlined by du Boulay and Luckin (2001), we can look at theories that explain how people learn new concepts in order to design the behaviour of the pedagogical agent. Learning theories can offer sufficient guidelines for human teachers but the principles derived from the theories may still be vague in terms of implementation in computer systems. Design principles to inform the design of the pedagogical agent will be presented in Section 2.3.5.

Other approaches to capturing teaching strategies proposed by du Boulay and Luckin (2001) are: observing how human teachers explain new concepts or how children learn new concepts in one-to-one interaction. However, considering the different nature of the learning process between a pedagogical agent and a child, e.g. text-based and human-computer interaction, observation of human teachers or students may give general teaching guidelines and not reflect the appropriate approach to learning new concepts using a pedagogical agent. Thus, we argue that, to design the agent, it is important to refer to an established theory that explains how people learn new concepts (discussed in the next section).

Considering the issue of designing computer tutors based on principles of learning theory from AIED research, we need to discuss the issue from the main closely-related research areas: Cognitive Psychology that discusses how human learn new concepts, Education that discusses how teachers teach new concepts and how students learn new concepts, and Computer Science that discusses relevant tutoring systems that intend to support students. These research areas provide theoretical foundations to the work described in this thesis.

2.3 Cognitive Foundations: Interpretation of New Concepts Using Schemas

Several researchers have referred to schema theory for the explanation of the processes that occur when students comprehend or do not comprehend new concepts they perceive; listen, look or read (Bartlett, 1958; Pressley, 1998; Rumelhart & Norman, 1978 & 1981; Schank, 1982). To inform the design of the pedagogical agent capable of supporting children's conceptual understanding, we thereby refer to the explanation provided by schema theory. Although schema-based reasoning has been extensively studied and the suitability of schema theory for the design of learning systems has been acknowledged (Marshall, 1995), to the best of our knowledge there are no computational architectures of intelligent tutors based on schema theory.

Our goal is to examine whether schema theory could be used as a basis for the design of a pedagogical agent and what the benefits of such an agent can be. In this respect, our work is original. It has to be acknowledged that schema theory is one possible model of the human learning. Other researchers have examined different theories, e.g. ACT* (Koedinger et al., 1997), Zone of Proximal Development

(Luckin & du Boulay, 1999) and constructivism (Akhras & Self, 2000). These theories and their applications will be discussed in Section 2.5.

2.3.1 Schema theory

Bartlett (1958) describes schemas in terms of human memory and explains the role of schemas in remembering stories or information; schema determines how a story is interpreted and remembered. To illustrate his idea of schema, Bartlett shows how different people have a variety of interpretations of a story *War of Ghosts*, and argues that the different interpretations of the story by the people are caused by the different prior knowledge of the listeners, and subsequently states that prior knowledge is the underlying factor of the multiple interpretations. In line with the idea of viewing human memory as schemas, Arbib (1989) reports that access to information in human memory can be enhanced if the memory is well connected. This discussion points to the need of a further analysis of the role of schemas in children's learning and the use of schemas in their reasoning, which can address the difficulties faced by children (discussed in Section 2.2) and hints the steps to help the children.

Researchers have given several definitions of schema:

- **Data structure.** Rumelhart (1980) defines schema as data structures for representing the *generic concepts* stored in memory, and describes the function of schema as *the building blocks of cognition*. Henk (1993) describes schema as representing the knowledge structures in the learner's minds, and these structures allow learners to connect new information to what they already know. This definition reflects the declarative nature of human knowledge.
- **Script**. Schank and Abelson (1977) describe schema in terms of a *script* that is a kind of data structure containing a specific sequence of events in familiar contexts, people's expectation from those contexts, and application to other similar contexts. Similarly, Holland and Quinn (1987) describe schema as *story-like chains of prototypical events that unfold in simplified worlds* that follow the ideas of a script. In contrast to the definitions given by Henk (1993) and Rumelhart (1980) discussed above, the script notion closely refers to procedural knowledge.
- **Frame**. Minsky (1975) defines schema as a *frame*, which is more closely related to knowledge representation in computer systems; as an alternative to other formalisms such as rule-based and logic-based. A closely related definition is given by Marshall (1995) who describes

schema as a means to store and process information in terms of schema attributes, values and instantiations.

In general, all the definitions describe schema in terms of knowledge representation and application in the human reasoning process. In case of supporting children learning new concepts, the assistance required by the children corresponds to supporting the mental processes of accessing the represented knowledge and its application to interpret the concepts, which would be informed by an analysis of schema theory.

This thesis follows the definitions given by Minsky (1975) and Marshall (1995) whose idea of schema can be executable in computer systems. In addition, the definitions are closely related to the Kalyuga's (2003) idea of student schematic knowledge represented as *function* (what a concept performs), *structure* (the components of the concept) and *process* (how the concept performs its function). This approach to assessing student schematic knowledge before and after an instructional event is compatible with the definitions of schema given by Marshall (1995) and Minsky (1975), and can be stored in a knowledge base. We will adopt this assessment approach to measure the student's schematic knowledge before and after learning new concepts with the agent as a way to evaluate the effectiveness and usefulness of the approach proposed in this thesis.

The underlying assumption of schema theory is that comprehension of new concepts is based on relevant prior knowledge or schema (Bartlett, 1958, Schank, 1982). In line with the idea, D'Andrade (1992) states that the application of an appropriate schema in a new context causes the comprehension. Thus, the explanations given by the schema theorists stress the crucial role of schemas in the reasoning process.

Kalyuga (2003) illustrates the process of interpreting new concepts in terms of activating already constructed schemas from a student's long-term memory (LTM) into short-term memory (STM), and modifying the activated schema. In a guided reading situation, this means the student has to interact with a teacher or some learning material to perceive a new concept and to perform tasks of activating relevant schemas and modifying them appropriately. The validity of the theoretical explanation offered by schema theory is emphasised by Schwartz et al. (1998) who refer to schema theory to study a student's knowledge structure and its relation to the recall process. A human or computer tutor must consider the mental process involved in a reading activity when supporting student's conceptual understanding.

The theoretical explanation of schema theory emphasises the mental process a child consciously or unconsciously performs when he reads domain concepts presented in a lesson (Marshall, 1995). Though the undertaken mental tasks are assumed to be the same for every child, the interpretation may be different based on the child's prior knowledge (Pressley, 1998).

Researchers have applied the ideas of schema theory in several areas, for example:

- Understanding and remembering stories (Bartlett, 1958; Black & Bower, 1980). Listeners understand and remember a story based on their prior knowledge. Depending on prior knowledge, a story may be understood and remembered differently by different people.
- Problem solving (Marshall, 1995). An individual uses schema to create a mental model about the current problem. The mental model influences how the problem is solved.
- Reading comprehension (Ajideh, 2003; Pressley, 1998). Readers use their schema to comprehend texts they read. The reader and writer may understand the text differently.
- Intercultural communication (Nishida, 1999). People understand a concept differently because of their different cultural background. The speaker should consider the cultural background of the listener to have an effective communication.
- Instructional design (Chou, 2000). Designers of instructional materials should consider the relevant prior knowledge of the students that may influence how the students construct new knowledge.

In brief, all the researchers agree with the idea of schema theory and conclude that human understanding depends on relevant schemas or prior knowledge. The existing work points to the importance of this research where a pedagogical agent will be designed as a new and potential application of schema theory in a computer-based learning environment. The development of the agent will require a novel design architecture that mainly incorporates the schematic representation of student knowledge and its application in conceptual understanding, as will be discussed in chapters 4 and 5.

Derry (1996) supports the application of schema-based teaching principles by claiming that the theory describes the objectives to achieve in instruction and how to achieve them. Adopting this idea, the goals of the agent's instruction are: to support learning following the principles of schema theory and to explain new concepts in a lesson in a way that simulates a human teacher's explanation to a child in a one-to-one interaction. The vision to have a schema-based pedagogical agent implies the requirement to present teaching principles based on schema theory in a precise way, which will enable their execution in a computer system. Accordingly, explanation strategies will be required to effectively explain new concepts in a computer-based learning environment.

Following the discussion above, a pedagogical agent designed based on the principles of schema theory is expected to help children relate their prior knowledge to the new concepts in a lesson and promote schema-based reasoning in interpreting new concepts. The expected benefits of the pedagogical agent will be discussed in Chapter 7, where the evaluation stage will examine both the potential effectiveness of the agent in explaining new concepts to children, and its usefulness in teaching and learning new concepts.

The fundamental issue is to understand in an integrated manner how children activate their existing relevant prior knowledge or schema and use it to understand new concepts.

2.3.2 Schema activation

Learning of new information is successful, i.e. comprehension occurs, if a person can connect new information to his *existing schema* (Marshall, 1995). The mental process of activating relevant prior knowledge explains how a schema stored in human memory is called for a reuse in the interpretation of a perceived concept, which Marshall describes in terms of instantiation; replacing the values of activated schema with particulars of the perceived concept. Adopting the theoretical claim, a schema-based pedagogical agent needs to activate appropriate schemas and change their values accordingly.

Dochy (1992) discusses that we should consider the nature of prior knowledge that may influence its activation; mainly the availability of prior knowledge and its accessibility during a text comprehension. In addition, Dochy also states that prior knowledge may contain misconceptions, i.e. conceptions that are wrong, such as being convinced that the sun orbits around the earth. In this thesis, it is assumed that the pedagogical agent helps student activate the *correct schemas* and guides the student's reasoning accordingly. The availability of relevant prior knowledge, its representation and application in generating explanations will be discussed in chapters 4 and 5.

Bransford (1979) highlights the need of a teacher in a learning process by reporting that learners do not normally activate and use schemas without explicit instruction to do so. The failure to effectively use schema in learning may be due to the children's lack of cognitive skills to perform the tasks (schema activation and modification) or simply being unaware of the relation between a new concept with existing schemas. To deal with this problem, researchers have proposed several pre-reading strategies to help the learners, for example the use of maps (Schwartz et al., 1998), thematic organizers (Risko & Alvarez, 1986) and concept maps (Novak & Gowin, 1984). These types of pre-reading strategies explicitly and graphically show how new concepts are interrelated and possible relationships between new concepts and existing prior knowledge of the learners. However, the strategies do not specify what prior knowledge the learners have and what actions to take when a relationship between a new concept and prior knowledge is found. Moreover, the use of these pre-strategies implies that teachers and learners need a separate and long explanation to show interrelationships between new concepts that may disturb a current guided reading session. In the case of developing a pedagogical agent, it will also need strategies to help children activate and use prior knowledge. In guided reading situations where teachers interact face-to-face with the children, the use of maps, concept maps or organizers may be effective. However, in the unique interaction style between the agent and the children, different strategies may be needed, as discussed in the next chapter.

2.3.3 Schema modification

Schema activation and modification are two distinct cognitive tasks, where a teacher can help with the activation but the student may not consequentially use the activated schema in his reasoning to understand domain concepts (Brandsford, 1985). In this thesis, we will follow Bransford and will utilise schema modification to refer to the use of the activated schema as a basis to understand new concepts.

To illustrate the idea of schema modification and relate it to a learning process, Piaget and Inhelder (1969) propose three constructs: *assimilation* - when a person interprets a perceived object or idea in terms of existing schemas, *accommodation* – when an activated schema is modified to provide consistency with the new object or idea, and *equilibration* - occurs when the person manages to assimilate or accommodate a new concept into his existing mental structure. Hence, these constructs not only define the use of existing schemas but also can explain two associated operations on the schema: the update of existing schemas and formation of new ones. The constructs reflect the dynamic nature of human knowledge and should be taken into account when developing a pedagogical agent based on the principles of schema theory, especially to inform the design of student and domain knowledge modules for the agent.

Rumelhart and Norman (1978, 1981) have extended the constructs discussed above, and proposed three learning processes to account for human reasoning based on *schema modification*: *accretion, tuning* and *restructuring.* Accretion occurs when an existing schema from the prior knowledge is directly used to interpret a new concept, e.g. `Mercedes` can be considered as another example of a `car`, i.e. the schema of `car` is instantiated to understand `Mercedes`. *Tuning* takes place when an existing schema has to be slightly changed in order to understand a new concept, e.g. a schema for `car` may be modified to include a new property – `engine_power` – or to modify the value of an existing property – `number_passengers < 6`). *Restructuring* occurs when an existing schema has to be significantly modified to create a new schema that will accommodate the new information, e.g. `truck` can be created by modifying `car` to include new properties – `carries_cargo` and `used_for_hauling`). In addition to these learning processes, Rumelhart and Norman (1978, 1981) also discuss the *creation* of a new schema when an appropriate schema in the prior knowledge cannot be found. Creation is based on stating the definition of the schema and, although very straightforward, rarely leads to meaningful learning. We will follow the detailed descriptions of the proposed learning processes to inform how our pedagogical agent performs schema modification.

Kalyuga (2003) depicts the schema activation process as a continuous retrieval of relevant schemas from human memory, and the schema modification process as an application of the activated schemas in new contexts or creation of new schemas. We will call these processes *cognitive tasks* to account for the fact that people (children in our case) undertake some cognitive activities to understand new concepts. Following the discussion above, we distinguish the following cognitive tasks that can be performed by children when assisted by a pedagogical agent adopting a schema-based teaching approach:

- *Activation*: The child activates some part of their prior knowledge needed for accretion, tuning, or restructuring.
- *Accretion*: The child uses an existing schema, without modification to its structure, as a mental model to interpret a new concept.
- *Tuning*: The child uses an existing schema, with a slight modification of a property or a property value (i.e. adding a property or changing a property value), as a mental model to interpret a new concept.
- *Restructuring*: The child uses one or more existing schemas, with a significant modification of its structure (i.e. adding or deleting properties), as a mental model to interpret a new concept.

- *Creation*: The child creates a schema of a new concept without using a mental model from his prior knowledge.

Schema activation and modification define the cognitive tasks that occur when a child is reading a lesson. A pedagogical agent that wants to help the child with learning of new concepts in the lesson has to support these cognitive tasks. Consequently, these cognitive tasks will be projected in the goals of a schema-based tutoring agent. Following the suggestion of Price and Driscoll (1997) to concentrate on schema construction strategies, we should focus on strategies to perform these tasks. In case of supporting children learning new concepts, the pedagogical agent needs teaching or explanation strategies, which will define how the agent interacts with a child. The strategies will be examined in Chapter 3.

2.3.4 The characteristics of children thinking

The activation and modification processes discussed in the section above apply to all learners. However, helping children learn new concepts is different from helping adults: explaining new concepts to young children requires supporting the reasoning at their cognitive development stage (Eisenberg, 1999; Ell, 2001; Piaget, 1977). Piaget reports that children aged seven to eleven years old *think logically but require concrete objects or ideas* while adults' reasoning is characterised by the ability to use abstractions, such as variables. These characteristics refer to the cognitive aspect of a guided reading exercise. Knowles (1984) highlights the characteristics of adults as being independent learners and having specific aims to achieve; because of their vast life experience that children lack. Therefore, pedagogical agent developers must consider how children think in order to generate adaptive explanations.

Several researchers have shown that children learn differently from adults, e.g. the need for reward to motivate the children (Montessori, 1973), lack of learning skills which can hinder their learning (Brna & Cooper, 2002), and use of learning tool to help them to construct knowledge (Papert, 1993). These characteristics refer mainly to the social aspect of a guided reading exercise that may influence how children think. Woolf (2003) reports that there are no intelligent tutors developed following the children's reasoning characteristics described by Piaget's (1977) *Genetic Epistemology*. The reasoning characteristics may influence how schema activation and modification occurs. Thus, we argue that a schema-based pedagogical agent should consider the learning characteristics in order to effectively support children.

2.3.5 Deriving design principles from schema theory

The major claims of schema theory are examined here to derive principles for the design of the pedagogical agent. The underlying idea of schema theory is that learning will be successful when new information is appropriately related to prior knowledge or existing schemas (Marshall, 1995). Hence, the first claim followed is:

> **Theoretical claim 1:** The understanding of a new concept is based on existing schemas.

Based on this claim, the first design principle for the agent was formulated as:

> **Design principle 1:** During the interaction with the child, the agent has to activate prior knowledge and use this knowledge to introduce new information.

The learners' reasoning involves schema activation and modification processes (Rumelhart & Norman, 1978; 1981). We called these processes *cognitive tasks*. The second theoretical claim we have followed is:

> **Theoretical claim 2:** The interpretation of a new concept is performed using one of the learning modes: accretion, tuning, restructuring, or creation.

Based on this claim, the second design principle for the agent was formulated as:

> **Design principle 2:** The agent should promote the cognitive tasks proposed by schema theory.

Children at the Concrete Operational Stage (mainly aged seven to eleven years old) think based on concrete objects or ideas (Piaget, 1977), as opposed to adults who think based on both concrete and abstract objects or ideas. Therefore, the third claim followed is:

> **Theoretical claim 3:** Children at the Concrete Operational Stage reason based on concrete objects or ideas.

This led to the third design principle:

> **Design principle 3:** The agent should provide concrete examples and avoid the use of abstract concepts.

While schema theory provides general guidelines of how to design the agent, it does not give sufficient understanding of what dialogue strategies should be used by a pedagogical agent to promote schema-based cognitive tasks. We, therefore, need to conduct study to capture strategies employed by human teachers to explain new concepts, as will be explained in the next chapter.

2.4 Educational Foundations: Viewing Teaching as an Interaction to Support Schema-based Cognitive Tasks

The importance of prior knowledge as a determining factor for reading comprehension is acknowledged by researchers, e.g. Anderson and Pearson (1984). Weibelzahl and Weber (2002) stress that prior knowledge shape how the students learn and what they will understand. The role of prior knowledge in reading comprehension is discussed in terms of schema theory. This section relates schema theory to reading comprehension and discusses how to support the reading process.

The support focuses on a *guided reading approach* where a teacher guides a student while he learns. One apparent characteristic of the reading approach is that it requires active participation from both the teacher and student (Mosenthal et al., 2001; Young, 2003), which is different from other reading approaches such as *reading alone* and *reading aloud.* A guided reading approach is normally practised in reading session classes where a small group of students take turns to read a book and explain what they understand from the lesson. The approach points to the necessity of an interaction between the teacher and student to guide the student's reasoning.

2.4.1 Reading comprehension

Schema theory explains reading comprehension as "a reader comprehends a message when he is able to bring to mind a schema that gives a good account of the objects and events described in the message" (Anderson, 1983). Similarly, Bransford (1985) as well as Norris and Phillips (1987), illustrate reading comprehension in terms of the reader's ability to understand the message conveyed by the writer. If the message *relates* to the reader's existing prior knowledge, he will be able to understand the message in a better way (Slavin, 1991). These arguments emphasise that reading comprehension largely relies on the reader's capability or *skill* to relate his prior knowledge to the new concept in the text, and interpret the text accordingly. These arguments point to the roles played by schema activation (to relate new concept to prior knowledge) and schema modification (to interpret the text), as discussed in Section 2.3, and the necessity to promote the skill.

Researchers have suggested several ways to support the reader, e.g. Milosavljevic (1997) proposes the use of analogy to help learners understand new concepts, i.e. by comparing existing concepts to the new one presented in the text, while Jitendra (2002) proposes the use of graphic diagrams to facilitate conceptual understanding. In interactive sessions, teachers can help the readers by showing how to link the lesson to the readers' relevant prior knowledge (Dye, 2000). These teaching strategies might be useful for the agent to utilise, and point to the need of a formal description of the strategies to inform the agent how to explain new concepts. Strategies to be employed by the agent will be discussed in Chapter 3.

In addition to having the teaching strategies, Barunger and Lewis (1997) emphasise that it is essential for the teachers to always consider both the general and specific prior knowledge that belongs to their students. This argument points to the necessity of the agent to represent a student's *general* prior knowledge relevant to a lesson, and his *specific* prior knowledge related to the main concepts in the lesson. This issue relates to student modeling, and will be discussed in Chapter 4.

As readers, the students should be equipped with appropriate reading skills. Bransford (1979) argues that teachers should provide clear guidelines to the students how to understand a text in terms of activating appropriate schemas and applying them during the reading process. Similarly, a pedagogical agent that follows the principles of schema theory should help the students perform the cognitive tasks during a reading session. This will ensure that the students are guided to activate appropriate schemas and perform schema modification accordingly.

The advantage of having a personal tutor is highlighted by Fielding and Person (1994) who state that if students are sufficiently guided in their reading process and can interact with their helpers (teachers, parents, peers etc), then the students will possibly acquire the strategies to comprehend a lesson by themselves. Though the solution to the problem faced by children seems clear, i.e. by providing helpers to the students, the real challenge is how these helpers should behave. We hereby argue that a pedagogical agent designed to guide the children's schema activation and modification is a potential approach to supporting reading comprehension because the agent can acts as a *computer helper* to the students in their guided readings.

2.4.2 Supporting student's cognitive tasks

In this thesis, we view learning as changes that happen to the learner's cognitive structure (Bos & Vaughn, 1998) and argue that the agent has to support the schema-based cognitive tasks that can lead to the changes, if appropriately performed.

Researchers have proposed several ways to promote students' learning. For example, Vygotksy (1978) introduces the notion of Zone of Proximal Development that describes a zone or an area between the level at which the student is currently achieving and the level at which the student can achieve if there is assistance or scaffold from a more knowledgeable person. Vygotsky stresses the fundamental role of social interaction to bring about the cognitive changes. The idea of software scaffolding applies here where the computer can provide the needed assistance (Luckin, 1998). In line with this idea, the student's schema-based cognitive tasks must be supported to enable the student to learn new concepts more effectively with the help of a more knowledgeable human or computer tutor.

Constructivists consider learning as an active process where the learner's cognitive structure changes due to the construction of new concepts (Bruner, 1996). Following the idea, the main objective of instruction is to help learners build new knowledge based on their existing knowledge structures. The description of learning as an active process that involves a teacher (in guided reading) points to the necessity to support the student's cognitive tasks that occur during the construction process.

In a closely related discussion to schema theory, Sweller (1998) illustrates the limitation of human memory in terms of Cognitive Load Theory. The theory explains how to manage the content of human memory to facilitate mental operations, i.e. memory storage, retrieval and application, which are assumed to occur every time a student learns. Primary schoolchildren may not be aware of these complicated mental operations, and they are not expected to understand these operations either. Teachers, however, are expected to support the mental processes, especially in reading sessions. This highlights the need to support the learners' mental operations that can cause changes to their cognitive structure.

In classroom settings, teachers support students when they have problems to understand new concepts in a lesson. The support is normally through some dialogue where the teachers introduce new concepts by tailoring the explanations to the reasoning ability and the prior knowledge of the children. To develop the agent, we focus on the support provided by the tutor using dialogue, which is a natural way of interaction for both teachers and children. Therefore, we aim to develop a pedagogical agent that is capable of engaging in a dialogue to help young children understand new concepts. It must be noted that in human-human learning situations, there are other aspects that have to be taken into account, e.g. motivational (Heffernan & Croteau, 2004; Mendez, du Boulay & Luckin, 2005; Qu, Wang &

Johnson., 2005; Wang et al., 2005) and emotional aspects of the children (Poel et al., 2004; Rotaru & Litman, 2005), which are being examined by other researchers and are outside the scope of our research.

Following Bloom's (1984) argument that one-to-one tutoring is the most successful form of instruction, we will view the approach of supporting conceptual understanding as a one-to-one interaction to support a learner's schema activation and modification: involving a pedagogical agent and a child. In this work, the agent needs a capability to guide the reasoning process using dialogue. We will elaborate on building computational models of interaction to support the cognitive tasks, presented in Chapter 5.

Though the importance of supporting cognitive tasks is acknowledged by researchers in Education and Psychology as discussed above, Pressley (1998) reports that the teachers in primary schools do not efficiently support children's mental processes that lead to reading comprehension. This shows the importance of providing support to the student's cognitive tasks by the pedagogical agent that intends to support children's conceptual understanding.

2.5 Computational Foundations: Relevant Computational Approaches

Studies (e.g. Aist, 2001; Zouaq, Frason & Rouane, 2000) show that computer generated explanations may fail to communicate successfully the meaning of words to children, and miscommunication may occur due to deficiency of appropriate teaching strategies or incapability to address the needs of each individual child. There are several studies on developing tutoring systems to support students' learning. In this section, we discuss relevant systems based on learning theories and dialogue-based pedagogical agents by pointing to the ways how the approaches can be applied and extended to suit the purpose of building schema-based pedagogical agent.

2.5.1 Theoretically inspired systems

Researchers have investigated the principles of learning theories as the basis of their approach to designing tutoring systems. Examples include: Geometry Tutor (Anderson, Boyle & Yost, 1985) that is based on ACT* Theory, Luckin and du Boulay's (1999) interpretation of Zone of Proximal Development to assist students in zone of available assistance and Akhras and Self's (2000) formalisation of interaction properties or affordances to support students' interaction from a constructivism perspective. The design approaches are discussed below by outlining main features that can be related to the schema-based approach.

Geometry Tutor

Geometry Tutor (Aleven & Koedinger, 2000; Aleven, Popescu & Koedinger, 2001; Anderson et al., 1985) is a computer tutor based on the principles of ACT* theory that explains learning process by describing how different types of knowledge are acquired. The main assumption of the theory is that knowledge is first acquired declaratively through instruction and then converted into procedures through experience. Geometry Tutor is developed to help adult students learn geometry via a *decomposition strategy*, through which the students are expected to better understand the difficult lesson.

To teach Geometry, the tutor makes the students' thinking process explicit by providing an interface that shows the declarative knowledge of the lesson in a spreadsheet. If a student has difficulty with a complex problem, the Geometry Tutor will suggest the steps to make the problem simpler: by decomposing it into comprehensible components. Thus, the student can identify the cause of the problem and solve it accordingly. As a result of the agent's intervention, the student's problem solving skill is promoted and he can acquire the procedural knowledge needed for the decomposition strategy.

The application of ACT* Theory to the development of the Geometry Tutor can suggest how to design tutoring systems based on schema theory. The ITS design approach provides an insight of how an agent can help the student construct knowledge. However, the ACT* theoretical explanations of the complex thinking process focus more on the steps to acquire difficult domains, which involve declarative and procedural knowledge, such as mathematics and geometry. Thus, the design principles based on ACT* theory may not be suitable for primary schoolchildren learning basic concepts in simpler domains taught in guided reading sessions. Moreover, the use of the new and unnatural interface, i.e. a spreadsheet, may cause other difficulties for the young students, such as an unnecessary increase of their cognitive load.

ECOLAB

ECOLAB (Luckin & du Boulay, 1999; Luckin & Hammerton, 2002, Mendez, du Boulay & Luckin, 2005) is a tutoring system based on the principles of Zone of

Proximal Development that explains learning in terms of a learner's possible achievement with the help from *a more able learning partner*. The main assumption of the theory is that learning can be enhanced with the help of others. ECOLAB aims to help children aged ten to eleven years old learn about food chains and webs.

The ECOLAB system provides learning activities to enable children to learn by interacting with the elements in a scientific investigation, in which the children are equipped with tools to conduct an experiment. Thus, the activities supplied by the system can guide the actions of the children to achieve a higher development zone, which is less likely without such help. In this case, the tutoring system is assumed to have adopted the role of a more able learning partner.

The ECOLAB's approach to designing systems that help children can be applied to designing tutoring systems capable of supporting children's learning new concepts. However, the approach does not sufficiently explain how to guide young learners to perform cognitive tasks that lead to conceptual understanding. In addition to that, the system focuses more on the complexity of the domain, which has been presented as being executable in a computer system in a similar way to a computer program. Should the design approach applied to tutoring systems supporting children's conceptual understanding, the children would focus more on the interaction to understand complex new concepts than constructing new knowledge based on existing schemas.

INCENSE

INCENSE (Akhras & Self, 2000) is an implementation of an ITS design approach based on the principles of constructivism that stress the importance of learners' *interaction and the process of learning*. The system supports knowledge representation, reasoning and decision making that are attuned to the principles of the learning theory. INCENSE supports adult students learning software engineering concepts.

The system analyses a *time-extended process of interaction* between a learner and a set of software engineering situations. It provides the learner with types of situations that afford the development of further courses of interaction that can support learning. As a result of the interaction, the learner can consult the system to construct models of the software engineering process.

The approach of designing tutoring systems based on constructivism is one possible application to designing systems capable of supporting children's conceptual understanding. Akhras and Self (2000) provide precise models of the interactions between the learner and system, but have not indicated teaching strategies to support cognitive tasks in constructing knowledge. Besides, INCENSE has been demonstrated with a complex domain, which suggests that the system is more suitable to adult learners instead of young learners at a primary school level. Moreover, INCENSE does not provide dialogue-like interactions and fully relies on the initiative and inquisitiveness of the learner. As discussed above, this may be valid for adult learners, while dialogue is critical for helping children.

The ITSs discussed above are all based on a thorough analysis of a particular learning theory and developed following the design principles derived from that theory. The systems are the implementation of the design approach that provides execution of the design principles in computer systems, and enables testing of the learning principles. To design pedagogical agents capable of supporting children's conceptual understanding, the same methodology will be followed. To the best of our knowledge, there are no ITSs or pedagogical agents whose design is based on schema theory. The review of the related systems shows an analysis is still required to understand schema theory and its relation with children's conceptual understanding, as presented in Section 2.3. The analysis will indicate what design architecture is needed to develop the agent, and a further analysis to understand how the agent should behave and teach.

2.5.2 Dialogue-based pedagogical agents

A recent trend in AIED follows the constructivist idea of the importance of interaction, and introduces pedagogical agents with conversational capabilities, capable of helping student by engaging in tutoring. The agents are expected to provide deeper learning compared to menu-based systems, for example by allowing the students to ask and answer questions. Current results from dialogue-based tutoring systems are promising (Person et al., 2001; Rosé et al., 2001) and suggest that the dialogue capability has made the systems more effective.

There are a number of successful examples of interactive tutors capable of engaging in effective pedagogical dialogue, e.g. Atlas (Freedman, 1999), AutoTutor (Graesser et al., 2000), CIRCSIM-Tutor (Evens et al., 2001), and Sherlock (Lesgold et al., 1992). We review these dialogue-based pedagogical agents by outlining their main features and how they are related to schema-based approach.

Atlas

Atlas (Freedman, 1999) is a dialogue manager that has the capability to engage students in mixed-initiative dialogues using interface that employs both textual and graphical elements. The tutor aims at increasing the opportunities for the students to construct their own knowledge by conversing with a natural dialogue-based tutoring system. The system is based on a plan that defines the actions the students should take: in what sequence and how. Thus, with the system's intervention, the students can plan how to solve a problem.

One possible application of Atlas is to inform the dialogue component of a pedagogical agent that intends to support children's conceptual understanding. In particular, the mixed-initiative characteristic of the system can engage learners in learning new concepts. The plan-based dialogue management system will enable dialogue participants to have a coherent dialogue. Atlas, however, has not specified dialogue strategies to perform cognitive tasks, which are required to define how a schema-based pedagogical agent will explain new concepts. This highlights the need to consider both a dialogue management component and a teaching expertise component.

AUTOTUTOR

AUTOTUTOR (Graesser et al., 2000) is a tutor that imitates the dialogue patterns of human teachers, both skilled and unskilled. During interactions with the student, the tutor employs dialogue strategies to encourage the student to contribute to the dialogue, for example by asking questions and giving hints. To analyse the contributions of the student, a comparison is made to the tutor's expected answer stored in a curriculum script. Consequently, the student can reach the answer by participating in the dialogue guided by the tutor.

AUTOTUTOR's approach to the design of a dialogue-based tutor can be applied to designing intelligent tutoring systems supporting children's conceptual understanding. Using asking strategies the children might be guided to understand new concepts presented in a lesson. However, AUTOTUTOR is more tutor-centred and focuses more on what a student understands instead of how the understanding process occurs. Besides, AUTOTUTOR concentrates more on the complexities of a domain instead of mental skill required to interpret new domain concepts.

CIRCSIM-Tutor

CIRCSIM-Tutor (Evens et al., 2001) aims to help medical students understand the human blood system. In particular, the tutor conducts a tutorial dialogue to inform the students about the mistakes they make when predicting changes of parameters in a blood pressure system. For each mistake, the tutor executes a pre-authored line of reasoning by asking closed-answer questions.

The ability of the CIRCSIM-Tutor to start reasoning with students can be applied to constructing tutoring systems capable of supporting children's cognitive tasks. Nevertheless, the agent is intended for adult medical students and a complex medical domain. The reasoning line used by the agent might not be suitable for children whose logical reasoning is characterised by concrete objects and ideas.

Sherlock

Sherlock (Lesgold et al., 1992) is an agent that provides *a coached practice environment* using a structured dialogue to detect problems in an electronics domain. The basis of the agent is an analysis of comprehensive interviews with experts in the domain. Consequently, the findings of the analysis inform the types of hints the agent should give to students.

The teaching knowledge of the Sherlock agent is represented as a hierarchy of subgoals, where each goal corresponds to a solution for a given problem. Thus, the teaching knowledge defines how an expert will troubleshoot problems in the domain. The help provided through the dialogue will enable the agent to guide the student to solve the same problem, for example by comparing the students' solution to an expert's solution.

The approach of developing the expertise of Sherlock through analysis of empirical data can inform how to provide expertise to tutoring systems that intend to support children's conceptual understanding. Interviewing experts, however, may not sufficiently reveal how to support each cognitive tasks involved in conceptual understanding. Though experts know how to explain new concepts to children successfully, they may not be able to explain the underlying mental processes that theoretically responsible for conceptual understanding.

The dialogue-based agents discussed above all show the importance of interaction between the agents and students and argue that dialogue is effective at supporting students. Research in reading comprehension (discussed in Section 2.4) indicates that teachers use dialogue to support children's learning, especially in a guided reading. The reviewed dialogue-based agents show a schema-based pedagogical agent can use dialogue to support the cognitive tasks. However, none of the dialogue-based agents is based on learning or cognitive theories. Differently from the agents, a schema-based agent follows the principles of schema theory that inform its design. Moreover, empirical data is needed to analyse schema-based explanation strategies.

Most of the agents are designed for university students whose reasoning characteristics differ significantly from children, as discussed in Section 2.3.5. Only few systems are designed for children, such as Reading Tutor (Aist, 2000) that augments texts with facts to help children in reading aloud sessions, StoryStation (Robertson & Wiemer-Hastings, 2002) that gives feedback to children on their creating writing, and Digital Puppets (Rizzo, Shaw and Johnson, 2002) that help children to structure their knowledge and explanations. However, these systems have not specified tutoring strategies to support children's schema-based cognitive tasks.

2.6 Issues to Examine in This Thesis

Following the above discussion, there are several design issues that we have to consider in order to inform the design of tutoring systems based on the principles of schema theory. We briefly discuss each of the issues here and point to further research which will be undertaken in this thesis.

2.6.1 Schema-based strategies used by human tutors to explain new concepts in one-to-one interaction with children

Though the importance of supporting children to understand new concepts has been emphasised by researchers in Education research (as discussed in Section 2.4), there is a lack of understanding of what strategies should be used by a pedagogical agent to promote schema-based cognitive tasks. These tutoring strategies are critical for the design of an interactive pedagogical agent that follows schema theory. Furthermore, a precise description of speech acts used and how they are combined into dialogue structures to accomplish certain tutoring strategies is required.

Human tutors' explanations can be analysed to derive the explanation strategies. The Wizard of Oz (Dahlback, Jonsson & Ahrenberg, 1993) approach to collecting data can provide a learning environment to collect examples of explanatory dialogue. This issue needs to be investigated further to provide the pedagogical agent with the explanation strategies to perform the schema-based cognitive tasks using dialogues, see Chapter 3.

2.6.2 Representation of the children's concrete prior knowledge

Schema theories have extensively discussed the nature of prior knowledge in terms of schemas. The theoretical explanation focuses on how prior knowledge is stored in the human memory and its retrieval process. To inform the design of a pedagogical agent based on schema theory, the prior knowledge needs to be represented in a format executable on a computer system. The agent needs a student's cognitive model in order to provide a personalised explanation to each individual.

While there are extensive studies on modelling students (see references to relevant work in Chapter 4), more work is needed to define a mechanism for extracting and maintaining a model of a child's concrete prior knowledge that reflects his thinking characteristics and is suitable for schema-based teaching. Frame-based representation methods that correspond to the idea of schematic knowledge structure can be employed to represent the learners' concrete prior knowledge. This issue needs to be examined further to provide knowledge base to the agent for adapting its explanation to learners, as described in Chapter 4.

2.6.3 Modelling the schema-based cognitive tasks

A pedagogical agent based on the principles of schema theory needs to simulate the tutors performing each cognitive task and adapt to the needs of a specific student at the same time. Models of the cognitive tasks are needed to inform the agent how to generate adaptive explanations.

Modelling of teaching principles based on learning theories have conducted by ITS researchers. However, the issue of modelling schema-based cognitive tasks needs further investigation in order to inform tutoring systems based on the theoretical principles of schema theory, see Chapter 5.

2.6.4 Maintaining schema-based explanatory dialogue

Explanatory dialogues involve complex interaction between a tutor and a child. The interaction normally consists of linguistic acts to perform cognitive tasks. The interaction may involve several turn-takings where both teachers and learners contribute to the knowledge construction process. A schema-based pedagogical agent should be equipped with an appropriate mechanism to manage the interaction process.

Dialogue management approaches can help the pedagogical agent to maintain the interaction process and achieve the dialogue goals. There are several studies on educational dialogue modelling. However, considering the specific type of cognitive tasks to support and young learners to interact with, the issue of modelling schema-based explanatory dialogue requires further investigation. This issue will be addressed in Chapter 5.

2.6.5 Validation of the design in studies with users

Theoretically-inspired tutoring systems are evaluated to show the effectiveness of the design approach. Potential users interact with the working prototypes of the design in an experimental setting. The effectiveness can provide evidence to the theoretical claims. In addition to that, the system can provide a testbed for the theoretical principles.

Traditionally, tutoring systems should be evaluated with learners. In the case of a schema-based pedagogical agent, the effectiveness of the design approach needs to be tested with the intended users. Integrating the pedagogical agent into existing teaching practice and testing with learners in real classroom settings can be useful in evaluating the pedagogical agent. An implementation of the proposed design approach in a multimedia educational system and its evaluation will be presented in chapters 6 and 7, respectively.

2.7 Conclusions

In this chapter, we have described the principles of schema theory to inform the design of a pedagogical agent that intends to support children's conceptual understanding. This chapter has provided cognitive, educational and computational foundations to the design approach, which give ideas how to support schema-based cognitive tasks to promote children's conceptual understanding. The support is anticipated to help children understand new concepts while reading text.

We have argued that the interaction plays a crucial role in supporting schema theory-based cognitive tasks responsible for conceptual understanding, and one-to-one dialogue communication is important in the development a pedagogical agent capable of simulating human explanation of new concepts. Although several aspects related to managing the interaction have been tackled, none of the existing pedagogical agents addresses the problem of supporting children explicitly. There is a lack of precise definitions of how to structure the behaviour of a schema-based pedagogical agent.

We have identified the main design issues that include teaching strategies to support children learning new concepts, formalisation of the theoretical principles, representation of the children's prior knowledge, and evaluation of the effectiveness of the design approach. For each issue, we have pointed out the required further investigation and referred to open questions.

This thesis will study the design of a schema-based pedagogical agent and its components that can form a computational framework for a schema theory-based approach of supporting students learning new concepts. Based on the computational framework, we will develop a prototype that will be evaluated with real users to validate the design principles and examine the benefits of the proposed approach.

Chapter 3

Deriving Schema-based Explanation Strategies: a WoZ Study

3.1 Introduction

Chapter 2 makes the case for interaction as a support for cognitive tasks performed by children as they learn to understand new concepts. In supporting children's conceptual understanding, human tutors normally use dialogue to interact with their students. We need to study the human dialogue strategies to inform the interaction strategies to be used by the *Schema Activation and Interpersonal Communication* (SAIC) pedagogical agent when explaining new concepts to children.

Human tutors employ a variety of strategies when they teach to activate relevant prior knowledge and then explain new concepts accordingly. Following the principles of schema theory discussed in Chapter 2, we are interested to examine dialogue strategies that help children perform schema activation and modification. It is hoped that this focus on a specific type of tutor-child dialogue will lead us to an understanding of how the SAIC agent might interact with young students to support their cognitive tasks.

The aim of this chapter is to examine how expert human tutors use dialogue to explain new concepts in terms of dialogue strategies. Through the analysis of the dialogue used by expert human tutors, we will aim to identify the communicative acts they use to perform schema activation and modification and to explain the dialogue goals for each dialogue exchange during the interaction.

To investigate how human tutors support children's conceptual understanding, we will identify an appropriate approach to collect example dialogues, prepare software needed for this, conduct a Wizard of Oz experimental study, analyse the collected dialogues, and present the findings of the analysis.

This chapter is structured as follows: Section 3.2 will provide justification of design decisions of our Wizard of Oz study. In Section 3.3 we will describe how the WoZ experimental study was conducted to collect dialogue corpora. Finally, we will present the results of dialogue analysis in Section 3.4.

3.2 Justification of Study Design Decisions

In the previous chapter, we have identified the principles of schema theory. These principles explain how humans understand new concepts based on existing prior knowledge or schemas. Our goal is to design a pedagogical agent that facilitates schema-based processes. While schema theory provides general guidelines of how to design the agent, it does not give sufficient understanding of *what dialogue strategies* should be used by a pedagogical agent to promote schema-based cognitive tasks even though schema-based reasoning has been extensively studied (e.g. Rumelhart & Norman, 1978; 1981) and the suitability of schema theory for designing learning systems has been acknowledged (Marshall, 1995). These tutoring strategies are critical for the design of an interactive pedagogical agent that follows schema theory. Furthermore, a precise description of speech acts used and how they are combined into dialogue structures to accomplish certain tutoring strategies is required.

To examine the dialogue strategies used by human tutors, we must examine relevant educational dialogues and identify how the tutors support each of the schema-based cognitive tasks in realistic learning situations. One possible approach is to use scenarios (Ferraris, Brunie & Martel, 2000; Melis, 2001; Sallaberry et al., 2004) where students or teachers can select a learning situation to simulate. Similarly, scenarios can be used to represent situations in which students have to perform schema-based cognitive tasks. These scenarios can be used to identify what strategies human tutors exploit to guide the student's reasoning leading to schema activation and modification.

We define a schema-based learning scenario as *a one-to-one interaction where a tutor and a student collaboratively perform a schema-based cognitive task*. The scenario-based interaction will provide key data about tutors' behaviour. Examples of these data include the tutors' actions to support each cognitive task, the linguistic acts the tutors perform in each task and the ways in which they respond to students' questions. To achieve realistic example dialogue based on the principles of learning theory, we will use predefined scenarios that explicitly reflect each of the main principles identified in schema theory. The scenarios will determine the characteristics of the cognitive tasks to be supported by the participants in the real setting.

Although the analysis of human-human example dialogues is a feasible way to analyse tutors' actions in supporting students (Chi, 1997; Graesser, Person & Magliano, 1995), we contend that the nature of human-human dialogue is different from human-machine interaction in terms of medium of instruction, which is textual instead of spoken format, and therefore propose instead to analyse empirical data obtained from human-machine interactions that reflect the pedagogical agent-student interactions.

One possible approach to achieving realistic dialogue from a human-machine interaction is to use a Wizard of Oz (WoZ) simulation technique (Dahlback, Jonsson & Ahrenberg, 1993; Ericsson, 1996; Paiva et al, 2003; Salber & Coutaz, 1993). The underlying idea of the technique is to simulate human-computer interaction with an intelligent system prior developing the required intelligence (Dahlback, Jonsson & Ahrenberg, 1993). WoZ is a powerful and efficient approach to gather design requirements and to make decisions about the system's behaviour. Researchers have used the Wizard of Oz simulation technique to collect example dialogues to analyse a range of topics including interface agent design (Paiva et al, 2003; Rizzo et al., 2005; Yang, Okamoto & Ishida, 2000), dialogue model (Bernsen, Dybkjær & Dybkjær, 1996) and multimodal interfaces (Batliner et al, 2003; Rapp & Strube, 2002; Salber & Coutaz, 1993). In contrast to other researchers who have applied WoZ to tune the design of ITS by *using a simulated tutor* and a real student (e.g. Paiva et al, 2003; Salber & Coutaz, 1993), we will *simulate the child.* By creating conditions where the simulated child requires the tutor's support to perform certain cognitive tasks we will be able to examine how human tutors perform cognitive tasks.

It must be noted that we initially opted to gather information about schema-based tutoring strategies by interviewing teachers. It soon became apparent that traditional approaches (such as interviews and questionnaires) for gathering such data were inappropriate and confusing for the teachers, who seemed to know how to explain but could not articulate the schema-based strategies they used. This led to a modification of the design. We decided to examine a rather non-traditional WoZ approach where not the system but the user was simulated. Having identified the main cognitive tasks needed for understanding new concepts and the reasoning characteristics of children, we could specify cognitive scenarios to simulate a child learning new concepts by interacting with a computer system via chat-like interface. This approach was efficient and appeared effective, since the interaction with the system provided empirical data of how human tutors explained new concepts in one-to-one communication and could be used as a basis for the design of the dialogue of a schema-based pedagogical agent.

3.3 Wizard of Oz Experimental Study

3.3.1 Objectives

The main objective of this study was to get an understanding of the tutoring strategies human teachers use to promote schema-based cognitive tasks when helping children at a concrete operational stage to understand and learn new concepts. The understanding gained should be sufficient to inform the design of an interactive pedagogical agent that follows schema theory. This includes:

- Identification of the main speech acts and the linguistic patterns used to express them;
- Description of the dialogue structure;
- Classification of the tutoring strategies used and detection of the dialogue patterns associated with these strategies.

A secondary objective of the study was to simulate the communication mode considered for the pedagogical agent in order to examine the appropriateness of the medium, as well as to ensure that the conditions under which the tutoring strategies are identified match the conditions in which those strategies will be applied. It must be stressed that the interaction medium has a significant influence on the communication. In one-to-one tutoring the teacher often knows the child personally and adapts his tutoring according to the perceived characteristics of the child. Moreover, face-to-face interactions involve a variety of modes, such as speech, gestures and facial expressions, which are incorporated by teachers into successful tutoring strategies. In contrast, the pedagogical agent we are designing is expected to operate in a fairly restricted communication environment that includes text-based interactions combined with references to teaching materials in the form of text and pictures. The agent does not know the child personally, so its model of the previous knowledge of the child may be incomplete or unreliable. These conditions have been mimicked in the study, as discussed below. Note that the need to deal with the constraints imposed by computer-human interactions and their effect on the pedagogical strategies used by teachers was also considered by Rizzo et al. (2005) in the design of a recent WoZ study for gathering affective tutoring strategies.

3.3.2 Experimental settings

In order to take into account the unique features of computer-human communications and to conduct realistic dialogues, we conducted a Wizard of Oz study (Dahlback, Jonsson & Ahrenberg, 1993).

Participants

Nine Malaysian primary school teachers, all with significant teaching experience[1], were asked to explain new concepts to a child who was geographically remote (in this case in the UK). The age group of the child was identified as seven to eleven years old. The child was simulated by an experimenter (the author of this thesis) based on knowledge gathered from the literature and his personal experience as a teacher-volunteer[2] working with young children for one year. The teachers were not aware that the child was simulated (although some of them realised this towards the end of their interactions). Each teacher was involved in a single one-to-one tutoring session that lasted approximately forty minutes.

Software

The tutoring process simulated in the experimental study comprised of four components: (1) a human tutor; (2) a wizard who simulates a child asking for explanation; (3) learning material in the form of texts and pictures; (4) an interaction channel.

The interaction was in the form of text typed into the chat-like interface that also contained information in the form of reading text and pictures. Space travel was chosen as a tutoring domain. A '*Going to the Moon'* prototype was constructed using a client/server model to enable geographically remote communication, see Figure 3.1 for the architecture and Figure 3.2 for a screenshot of the interface.

A server-based application was used to connect the tutor and the wizard-student. It managed the textual input from both the tutor and the wizard-student. A copy of the system prototype is run at the server to register all the textual messages and automatically record the dialogues. The tutor and the wizard simulating the child connected to the server by entering the URL address of the server application into a web browser. The teachers and the child had a similar interface (see Figure 3.2) but the teacher's interface included additional information about the activities of the child, e.g. 'The student is typing', 'The student clicks on 1- Training at NASA' and 'The student clicks on the picture'.

---

[1] All of the teachers had several years of teaching experience in primary schools and were recognised as expert science teachers by the Ministry of Education. They had Internet access and were equipped with laptop computers by the ministry.

[2] The teacher-volunteering period was done in parallel with the work on this thesis. It was used as a field study that enabled the author to become aware of the problems and needs of teachers and children, and to get insights for the design of the system. It also brought significant benefits for conducting an evaluation study of the developed system in realistic learning conditions, see Chapter 7.

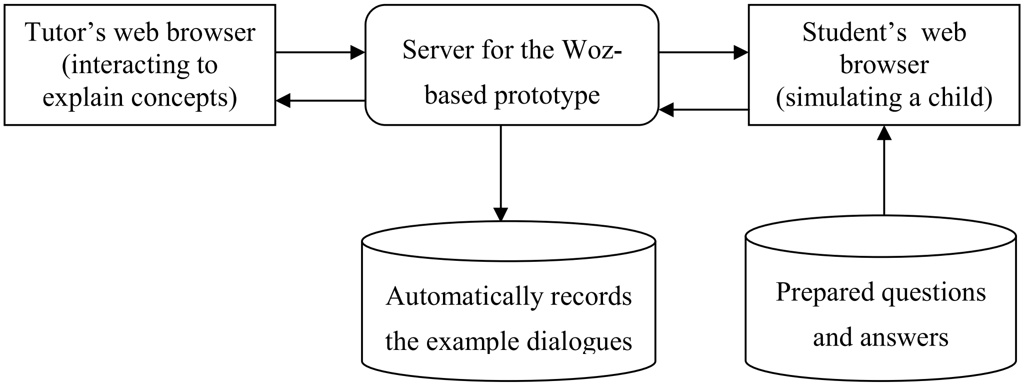
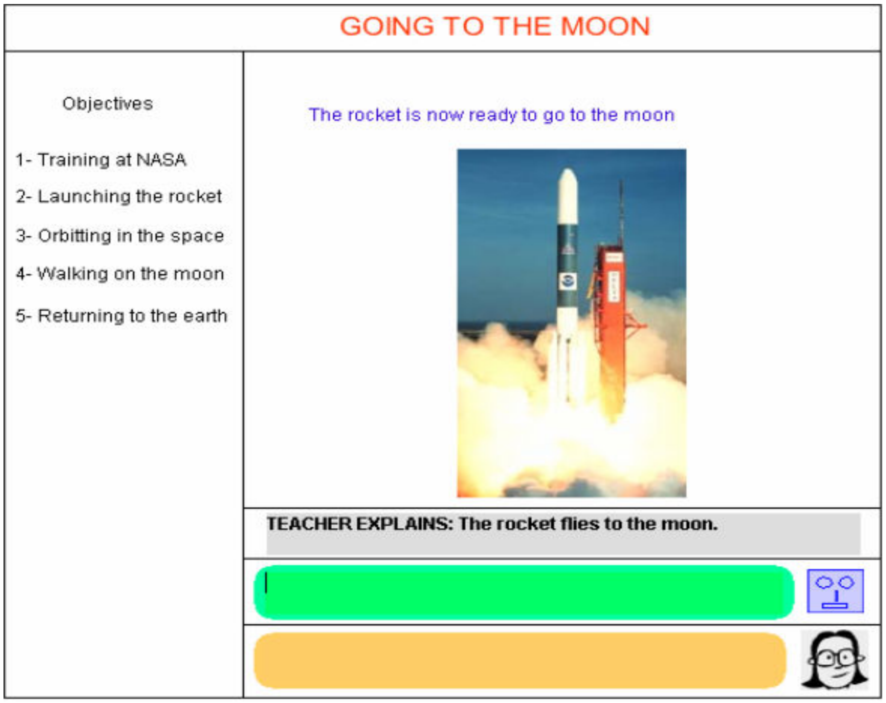

Tutor’s web browser (interacting to explain concepts)
Server for the Woz-based prototype
Student’s web browser (simulating a child)
Automatically records the example dialogues
Prepared questions and answers

Figure 3.1 The components of the Woz-based system architecture

In addition to typing the questions and answers using the keyboard, the wizard-child had the option of using preset questions and answers (based on scenarios) from a dropdown menu. This allowed the wizard to ask the same questions of, or give the same answers to, different teachers resulting in the collection of a number of different tutoring approaches for each schema-based cognitive task.



GOING TO THE MOON

Objectives

1- Training at NASA
2- Launching the rocket
3- Orbitting in the space
4- Walking on the moon
5- Returning to the earth

The rocket is now ready to go to the moon

TEACHER EXPLAINS: The rocket flies to the moon.

Figure 3.2 A screen shot of the prototype used in the study

Task

The teachers were told that the child was interested in learning about manned space flight but needed support to understand some of the new concepts. When the child "asked" for an explanation of a concept, the tutor had to explain the concept in a chat-like interaction. The explanations could refer to text or pictures from the lesson.

The wizard-child "created" conditions under which, according to schema theory, certain cognitive tasks would be required. When the child faced an impasse, he asked for explanation from the tutor by typing in the chat-like interface. The schema based cognitive tasks were invoked to see how the teachers would react. The situations simulated by the wizard fell in the following basic categories:

- schema is known/unknown (e.g. know `aeroplane`, never heard of `astronaut`),
- schema is created and some properties are known/unknown (e.g. know `rocket` and that it `can fly`, but unaware that `rockets` have to be repaired)
- schema created, properties known, values known/unknown (e.g. know `moon` and that it is visible during the night but unaware that it is far away).

3.3.3 Data Analysis

The data collected consisted of the dialogue transcripts of nine tutoring sessions recorded by the '*Going to the Moon* prototype'. The transcripts were analysed to identify the speech acts used by the tutors and to classify dialogue episodes for each of the identified schema-based cognitive tasks (as defined in Section 2.3, Chapter 2), as well as to identify other types of dialogue episodes that may occur during the interaction. Based on the analysis, the strategies employed by the expert tutors in each of the dialogue episodes have been examined.

To analyze the example dialogues, we have decomposed each tutoring session into *episodes* and *exchanges* (Katz, O'Donnell & Kay, 2000; Knesser, Pilkington & Jones, 2001). Episodes represent the goals of the tutors either to support a child's cognitive task (activation, accretion, tuning, restructuring and creation) or to manage the dialogue (meta-tasks). Exchanges represent short dialogue chunks that address a sub-goal and comprise of initiation, reply and feedback (Knesser, Pilkington & Jones, 2001)[3].

---

[3] Most of the exchanges included initiate, reply, and feedback. However, we also found several exchanges which did not end with a feedback; instead the tutor initiated a new exchange after the student's reply. The exchanges that did not include feedback referred mainly to activating well-known concrete concepts, e.g. `bus, car` and `policeman.`

The dialogues included both tutors' turns and wizard's turns. The wizard's turns related to the support he wanted from the tutors. The wizard's dialogue goals were to simulate a specific cognitive scenario, as described above. Following this, the dialogue analysis focused primarily on the tutors' turns, which have been analyzed in several steps:

- Parsing the tutors' turn to identify the speech act used.
- Grouping sequences of atomic speech acts into complex speech acts (Katz, O'Donnell & Kay, 2000), when appropriate. This finer division enables specifying the intent of the human tutors in a more precise manner.
- Classifying the speech acts into initiation, reply and feedback, and marking the dialogue exchanges.
- Grouping a series of exchanges into dialogue episodes.

The dialogue transcripts were first analyzed by three markers. Each marker was given instructions with the definitions of the cognitive tasks, a list of atomic speech acts following DISCOUNT (Pilkington, 1999), and a description of the dialogue structure. The markers marked sample dialogues individually and had a discussion to clarify their choices. There were few disagreements about the speech act categories. The markers quickly agreed on the grouping of atomic speech acts into complex ones. While the dialogue structure was fairly easy to identify, significant time was required to clarifying the goals and sub-goals of episodes and exchanges, respectively. As a result of such debate, the markers agreed on a set of criteria for classifying the dialogue episodes (see Section 3.4.3). It was also noted that the dialogue included episodes that were not necessarily aimed at supporting cognitive tasks, e.g. to diagnose the student's knowledge or to open/close a topic (see Section 3.4.3). The dialogues were then analyzed by two of the markers following the agreed upon categories, marking independently, followed by a discussion to agree upon the final result. The transcript analysis is illustrated below with an example of the marking of a schema activation and restructuring episodes.

[ACTIVATION conditions = Teacher believes that the student]
+ Has the schema ‘moon’
+ Knows its property ‘visible’: moon(visible, -)
+ Knows ‘visible’ possible values: daytime and night time
+ Knows its property ‘shape’: moon(shape, -)
+ Knows ‘shape’ possible values: like a box and like a ball

[EPISODE Activation. Goal: activate property values of moon: moon(visible, ?) and moon(shape, ?)]

[EXCHANGE 1. Sub-goal: activate moon(visible, nightime)]
[**INITIATE**]
[**Ask-Suggest**]
[**Inquire** property value: moon(visible,?)]
[**Suggest** property value: {daytime, night time}]
Teacher: when do you usually see the moon? Daytime or night time?
[**REPLY**]
[**Inform** property value: moon(visible, night_time)]
Student: Night time.
[**FEEDBACK**]
[**Confirm** property value: moon(visible, night_time)]
Teacher: You’re right.

[EXCHANGE 2. Sub-goal: activate moon(shape, like_a_ball)]
[**INITIATE**]
[**Ask-Suggest**]
[**Inquire** property value: moon(shape,?)]
[**Suggest** property value: {like_a_box, like_a_ball)]
Teacher: What’s the shape of the moon? Is it like a box or like a ball?
[**REPLY**]
[**Inform** property value: moon(shape,like_a_ball)]
Student: It is like a ball
[**FEEDBACK**]
[**Confirm-Praise**]
[**Praise**]
[**Confirm** property value: moon(shape,like_a_ball)]
Teacher: Very good. You’re right.
[END EPISODE Activation]

[Conditions - Restructuring]
+ Student has the schema ‘rocket’
+ Student has the schema ‘aeroplane’
+ Student knows the isa of the two schemas
+ Student does not know the difference between the two schemas

----

[EPISODE Restructuring. Goal: Restructure aeroplane to become a rocket]
[EXCHANGE 1. Sub-goal: Compare the new schema with an old schema]
[**INITIATE**]
[**Probe** difference between a new and old schema]
Teacher: What’s the difference between a rocket and an aeroplane?
[**REPLY**]
[**Inform** property value (not known)]
Student: I don’t know.
[EXCHANGE 2. Sub-goal: Highlight the difference between the two schemas]
[**INITIATE**]
[**Contrast** old schema and new schema]

`[`**`Inform`**` property value of the old schema aeroplane(destination, airport)]`
`[`**`Inform`**` property value of the new schema rocket(destination, moon)]`
Teacher: an aeroplane flies from one airport to another airport
Teacher: a rocket flies from the earth to the moon
`[END EPISODE Restructuring]`

3.4 Results

3.4.1 The use of prior knowledge to help interpret a new concept

The results of the analysis showed the extensive use of prior knowledge by the tutors, which is consistent with the principles of schema theory discussed in Chapter 2. They also showed that the tutors dynamically and adaptively supported students' learning modes during their explanations. The tutors mainly guided the student to reason about how to understand new concepts in the lesson based on existing schemas in the student's prior knowledge. We noticed that knowing the age group of the child helped the teachers create some rough assumptions of the child's prior knowledge (i.e. knowing `policemen`, `teacher`, `car`, `aeroplane`, etc.) which they used in the dialogue.

The role of the tutors was to help activate relevant schemas which might assist the student's understanding, and to select the most appropriate learning mode to explain new concepts. The role of the student was to confirm his or her knowledge in the reasoning in terms of a schema, its properties and values. The interaction process included:

1. **Perceiving the new concept in the lesson.** The student asked the tutors if he did not understand the new concept or the tutors asked the student about the new concept to check the student's understanding.
2. **Activating relevant prior knowledge.** The tutors activated relevant schemas from the student's prior knowledge.
3. **Selecting a schema-based mode.** Based on their perception of the cognitive state of the student and the activated prior knowledge, the tutors conducted dialogue to support one of the modes: accretion, tuning, or restructuring.
4. **Creating new schema.** If existing schemas from the student's prior knowledge were not appropriate to explain the new concept, the tutors provided a definition to support schema creation.

3.4.2 Speech acts of human tutors

Several types of speech acts were employed in the dialogue episodes. These are summarised in Table 3.1 (for the atomic speech acts which are not divisible into other speech acts and Table 3.2 (for the complex speech acts which combine two or more atomic acts).

Table 3.1 Atomic speech acts used by the tutors in the WoZ study. The acts are adapted from DISCOUNT (Pilkington, 1999)

Speech act	Type	Example
Inquire (asking question)	Inquire Schema Inquire Property-value Inquire Meta-move	*What do we call the people who go to the moon?* *What's the shape of the moon?* *Have you clicked on the menu?*
Inform (making statement)	Inform Schema Inform Property-value Inform Meta-move	*The man who goes to the moon is called astronaut.* *A rocket flies from the earth to the moon.* *Today we'll learn a new lesson.*
Confirm (confirming fact)	Confirm Schema Confirm Property-value Confirm Meta-move	*The space shuttle flies from the earth to the moon.* *You also know that moon looks like a ball.* *You already know a lot about 'Going to the moon'.*
Suggest (offering possible answers)	Suggest Schema Suggest Property-value	*Is he a tutor, a policeman, or a doctor?* *In the morning, afternoon or at night?*
Praise (motivating the student)	Praise Correct answer	*Very good.*
Show (referring to a picture)	Show Picture	*I want to show you some pictures of the moon.*
Probe (diagnosing the student's knowledge)	Probe Schema Probe Property-value Probe Schema-Difference Probe Schema-Similarity	*Have your ever heard the word astronaut?* *Do you have any idea how the astronaut goes to the moon?* *Do you know the difference between rocket and aeroplane?* *Do you know what is the similarity between rocket and aeroplane?*

Table 3.2 Complex speech acts used by the tutors in the WoZ study

Speech act	Definition	Type
Ask-suggest (inquire and suggest possible answers)	Ask-suggest Schema *Ask-suggest Property-value*	*Do you know what we call the man who goes to the moon? Is he a soldier?* *When do you usually see the moon? Daytime or night time?*
Confirm-praise (confirm correct knowledge, praise)	Confirm-praise	*Yes. You are clever.*
Compare (similarity between property values)	Compare Property values	*Rocket can fly to the moon. Aeroplane also can fly.*
Contrast (difference between property values)	Contrast Property values	*Rocket flies to the moon. Aeroplane flies to airport.*

The tutors' utterances were analysed to identify regularities of the linguistic forms of the speech acts and the correspondence between a linguistic form and the domain presented in terms of schemas. The linguistic forms are required to form a basis for the composition of the utterances of pedagogical agent interacting with a child. Some example speech act templates are shown below.

Inquire

When do you usually see-<property: visible> the moon-<schema: moon>?

What does postman-<schema: postman> do-<property: function>?

What is the difference between a rocket-<schema: rocket> and an aeroplane-<schema: aeroplane> ?

Inform

Rocket-<schema: rocket> is a vehicle-<property: isa><value: vehicle> like a car-<schema: car>.

The man-<schema: man> who goes to the moon-<property: function> <value: goes to the moon> is called astronaut-<schema: astronaut>.

We can see-<property: visible> the stars-<schema: stars> at night-<property: visible><value: at night>.

Confirm

Yes, you're right!

You know about the time-<value: `night time`> when we can see-<property: visible > the `moon` -<schema: the `moon`>.

I think you understand now why `astronaut`-<schema: the `astronaut`> does `training`-<property: `activity`> <value: `training`>.

Suggest

Is he a `teacher`-<schema: `teacher`>, a `policeman` <schema: `policeman`> or a `doctor`-<schema: `doctor`>?

Praise

Brilliant!

You are clever!

Show

I want to show you some pictures of the `moon`-<schema: the `moon`>.

Now click 'Start > 2- Launching the `rocket'` <actions> to see a picture.

Probe

Have you ever heard the word `astronaut`-<schema: `astronaut`>?

Do you have any idea how an `astronaut`-<schema: `astronaut`> `goes to the moon` -<property: function> <value: `goes to the moon`>.

3.4.3 Dialogue episodes

The tutors' explanation of new concepts can be classified into dialogue episodes, as discussed in Section 3.4. Schema-based dialogue episodes aim to support the schema-based cognitive tasks, defined in Section 2.3, Chapter 2. The following schema-based episodes have been identified in the example dialogues:

Table 3.3 Schema-based dialogue episodes identified in the WoZ study

Dialogue episode	**Definition**
Activation **(13%)**	A dialogue episode guided by the tutor to facilitate the activation of some parts of the student's prior knowledge
Accretion **(14%)**	A dialogue episode guided by the tutor to facilitate the creation of a new schema based on existing schema structure. There are no structural changes to the existing schema.
Tuning **(26%)**	A dialogue episode guided by the tutor where an existing schema from the student's prior knowledge is modified by changing the value of a property in that schema or adding a property and its value without analogy to other schemas.
Restructuring **(8%)**	A dialogue episode guided by the tutor where an existing schema from the student's prior knowledge is modified by adding or deleting a property or a set of properties in that schema through analogy to one or more schemas from the student's prior knowledge.
Creation **(10%)**	A dialogue episode guided by the tutor where a new schema is created in the student's prior knowledge without using schemas from the student's prior knowledge because they are not appropriate.

The dialogues not only contained episodes for supporting schema-based cognitive tasks but also episodes for managing the dialogue: to introduce the lesson or a new topic, diagnose student's knowledge, and summarize explanation and confirm student's knowledge, as defined in Table 3.4.

Table 3.4 Dialogue episodes to manage dialogue and their definitions

Dialogue episode	**Definition**
Introductory **(8%)**	A dialogue episode guided by the tutor to introduce the lesson or a new topic.
Diagnose **(15%)**	A dialogue episode guided by the tutor to check whether the student has a schema or knows its property value.
Summarizing and confirmation **(6%)**	A dialogue episode guided by the tutor to confirm student's knowledge of a schema

3.4.4 Tutoring strategies for the dialogue episodes

The analysis of the dialogue transcripts enabled us to identify strategies employed by the tutors to perform the dialogue episodes, see Table 3.5.

Let us consider again the example dialogue shown earlier in the section (see p. 39) in Table 3.3. It illustrates strategies for activation and for restructuring. The activation is done by asking property values of a schema. The restructuring shows contrasting a new schema with an existing schema. It is based on the cognitive state of the student, namely having both the schemas `rocket` and `aeroplane` but not knowing their difference. The understanding of the new concept `rocket` is, therefore, generalised from the schema `aeroplane`; a vehicle like an `aeroplane` but flies to the `moon` instead of to an airport.

Table 3.5 Tutoring strategies used by the tutors to perform dialogue episodes

Dialogue episode	**Strategies**
Activation	Asking property value of a schema Asking name of the schema with a property value Showing picture of a schema Informing the *isa* category and its instance
Accretion	Asking instance of the new concept Informing the *isa* of the new concept Comparing the new concept with its *isa* instances Informing an instance of the new concept *isa* Asking whether the student can see the similarity between the new concept and its parent
Tuning	Adding a new property to the new concept which has just been created Informing a wrong value of the new concept Contrasting the new concept with schemas under the same *isa* category Informing the *isa* category of the new concept
Restructuring	Comparing and contrasting the new concept with an existing schema
Creation	Informing the property of the new concept Showing a picture of the new concept Informing its *isa* category
Introductory	Informing that the lesson is new Informing that the lesson is interesting Informing what the student will learn Informing how the student will learn
Diagnose	Asking whether the student is familiar with the new concept Asking whether the student knows properties of the new concept
Summarizing and confirmation	Confirming the knowledge of the student Informing the properties of the new concept and confirming the knowledge of the student

3.5 Conclusions

The chapter presents an approach for capturing human tutors' strategies to shape the design of the SAIC pedagogical agent. A WoZ experimental study was conducted with a simulated child who created conditions where teachers had to support the child's schema-based cognitive reasoning. This enabled the collection of sufficient data to analyse the behaviour of human tutors in order to build a dialogue model of the SAIC agent. The study confirmed that tutors explained new concepts using the student's prior knowledge as a mental model to help interpret new concepts, as explained by schema theory. Tutoring strategies for supporting schema-based cognitive tasks have been identified. In addition, the speech acts used have been defined.

The results presented here are based on a sample domain (introductory astronomy) as taught by nine experienced tutors. Due to the fact that schema-based cognitive tasks are domain-independent, and the theoretical principles of the conceptual explanations are the same, it can be expected that the results from the study can be applied to the design of schema-based pedagogical agents in different domains. Furthermore, some of the strategies identified in the chapter can be applied to other pedagogical agents that support conceptual understanding.

It must be stressed that the study was aimed at the design of a pedagogical agent and not at capturing a general model of tutoring strategies based on schema theory. This justified the constraints imposed on the experimental conditions, such as using chat-based communication and a simulated child. The schema-based strategies identified are by no means comprehensive: the results have been based on nine dialogues but other teachers might exhibit different behaviour and there may also be some cultural bias. However, there has been a good coverage of strategies that not only confirmed the application of schema theory in the tutoring practice but also enabled us to produce a description of the dialogue patterns.

Next in this thesis we will present the development of the SAIC agent to replace the human tutors in the '*Going to the Moon'* prototype used in the WoZ settings. Chapter 4 will present a design architecture as a computational framework of designing the agent and Chapter 5 will present how to generate adaptive explanation and provide a computational model of the SAIC dialogue.

Chapter 4

The SAIC Architecture and Student Modelling Mechanism

4.1. Introduction

The goal of our work is to develop a pedagogical agent capable of supporting children's conceptual understanding by formalising the interactive process involved in schema activation and modification. To provide a computational framework for the development of a SAIC agent, we define a SAIC architecture that is based on investigation of the principles of schema theory, discussed in Chapter 2, and examination of human teachers' tutoring strategies to explain new concepts, discussed in Chapter 3.

In this chapter, we will present the proposed SAIC architecture by illustrating its components, describing the function of each component, and explaining how they work. As a knowledge-based system, a SAIC agent requires knowledge to perform reasoning and to adapt its interactive explanations. Based on the use of schema theory to support children to learn new concepts, the techniques to represent domain and student schematic knowledge will be presented.

To explain the proposed SAIC architecture, we start by describing the issues pointed by the WoZ experimental study presented in Chapter 3 that identified the functionality of the SAIC agent. Based on these issues, the main components of the architecture will be introduced. Formal knowledge representation will be discussed to show how schematic knowledge can be represented in SAIC. Following this, the processes of student modelling and updating student schematic model will be described.

This chapter will first present (in Section 4.2) the issues that lead to the SAIC architecture and will provide an illustration of the architecture to show its components and their relationship. The representation of schematic knowledge in SAIC and the reasoning used is presented in Section 4.3. The approach to the student modelling in SAIC and the mechanisms employed to update the student the student model will be presented in sections 4.4 and 4.5, respectively.

4.2. The SAIC Architecture

As discussed in Chapter 3, in order to design a pedagogical agent capable of supporting children's conceptual understanding based on the design principles discussed in Chapter 2 (Section 2.3), we must identify how human teachers apply the principles of schema theory and explain new concepts to students in one-to-one interaction. The understanding gained by the WoZ study presented in Chapter 3 will be the basis for specifying the behaviour of the SAIC agent.

4.2.1. Issues illustrated by the WoZ study

The findings of the WoZ study (see Chapter 3, Section 3.4) pointed at several important issues to consider in designing a pedagogical agent that aims to support children learning new concepts.

Domain knowledge

The teachers referred to their knowledge of the domain when trying to explain new concepts to the child. It was noted that this knowledge included both abstract concepts to be introduced in the lesson and concrete concepts from everyday life that were employed in the schema-based cognitive tasks. Formal method to represent domain knowledge in the SAIC agent will be discussed in Section 4.3.

Student model

The teachers seemed to form some picture of the student's conceptual understanding, i.e. student model, and took this model into account in the explanatory dialogue. This student model is *not necessarily complete* or *accurate* but should be sufficient to decide about what cognitive strategies to employ and how. The representation of the student model will be discussed in Section 4.4.

Student modelling algorithms

The teachers made some initial assumptions of what previous knowledge the child might have based on their prior experience with children at a similar stage. The initial assumptions concerned mainly concrete schemas that the children could have built in their previous experience. This points at the idea of using some kind of stereotype in order to initialise the student model.

Throughout the dialogue, the teachers updated their assumptions about the student based on what has been already explained. In addition, on several occasions, the teachers initiated diagnostic episodes to check whether the student had built a schema or acquired a schema property and its value. This concerned mainly new, abstract concepts introduced in the lesson. A mechanism for updating the student model in a SAIC pedagogical agent will be described in Section 4.5.

Dialogue planner

The dialogue was structured as a sequence of episodes, i.e. dialogue games, of two kinds: schema-based episodes and episodes to manage the dialogue. An episode, that included several speech acts, was initiated when certain conditions were detected, based on the teacher's beliefs about the student's previous knowledge (i.e. the student model), as well as on some pedagogical strategies. The dialogue was guided by the teachers and adapted to the student. The mechanism for maintaining dialogue in the SAIC will be discussed in Chapter 5.

Dialogue utterances

The analysis of the WoZ interactions shows that there were some linguistic patterns used by the teachers to express dialogue utterances. The patterns can be used as templates for generating dialogue utterances, instead of sophisticated natural language generation techniques. The generation of SAIC template-based utterances derived from the WoZ study will be discussed in Chapter 5.

The above issues have been taken into account in the design of the SAIC architecture. The knowledge representation in SAIC and the student modelling will be discussed in detail in this chapter, the dialogue planning and adaptive explanation will be discussed in the next chapter.

4.2.2. The proposed architecture

Considering the design principles discussed in Chapter 2 (Section 2.3) and the identified issues discussed above, we have proposed a *Schema Activation and Interpersonal Communication* (SAIC) architecture of a pedagogical agent that simulates the help provided by human teachers when supporting children's conceptual understanding. The architecture combines the cognitive principles of schema activation and modification, and the interactive nature of learning with a pedagogical agent, i.e. interpersonal or one-to-one communication. Accordingly, a pedagogical agent that follows the proposed architecture will be called a *SAIC pedagogical agent*. The SAIC architecture is illustrated in Figure 4.1.

The SAIC architecture extends a traditional multimedia system with a SAIC pedagogical agent that explains new concepts via one-to-one dialogue with a child. The proposed architecture implements the metaphor of an *interactive reading session* where the child goes through some multimedia resources and can ask someone (in this case a pedagogical agent) for an explanation when the child faces problems to understand new concepts. The SAIC agent works as an integrated part of the learning environment.

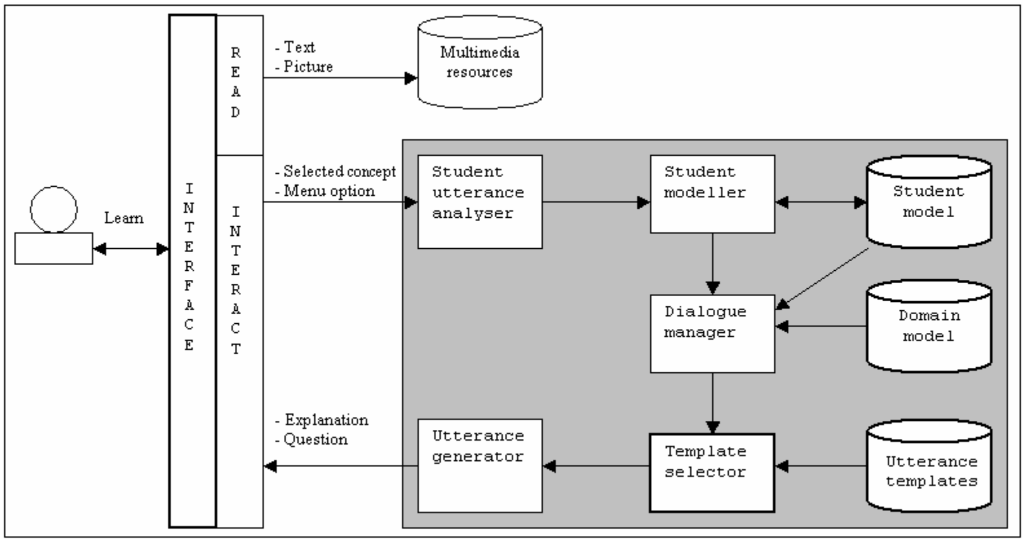


Figure 4.1 The proposed architecture of a SAIC pedagogical agent integrated in an educational multimedia system. The components of the SAIC agent are shown in the shaded area.

The SAIC architecture consists of the following main components:

Interface

The student reads the lesson provided in form of a multimedia presentation. For example, to illustrate the application of the SAIC agent, we will use a multimedia system comprising of texts, pictures and navigation buttons, as discussed in Chapter 6 (Section 6.3). The student initiates an interaction with the SAIC agent by asking for an explanation when facing difficulty to understand domain concepts. The interface should enable him to communicate with the agent using an interaction language that is understandable by both dialogue participants, i.e. the agent and the student. The student will see utterances that are composed of texts and pictures, as used by human teachers in the chat-style WoZ interaction. Using the interface, the student can contribute to the interaction.

Student utterance analyser

The SAIC agent uses an interaction language (to be presented in Section 5.7, Chapter 5) that requires that the intention of each student contribution to be explicitly specified in terms of a speech act (see Section 3.4.2, Chapter 3). The function of this component is to classify a student's contribution into: answer, question or confirmation. The component also determines the schema, property or value (see Section 4.3.3) of the student's contribution.

Domain model

A domain model contains facts about the domain knowledge that defines what to teach and is represented in a formalism suitable for schematic knowledge. The facts are stored in a knowledge base that is queried by the dialogue manager to find out which domain concept to explain. The representation of domain knowledge in SAIC will be discussed in detail in Section 4.3.

Student model

A student model contains facts about a student in the form of abstract domain concepts he has already acquired and concrete prior knowledge that together represent the student's schematic knowledge. This cognitive model[4] follows the teachers' assumption of the student's existing knowledge that may be used to help the student understand new concepts and explicit information from the student to tune his model. The student model is consulted by the dialogue manager component to decide how to adapt the explanation to the student's previous knowledge. The student model will be discussed in detail in Section 4.4.

Student modeller

This module dynamically models the student by examining and updating the student model to reflect the changes caused by the interaction with the agent. Based on the current student contribution, knowledge extracted from the domain model and the examination of the student model, the student modeller passes the recent state of a

---

[4] In our work, a student cognitive model refers to the student's knowledge of the domain concepts represented as a combination of some acquired domain concepts and other concepts related to the lesson, which are described as *concrete* (discussed in Section 2.4, Chapter 2). In different research areas, such as Human Computer Interaction, the phrase "cognitive model" may have different meanings.

concept's schematic knowledge to the dialogue manager. This component will be discussed in sections 4.4.4 and 4.4.5.

Dialogue manager

This component plans the interaction between the agent and the student in terms of speech acts and dialogue episodes (see sections 3.4.2 and 3.4.3, Chapter 3). The dialogue manager addresses the dialogue goals of the SAIC agent: to support a student to perform schema-based cognitive tasks or to manage the dialogue (meta-tasks). The dialogue goals of the SAIC agent correspond to the dialogue goals of human tutors discussed in Section 3.4.3, Chapter 3. This component will be discussed in Chapter 5, Section 5.6.

Template selector

This module selects a suitable template from a template library and passes it to the utterance generator to compose the agent's utterance. Each template represents the strategy related to a cognitive task that is specified by the student modeller. The template structure and the selection of an appropriate template are based on heuristics defined from the WoZ study presented in Chapter 3.

Utterance generator

This component composes the agent's utterances based on the template identified by the template selector. The dialogue goal is performed using a speech act that is externalised in the form of a sentence composed based on the templates. The generation of dialogue utterances using templates will be discussed in Chapter 5, Section 5.7.

The SAIC design assumes that the interaction is initiated by the student, i.e. the student starts by *asking* for an explanation of a new concept (e.g. the student may click on a new concept in the multimedia presentation to ask for an explanation, see the SAIC prototype presented in Chapter 6). The student-agent interaction can be explained in terms of input, process and output:

> **Input**: The student's contribution to the interaction, domain and student models plugged into the agent.
>
> **Process**: When performing a schema activation, the agent selects the most relevant prior knowledge or schema to activate. The selection is based on the student's cognitive model and answers explicitly given by the student during

the interaction. When performing schema modification, the agent will select the most suitable learning mode - accretion, tuning, restructuring or creation – that will determine how to explain a new concept chosen by the student. To perform the cognitive tasks, the agent employs speech acts and tutoring strategies derived from the WoZ experimental study. The processes are externalised as dialogue utterances by which the agent interactively guides the process of schema activation and modification. The agent selects a suitable template to generate the utterances that represent the tutoring strategies used by human teachers in the WoZ study. During the interaction, the agent may require further inputs from the student in the form of confirmation and answer. These inputs are used by the agent to best adapt its explanation.

Output. Based on the domain and student knowledge provided to the system, and the student's utterance, the agent performs some dialogue in order to generate adaptive explanation. The agent's beliefs about the learner's schematic knowledge are used to update the student model to reflect the changes to the student's schematic knowledge.

4.3. Knowledge Representation in SAIC

4.3.1. Knowledge representation in ITSs

In order to have a schema theory-based pedagogical agent, we need to have a knowledge representation that will support how the learning theory views learning of new concepts. We will review relevant techniques from intelligent tutoring systems to identify a suitable way for representing domain and student knowledge in SAIC.

ITSs normally consist of four main modules: the student interface, the student model, the teaching model and the domain model (Murray, 1999; Virvou & Mandridou, 2001; Wenger, 1987). Andriessen and Sandberg (1999) contend that a domain model is required to record the concepts, relations and facts of a domain. In the case of schematic knowledge, Kalyuga (2003) proposes that it should be represented in terms of a concept's *function, structure* and *process*. The contents of the schematic knowledge can define the relations and facts of a domain that the SAIC agent wants to teach. We adopt Kalyuga's approach to representing schematic knowledge to represent the SAIC domain and student knowledge.

The SAIC pedagogical agent requires information to support children. There are two approaches to providing adaptive explanations: *generate* or *search* (Baker, 2000). The *generate* approach, which is the main concern of a SAIC agent, describes that the content of an adapted educational intervention is to be generated dynamically on the basis of a general underlying knowledge representation and student knowledge. A SAIC agent must generate an explanation of new concepts dynamically to suit the needs of a student. For this reason, the knowledge in a SAIC agent has to be represented formally in a way that will enable the generation of dynamic explanations of new concepts. On the other hand, the *search* approach assumes the availability of a huge number of predefined explanations that is not applicable to the SAIC agent. A search approach has been followed in AutoTutor (Graesser et al., 2004), for example.

ITS designers need to have a knowledge representation format that is exchangeable between systems to enable knowledge sharing and reuse, which can enhance the interactivity and adaptivity of the systems (Melis et al., 2001). To have a general SAIC architecture that is not restricted to a certain knowledge representation format, it is necessary to represent its knowledge using a conventional Artificial Intelligence formalism. This can ensure that a SAIC agent can be adapted in educational environments with relative ease.

Melis et al. (2001) stress the separation of the knowledge representation from the functionality of a system. Taking into account the issues pointed by the WoZ experimental study, knowledge in a SAIC agent is separated into: *what to teach* (domain model), *who to teach* (student model), and *how to teach* (interaction strategies to perform schema-based cognitive tasks). In a SAIC agent, any knowledge representation that adheres to the idea of schematic knowledge representation should be executable. Hence, to adopt the SAIC architecture in designing pedagogical agents, one may represent knowledge in domain and student models in schema-based formats and attach the knowledge to the agent. Therefore, the explanatory functionality of a SAIC agent will not be restricted to a specific knowledge representation format.

4.3.2. Representation of schematic knowledge in SAIC

Schema theorists, for example (Ausubel, 1967; Bartlett, 1958; Rumelhart, 1980; Schank, 1982) describe schema theory in terms of a framework for representing prior knowledge (see Section 2.3, Chapter 2). Ausubel (1967) proposes the idea of hierarchical organisation of schematic knowledge where a student interprets new concepts according to the existing hierarchy. This approach to representing schemas can inform how to link the schemas in knowledge bases. We adopt this approach to represent domain and student knowledge hierarchically.

Minsky (1981) proposes *frames* as knowledge structures to represent schematic knowledge in computer systems, in which facts about a schema are clustered. Minsky's constructs *frame, slot* and *value* directly correspond to the constructs of schema theory: *schema*, *property* and *value*, respectively. Frames are suitable to represent typical situations that have some generic structures (Bratko, 1997).

Frames have been widely applied in intelligent systems. The formalism is the basis of some languages for representing conceptual knowledge, e.g. Frame Logic (Kifer, Lausen & Wu, 1995), DAML+OIL (Horrocks et al., 2001), and RDFS (Brickley & Guha, 2002). A frame is generally defined as a data structure having components that are called slots. The values of the slots can be filled with information of various kinds: simple values, reference to other frames and also procedures that can be called to compute the values of the slots. Frame-based reasoning mainly concerns with filling the values of the slots using direct retrieval or inference (Bigus & Bigus, 1997; Bratko, 1997; Le, 1993).

Slots can have *default values*; it is not always necessary to provide the details of the facts about a given entity or object. The default values will be used whenever the information for the slot is omitted. The capability is very useful when not all of the knowledge is available or provided. Moreover, frames can be dynamically linked to other frames by using references to the frames. This capability allows a representation of hierarchical knowledge about a problem.

To sum up, frames are suitable for representing schematic knowledge and can be used to develop complex knowledge structures in knowledge-based systems. Hence, they have been chosen as the formalism for representing domain and student schematic knowledge in SAIC.

For programming purposes, the knowledge in the frames can be represented as a set of facts normally represented as triples:

```
frame_name.slot: slot_value
```

For example, some knowledge about a `space shuttle` can be represented as a set of facts, as follows:

```
Space_shuttle.a_kind_of: vehicle

Space_shuttle.has: wings

Space_shuttle.can: fly

Space_shuttle.destination: moon

Space_shuttle.function: transport_astronaut_into_outer_space

Space_shuttle.process: blast_off_into_outer_space
```

Frame-based reasoning refers to retrieving knowledge from a set of slots; usually the queries refer to a slot and its value. In a declarative programming language like Prolog, a query procedure is normally created. For example, a query to get the values of a slot can be represented as follows:

```
Value(frame.slot: value)

     query = frame, slot, value

     query = parent (frame, slot, value)

     query = instance(frame, slot, value)

     call(query).
```

The value of the slot can either be directly retrieved from the frame or indirectly from its parent or superclass through inheritance method. The *call* command checks available facts to retrieve the required value.

For instance, to query a frame-based student model for an existing frame or frames related to a `space shuttle`, the agent will refer to the parent of the frame using `parent (frame, slot, value)`. Using the parent frame, SAIC can find the class members of `space shuttle`, i.e. other frames whose *isa* slot has a value `vehicle`, for example `bicycle`, `car`, `aeroplane` and `bus`. To find a closely-related frame, SAIC will check the class members to find a frame that shares the same property and value with space shuttle, for example `aeroplane` shares the property `has: wings` and `can: fly` with a `space shuttle`.

4.3.3 SAIC reasoning to support the explanation strategies derived from the WoZ study

In this section, we present the required reasoning to perform the tutoring strategies discussed in Chapter 3 (Section 3.4.3). Using the data in a student model, the agent can make an informed decision about which tutoring strategies to select for each dialogue episode. Here we describe nine queries that can be employed to implement the tutoring strategies in SAIC (see Section 3.4.3, Chapter 3) and provide examples of their application.

Query 1: *Search for frame*. This query is needed to check whether a frame is found in the domain or the student model. SAIC will check all the facts in the knowledge base to search for a specific frame with any slot and value. The query produces a `Yes/No` result. For example, the result of the search can be used to implement a creation strategy ‘informing the property of the new concept’.

`?(frame, slot, value)`

Result: `Yes`

Example: `?(space centre, isa,  learning_centre)`

Result: Yes

The query can be sent to the knowledge base and the result shows that there is a fact about `space centre` in the knowledge base (e.g. the dialogue modeller may have to check whether the student knows this fact). Alternatively, the query can be sent to the student model to check whether the student has some knowledge about `space centre`.

Query 2: *Search for slot value.* This query is used to check the value of a slot, i.e. when the frame and slot are known.

`?(frame, slot, ?)`

Result: `value_V`

Example: `?(space shuttle, destination, ?)`

Result: `moon`

The example shows that the query is intended to find out the value of `space shuttle.destination`, that is `moon`. The query can be sent to the domain knowledge base and the result can be used by the dialogue modeller to implement an activation strategy ‘asking property value of a schema’.

Query 3: Search for frame by slot and value. This query is used to find a frame with a certain slot and value. The query will return none or all the frames that have the specified slot and value. It is used by the dialogue modeller to find similar schemas.

`?(?, slot, value)`

`Result: {frame_F1, frame_F2,... frame_Fk}`

Example: `?(?, has, wings)`

Result: `{bird, aeroplane, space shuttle}`

The example shows a search for objects that `have wings`. The result of the query can be used by the dialogue modeller to decide what schemas from the student's previous knowledge to activate when explaining `space shuttle`.

Query 4: Search for picture. This query checks the domain knowledge if there is a picture of a specified frame. The query produces a result of either *No* or the name of an external picture file. This query is a specification of *Query 2* applied for a specific slot called `picture`.

`?(frame, picture, ?)`

Result: `{picture_P1, picture_P2, ... picture_Pk}`

Example: `?(space shuttle, picture, ?)`

Result: `space shuttle.jpg`

The example shows that SAIC searches the knowledge base for a picture of the frame `space shuttle`. The result shows that there is an externally linked picture file `space shuttle.jpg` that is a picture of the frame. For instance, the result of the query can be used to implement an activation strategy 'showing picture of a schema'.

Query 5: Search for a superclass and its class member. To find out the superclass of a frame and its class member, SAIC will check the superclass of a frame, i.e. the value of the frame's *isa*, and then search for its class member, i.e. other frames that have the same isa value. The query will return none or all the class members.

`?(frame,isa, X) & ?(?, isa, X)`

Result: `value X & {frame_F1, frame_F2, ...frame_Fk}`

`Example (landing on the moon, isa, X) & ?(?, isa, X)`

Result: `activity & {swimming, activity}`

The example shows a query for the superclass of `landing on the moon` that equals to `activity`, and the class members, i.e. `landing on the moon` and `swimming`. For example, the result of the search can be used to implement an activation strategy 'informing the *isa* category and its instance' (see Chapter 5).

Query 6: Search for similarity between two frames. It compares two frames that are under the same *isa* category in terms of a property and its value, i.e. a same property and value. The query produces a result of none or an isa instance and a similar property-value shared by the frame and the instance.

`?(frame, isa, X) & (F, isa, X) & (frame, Y, Z) & (F, Y,Z)`

Results: `value_X & frame_F,value_X & slot_y, value_Z & frame_F, slot_Y, value_Z`

Example: `?(space shuttle, isa, X) & (F, isa, X) & (space shuttle, Y, Z) & (F, Y, Z)`

Results: `vehicle & aeroplane, vehicle & space shuttle, has, wings & aeroplane, has, wings`

The example shows that `space shuttle` and `aeroplane` are instances of `vehicle`. The frames have a similar slot-value `has wings`. The query searches for the *isa* of the frame `space shuttle`, i.e. `vehicle`, and find an instance of `vehicle`, i.e. `aeroplane`. Then the query finds a slot-value that is shared by `space shuttle` and `aeroplane`. For example, the results of the search can be used to implement the accretion strategy, ‘comparing the concept with its *isa* instance’.

Query 7: Search for a slot-value. To check if a frame has a certain slot and value. SAIC will check if the student already knows about a slot and value of a frame. The query produces a result of `No` or `the slot value`.

`?(frame, slot_S, value_V)`

Result: `slot_S, value_V`

Example: `?(moon, slot_s, value_v)`

Result: `moon, function, reflects sunglight`

The example shows that SAIC checks whether a moon concept has the slot value `function, reflects sunlight`. For example, the result of the query can be used to implement a tuning strategy ‘adding a new property to the new concept that has just been created’, if the slot-value is not found (see Chapter 5).

Query 8: Search for two frames that share a slot but with different values. It contrasts two frames in terms of a property and its value, i.e. the same property but with different values. The query produces a result of a frame, slot and two different values or `No` that indicates that there are no frames that share a slot with different values.

`?(frame, S, T) & (F, S, U)`

Results: `slot_S, value_T & frame_F, slot_S, value_U`

Example: `?(space shuttle, S, T) & (F, S, U)`

Results: `destination, moon & aeroplane, destination, airport`

The example shows that `space shuttle` and `aeroplane` share a slot, i.e. destination, but having different values, `moon` and `airport`, respectively. The query searches for a slot-value that belongs to `space shuttle` and `aeroplane` and then finds a frame with the same slot but a different value. For instance, the result of the query can be used to implement a tuning strategy 'informing a wrong value of the new concept' (see Chapter 5).

Query 9: Search for isa instances that share a slot but with different values. It contrasts two frames that are under the same *isa* category in terms of a property and its value, i.e. the same property but with different values. The query produces a result of none or an *isa* instance and a slot shared by the frame and the instance but having different values. This is a specification of Query 8 applied for specific frames having the same *isa* value.

`?(frame, isa, X) & (F, isa, X) & (frame, S, T) & (F, S,U)`

Results: `value_X & frame_F,value_X & slot_S, value_T & frame_F, slot_S, value_U`

Example: `?(astronaut, isa, X) & (F, isa, X) & (space shuttle, S, T) & (F, S, U)`

Results: `profession & teacher, profession & astronaut, working place, space centre & teacher, working place, school`

The example shows that `astronaut` and `teacher` are instances of `profession`, and both share a slot, i.e. `working place`, but with different values, `space centre` and `school`, respectively. The query searches for an instance of the *isa* and compares them to find a shared slot but having different values. For instance, the result of the query can be used to implement a tuning strategy 'contrasting the new concept with schemas under the same *isa* category' and a restructuring strategy 'comparing and contrasting the new concept with an existing schema' (discussed in Chapter 5).

Based on the facts provided to the system and frame-based reasoning to perform the tutoring strategies as described above, the agent can simulate the reasoning taken by human tutors while explaining new concepts to students.

4.4. Student Modelling in SAIC

This section provides an overview of relevant student modelling approaches in ITS and then explains how to represent a student's schematic knowledge.

4.4.1. Student modelling in ITSs

We follow the argument by Greer and McCalla (1994) that a student model is necessary to generate adaptive behaviour, which is interactive explanations in our case. Student modelling is normally described in terms of the process of *representing specific aspects of student knowledge* that the system intends to support (Kay, 2000). In SAIC, student modelling involves the representation of a student's prior knowledge or schemas that are used in the interpretation of new concepts.

Ideally, a student model should capture *all* the knowledge the student employs in the learning (Gluck, Anderson & Douglass; 2000) and should be similar to how a human tutor dynamically models his student. A formal approach to student modelling is needed to enable computation of the student knowledge in tutoring systems (Self, 1994). The SAIC agent needs to model the student to be able to generate adaptive explanation as performed by a human tutor. Consequently, we need to consider how the agent will initialise and update the student model (discussed in section 4.4.4 and 4.5, respectively).

Intelligent Tutoring Systems diagnose students for different purposes: for example, to choose the pedagogical plan that best responds to a particular diagnosis (Heffernan & Koedinger, 2002), to decide what knowledge to teach (Albacete & VanLehn, 2000; Sleeman et al., 1989; VanLehn, 1988) and to provide the tutor with the information necessary to select a suitable instructional action (Aïmeur, 2002; Kosba, Dimitrova & Boyle, 2004). In the SAIC architecture, the student diagnosis is required to decide which schema to activate, which learning mode to perform and which explanation strategies to select.

Researchers agree that student modelling is a challenging process (Holt et al., 1994; Martin, 1999; Self, 1990). Sleeman et al. (1989) even suggest that feedback tailored to the need of individual student is no more useful than generic feedback. However, recent ITSs have shown that systems with student models can produce higher learning gains, for example AutoTutor (Graesser et al., 2004; Graesser, Person & Harter, 2000), ECOLAB (Luckin & Boulay, 1999) and Algebra Tutor (Anthony et al., 2004). Stern, Beck and Woolf (1996) agree with the complexity of student modelling but contend that a student model may be useful even though it is not complete or accurate. Moreover, advancement in artificial intelligence techniques may automate the construction of a student model (Beck, 2000; Sison & Shimura, 1998). In addition, the use of authoring tools to develop ITSs (for example, Murray, 1999) can reduce the time to construct student model. The benefits of having student models, as shown by recent ITSs, and the findings of the WoZ study (presented in Chapter 3) justify the use of a student model in SAIC.

4.4.2. Overlay student model

The overlay student model is the most common student modelling approach in AIED research (Brusilovsky, 1996 & 2003). The assumption of an overlay student model is that the student's knowledge is a subset of the expert's knowledge in one domain (Carr & Goldstein, 1977). This modelling approach implies that learning is related to the process of acquiring the expert's knowledge that is not available in the student model. To support the learning, overlay-based ITSs normally provide interactions with the student to facilitate the acquisition of the expert knowledge.

The overlay approach has been applied in many ITSs. For example, REDEEM (Ainsworth, Williams & Wood, 2003) employs a basic overlay model to record the system's understanding of the student's knowledge of an area, SQL Tutor (Mitrovic & Ohlsson, 1999) in which the student model is an overlay of the domain module, and CLARISSE (Aïmeur et al., 2002) that represent the learner's knowledge of the domain in terms of what the student knows and does not know.

The main advantage of the overlay approach is that it is easy to implement and in most cases sufficient for planning adaptive system behaviour. The comparison between a student's and an expert's model allows a simple mechanism to support reasoning or inference about the student's cognitive state. When the system has the teaching expertise of a domain, the reasoning of the system is relatively easy as the overlay model points out *what the student has already learned* and *what he should now learn*.

The main disadvantage of overlay approach is that it is unable to give an account of the student's misconceptions in the domain; it is limited to representing the student's knowledge according to the scope on an expert model. The assessment of the student's answers in interactions with overlay-based systems is performed by comparing the student's knowledge with the expert's. This implies that any knowledge outside the expert's knowledge is not considered by the system in its tutoring, or even might be viewed by the system as incorrect.

A buggy model is an approach proposed to deal with the limitations of the overlay model, in which libraries or collections of misconceptions are represented in the student model. The idea of buggy model is that the student's misconception should be represented to allow the system to deal with student's misconception. The inclusion of misconceptions into the student model gives more information about the student to the system. However, constructing buggy model is normally time-consuming and difficult and this limits its application in ITSs (Virvou, 2000). The findings of the WoZ study (presented in Chapter 3) show that human tutors generally avoided explaining new concepts by using student's misconceptions, instead the tutors used the most relevant schemas and focused on schematic knowledge of the schemas. Examples of ITSs that employ buggy models are DEBUGGY (Brown & Burton, 1978), PROUST (Johnson, 1990), and LUCY (Goodman et al., 1998). The discussion of the buggy model as an approach to deal with the limitation of the overlay model points to the requirement to extend the overlay approach to include the student's relevant prior knowledge.

Following the idea of schema theory that states that interpretation of a new concept is based on relevant prior knowledge or schemas, we have to represent both the prior knowledge of the student and the domain concept in SAIC to enable the explanation of the domain concept following the theoretical principles discussed in Chapter 2. The idea of what the student needs to learn (the domain concepts) corresponds with the main idea of the overlay model approach. Thus, we can use the overlay approach to represent the domain concepts in the form of a student's schematic knowledge. The next step is to consider how to represent the student's relevant prior knowledge, discussed in Section 4.4.3.

4.4.3. Representation of student schematic knowledge in SAIC

A SAIC student model consists of two parts: a student's *relevant prior knowledge*, that is an assumption of the concrete prior knowledge the student has, and the student's *already acquired domain concepts*.

- ***Student Relevant Prior Knowledge:*** This part of the student model is used to specify existing schemas that can be used in the interpretation of the domain concepts. This depends mainly on the stereotype *concrete* concepts (discussed in Section 4.4.4) assumed to have been mastered by the students. The stereotype concepts are derived from students' answers in a pre-test and further refined by their class teachers to form individual student models, presented in Chapter 7. `Relevant` is defined as:

  ```
  relevant(S, p):
      p = C,
      isa(p,x) & isa(C, x),
      property(p, y) & property(C, y) or
      property-value(p,y,z) & property(C,y,z).
  ```

  i.e. a student's `S` prior knowledge `p` as is r`elevant` if the prior knowledge is equal to the new concept `C`, it is comes under the same *isa* category with the new concept, shares a property with the new concept or shares a property-value with the new concept.

- ***Student Domain Knowledge:*** This part of the student model is used to determine what schematic knowledge of a *domain concept* a student has acquired. This depends mainly on the information derived from the student pre-test used to initialise the student model, discussed in Chapter 7.

For each *concrete* concept *c*, two values are defined: schematic knowledge of the concept and the *familiarity level* of a schematic knowledge that is later updated based on the student interactions with the agent, see Section 4.5. The knowledge status of the *concrete* concept is used to compute the general student mastery of the schematic knowledge.

To illustrate the knowledge representation in SAIC, we provide an example of how some knowledge about a domain concept `space shuttle` can be represented as frames in the student model.

```
Space_shuttle.isa: vehicle, 4

Space_shuttle.has: wings, 5

Space_shuttle.can: fly, 3

Space_shuttle.destination: moon, 1

Space_shuttle.function: transport_astronaut_into_outer_space, 2

Space_shuttle.process: blast_off_into_outer_space, 3
```

This frame states that the student has acquired a frame `space shuttle` and its schematic knowledge. The `a kind of slot` states that `space shuttle` is a `vehicle` that is a superclass of `space shuttle`. The frame also states that `space shuttle has wings`, `can fly` and `destination` equals the `moon`. The process slot refers to how the function slot `transports astronauts into outer space` is performed, i.e. `blasts off into outer space`.

The student's knowledge level of each schematic knowledge is indicated by its *familiarity number* stated after each slot value. The numbers are assigned on the basis of a class stereotype and its refinement following the student's answers to a pre-test. A low number, for example 1 in "`Space_shuttle.destination: moon, 1`" can imply that the student is not familiar with a specific concept and, therefore, suggests that it has a higher priority to be learned by the student than other schematic knowledge with a higher number. On the other hand, a high number for example 5 in "`Space_shuttle.has: wings, 5` " can imply that the student is very familiar with the schematic knowledge and, therefore, should be used in activation process.

We also need to represent relevant concrete schemas in the student model that can be used in the interpretation of the concept `space shuttle`. Examples of relevant concrete schemas are: `aeroplane`, `car` and `bicycle`. We illustrate here how some knowledge about the schemas can be put into frames.

```
aeroplane.isa: vehicle, 3

aeroplane.has: wings, 4

aeroplane.can: fly
aeroplane.destination: airport, 2

aeroplane.function: transport_people_to_distant_places, 2

aeroplane.process: fly_to_distance_places, 3
```

The frame `aeroplane` is stated as a kind of `vehicle`, which is also a superclass of a `space shuttle`. Both `airplane` and `space shuttle` have two slots with the same values - `has: wings` and `can: fly`. However, the familiarity values can be the same or different. The function of `aeroplane` is to `transport people to distant places` and the process is `to fly to distant places`. The schematic knowledge shows that the frame `aeroplane` is closely related to the frame `space shuttle` because they are of the same class, i.e. vehicle, share two structural slots with the same values, i.e. `has: wings` and `can: fly`, and a structural slot with different value, i.e. `Space_shuttle.destination: moon` versus `aeroplane.destination: airport.`

```
Car.isa: vehicle, 5

Car.has: 4_tyres, 2

Car.function: transport_people_to_places, 3

Car.process: people_drive_cars_to_go_to_places, 3
```

The frame `car` is a kind of `vehicle` like `space shuttle`. The frame has a structural slot `has: 4 tyres`. The function of frame car is to `transport people to places` and its process is `people drive cars to go to places`. The schematic knowledge shows that both `car` and `space shuttle` are of the same class, i.e. `vehicle` but are not closely related because of the different structural slots and different slot values.

```
Bicyle.isa: vehicle, 3

Bicyle.has: 2_tyres, 3

Bicyle.function: transport_people_to_places, 3

Bicyle.process: people_ride_bicycles_to_go_to_places, 3
```

The frame `bicycle` is also a kind of `vehicle` like `space shuttle`. The frame has a structural slot `has: 2 tyres`. The function of frame `bicycle` is to `transport people to places` and its process is `people ride bicycle to go to places`. The schematic knowledge shows that both `bicycle` and `space shuttle` are of the same class, i.e. `vehicle`, but are not closely related because of structural differences and different slot values.

Although using frames we can represent knowledge about the superclass of `space shuttle, i.e. vehicle`, the analysis of human teachers' tutoring transcripts discussed in Chapter 3 has shown that human teachers did not normally perform schema-based cognitive tasks using a superclass, for instance the function of vehicle is to `transport people from one place to another`. Frame also can include information about a subclass of `space shuttle`, for example `Challenger` and `Discovery`. However, human teachers did not normally go deeper into details of a subclass in their explanation of new concepts, as shown in Chapter 3 (Section 3.4). One possible explanation is that a superclass or subclass is abstract whereas its class members, for instance `aeroplane, car` and `bicycle`, are concrete for the student. Hence, we have focused on the representation of a class and members of the class.

4.4.4. Initialising student schematic knowledge using stereotype

An overlay student model does not capture the student's relevant schemas that are required in the interpretation of new domain concepts. Following the idea of schema theory discussed in Chapter 2 about the crucial role of relevant prior knowledge in an interpretation process, we adopt Virvou and Moundridou's (2001) approach of *combining an overlay and a stereotype student model*, which can represent both the student's relevant schema and acquired domain concepts in a student model.

The stereotype approach is a useful mechanism for building student models. Rich (1989), Tsiriga and Virvou (2002) describe the use of stereotypes to infer default assumptions about a student's knowledge. In line with the idea of a student model, Kass and Finin (1988) describe a possible application of stereotypes: to model the student's belief, which implies the development of a student model on the basis of default assumptions (Kay, 2000). This discussion highlights the idea of a student's default beliefs that are used by tutoring systems to build student models.

The stereotype representation approach implies that when the system knows nothing about a specific student, it could use an initial student model for a typical student. Many systems have used stereotypes to model students, for example GRUNDY (Rich, 1979) that is a system that can recommend books to users based on the users' characteristics, UC (Wilensky et al., 2000) that has a set of stereotypes for representing the user's expertise in UNIX, WEAR (Virvou & Moundridou, 2001) and ICALL (Tsiriga & Virvou, 2003) that use stereotypes to initialise a student model in a language tutor. Stereotype student models provide a richer representation of the student's prior knowledge or schema required in the explanation of new concepts. Therefore, in the SAIC agent we adopt a combination of an overlay and stereotype approach to represent student knowledge that can support the reasoning in the agent's explanation process.

Following the above discussion, a stereotype model in SAIC is defined as frequently occurring student's schematic knowledge of a concept. It can be, for example, based on a pre-test (discussed in Section 7.5, Chapter 7). For instance, a new concept `computer` is represented as follows:

```
Computer.isa: electronic device

computer.function: performs calculation

computer.structure – has: monitor

computer.structure – has: keyboard

computer.process: receives input and show output
```

This example shows a student's stereotype knowledge of a new concept `computer` that is represented in his student model. The knowledge represents frequently occurring schematic knowledge a student normally has about `computer`, i.e. the `isa,` `function,` `structure` and `process` of a `computer`. The represented knowledge is *assumed to be true* until the student *explicitly informs* he does not know a schematic knowledge when probed by the agent in the interaction, or he incorrectly answers a question about such schematic knowledge. For example, using the represented schematic knowledge, the agent can search for other instances of `electronic device` and other frames that have `monitor` that can be employed to generate adaptive explanations.

4.4.5. Opening student models to teachers

The main issues in constructing user models are incorrectness and incompleteness (Kay, 2001). These problems arise especially when the user model has been built from limited observations of the student or initiated with stereotypes. Using an incorrect student model, systems will not be able to adapt instruction effectively. The correctness and validation of the student model should be checked before the model is incorporated into a tutoring system.

To address the problem of providing an incorrect student model to ITSs, AIED researchers have proposed the idea of opening the model to the student (Bull & Broady, 1997; Bull, 2004; Dimitrova, Self & Brna, 2001; Zapata-Rivera & Greer, 2003). Researchers (e.g. Bull & McKay; 2004) have discussed the benefits of opening student models, for instance to allow students to perform self-assessment and undertake reflection. Hartley and Mitrovic (2002) also discuss the educational benefits of opening the student model to promote reflection that enhances learning.

In our case, where the students are aged seven to eleven years old, it seems most appropriate that the model should be checked and corrected by teachers who know their students well[5]. The teachers have years of experience interacting interpersonally with the young students especially in reading sessions. Thus, opening the student model to teachers enables them to assess their students and can verify the model to make it more accurate.

An inspectable student model implies that the adopted student modelling approach should be relatively easy to understand (Zapata-Rivera & Greer, 2000). Frame knowledge representation formalism discussed earlier should be understandable to the teachers. The formalism represents overlay and stereotype knowledge in the form of slots and values that can be easily explained as *facts* about objects without describing its reasoning mechanism or programming technique.

4.5. Updating Schematic Student Model

Section 4.4 showed how a schematic student model would be constructed and initialised based on overlay and stereotype approaches. This section will explain how the student model in SAIC will be updated.

4.5.1. Approach to update SAIC student model

A student model is updated to reflect the changes in the student's knowledge that reflects the system's estimate of the student knowledge (Henze & Nejdl, 2001; Wong & Chan, 1997). Similarly, a SAIC student model is updated to reflect the changes in the student's schematic knowledge. As in Mizoguchi and Bourdeau (2000), the SAIC student model should be represented declaratively in order for the system to update the model and interpret new concepts accordingly. The update is necessary to record changes to the student model caused by interaction with the agent; i.e. the student model will be dynamic in nature.

When a student interacts with the SAIC agent, the student model can be updated based on the right and wrong answers given by the student, as suggested by (Blessing, 1997; Hsieh, Halff & Redfiled, 1999). To update a student model, Heift and Nicholson (2001) keep a score that can go up and down, for each node in their student model. Following these update approaches, the estimation of a schema in the SAIC student model can be updated based on the answers given by the student and

---

[5] In general, children at this age may not have well-developed self-assessment skills. This, however, does not exclude that open learner models can be used with children; see Bull and McKay (2004) for a case study.

using a score that can reflect the student's familiarity with the schema. This implies that the SAIC agent does not make an update to the student model *merely* after the student sees a multimedia presentation that contains new concepts; the update is based on the observation of the student behaviour throughout the whole interaction.

In SAIC, the update process is performed by the *student modeller* component. When the student answers the questions asked by the SAIC agent or confirms a statement, the *student utterance analyser* component sends schematic elements of the utterance - frame, property and value - to the *student modeller* that analyses the student contribution and updates the student model appropriately. First, each student input is matched to the facts in the domain model. If the input matched with one fact in the domain model, it is classified as correct. Hence, an update is performed every time the student answers a question, which is an observable verbal act of the student.

Following the dialogue episode strategies presented in Chapter 3 (Section 3.4.3), a SAIC student model is updated to reflect the changes to a *schema, property* and *value* that correspond respectively to *frame, slot* and *value* in frame representation. To record the student's familiarity level with the schematic knowledge, *a numerical value* is associated with each frame. The value will be used by the agent to select which frame to activate and which frame to be used in its utterances.

Adding a new frame

If the frame for the new concept selected by the student is not found in the knowledge base, a new frame will be added. This determines which schema to include in the student model.

Let us assume that the student is interacting with the SAIC agent to learn about `space centre`. After examining the student model, the agent finds that the frame for the new concept is not found in the student model, and therefore `space centre` is considered as not known by the student. Following one of the creation strategies 'Informing its *isa* category' and using a template (discussed in Section 5.7), the agent informs the student that 'A space centre is a learning place'. As a result, the fact or frame is added to the student model. For example:

```
Space centre.isa: learning place, 1
```

This update implies that the new concept is integrated into the student's mental structure next to other concrete instances of learning place, for instance, nursery, school and university. However, a better understanding of the new concept is possible if the student manages to relate it to one of these relevant concrete concepts that should have been activated from the student's LTM (see Chapter 5, Section 5.3).

Adding a new property and value

If the frame for the new concept is already found and the student is learning a new property, a new property is added.

Let us assume that the student is learning about a domain concept `space centre`, and from the examination of the student model, the agent finds a frame or frames about the concept. Following one of the tuning strategies 'Adding a new property to the new concept' (discussed in Section 5.7), the agent informs the student that 'The student of space centre is astronaut'. As a result, the fact or frame is added to the student model to indicate that the student knows one more fact about `space centre, i.e.` in addition to an existing fact or facts about the concept already acquired by the student. For example:

```
Space centre. student: astronaut, 1
```

Changing the familiarity level of a property value

If a frame already has the slot that is in the focus of the current discussion, the familiarity level of the slot will be updated. Based on the status of the user input in the interaction with the agent, the familiarity level of the frame will be modified.

> If the student answers a question about the slot *correctly* or confirms a statement, the familiarity level is increased. To avoid a dramatic change to the level, it is increased by one every time a correct answer is given. An initial value is given to a frame when it is based on a stereotype model. That value is increased by 1 if it is explicitly written by the student in a word-association pre-test (discussed in Chapter 7, Section 7.5). Thus, a high number implies a student's familiarity with a frame.
>
> If the answer is *incorrect*, the familiarity level is decreased by one. A low number, for example 1 and 2, implies that the student is not familiar with the frame. 0 is set as the lowest possible number of the familiarity level of a frame. It indicates that the frame should not be used in a schema activation process.

Depending on the student input, the score will go up or down. The student modeller maintains the student model update by updating the familiarity level for each frame and adding new frames, when needed.

4.6. Conclusions

In this chapter, we have presented the proposed SAIC architecture that is based on schema theory principles and an examination of human teachers' tutoring strategies to explain new concepts. We have shown how the issues illustrated by the WoZ study have led to the main components of the SAIC architecture. We also have shown how schematic knowledge of the domain model and student cognitive model can be represented in the architecture. The architecture is the basis of a schema theory-based computational approach to support children's conceptual understanding.

We have discussed how the components of SAIC work as a system. The schematic knowledge represented in the domain model and the stereotype student model are not restricted to any specific domain or student. Therefore, we argue that a range of domains that can be represented as frames should be suitable to be employed in the SAIC architecture. The characteristics of potential domains will be discussed in Chapter 6. Student models that follow the frame formalism should also be suitable to be used in the architecture.

In order to illustrate how schematic knowledge can be represented in SAIC, we have justified why frames are used to represent schematic knowledge in the domain and in student models. The process of how to update the student model has also been described to show the modelling of changes in student schematic knowledge.

In the next chapter, we will provide a formal description of the schema activation and modification processes. Then, a mechanism to implement the schema-based dialogue and manage the interaction will be presented.

Chapter 5

Dialogue Planning and Adaptive Explanations

5.1. Introduction

In the previous chapter, we presented the architecture of the SAIC agent and discussed its knowledge representation and student modelling mechanisms. This chapter presents the decision-making mechanisms of the SAIC agent to perform schema activation and modification. To provide a computational framework for the cognitive tasks outlined in Chapter 2 and to maintain schema-based dialogue, precise descriptions of the schema activation and modification processes, as well as an appropriate dialogue planning mechanism, are needed. The dialogue of the SAIC agent will be designed to simulate the patterns of human tutors' explanations identified in the WoZ study presented in Chapter 3.

The aim of this chapter is to define the schema activation and modification processes and to describe the schema-based dialogue process to produce formal models executable at a system level. The principles of the learning theory and the dialogue processing have to be defined precisely so that the SAIC agent can make inferences and take decisions to simulate the human teachers' tutoring strategies observed in the WoZ study.

To address the aim, we will start by describing how to model the tutors' support of schema activation and modification following the information processing model. Then, we will provide formal definitions of support for schema activation and modification by defining the cognitive processes of each task. We will also describe the characteristics of the SAIC dialogue. To plan the SAIC dialogue and to represent schema-based dialogue episodes and strategies, we will use a dialogue planning approach known as *dialogue games*. Finally we will explain how to generate adaptive explanations by using template-based linguistic patterns that have been derived in the WoZ study presented in Chapter 3.

This chapter is structured as follows: in Section 5.2, information processing models of schema activation and modification will be discussed. Sections 5.3 and 5.4 will define schema activation and modification, respectively. Section 5.5 will describe the characteristics of the SAIC dialogue and Section 5.6 will present definitions of SAIC dialogue episodes as dialogue games. Finally, Section 5.7 will explain how the agent's dialogue utterances are generated.

5.2 Information Processing Models of Schema Activation and Modification

In this section, we discuss how to model the schema activation and modification processes in order to produce executable models for supporting the cognitive tasks defined in Chapter 2. A brief overview of modelling cognitive tasks in ITSs is presented, and then an explanation of how schema activation and modification are modeled using the Information Processing Theory (Gagne, 1987; Miller, 1956) is provided. The next sections will present the definition of each task with clarification of how the task can be performed using speech acts.

A model can help people to understand and envision what will happen in a learning situation under investigation (Baker, 2000). A computational model of a learning process is developed to understand how the process works. To better understand the cognitive processes and to simulate the support in SAIC, we need a computational model of the tutor's support. This implies the need to describe the support precisely in such a way that it is executable in a computer system.

A cognitive model is the basis of some tutoring systems; for example AlgeBrain (Alpert, Singley & Fairweather, 1999) has a cognitive model of an ideal student problem solver to simulate an expert equation solver; SE-Coach (Conati & VanLehn, 2000) has a cognitive model of a solution process and Help tutor (Aleven et al., 2005) has a cognitive model of the student's help-seeking process. The SAIC agent must simulate an expert tutor who interacts with the student to support his conceptual understanding. Hence, it needs a cognitive model of how a tutor interactively supports a student's schema activation and modification.

Schema theory explains how humans understand new concepts but does not inform how to support students to understand new concepts in computer-based learning environments. Following the ITS development methodology that suggests the system should be designed based on precise descriptions and models (Self, 1999), we need to have a precise model of the tutor's support in order to enable the agent to generate schema-based adaptive explanations.

The processes of schema activation and modification have been depicted as a continuous retrieval of schema from the LTM and the application of the schema in new situations (Kalyuga, 2003). We define the tutor's support as his effort to help a student to activate a relevant schema from the student's LTM and to use the activated schema in the schema modification process. The support is closely related to the Information Processing Theory of human cognition (Gagne, 1987; Miller, 1956) that explains human cognition in terms of *encoding, storage* and *retrieval* (Gross, 1992; Kellogg, 1995): *encoding* refers to the process of translating the incoming information into a mental representation that can be stored in the STM, *storage* is the process of holding information in the LTM, and *retrieval* is the process of recalling information from the LTM. Using the principles of information processing theory, we can explicitly describe the steps in schema activation and modification to produce formal models of supporting the schema-based cognitive tasks, i.e. in terms of encoding of a new concept, storage of an explained concept and retrieval of relevant prior knowledge from the student's LTM.

In a guided reading session, new concepts in a book or multimedia presentation are encountered by the student, i.e. the student will read the lesson and select new concepts for which he requires some explanation. The lack of understanding of a concept can be caused by the child's inability to relate the concept to something familiar, i.e. to the structures in the previous knowledge, see Section 2.3 in Chapter 2. The role of the tutor is to help the student to activate relevant prior knowledge from the LTM into the STM, i.e. schema activation. Then, the tutor needs to decide a learning mode to modify the activated schema by using explanation strategies. The student can activate the relevant prior knowledge and link it to the new schema by following the tutor's dialogue. Hence, from the information processing theory perspective, the role of the tutor is to support the cognitive processes by helping the child to retrieve the appropriate schemas from the LTM and relate them to information in the STM.

We illustrate here the process of supporting schema activation and modification by giving an example of how a new concept is explained.

5.2.1 Supporting schema activation

Consider that in a guided reading session, a student selects a new concept and asks the SAIC agent for an explanation. Following the principles of schema theory, relevant prior knowledge needs to be activated before interpreting the new concept. From the information processing model viewpoint, the new concept corresponds to information *perceived* from the environment that is translated (encoded) into a mental representation and is stored in the STM. The dialogue will discuss relevant schema(s) from the previous knowledge, and thus information will be transferred (retrieved) from the LTM into the STM. Figure 5.1 illustrates this with an example.

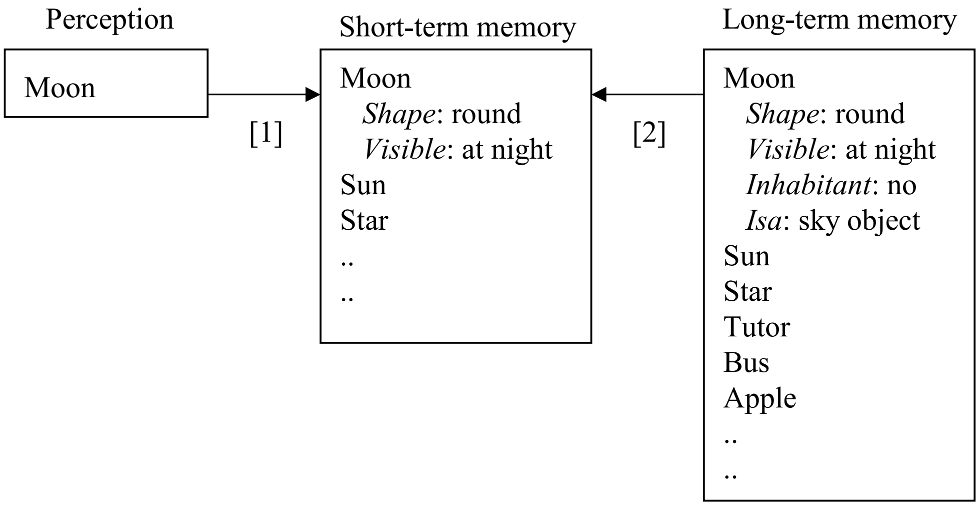


Figure 5.1 Activating relevant prior knowledge from the LTM into the STM– an example

In the example given in Figure 5.1, the student selects a new concept `moon` to get an explanation from the SAIC agent (`moon` is introduced in the STM, see arrow [1]). The student model that represents the student's LTM shows that the student already has some knowledge about `moon` (as well as many other facts that are relevant and irrelevant to the new concept). The role of the SAIC agent is to:

- Look for relevant facts about the new concept in the student's LTM (represented in the student model) using schema-based reasoning, as defined in Chapter 4 and
- Bring them into the student's STM for modification[6] (arrow [2])

5.2.2 Supporting schema modification

When relevant prior knowledge has been brought into the student's STM, it is ready for modification. Schema theory states that the modification involves the use of one of the learning modes: accretion, tuning, restructuring or creation (see Chapter 2, Section 2.3). Modification corresponds to the processing required to change the information in the STM and transfer the processed information into the LTM for storage. Figure 5.2 illustrates this with an example.

---

[6] In principle, it is possible for relevant information to be transferred from the long-term memory to the short-term memory without the help of the agent. We do not exclude such cases but our primary focus is on the information that is brought to the short-term memory with the help of the agent and it is assumed that the information processing model of modification relies only on that information.

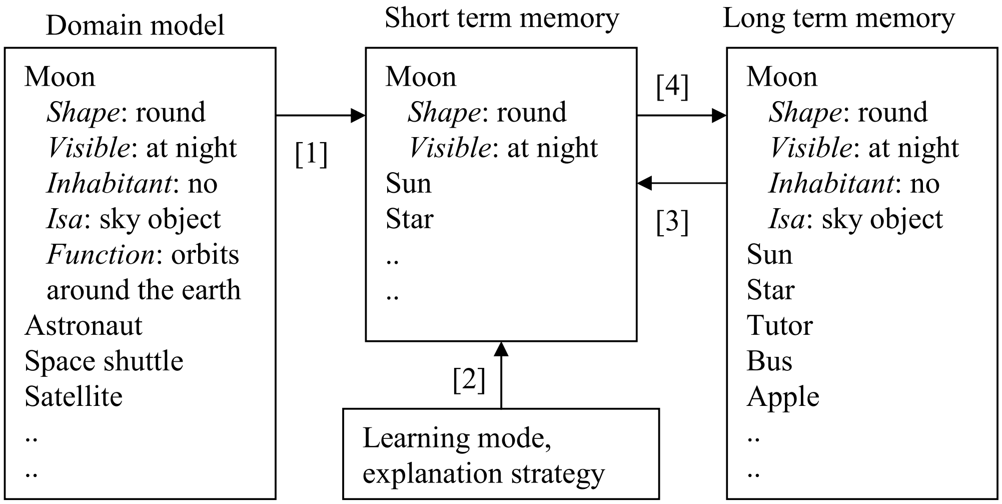


Figure 5.2 Modifying the activated schema – an example

In the example given in Figure 5.2, when the student model is compared to the domain model, the student is found to have an incomplete schema about `moon`. In this case, the agent selects a learning mode *tuning* to explain the new concept. The agent decides to explain the *function* of `moon` ‘`orbits around the earth`’. Consequently, the fact is brought into the STM for processing. Hence, the role of the agent in schema activation is to:

- Find facts about the selected concept in the domain model (arrow [1]).
- Compare the facts with the activated knowledge in the STM
- Select a learning mode and choose an explanation strategy (arrow [2])
- Activate more facts in the student model if required by the strategy (arrow [3])
- Update the student model with the modified schema, see Chapter 4, Section 4.5, (arrow [4]).

Note that the agent assumes that once the schema has been properly modified in the student’s STM, it will be stored in his LTM. This assumption is simplified for the sake of developing a formal description of the processes and a mechanism for updating the student model. In reality, remembering may not involve all knowledge that has been processed in the STM and it may well be the case that certain aspects of the modified schema have been explained by the agent but have not been stored in the student’s LTM. This is taken into account in the design of the dialogue of the agent: throughout the interaction, the agent brings facts from the student model and checks with the child that these facts have indeed been stored, e.g. from previous experience or earlier interactions with the agent.

5.3 Defining the Schema Activation Process

The SAIC agent needs some intelligence to take decisions as to how to support the student's schema activation. This section provides a formal definition of supporting activation cognitive tasks following the discussion in the previous section. In this section we define the activation process (discussed in Chapter 2, Section 2.3.2) to produce a *formal information processing model of supporting schema activation* and define activation tutoring strategies as events in terms of speech acts (discussed in Chapter 3, Section 3.4).

5.3.1 Notations

Self (1999) suggests that a theory of learning can be formalised as a set of statements that describe how the cognitive state of a student changes as a consequence of instructional actions or other events. Self proposes an approach for formalising cognitive processes for designing ITSs by using statements including: `<cognitive_state, event, effect>`. We will adapt Self's model of cognitive change to define the *information processing model of schema activation* by describing the activation tutoring strategies (see Chapter 3, Section 3.4). The structure proposed by Self will be followed while the components will be further clarified:

- **State**. The cognitive state before the instructional event that is denoted as: <`focus_concept, belief, state_of_affair`> where the *focus concept* is a domain concept that is the aim of the agent's explanation, *belief* represents prior knowledge of the student relevant to the focus concept, and *state of affair* is the relevance between the focus concept and a schema.
- **Event**. The instructional event to activate the relevant prior knowledge that is denoted as: `<strategy>` where strategy refers to one of the schema activation strategies.
- **Effect**. The cognitive state of the student after the instructional event that is denoted as `<activation>`, where activation corresponds to what information has been activated in the STM after performing the activation event.

5.3.2 Description of the information processing model for schema activation

Using the above notations, the schema activation strategies (see Chapter 3, Section 3.4), will be described in a declarative way (where small letters will indicate variables and capital letters will indicate constants). Note that the predicate `relevant` used in the definitions below has been defined in Section 4.4.3, Chapter 4 where representation of student schematic knowledge was discussed.

No relevant schema

```
state(focus(C)&Believes(S,r)& not-relevant(r,C)) &
event(no-strategy(A)) →
effect(activated(S,[]))
```

i.e. the student `S` has some prior knowledge before an instructional event but the prior knowledge is *not relevant* to the focus concept `C` then the SAIC agent `A` cannot perform an activation strategy and no relevant knowledge will be activated in the student's STM. Note that the definition above implies that all existing schemas `r` in the student model are checked for relevance with the focus concept `C`.

Asking a property value

```
state(focus(C)&believe(S,r)&relevant(r,C)&property-
value(r,x,v))&property-value(C,x,v) &
event(ask-property-value(A,x,v)&inform(S,r,x,v)) →
effect(activated(S,[r, property-value(r,x,v)])
```

i.e. if the student's prior knowledge includes a schema `r` relevant to the focus concept `C` where `r` and `C` share the same property `x` and value `v`, then if the SAIC agent `A` asks the value of the property for `r`, the student will activate in terms of its property `x` and value `v` into his STM. Thus, r can be used in a follow up modification strategy, as will be defined in the next section.

For example, if the SAIC agent finds out that the new concept `moon` is related to the schema `sun` from the student's prior knowledge and they share the property `location` and its value `in the sky`, the agent will ask the value of the property '`sun.location: ?`' or is '`sun.location: in the sky`?' (see Section 5.7 for the generation of the SAIC utterances). As a result, the student will activate the schema `sun` in terms of its `location` property and its value `in the sky`, i.e. the schema, its property and value have been retrieved from the LTM into the STM and ready for schema modification, presented in the next section.

Asking name of the schema with a property value

```
state(focus(C)&believe(S,r)&relevant(r,C)&property-value(r,x,v)) &
event(ask-schema-name(A,r,x,v)&inform(S,r,x,v) →
effect(activated(S,[r, property-value(r,x,v)]))
```

i.e. if the student's prior knowledge includes `a` schema `r` relevant to the focus concept `C` where `r` and `C` share the same property `x` and value `v`, then if the SAIC agent `A` asks the student to name a schema that has property `x` and value `v` and the student names `r`, the student will activate r in terms of its property `x` and value v into his STM. Note that the description of this activity differs from the previous one only by the way the tutor asks about the schema r.

For example, if the student has some prior knowledge and if SAIC finds out that the new concept `satellite` is related to the schema `moon` from the prior knowledge, as both have the property `function` and the same value `moves around the earth`. The SAIC agent will ask about the name of a schema whose `function` equals to `moves around the earth`. If the student answers `moon` it will be activated (i.e. moved to the student's STM) in terms of having property `function` with value `moves around the earth`.

Showing a picture of a schema

```
state(focus(C)&believe(S,r)&relevant(r,C)&property-
value(r,"picture",p) &
event(show-picture(A,r,p)) →
effect(activated(S, [r, visual-appearance(r)]))
```

i.e. if before an instructional event the student `S` has in his prior knowledge a schema `r` that is relevant to the focus concept `C`, and there is a picture `p` of `r` and the SAIC agent shows `p` to the student, then the student will activate the schema `r` in his memory.

For example, if the student has some prior knowledge and if SAIC finds out that a new concept `satellite` is related to prior knowledge because there is already a schema named `satellite` that has property `picture` then if SAIC shows the picture of the `satellite`, the schema satellite and its visual appearance will be retrieved from the LTM into the STM.

Informing the isa category and its instance

```
state(focus(c)&believe(S,r)&Relevant(r,C)&isa(r,x)&instance(x,j)) &
event(inform-isa-instance(A,x,j) →
effect(activated(S, [r, isa(r,x), instance(x,j)])
```

i.e. if before an instructional event the student s has in his prior knowledge a schema `r` that is relevant to the focus concept `C` and if the SAIC agent informs the students that the *isa* of the schema and an *isa* instance `j`, then after the instructional event, the student will activate the *isa* and the *isa* instance of the schema into his STM.

For example, if the student has some prior knowledge and if SAIC finds out that a new concept `landing on the moon` can be related to the student's prior knowledge , and if SAIC informs the *isa* of the new concept '`isa: activity`' and other instance or instances of `activity` from the previous knowledge such as `returning to the earth`, `swimming` and `training`, the student will activate the schema `landing on the moon` in terms of being an `activity` and will also relate to the other familiar activities.

5.4 Defining the Schema Modification Process

We will define the modification process (discussed in Chapter 2, Section 2.3) to produce a formal information processing model of supporting schema modification and to define the update of the student model (see Section 4.5, Chapter 4 for the process of updating the student model).

The basic assumption of schema modification is that the student needs to modify the activated schema in order to interpret and understand the new concept. As discussed in Section 5.2, the teacher's support corresponds to helping the student to modify the activated schema that is temporarily stored in his STM.

Similar to the formal definition of schema activation in the previous section, schema modification can be formalised as a set of statements that describe how the activated schema changes as a result of instructional actions or other events.
We will define *information processing model of supporting schema modification* by describing the changes in STM and LTM after performing the learning modes (accretion, tuning, restructuring and creation) with the corresponding dialogue episode strategies. The learning modes are defined in Table 3.3 in Chapter 3, and the modification tutoring strategies are discussed in Section 3.4 in the same chapter.

5.4.1 Defining accretion

The basic assumption of accretion is that an activated schema is used to interpret a new concept without changes to the schema's property and value. Before an accretion takes place, it is assumed that a schema has already been activated; otherwise no effective modification will occur. To formalise accretion, a set of possible *events* derived from accretion tutoring strategies are defined as follows:

Asking instance of the new concept

```
state(activated(S,R)&isa(x,R)) &
event(ask-concept-instance(A,R&inform(S,x)) →
effect(accretion(S,R)&believe(S,[isa(x,R)]))
```

i.e. if the student `S` has activated a schema `R` before an instructional event and if the SAIC agent `A` asks an instance of `R`, then if the student answers with a correct instance, the schema will be modified by changing the *familiarity level* of the slot (see Section 4.5.1, Chapter 4) and the information will be stored in the student model (i.e. LTM).

For example, if the student has activated the schema `planet` then if SAIC asks for an instance of `planet` and the student answers `earth`, the student understands that `earth` is an instance of `planet` and this is recorded in the student model. The new concept `planet` and prior knowledge `earth` have a *parent-child* relationship.

Informing the *isa* of the new concept

```
state(activated(S,R)&isa(R,x)) &
event(inform-isa(A,R,x)) →
effect(accretion(S,R)&believe(S, [isa(R,x)]))
```

i.e. if the student has activated the schema `R` for the new concept before an instructional event and then if SAIC informs the student about the *isa* of the new concept, after the instructional event the student will modify the schema and the student model will be updated accordingly.

For example, if the student has activated a schema `space centre` then if SAIC informs that it is a kind of `learning place` then the student will believe that `space centre` is a kind of `learning place`. The new concept `space centre` and prior knowledge `learning place` have a *child-parent* relationship.

Comparing the new concept with its *isa* instances

```
state(activated(S,R)&isa(R,j)& isa(x,j)&property-value(R,p,v)&
property-value(x,p,v)) &
event(compare-with-instance(A,R,x,p,v))→
effect(accretion(S,R)&believe(S, [property-value(R,p,v),
property-value(x,p,v)]))
```

i.e. if the student `S` has activated a schema `R` about the new concept before an instructional event and if the SAIC agent `A` compares the new concept with its *isa* instance (by informing the student of a similarity between the new concept and *isa* instance), then after the instructional event the student will modify the schema and update the student model.

For example, if the student has activated a schema `space shuttle` and then if SAIC compares it with a schema `car` that is a kind of `vehicle` like `space shuttle`, the student understands that `space shuttle` is similar to `car` in terms of the compared property-value, e.g. `has engine`.

Informing an instance of the new concept *isa*

```
state(activated(S,R)&isa(x,R)) &
event(inform-instance(A,x,R)) →
effect(accretion(S,R)&believe(S, [isa(x,R)])
```

i.e. if the student has activated a schema `R` about the new concept before an instructional event and if SAIC informs the student about the instance of the concept, after the instructional event the student will modify the schema and update the student model.

For example, if the student has activated a schema `space centre` then if SAIC informs the student that the schema `school` is also a `learning place` like `space centre`, the student model will indicate that the student believes `space centre` is a `learning place`.

Asking whether the student can see the similarity between the new concept and its parent

```
state(activated(S,R)&isa(R,x)&property-value(R,p,v)&
property-value(x,p,v)) &
event(compare-with-instance(A,R,x,p,v))&inform(S,R,x,p,v))→
effect(accretion(S,R)&believe(S, [property-value(R,p,v),
property-value(x,p,v)]))
```

i.e. if the student has activated a schema `R` about the new concept before an instructional event and if SAIC asks whether the student knows any similarity between the new concept and its superclass, then after the instructional event the student will modify the schema and update the student model.

For example, if the student has activated a schema `space centre` that is a kind of a `learning place` then if the SAIC agent asks the student whether he knows any similarity between `space centre` and a `learning place`. The agent will inform a property-value that is shared by a `space centre` and a `learning place`, e.g. `has teacher`, and the student model will record the new concept with the property-value.

5.4.2 Defining tuning

The basic assumption of tuning is that an activated schema is used to interpret a new concept with slight changes to the schema's property and value. To formalise tuning, a set of possible events derived from tuning tutoring strategies are defined as follows:

Adding a new property to the new concept that has just been created

```
state(activated(S,R)&property-value(R,p,v)) &
event(add-property(A,R,p,v)) →
effect(tuning(S,R)&believe(S, [property-value(R,p,v)]))
```

i.e. if the student `S` has activated a schema `R` before an instructional event and the SAIC agent informs the student about a new property to the activated schema, the schema will be modified by adding a new property and its value.

For example, if the student has activated the schema `satellite` that has been created and the SAIC agent informs the student that the `function` of a `satellite` is to `orbit around the earth`, the student understands what a `satellite` does and this property-value is added to the student model.

Informing a wrong value of the new concept

```
state(activated(S,R)&property-value(R,p,v)&
property-value(x,p,w)) &
event(inform-wrong-value(A,R,p,w)) →
effect(tuning(S,R)&believe(S, [not-property-value(R,p,w)]))
```

i.e. if the student S has activated a schema before `R` an instructional event and the SAIC agent informs the student about the wrong value of a property, the schema will be modified by stating that the schema does not have the specific property-value.

For example, if the student has activated the schema `astronaut` then if the SAIC agent informs him that the *function* of `astronaut` is not `treats_the_patients`, which is a property-value of schema `doctor`, then the student will believe that `astronaut` does not have the property-value.

Contrasting the new concept with schemas under the same *isa* category

```
state(activated(S,R)&isa(R,x)&isa(j,x)&
property-value(R,p,v)&property-value(j,p,w)) &
event(contrast-concept-schema(A,R,j,p)) →
effect(tuning(S,R)&believe(S, [property-value(R,p,v),
property-value(j,p,w)]))
```

i.e. if the student `S` has activated a schema `R` before an instructional event and the schema has a property that is shared by schema `j` but having different values, then if the SAIC agent informs the student about the property-value of the new concept and schema `j`, the schema will be modified by adding the property and value.

For example, if the student has activated a schema `astronaut` before an instructional event and the SAIC agent informs him that the `vehicle` of an `astronaut` is a `space shuttle` but the `vehicle` of a `teacher` is a `car` (where both `astronaut` and `teacher` are a kind of `profession`), the student understands that the `vehicle` of an `astronaut` is a `space shuttle`, while a `teacher`'s vehicle is a `car`.

Informing the *isa* category of the new concept

```
state(activated(S,R)&isa(R,x)) &
event(inform-isa(A,R,x) →
effect(tuning(S,R)&believe(S,[isa(R,x)]))
```

i.e. if the student `S` has activated a schema `R` before an instructional event and the SAIC agent informs the students about the *isa* of the schema, the activated schema will be modified by adding the isa link.

For example, if the student has activated the schema `telescope` that is a kind of `image magnifier` and the SAIC agent informs him about the *isa* of `telescope`, the student understands that `telescope` is a kind of `image magnifier`.

5.4.3 Defining restructuring

The basic assumption of restructuring is that an activated schema is used to interpret the new concept with major structural changes to the schema. To formalise restructuring, a set of possible events derived from the restructuring tutoring strategy are defined as follows:

Comparing and contrasting the new concepts with an existing schema

```
state(activated(S,R)&property-value(R,p,v)&property-value(j,p,v)&
property-value(R,w,y)&property-value(j,w,z))&
event(compare(A,R,j,p)&contrast(A,R,j,w)) →
effect(restructuring(S,R)&believe(S,[property-value(R,p,v),
property-value(j,p,v),property-value(R,w,y],property-value(j,w,z]))
```

i.e. if the student `S` has activated a schema `R` before an instructional event and the schema has a property-value that is shared by an other existing schema, and another property that is also shared but having a different value, then if the SAIC agent compares and contrasts the schema with the other existing schema, the activated schema will be modified by adding the shared property-value.

For example, if the student has activated the schema `earth` then the SAIC agent compares it with `venus` that are both `isa planet` and the SAIC agent contrasts the two instances of planet: `has habitants` and `has no habitants`, respectively, the student understands that `earth isa planet` and `has habitants`.

5.4.4 Defining creation

The basic assumption of creation is that no activated schema is used to interpret a new concept `C`. To formalise creation, a set of possible events derived from creation tutoring strategies are defined as follows:

Informing the property of the new concept

```
state(not-activated(S,R)&property-value(C,p,v)) &
event(inform-property(A,C,p,v)) →
effect(creation(S,C)&believe(S,[property-value(C,p,v)]))
```

i.e. if the student has not activated relevant prior knowledge to a new concept and the SAIC agent informs the student about a property-value of the new concept, a new schema will be created and stored in the student model.

For example, the student does not have relevant prior knowledge about `telescope` and the SAIC agent informs the `function` of a telescope is to `magnify the image of distant objects`, then the student will create a schema about `telescope` with the property-value.

Showing a picture of the new concept

```
state(not-activated(S,R)&property-value(C,"picture", v)) &
event(show-picture(A,v)) →
effect(creation(S,C)&believe(S,[visual-appearance(C)]))
```

i.e. if the student S has not activated a schema of the new concept, and there is a picture `v` of the concept and the SAIC agent shows `v` to the student, then the property picture and its visual appearance will be stored in the student model.

For example, if the student has no relevant prior knowledge about `satellite` then if the SAIC agent shows a picture of `satellite`, the schema `satellite` will be created.

Informing its *isa* category

```
state(not-activated(S,R)&isa(C,x)) &
event(inform-isa(A,C,x)) →
effect(creation(S,C)&believe(S,[isa(C,x)])
```

i.e. if the student `s` has not activated a schema of the new concept before an instructional event and the SAIC agent informs the student about the *isa* of the new concept, the schema will be created by adding the *isa* link.

For example, if the student has not activated schema relevant to `space shuttle` that is a kind of `vehicle` and the SAIC agent informs him the *isa* of the concept, the student understands that the `space shuttle` is a kind of `vehicle`.

5.5 Schema-based Dialogue

5.5.1 Characteristics of the SAIC dialogue

To define the characteristics of the SAIC dialogue, we will follow the findings of the WoZ study presented in Chapter 3. Three main characteristics of the SAIC dialogue can be derived: supporting schema-based cognitive tasks, collaboration, and mixed initiative. For each characteristic, we will discuss how a dialogue management system can achieve the characteristic, and explain its effect on the student's conceptual understanding.

Supporting schema-based cognitive tasks

Supporting schema-based cognitive tasks is the main objective of interactive support of a student's conceptual understanding. It implies supporting schema activation and modification that, according to schema theory, promote conceptual understanding. Interaction makes this support possible by enabling the SAIC agent to help the student to activate relevant prior knowledge and perform schema modification. In addition, a well-planned dialogue would guide the reasoning and facilitate a student's schema activation and modification. Thus, the SAIC dialogue management mechanism should allow the agent to employ suitable explanation strategies for each cognitive task. Following this idea, the interaction should focus on performing each step in the cognitive tasks based on the student's cognitive state.

The aim of supporting schema-based cognitive tasks is to help the student to activate appropriate and relevant prior knowledge or schemas, and to modify the schemas to interpret a new concept. Therefore, the support involves a decision making process to find out which prior knowledge to use and which learning mode (accretion, tuning, restructuring or creation) to apply in a learning situation. The support mechanism should separate the decision making processes into two different steps: selection of an appropriate schema, i.e. what to talk about, and selection of a learning mode, i.e. how to say it. The separation of these processes is normally performed in every dialogue-based system, as discussed by Zinn, Moore and Core (2002).

Collaboration

The schema-based cognitive tasks are distributed between dialogue participants, i.e. the concept of distributed cognition can be relevant. Hutchins (1994) argues that human cognition is distributed across individuals, tools and artifacts in the environment, rather than being solely confined to the boundaries of an individual. Following Salomon's (1993) idea of distributed cognition, Burton, Brna and Pilkington (2000) suggest that sharing of cognitive processes can be interpreted as collaboration, and emphasise the importance of dialogue in such collaboration together with the need for allocating appropriate roles to each participant.

When explaining the use of a learning theory to derive teaching strategies, du Boulay and Luckin (2001) also discuss the idea of distributing cognitive processes as roles between a learner and the other dialogue participant (a system in their case). In this line of thought, we can consider that during the interaction between a student and the SAIC agent, the cognitive process involved in schema activation and modification can be shared between the dialogue participants as dialogue roles. The distribution of the cognitive processes between the agent and the student means that the dialogue roles of dialogue participants are determined before the interaction starts.

The collaborative interactions with SAIC will consider the student as an active participant who uses dialogue to seek understanding of a new concept, i.e. he plays the role of a *help seeker*. On the other hand, the SAIC agent can be considered as a *dialogue partner* who helps the student to understand the new concept. The allocation of the roles can help students, especially the lower achieving ones, to become effective help seekers. Several researchers address the issue of helping students to acquire help-seeking skills, e.g. Aleven et al.(2003, 2005). As stated by Wood (2001), students, especially the low achieving ones, are not effective at seeking help. Hence, a dialogue mechanism should enable the SAIC agent to encourage the student to be more proactive and ask for help if they do not understand a new word or cannot follow the explanation given by the agent. Furthermore, by activating the student's previous knowledge and tailoring the dialogue utterances to the student's cognitive state, the agent can encourage the student to become an active participant in the dialogue and to realise the benefit of asking for help.

Mixed-initiative

The interactive support is conducted in a mixed-initiative environment where the initiative and control of the dialogue are distributed between the pedagogical agent and the student. The SAIC agent will guide the dialogue based on its tutoring strategies to perform dialogue episodes but requires information from the student for its reasoning. The student can take the initiative to ask either when he does not understand a question asked by the agent or when he does not know the answer for that question. This characteristic allows the student to have more control over his learning. In addition, the agent can verify its belief about the student's knowledge using the student's dialogue contribution. This implies that the SAIC dialogue management needs a mechanism to regulate turn taking in the dialogue.

Mixed initiative dialogue can facilitate the student's schema activation and modification. For example the student can confirm which schema is used in an activation process, e.g. by telling that he does not know the answer to a question that can cause the agent to inform him which schema is being activated. The student may alter the line of the SAIC agent's reasoning while performing schema modification, e.g. when the student informs the agent that he does not know the answer or does not understand the explanation, the agent can decide to change the strategy of how a selected concept is explained.

5.6 Using Dialogue Games to Plan the Dialogue in SAIC

We want to manage schema-based dialogues that have episodes as found in the analysis of WoZ scripts. As discussed in Chapter 3, each episode has a well-defined goal and can be modeled as a dialogue game, discussed in Section 5.6.1. For the SAIC dialogue management, we will extend the STyLE-OLM (Dimitrova, 2003) dialogue management approach. While STyLE-OLM handles only student modeling episodes, SAIC includes also schema-based cognitive and management episodes, discussed in section 5.6.3 and 5.6.4.

5.6.1 Mechanisms for planning tutoring dialogue

There are several approaches to plan tutoring dialogue, among which the most widely used are reactive planning, sharedPlans and dialogue games. We will briefly outline these approaches and will then present the approach adopted in SAIC.

Reactive planning

Reactive planning (Georgeff & Ingrand, 1989) is an approach that supports dynamic planning with unexpected changes to the plan. A system is called reactive if it can react in an acceptable amount of time to any changes that occur in the world while the system is running (Wilkins, 1988.) A reactive planning system can react to events that have not been foreseen at the planning stage for different reasons (e.g. because the events were not known at the planning stage or it is too difficult to consider all possible events).

The area of ITS can be used as a domain for developing and testing ideas of the reactive planning approach. There are several reasons to support the application of the approach to ITS: a number of possible effects of the action taken by the tutoring system, the dynamic nature of the learning environments and no guarantee the system knows every single event in the learning environment. Reactive planning has been applied in ITSs, for example Atlas (Freedman, 1999) that is a dialogue manager based on reactive planning that allows the computer and a person to conduct a mixed-initiative dialogue, and TOBIE (Vassileva, 1997) which uses reactive planning to plan contents in instruction.

One cannot fully plan a conversation in advance because it is impossible to predict what the other dialogue participant is going to say. However, SAIC dialogue is designed for the specific purpose of supporting schema-based cognitive tasks. To avoid the complexity of natural language processing, SAIC uses a template-based dialogue where the student selects a menu option to contribute to the dialogue, where all the alternatives are enumerated in advance. If SAIC were to receive free text input from the student, the reactive planning approach would be beneficial. Moreover, all the instructional events of the SAIC agent in a schema-based dialogue have been defined following the episode strategies (see sections 5.3 and 5.4). For these reasons, the capability of reactive planning approach is not needed as the underlying model for its dialogue manager component.

SharedPlan

The sharedPlan theory (Grosz & Sidner, 1990; Lochbaum, 1998) is a formalism for modelling teamwork and collaboration. In tutoring dialogues, participants collaborate to coordinate their actions in order to achieve shared goals (Garland, Lesh & Rich, 2003). Tutoring is a kind of collaboration in which the tutor participant normally has greater expertise and initiative. The representation of the learning task affords the construction of shared plans and joint intentions in order to complete the task with a student. This implies the need for a shared mental model of the task.

The SharedPlan approach is useful in planning tutoring dialogues when dialogue participants know the task to perform and pursue a common solution path. This approach has been applied in several systems, for example COLLAGEN (Rich & Sidner, 1998), CAST (Yen et al., 2001) and STEAM (Tambe, 1997). The shared plan approach has been applied mostly for task-based dialogues where participants have a shared plan about how to perform the task (e.g. buying a travel ticket).

The approach assumes an equal role between the team members. In SAIC, the agent and student share a general goal, i.e. to interact and for the student to understand new concepts in a lesson. However, the student is not expected to know how to perform schema-based cognitive tasks that requires an understanding of cognitive psychology and education. Instead, the SAIC agent is expected to guide the reasoning and make decisions based on inputs from the student. Therefore, this approach does not closely suit the nature of the SAIC collaboration.

Dialogue games

The formal approach is based on Levin and Moore's (1977) linguistic model that considers dialogue games as knowledge structures that represent dialogue patterns organized around specific goals. Burton, Brna and Pilkington (2000) describe the functions of dialogue games as "to enable the focus of the dialogue to be maintained, to support decision making about what moves are available and to help predictions to be made about what might be said next." Burton, Brna and Pilkington define a dialogue game as a state machine that represents all possible dialogue utterances and the sequence of their occurrence. The role of an utterance in a dialogue game is to move a game from one state to another. Thus, if a game is played correctly to the end, it is assumed to be successful.

The dialogue games approach is useful and suitable for tutoring dialogues that have a well-defined structure. Several dialogue management systems adopt this approach, for example CLARISSA (Burton, Brna & Pilkington, 2000), CoLLeGE (Ravenscroft & Pilkington, 2000), STyLE-OLM (Dimitrova, 2003) and MarCo (Tedesco, 2003).

The SAIC dialogue is based on an analysis of WoZ data where dialogue episodes, strategies for each episode, and speech acts for each strategy were identified. Moreover the utterances of human tutors were analysed to examine how the tutors use dialogue to support cognitive tasks. The SAIC dialogue episodes can be represented as games and strategies can be represented as a sequence of moves to take. Therefore, the dialogue game approach is suitable to plan the SAIC dialogue

Most of the dialogues in SAIC are guided by the tutor and take into account the student's questions. When the student is in control of the conversation, the tutor's main priority is to answer the questions asked by the student, when the tutor is in control, there is an agenda to follow in order to help the student to perform the current cognitive task. The tutor leads the student through a planned multi-turn strategy (dialogue game). However, the tutor may need to change or abort a plan depending on the student's responses. A game may become inappropriate if the student cannot follow the desired reasoning. To plan a coherent conversation, the agent needs to evaluate the student and domain model and define a multi-turn game. Depending on the student's input, the agent might continue the game, modify it or initiate a new game based on new circumstances.

The SAIC agent has a dialogue manager component that implements the dialogue games. The goal of the component is to conduct a conversation. When given conditions are satisfied, a dialogue game is triggered. Each dialogue game consists of strategies about how to perform the game in terms of speech act sequences. Hence, the SAIC dialogue manager needs to have the ability to generate utterances to the user to make statements, ask questions and display pictures on the interface.

5.6.2 Definitions of the main concepts in the SAIC dialogue model

Following STyLE-OLM (Dimitrova, 2003), a dialogue episode in SAIC is modelled as a game defined by its *parameters* (specific values for the game, such as the game goal), *specifications* (that describe conditions when a game can be triggered), and *components* (that represent the participants' sub-goals and are structured in a sequence of actions, i.e. plans to accomplish the sub-goals). SAIC will use the same dialogue structure but will extend it to include schema-based dialogue strategies and templates based on the tutors' linguistic forms of speech act identified in Chapter 3.

Based on the STyLE-OLM framework, we define some main concepts to be used in the formalisation of the SAIC dialogue to produce a formal framework for managing the interaction: dialogue participants, communicative acts and dialogue rules.

Dialogue participants

Interactive support of children's conceptual understanding involves two dialogue participants:

- a SAIC pedagogical *agent*, denoted as `A` and
- a *student*, denoted as `S`.

The roles of the participants are determined by the cognitive tasks collaboratively performed during the interaction. The agent participant guides the interaction, and the student participant answers the agent's questions, asks questions or confirms an agent's statement.

Communicative acts

A communicative act is the basic unit of interaction and consists of a *proposition* and an *illocutionary force* (denoted here as dialogue move). A communicative act is defined in terms of: speaker, hearer, move and proposition.

```
<Speaker, Hearer, Move, Proposition>
```

The following eleven dialogue moves are based on the results of the WoZ study presented in Chapter 3. While the agent participant uses all of the moves, the student participant only uses three of them, i.e. *inquire*, *inform* and *confirm*. The moves and their communicative act representations are:

[1] *Inquire*. The speaker asks about a proposition.

`<A, S, inquire, p> or <S, A, inquire, p>`

[2] *Inform*. The speaker informs the hearer about a proposition

`<A, S, inform, p> or <S, A, inform, p>`

[3] *Confirm*. The speaker confirms an informed text is known or understood.

`<A, S, confirm, p> or <S, A, confirm, p>`

[4] *Suggest*. The speaker offers the hearer possible answers to a question.

`<A, S, suggest, p>`

[5] *Praise*. The speaker praises the hearer.

`<A, S, praise, p>`

[6] *Show*. The speaker shows a picture to the hearer.

`<A, S, show, p>`

[7] *Probe*. The speaker asks a question to diagnose the hearer's knowledge.

`<A, S, probe, p>`

[8] *Ask-suggest*. A complex speech act where the speaker inquires and suggests possible answers. It comprises two sequential simple acts.

`<A, S, ask, p> & <A, S, suggest, p>`

[9] *Confirm-praise*. A complex speech act where the speaker confirms the hearer has correct knowledge and praises this. It comprises two sequential simple acts.

`<A, S, confirm, p> & <A, S, praise, p>`

[10] *Compare*. A complex speech act where the speaker makes a statement about similarity between *property values*.

```
<A, S, inform, p1> & <A, S, inform, p2> ... &
<A, S, inform, pk>
```

[11] *Contrast*. A complex speech act where the speaker makes a statement about difference between *property values*.

```
<A, S, inform, p1> & <A, S, inform, p2> ... &
<A, S, inform, pk>
```

Dialogue rules

Dialogue rules are needed to specify the sequence of dialogue moves of the agent (see Table 5.1) and the student (see Table 5.2) in a dialogue turn to ensure the coherence of the dialogue. A rule is defined in terms of a move and its next move:

```
(speaker, hearer, move1, p1) → (speaker, hearer, move2, p2)
```

i.e. the dialogue move `move2` and p2 can occur after `move1` and `p1`.

Table 5.1 Agent dialogue rules

Previous move	Current move
`<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,inquire,p>`
`<S,A,inquire,q>, <S,A,inform,q>, <S,A,confirm,q>`	`<A,S,inform,p>`
`<S,A,inform,q>`	`<A,S,confirm,p>`
`<S,A,inquire,q>,<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,suggest,p>`
`<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,praise,p>`
`<S,A,inquire,q>,<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,show,p>`
`<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,probe,p>`
`<S,A,inquire,q>,<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,ask-suggest,p>`
`<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,confirm-praise,p>`
`<S,A,inquire,q>,<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,compare,p>`
`<S,A,inquire,q>,<S,A,inform,q>, <S,A,confirm,q>`	`<A,S,contrast,p>`

Table 5.2 Student dialogue rules

Previous move	Current move
`<A,S,inform,q>, <A,S,suggest,q>, <A,S,probe,q>,<A,S,ask-suggest,q>, <A,S,compare,q>, <A,S,contrast,q>`	`<S,A,inquire,p>`
`<A,S,inquire,q>, <A,S, confirm, q>, <A,S,suggest,q>, <A,S, probe, q>, <A,S,ask-suggest,q>, <A,S,confirm-praise,q>, <A,S,compare,q>, <A,S,contrast,q>`	`<S,A,inform,p>`
`<A,S,inform,q>, <A,S,praise,q>, <A,S,show,q>`	`<S,A,confirm,p>`

5.6.3 Schema-based dialogue games

Schema-based dialogues are organised as a sequence of dialogue episodes conducted to perform certain schematic and meta-dialogue goals. Each of the dialogue episodes is represented as a dialogue game.

Dialogue game

A dialogue game represents a dialogue episode concerning a particular goal and discussion topic. A dialogue game is defined in terms of *parameters, specification* and *components.*

- *Parameters* represent values specific for the game, which in our case is the goal of the game.
- *Specification* indicates the conditions that must hold in order for the game to take place.
- *Components* describe the algorithm to extract knowledge required for the game and the strategies to perform the game.

Schema-based dialogue games are required to perform schema activation and modification. Schema-based episodes: *activation*, *accretion*, *tuning*, *restructuring* and *creation* are defined as the following dialogue games:

```
activation_game(C) // where C is the focus concept
      Parameters
            Goal: [activated,C]
            Focus: [frame,property,value]
      Specification:
            Believe(S, [r,relevant(r,C)])
      Components
            Relevant_Propositions: frame-property-value(r,x,v)
            List_of_communicative_acts: activation_strategies
            // The activation strategies are described in 5.3.2.
            // Queries to extract knowledge are described in 4.3.3
            in Chapter 4.

accretion_game(C)
      Parameters
            Goal: [accretion,C]
            Focus: [frame,property,value]
      Specification:
            Believe(S, [property-value(r,x,v)&(C,x,v]) // the
            activated schema can be used without changes to its
            property and value
      Components
            Relevant_Proposition: property_value(r,x,v)
            List_of_communicative_acts: accretion_strategies
            // The accretion strategies are described in 5.4.1.

tuning_game(C)
      Parameters
            Goal: [tuning,C]
            Focus: [value]
      Specification: //see Section 5.4.2
            Believe(S,[property-value(r,x,v)&(C,x,w)])
      Components
            Relevant_Proposition: different_value(p)
            List_of_communicative_acts: tuning_strategies
            // The tuning strategies are described in 5.4.2.

restructuring_game(C)
      Parameters
            Goal: [restructuring,p]
            Focus: [property, value]
      Specification:
            Believe(S,[property-value(c,x,v) & not(r,x,v)])
```

```
        Components
              Relevant_Proposition: different_property_value(p)
              List_of_communicative_acts: restructuring_strategy_acts
              // The restructuring strategies are described in 5.4.3.

creation_game(C)
        Parameters
              Goal: [creation,p]
              Focus: [C]
        Specification:
              not-activated(S,r)
        Components
              Relevant_Proposition: create_new_schema(p)
              List_of_communicative_acts: creation_strategies
              // The restructuring strategies described in 5.4.4.
```

To illustrate the use of dialogue games to represent the episodes, we show here how an *accretion episode* is represented as a dialogue game.

```
accretion_game(astronaut)
        Parameters
              Goal: [accretion, astronaut]
              Focus: [astronaut, isa, profession]
        Specification:
              Believe(Student, [astronaut,isa,profession])
        Components
              Relevant_Proposition: property_value(isa, profession)
              List_of_communicative_acts: [inform(property,value)]
```

i.e. the accretion game is defined in terms of parameters: the goal of the game, i.e. to interpret `astronaut` following the accretion learning mode and using the activated schema. The game is triggered when its conditions are met: the agent believes that the student knows about `astronaut`. The agent then extracts the necessary knowledge required for the game: e.g. `astronaut`, `isa` and `profession`. Then it uses one of the accretion strategies (described in Section 5.4.1) to explain the concept `astronaut`.

5.6.4 Dialogue games to manage the dialogue in SAIC

Dialogue management games are required to ensure smooth transition between dialogue games. SAIC dialogue consists of three episodes to manage the interaction: *introductory, diagnose*, and *summarising and confirmation*. The dialogue games of the episodes are defined as follows:

```
introductory_game(L) // where L is a new lesson
      Parameters
            Goal: [introduce(L)]
            Focus: [lesson(L)]
      Specification:
            Believe(S,L)
      Components
            Relevant_Proposition: lesson(L)
            List_of_communicative_acts: introductory_strategy_acts =
                  inform(A,S, [”new”, L]),
                  inform(A,S, [”interesting”, L]),
                  inform(A,S, [”will learn”, L) or
                  inform(A,S,L)

diagnose_game(C)
      Parameters
            Goal: [diagnose,S,C]
            Focus:[C,property-value]
      Specification:
            Believe(A, [not-believe(S,C)])
      Components
            Relevant_Proposition: frame
            List_of_communicative_acts: diagnose_strategy_acts =
                  inquire(A,S, [believe(S,C)]) or
                  inquire(A,S, [believe(S,C,x,v)])

summarising_confirmation_game(C)
      Parameters
            Goal: [summarise_confirm,C]
            Focus: [C, property-value]
      Specification:
            Believe(A, [believe(S,[C,x,v])])
      Components
            Relevant_Proposition: property-value(C,x,v)
```

```
List_of_communicative_acts: summarising_strategy_acts =
      confirm(A,S, [believe(S,C)]) or
      inform(A,S, [believe(S,C,x,v)])
```

5.6.5 Switching between dialogue games

From the above descriptions, we can see that the SAIC dialogue is represented as a series of dialogue games that trigger corresponding schema-based cognitive tasks. At any time during a conversation with the SAIC agent, a student is involved with one of the dialogue games. Hence, a mechanism is required to decide how to switch between the games. A high-level planning loop consists of the following steps:

1. Waiting for a student's input.

The SAIC agent waits for an input from the student.

2. Finding out what the student said.

The student's input is interpreted by the student utterance analyser component. When the student interacts with the agent, the communicative act of his utterance is analysed by comparing the communicative act with the current dialogue state. SAIC examines the consistency of the act with the focus of the active game to relate the proposition of the act to the focus space of the active game.

The examination processes require an inference about the student's cognitive state using a formal model of schema activation and modification (as described in Sections 5.3 and 5.4). To examine the correctness of the student's statement, SAIC requires domain reasoning using domain knowledge (see Chapter 4).

3. Figuring out the intention of the student's speech act.

This step equals to what action has been taken by the student. The student input is categorised into three types of moves: inquire, inform and confirm.

4. Deciding how to respond.

The agent decides the current dialogue game; one of following four categories of dialogue games is active at a time:

- a- *Introductory and diagnose*. The student has clicked on a new concept to learn.
- b- *Activation*. The agent is activating the student's relevant prior knowledge. The agent is performing schema activation using one of the schema activation strategies.

c- *Schema modification*. The agent decides one of the learning modes: accretion, tuning, restructuring and creation. The conditions that trigger a learning mode are determined by the student's cognitive state and the answers he gives to the agent's questions.

d- *Summarising and confirmation*. The agent is completing the interaction by summarising the interaction and confirming the explanation.

These categories of dialogue games are completed sequentially though human tutors may implicitly skip one of the games in their explanations. A game strategy normally involves turn-taking consisting of: initiate, reply and feedback. At the basic level, all decisions about how to respond to the student, including the content, the medium (text or graphics), and whether to retain the initiative or surrender it to the student, are made following the active plan of the current game.

Dialogue management deals with possible operations over an active dialogue game: to change the current game, to proceed to the next game or to terminate the current game. In addition, the agent has to decide which tutoring strategies to employ for each game.

5. Executing the turn.

Responses are processed by the utterance generator that can generate sentences and display pictures (this will be presented in Section 5.7).

6. Return to step 1.

The conversation is complete when the agent has finished performing schema activation and modification tasks and the student has finished asking questions, or if the student aborts the current dialogue.

5.7 Dialogue Utterances Templates

Here we explain how the SAIC agent interacts with the student using template-based utterances. The process of generating utterances involves three components of the SAIC architecture: templates, template selector and utterance generator.

5.7.1 Choosing a language to interact

To implement the communication between the SAIC agent and the student, we need to define appropriate interaction language suited for the context of supporting children's conceptual understanding. We will outline issues to consider when designing a suitable interaction language for the SAIC agent.

The focus on supporting children's conceptual understanding highlights the importance of how to support each of the schema-based cognitive tasks using an interaction language. In the WoZ experimental study presented in Chapter 3, human tutors have supported children’s schema-based cognitive tasks using text in natural language combined with references to supporting pictures. Ravenscroft and Pilkington (2000) highlight the requirement of an explicit interaction language to understand the intention conveyed by the dialogue participants. Following this line of argument, we need a well-defined interaction language that will provide both the agent and the learner with a convenient means to interact and collaboratively perform their tasks.

Oates and Grayson (2004) discuss the inseparability of children's cognitive and language development. Based on the idea that tutors normally adapt their language to the children's cognitive and language development, in the design of the interaction language with the SAIC agent we will follow as closely as possible the main characteristics of the language used by tutors. Hence, the simulated language should provide a way for the pedagogical agent to explain new concepts, and the learners to ask questions or give feedback. *Natural language interface* is one possible approach to simulate a dialogue between a human tutor and a student in a learning environment. Considering that it will be time-consuming for children; as end users of the system, to learn a new form of interaction language that may increase their cognitive load, a natural language interface would be sufficient as a medium of interaction. Moreover, this interface style can promote the use of everyday language as a tool to learn new concepts.

Several researchers, for example Hill (1983), Ratnaparkhi (1998) and Ciaramita and Johnson (2000) highlight the issue of ambiguity inherent in natural language to be considered when employing natural language for human-computer communication. This is critical in designing dialogues with children who may not be able to express what they think or are confused with using clearly defined utterances. If the child’s utterances are misunderstood, inappropriate cognitive tasks may be activated and the interaction may become confusing for the child. We will use a natural language interface that enables expression of utterances using a template and menu to avoid the ambiguity of free style natural language input and the complexity of understanding children’s utterances.

5.7.2 Generating dialogue utterances in SAIC

Deemter, Theune and Krahmer (2003) argue that syntactic template-based approach to generating natural language resembles a linguistic approach. The template-based natural language generation approach can reduce the complexity posed by the syntax of human language in constructing utterances using texts and graphics. Hence, a sufficient and computationally tractable interaction language that resembles natural language interactions can be achieved.

The main focus of the schema-based dialogue in SAIC is to explain new concepts in terms of a relevant schema, its property and property value that are represented in the human tutors' utterances. Researchers have suggested approaches to model human utterances, for example based on intentions and responses (Yamada et al., 1993), a small number of dialogue moves (Ginzburg, 1994), and classification of communicative actions (Pulman, 1999). To model human tutors' utterances, SAIC adopts a linguistic approach proposed by Poesio and Traum (1997) that can represent schema, property and property value in the tutors' utterances. Using this approach, the SAIC agent can compose utterances consisting of speech acts to handle these schematic constructs.

Poesio and Traum (1997, 1998) model dialogue utterances using the notation of Discourse Representation Theory (Kamp & Reyle, 1993) where referents are used to describe utterances: *conversational events* and *contents of utterances*. In the case of the SAIC agent, the referents can be used to represent speech acts and schematic knowledge in SAIC utterances, respectively. The approach also can handle the pragmatic issue of constructing the utterance using a combination of text and graphics.

Following Poesio and Traum's (1997, 1998) utterance modelling approach, human tutors' utterances are defined in terms of dialogue referents: *cognitive state, utterances, events* and *content of utterances*. A human tutor's utterance consisting of a speech act to deal with a schema, its property and property value can be represented using *dialogue referents*.

- *Cognitive state* refers to what the learner already knows: schema, property, property value or their combination.
- An *utterance* refers to a sentence consisting of events and the content of utterances.
- *Events* are communicative acts to perform linguistic actions and
- *Contents of referents* refer to a relevant schema, its property and property value.

The use of templates implies that, to ask for an explanation of a new concept, the student needs to explicitly specify his dialogue moves. The agent constructs its utterance using a suitable template to fill its variables with values from the schema. Therefore, the agent requires explicit templates for each utterance in a dialogue episode. For example, to generate an utterance for an *introductory episode* using a strategy, the utterance is represented as:

Episode: Introductory
Strategy: Informing what the student will learn
Template: Good, you want to learn about `newConcept`.

In this case, using such a template, the agent can generate utterances by filling the value of the `newConcept` variable with suitable values from the lesson, for example, *`space shuttle, satellite`* or *`oxygen`*. All templates have to be prepared for each of the tutoring strategies identified in the WoZ study, and a strategy may have more than one template to cater to utterance variants. This implies that the use of the templates can avoid the complexity of processing natural language utterances that is inheritently ambiguous and unpredictable.

5.7.3 Generating SAIC utterances

Using the templates, the SAIC agent can generate a variety of utterances. To illustrate the actions taken by the agent to compose an utterance using templates, let us assume a student already has a schema `astronaut`.

```
Agent: Do you know how an astronaut goes to the moon?
      Drives a car, drives a bus or drives a space shuttle?
Student: Drives a space shuttle
Tutor: Good.
```

To generate the utterances, the referents of the utterances are represented as follows:

```
Cognitive state: S,C,r
Utterances: u1, u2, u3
Communicative events: e1, e2, e3
Content of utterance: cu1, cu2, cu3

u1: utter(Tutor, ‘Do you know how an astronaut goes to the
moon?’
e1: <A,S,ask-suggest,p> //where p equals to cu1, the
suggestion part is provided in the e2 below.
cu1 = [x, y, z]
      - astronaut(x)
      - goes-to-the-moon(y)
      - do-you-know-how(x, y)
generate (u1, e1, cu1)

u2: utter(Tutor, ‘Drives a car, drives a bus or drives a
space shuttle?’)
e2: <A,S,ask-suggest,p> // where p equals to cu2
```

```
cu2 = [j, k, l]
      - drives a car(j)
      - drives a bus(k)
      - drives a space shuttle(l)
generate (u2, e2, cu2)

u3: utter(Student, ‘Drives a space shuttle’)
e3: <S,A,inform,p> //where p equals to cu3
cu3 = [l]
      - Drives a space shuttle(x)
generate (u3, e3, cu3)

u4: utter(Tutor, ‘Good’)
ce4: <A,S,praise,p> //where p equals to cu4
cu4 = [p]
      - good(p)
generate (u4, e4, cu4)
```

Using the templates, the agent can generate similar utterances when explaining other concepts, for example:

```
Do you know how(x,y)?
        - policemen(x)
        - reduces-crime(y)
(j,k,l)?
        - explains lessons to students
        - prescribes medicines to patients
        - catches thieves
Property value(l)
        - catches thieves
Praise(p)
        - Good
```

5.8 Conclusions

In this chapter, we have presented formal models supporting schema activation, modification and SAIC dialogue that have informed the design of a SAIC agent. The models specify how the SAIC agent makes decisions about the explanation process and updates the student's cognitive state.

The characteristics of the tutors' support have been described in terms of information processing model to explain how the agent views the support. Then we have defined the schema activation and modification to produce executable models of the cognitive tasks. A set of axiomatic statements was used to describe each process in both tasks. Therefore, the formal definitions can be used by the agent to make inferences and decisions based on the knowledge provided to the system.

The characteristics of the SAIC dialogue have been described to clarify the nature of SAIC dialogue. In order to plan a schema based dialogue, we have represented the dialogue using dialogue games. The SAIC dialogue episodes have been represented by dialogue games that have enabled the implementation of the tutoring strategies identified in the WoZ study. Finally, we have explained how the agent can generate utterances using templates.

In the next chapter, we will present a walkthrough and implementation of the SAIC agent in a multimedia educational system '*Going to the Moon'* to validate the proposed architecture. Chapter 7 will present the evaluation of the SAIC agent conducted in classroom settings with real students.

Chapter 6

Implementation of SAIC in '*Going to the Moon'*

6.1 Introduction

In chapters 4 and 5, we have discussed the SAIC architecture, representation of schematic knowledge, dialogue planning and generation of adaptive explanations. It is necessary to demonstrate the applicability of the SAIC architecture to show ITS designers how the formalisations described in chapters 4 and 5 are realised as a pedagogical agent, and how adaptive explanations are generated during an interaction to illustrate the role of the agent.

This chapter aims to present an implementation of the SAIC agent integrated in a multimedia educational system for teaching basic Astronomy to children. To demonstrate the SAIC design approach, we will select a domain for a reading session in the form of a book and then author a multimedia system based on a book that is graded as suitable for children aged seven to eleven years old who are potential users of the SAIC agent. A pedagogical agent is then developed following the SAIC architecture to help the children understand new concepts in the lesson. Finally, we will describe the integration of the agent into the multimedia system and will show examples of explanations generated by the SAIC agent.

This chapter will first discuss in Section 6.2 the introductory Astronomy domain and the chosen lesson '*Going to the Moon'* outlining the new concepts to be learned by the users. Section 6.3 will present a multimedia educational system '*Going to the Moon'* developed based on the lesson. Section 6.4 will describe in detail how the SAIC agent is developed. In Section 6.5, the integration of the SAIC agent into the multimedia system will be discussed, and finally Section 6.6 will illustrate the role of the SAIC agent.

6.2 The Domain: Introductory Astronomy

This section describes an introductory Astronomy domain and the lesson '*Going to the Moon'* selected for the demonstration of the SAIC agent, justifies why the domain was used in the implementation of the SAIC architecture, and describes the main concepts.

6.2.1 The suitability of the domain for a reading session

A domain is suitable for SAIC if it has the following characteristics:

- *It is suitable for children*. The SAIC agent is based on how children think and teaching strategies to help children understand new concepts.
- *The domain can be represented as frames*. The domain concepts should be declarative in nature.
- *There should be suitable multimedia materials*. ITS designers should be able to present the domain concepts using multimedia elements.

Introductory astronomy is a Science subject that has the above characteristics. It is normally taught in reading sessions to introduce basic astronomical concepts to young students. The '*Going to the Moon'* lesson is based on a book *Space Mission* that explains how and why astronauts go into outer space (Potter, 2000). The book is graded following the criteria set by the University of Cambridge Local Examinations Syndicate UCLES (PET, 2004; UCLES, 2001): the choice of words in the book (vocabulary) and its grammar constituents. Hence, the book contains materials suitable for children, and reflects the *Movers* level of the UCLES, i.e. children aged from 7 to 11 years old.

Seeds (2005) explains that introductory Astronomy teaching has two goals: firstly to make students understand where they are in the cosmos, and secondly to make them understand science as a way of learning and knowing about nature. Seeds argues that teachers should not find it difficult to introduce the lesson to students as the aim of an astronomical lesson is not to attract students to become astronomers in the future, and new concepts in the domain should be explainable using everyday language. Similarly, a pedagogical agent developed to explain new concepts using an interaction language based on human tutor-student interactions should be able to guide children's reasoning in understanding basic astronomical concepts.

The lesson '*Going to the Moon'* can provide a means to introduce astronomical concepts, such as, `gravity`, `satellite` and `orbit`. This lesson can be seen as a container of new concepts that a student requires both his relevant prior knowledge and a teacher's help to understand. Therefore, the lesson is suitable to illustrate the SAIC approach. Moreover, the lesson is categorised as non-fiction that encourages curiosity and the desire to read further (Potter, 2000).

As a science domain, the concepts in the *'Going to the Moon'* lesson are presented in a logical sequence according to the stages of the scientific exploration; i.e. astronauts going into outer space. Therefore, the lesson has a structure that is relatively easy to follow if compared to ill-structured lessons, for example language and history. Hence, the students should not have difficulties to follow the lesson during a reading session with a SAIC agent.

6.2.2 Preparation of the lesson

The lesson was prepared to fit a reading session lasting about 20 to 30 minutes. For this reason, the lesson has been shortened to ensure that students should finish their reading within the time constraints, and to allow sufficient time for children to read and interact with the SAIC agent during the session.

The lesson is structured as a sequence of pages. The function of each page is to provide context for the new concepts. Each page contains a few new concepts that a student can choose to learn. Therefore, a page may provide a context in which the student should understand a new concept. The concepts used in the shortened lesson are shown in Table 6.1. The concepts have been ordered according to their appearance in the lesson[7].

Table 6.1 List of 18 new concepts in the lesson '*Going to the Moon*'

1	Astronaut
2	Moon
3	Space shuttle
4	Earth
5	Telescope
6	Star
7	Planet
8	Space
9	Space centre
10	Training
11	Interviewing
12	Gravity
13	Swimming
14	Taking off
15	Computer
16	Satellite
17	Floating
18	Oxygen

For each new concept, we need to prepare its schematic knowledge in terms of *function, structure* and *process* (Kalyuga, 2003) and also to specify its *isa*

---

[7] The number is relatively small as normally a reading session is restricted to a limited number of new words. However, each new concept can have several relevant schemas that significantly increase the complexity of the domain.

relationship (Bratko, 1990). For example, we can use Lingo, an object-oriented scripting language for Macromedia Director[8], to present the schematic knowledge of an `astronaut` taken from the domain knowledge of the agent as the following Lingo facts:

```
astronaut(isa, profession).
astronaut(function, goes-to-the-moon).
astronaut(process, pilots-space-shuttle).
astronaut(can, walk-on-moon).
astronaut(has, uniform).
astronaut(workplace, space-centre).
astronaut(vehicle, space-shuttle).
```

A list of all new concepts and examples of their defined schematic knowledge can be found in Appendix-A. At this stage, the domain for the implementation and schematic knowledge of its new concepts have been specified. The next stage in the implementation is to convert the domain into a lesson in the form of a multimedia system, as will be discussed in the following section.

6.3 Authoring a Multimedia System '*Going to the Moon*'

This section discusses several issues to consider when authoring multimedia educational system and then describes how a multimedia system is authored to integrate the SAIC agent.

6.3.1 Issues to consider when authoring multimedia educational system

Multimedia offers the potential for learners to have access to, and control of, their interactions. Learners could benefit from a rich and varied learning experience. However, their attention can be distracted away from the educational focus (Luckin et al, 2001). Sweller and Chandler (1994) highlight that a multimedia show can split the attention of the student. Hence, we have to consider this trade-off when developing a multimedia system for the selected domain.

Producing educational multimedia systems can be done manually using general-purpose programming languages, for example C++ and Java. Developers of educational multimedia systems normally use commercially available multimedia

---

[8] Macromedia Director is a commercial multimedia authoring tool capable of producing animations, presentations and fully interactive multimedia programs. Its capability can be extended using two scripting languages JavaScript and Lingo

authoring tools, such as Authorware[9], ToolBook[10] and Director[11]. Commercial multimedia authoring packages allow designers to easily create interfaces with sophisticated widgets, for example pop-up menus and radio buttons that could be used as student interface in tutors (Blessing, 1997; Wang & Chan, 2000). The lesson *Going to the Moon* needs to be converted into a visually appealing multimedia system with easy navigation from one page to another. For this reason, a commercial authoring tool has been used to produce a multimedia version of the book.

There are tools developed for authoring multimedia tutoring systems, as discussed in detail by Murray (1999). However, the authoring tools are intended for specific purposes such as curriculum development (CREAM-TOOLS by Nkambou et al, 1996), model tracing tutors (DEMONSTR8 by Blessing, 1997) and approximate reasoning (DIAG by Towne, 1997). The tools will, therefore, not be suitable to produce a SAIC agent that requires its own specific intelligence to support schema-based cognitive tasks.

Considering that multimedia production is an expensive and time-consuming undertaking, there is a strong tendency for designers to select a media element based on design efficiency (Collins, Neville & Bielaczyc, 2000). Norman (1998) highlights the issue of which media element to use and how to make the most of the media. The SAIC agent needs to simulate how human teachers explain new concepts using texts and pictures in one-to-one interaction. A multimedia system that uses textual and graphical media elements will be authored so that the system can provide a learning environment that facilitate the SAIC agent explanations.

6.3.2 Authoring process of the multimedia system

The implementation of the SAIC agent uses a multimedia system called '*Going to the Moon*' that consists of a combination of texts and pictures. To produce a multimedia version of the lesson, we have used the Macromedia Director authoring tool and followed a multimedia authoring process suggested by Gross and Gross (2002), as follows:

1. **Assemble media elements.** Three media elements are employed in *'Going to the Moon'*: texts, pictures and buttons. The texts are taken from a book *Space Mission* and shortened to fit a 30-minute reading session, as discussed in sections 6.2.1 and 6.2.2. Most pictures are based on several other

---

[9] Authorware. www.macromedia.com/software/authorware
[10] ToolBook. www.toolbook.com
[11] Director. www.macromedia.com/software/director

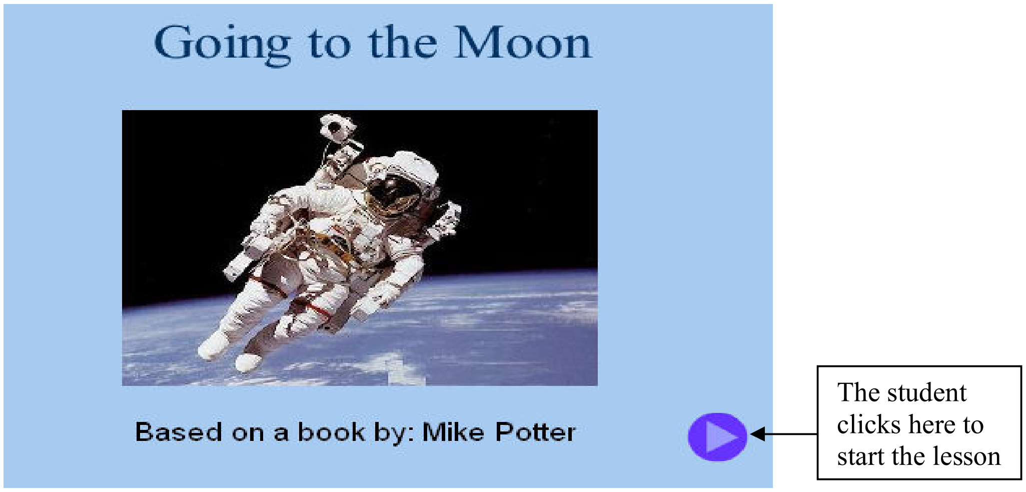
introductory astronomy books written for children (Atkinson, 1990; Mahoney, 2001; Muirden, 1987). The buttons for both navigation and interaction are created using the Director authoring tool.

2. **Position the media elements on stage.** The media elements are arranged in a uniform order throughout the multimedia presentation, see Figure 6.2. The texts are put on the top left side of the screen, the pictures on the top right side of the screen, and the navigation buttons on the bottom right side of the screen. Consequently, students should easily be able to see the relation between the text and the accompanying picture, and the role of the navigation buttons to move backward and forward during a multimedia presentation.

3. **Sequence the media elements**. The multimedia presentation resembles a book page where the story of astronauts going to the moon develops chronologically. Accordingly, the texts and pictures have the same logical sequence so that students can follow the story from the beginning (preparation before departure), middle (landing on the moon), and end (returning to the earth).

4. **Add interactivity.** Interactivity is required to navigate the student through different part of the multimedia presentation. Buttons are used to enable the navigation and also to activate the SAIC agent. Interactivity is possible in multimedia systems through scripting or programming that extends the capability of a multimedia authoring tool.

Screenshots of ‘Going to the moon’ are provided in figures 6.1 and 6.2. Figures 6.3 and 6.4 will show screenshots of the SAIC agent and the integration of the agent into the multimedia system, respectively.

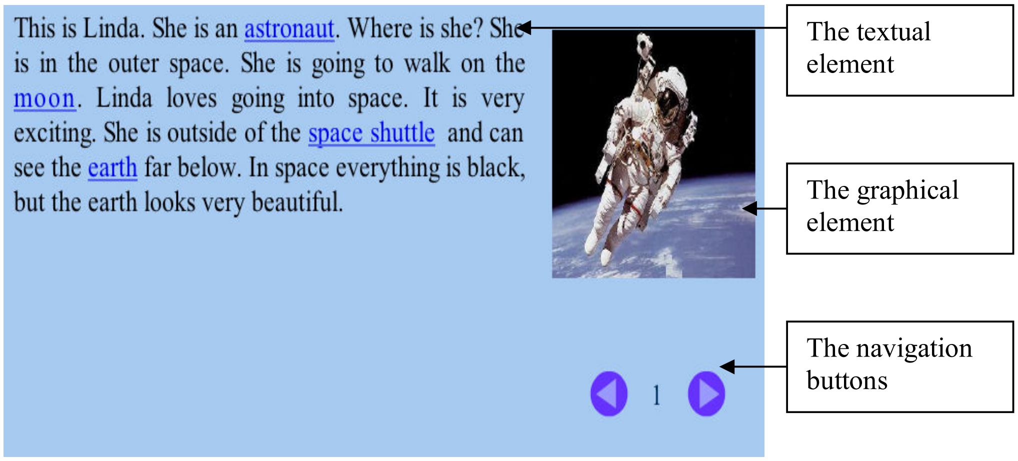


Going to the Moon

Based on a book by: Mike Potter

The student clicks here to start the lesson

Figure 6.1 The front page of the multimedia system



This is Linda. She is an astronaut. Where is she? She is in the outer space. She is going to walk on the moon. Linda loves going into space. It is very exciting. She is outside of the space shuttle and can see the earth far below. In space everything is black, but the earth looks very beautiful.

The textual element

The graphical element

The navigation buttons

Figure 6.2 A screenshot of the multimedia system and the components of the multimedia presentation. When the student clicks on an underlined word, a dialogue box with the SAIC agent is initiated, as shown in figures 6.3 and 6.4 (pg. 119 and 120).

6.3.3 Textual and graphical media elements in '*Going to the Moon*'

Textual elements are the main reading material in the multimedia presentation. Each presentation page consists of texts that present part of the whole story. Each page includes a few new concepts that are underlined, on which the student can click to call the SAIC agent. The click is comparable to a hand raised in traditional reading session to get an explanation from a human teacher. The multimedia system interprets the click event as a request to start a dialogue with the agent and passes the name of the concept as a parameter to specify which new concept needs explanation.

In addition to text, each presentation page consists of a picture to illustrate the text. This follows the WoZ findings reported in Chapter 3 that show human teachers refer to pictures in their explanation, which is one of their explanation strategies. All pictures in the multimedia system are located at the same location for each page to ensure that students should have a quick familiarity with the structure of the learning materials, i.e. the multimedia presentation format. Therefore, when a student progresses to a new page, he will see the text and picture of that page at the same locations as in the previous page.

To simulate a page turn when reading a book, a student can navigate to the next or previous page using navigation buttons (see Figure 6.2). The goal of the navigation is to provide easy access to all pages. Therefore, the system should not have complex menus, such as drop-down and pop-up, that students may take a long time to be familiar with. The focus of the interaction should be to understand new concepts in the lesson instead of learning how to use the system. Hence, we provide a quick access to the SAIC agent using a hyperlink mechanism that calls the agent when an underlined word is clicked, and returns to the current page when an interaction with the SAIC agent is completed or ignored.

6.4 Developing a SAIC Agent

The SAIC prototype, available online at saic.thinkwhyhow.com, was developed using Lingo. The prototype includes 1667 lines and 71 files. The main files are:

- A file to activate the agent using a hyperlink: invoke.dir.
- 9 query files to search for facts in the student and domain models: query1.dir … query9.dir.
- 2 converter files to convert the Lingo facts into Prolog facts when reading the knowledge bases and vice versa when writing the knowledge base: convertToLingo.dir and convertToProlog.dir
- 8 template files to construct utterances based on linguistic patterns: template1.dir … template8.dir
- 8 dialogue game files to manage dialogue episodes: game1.dir … game8.dir.

The knowledge base for the domain and student model was stored as text files. The student cognitive model and domain ontology were built using Protégé (Protégé, 2006) and then converted to frame-based representation followed in the SAIC agent.

The prototype was implemented as an extension to the educational multimedia system, as discussed in Section 6.3, and followed the architecture described in Chapter 4.

In this section we will discuss the development of the main components of the architecture: student cognitive module, dialogue modeller and interface, as illustrated in Chapter 4.

6.4.1 Student cognitive module

The knowledge base for the student cognitive model consists of schematic knowledge represented as frames built on the basis of a stereotype model. To produce a student model that best represents a student's schematic knowledge, we refine each model on the basis of his answers in a word-association test, as will be discussed in Section 7.5 in Chapter 7.

A customised student cognitive model is a subset of the domain model. In addition, it also consists of relevant schemas used to explain new concepts. Therefore, every individual student will have a cognitive model that best represents his schematic knowledge of the new domain concepts. An inference engine in SAIC will consult the knowledge base to decide which schematic knowledge to explain in an interaction and which are relevant to the new concept and might be useful in the explanation. For example, the query to check the *isa* of a new concept uses the following Lingo codes:

```
--function: search for concept isa, parameter = concept name
--input   : new concept
--process : if frame = new concept, if property = ISA
--output  : found or not, if found - the value of the ISA property

on qConceptISA(newConcept)

  --1 read the domain knowledge base
  openDomain
  domainKnowledge = member("domain")
  lastLine = domainKnowledge.line.count

  --2 initialize variable
  conceptISA = "NOT FOUND"

  --3 check the input against all frames in the prior knowledge
  if lastLine > 0 then
    repeat with i = 1 to lastLine
      currentFrame= domainKnowledge.line[i]
      lingoFact = convertFrameLingo(currentFrame)
      myFrame    = lingoFact.word[1]
      myProperty = lingoFact.word[2]
      myValue    = lingoFact.word[3]
      --4 query the prior knowledge
      if myFrame = newConcept then
        if myProperty = "isa" then
          conceptISA  = myValue && myFrequency
        end if
      end if

    end repeat
  end if

  --5 return the output
  return conceptISA
 end
```

These Lingo codes open the domain model from an external file named domain.txt. Then, the agent searches for the *isa*-value of the new concept. Using the *isa*-value of the new concept, the agent can search for *isa* instance in the student cognitive model. To find the most familiar prior knowledge, the following Lingo functions are used:

```
--function: to find the schema with highest frequency number
--input: a list of complete frame
--output: return maximum in descending order


global saved
on qHighest(completeFrame)
    --1 how many frames?
  frameNo = completeFrame.count
    --2 initialize list
  saved = []
  highest = []

  --3 check with each element
  if frameNo <> 0 then
    repeat with i = 1 to frameNo
      tmpMax = 0

      --4 compare the frequecy of the frames
      repeat with j = 1 to frameNo
        currentNo = integer(completeFrame[j].word[4])

        tmpFunction = functionCheck(j)

        --5 if an element is still not compared
        if tmpFunction = TRUE then
          if currentNo > tmpMax then
            tmpMax = currentNo
            tmpElement = j
            currentFrame = completeFrame[j]
          end if
        end if

      end repeat
      --6 saved the compared line and the highest
      add saved, tmpElement
      add highest, currentFrame
    end repeat
  end if
  return highest
end

--to check if an element already compared
on functionCheck(j)
  checkElement = TRUE
  savedNo = saved.count

  if savedNo > 0 then
    repeat with k = 1 to savedNo
      currentSaved = saved[k]
      if currentSaved = j then checkElement = FALSE
    end repeat
  end if
  return checkElement
end
```

These two Lingo functions check the *isa* instances to find out the most familiar schema to the student to be used in the explanations.

6.4.2 Dialogue modeller

The dialogue modeller module is the reasoning part of the architecture. It uses student inputs, elicits knowledge from the domain module and consults student cognitive models to infer how to explain a new concept selected by a student in an interaction. After the reasoning process, it will select a template to generate utterances.

In order to support schema-based cognitive tasks, the SAIC agent has to make decisions at three main stages: schema activation, schema modification and generating text-based natural language output. To perform the decision making process, the agent uses speech acts, as described in Chapter 3, and engages a student in a dialogue. The verbal actions are externalized as utterances that are understandable by the student. The student will read utterances that have been composed based on templates and provide input using mouse click.

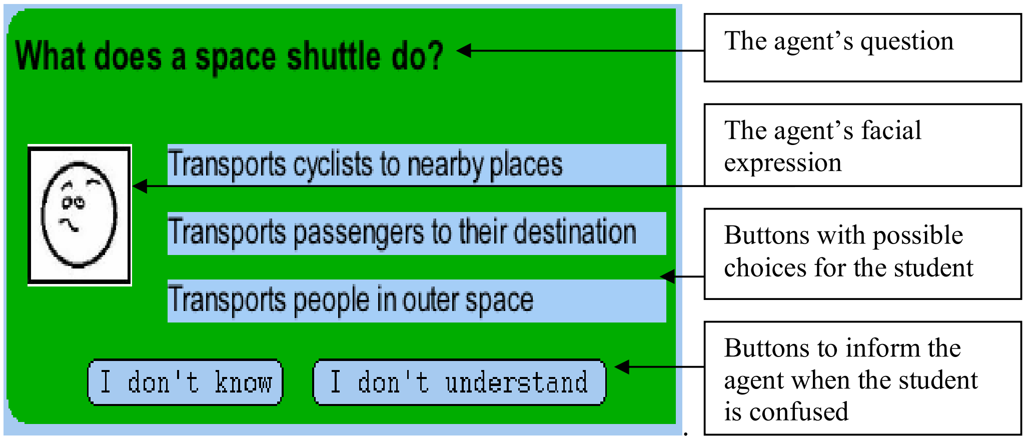
For example, the decision to select an accretion strategy and its matching dialogue game is coded in Lingo as follows:

```
--Function: to decide an accretion strategy
--Based on frequency of a schematic knowledge

global newConcept, gDialogueGame
on accretion_game
  gDialogueGame = "accretion_game"
  gUtterance = 1
  strategy1 = TRUE  -- set 1 to TRUE for initialisation
  strategy2 = FALSE
  strategy3 = FALSE
  strategy4 = FALSE
  strategy5 = FALSE

  --1 inform ISA
  conceptISA = qConceptISA(newConcept)
  if conceptISA <> "NOT FOUND" then
    conceptISAvalue = conceptISA.word[1]
    conceptISAfrequency = conceptISA.word[2]
    if conceptISAfrequency <= 5 then strategy1 = TRUE
  end if

  --2 inform ISA instance
  frameInstance = qInstance(newConcept)
  frameInstanceNo = frameInstance.count
  if frameInstanceNo < 5 then strategy2 = TRUE

  --3 check the frequency number
  if frameInstanceNo >= 5 then strategy3 = TRUE

  --4 check the frequency number of the shared property
  shareProperty = qInstanceProperty(newConcept)
  sharePropertyNo = shareProperty.count
  if sharePropertyNo > 0 then
    shareProperty = qHighest(shareProperty)
    tmpFrame     = shareProperty[1].word[1]
    tmpProperty  = shareProperty[1].word[2]
    tmpValue     = shareProperty[1].word[3]
    tmpFrequency = shareProperty[1].word[4]

    if tmpFrequency < 5 then strategy4 = TRUE
    if tmpFrequency >= 5 then strategy5 = TRUE

  end if

  if strategy1 = TRUE then gDialogueGame = "accretion1"
  if strategy2 = TRUE then gDialogueGame = "accretion2"
  if strategy3 = TRUE then gDialogueGame = "accretion3"
  if strategy4 = TRUE then gDialogueGame = "accretion4"
  if strategy5 = TRUE then gDialogueGame = "accretion5"
  exit
end
```

This example shows how SAIC decides on an activation strategy based on a student’s familiarity with a schematic knowledge, as indicated in the student model. The decision sets a current dialogue game `gDialogueGame` that is called accordingly.

6.4.3 Interface of the SAIC agent

The agent uses a text-based interface to communicate with the student. Researchers have shown that facial expressions can improve agent-student interaction (Lester et. al, 1997; Pelachaud, 1996; Person et. al, 2000). Following the interface approach, the SAIC agent is shown to the student as a face whose appearance changes based on types of task the agent is performing and the student's input: happy when explaining new concepts, puzzled when asking question, smiling when the student answers correctly, and confused when the answer is incorrect. For example, when asking a question the agent looks puzzled, and then it looks happy if the student has answered a question correctly, see Figure 6.3.



What does a space shuttle do?

Transports cyclists to nearby places
Transports passengers to their destination
Transports people in outer space
I don't know
I don't understand
The agent's question
The agent's facial expression
Buttons with possible choices for the student
Buttons to inform the agent when the student is confused

Figure 6.3 The components of the SAIC agent's interface

Figure 6.3 shows that the student is provided with options to select:

- One of the three answers
- A button to inform that he does not know the answer, and
- A button to inform that he does not understand the question.

The decision of what question to ask depends on the student cognitive model. During an *activation episode*, the agent will ask about schematic knowledge that the student already has, as indicated by the student model and explicit student answers during the interaction. During a *learning mode* episode (accretion, tuning, creation and restructuring), the agent will ask the student about schematic knowledge that a student does not have or is not familiar with, also as indicated by the student model and explicit student answers during the interaction.

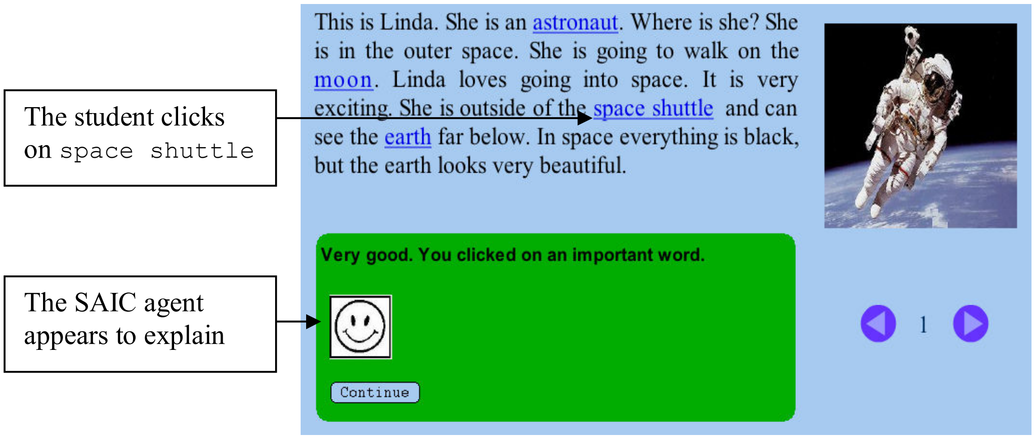
The options are based on the student cognitive model. Different students may have different option items based on their current knowledge as represented in the cognitive model. One student should have different questions and option items when his student model changes that reflect changes in his cognitive model. Templates are used to generate questions and answers (discussed in Section 5.7, Chapter 5).

6.5 Implementation of SAIC in an Educational Multimedia System

This section presents the tasks carried out to implement the SAIC design approach as an extension to a conventional multimedia educational system. We will show that it is possible to apply the SAIC architecture within conventional educational multimedia systems that can be utilised in reading sessions. A multimedia system can be easily applied to a reading session as an interactive learning material that is a logical step to reading ordinary books.

6.5.1 Integrating the SAIC agent into *'Going to the Moon'*

The SAIC agent is developed as an independent component or sub-system from the multimedia system '*Going to the Moon*'. A student should not have any difficulty to call the agent or to bother whether the agent is part of the learning material. Instead, he can easily realise that a pedagogical agent will appear if he clicks on a new word underlined in the multimedia system, see figures 6.1, 6.2 and 6.3. Therefore, his focus can be directed to the explanation provided by the agent. This ensures that he is engaged in the interaction and his reasoning is guided as to how to understand new concepts in the lesson.



This is Linda. She is an astronaut. Where is she? She is in the outer space. She is going to walk on the moon. Linda loves going into space. It is very exciting. She is outside of the space shuttle and can see the earth far below. In space everything is black, but the earth looks very beautiful.

Very good. You clicked on an important word.

The student clicks on `space shuttle`

The SAIC agent appears to explain

Figure 6.4 This screenshot shows the SAIC agent is activated and integrated into the multimedia presentation

Figure 6.2 shows the multimedia system without the SAIC agent being activated. Figure 6.3 shows the agent appears when the student clicks on one of the new concepts, i.e. `space shuttle`, and Figure 6.4 illustrates how the student interacts with the agent.

6.5.2 Calling and interacting with the SAIC agent

To implement the metaphor of an *interactive reading session* (discussed in Section 4.2.2, Chapter 4) a student interacting with the multimedia system always has the opportunity to ask for an explanation of a new concept. In normal reading sessions, where a human teacher reads a book with a group of students, a student can ask questions about difficult words by raising his hand. Similarly, when interacting with the multimedia system, a student can ask by clicking on a specific word in the lesson. However, unlike human teachers, the agent will always have time to entertain the student's request for help.

The instructional intervention provided by the agent may help the student understand new concepts in the lesson. Viewed as a cognitive tool to support a very specific task, i.e. to help children understand new concepts, the agent is not intended to replace the role of human teachers in the reading session. Instead, the agent can be regarded as a supplementary learning tool that can be adopted without changing existing teaching practice.

The agent uses the answers and confirmations provided by the student as input in its reasoning process that enables the agent to make decisions about which schema to activate, which dialogue game to execute and which explanation strategy to use. When the student follows the dialogue, the agent generates explanations adapted to the student's cognitive model. Akin to a traditional reading session, the student can stop or ignore a dialogue with the agent and can continue with reading the multimedia material.

6.6 Examples of SAIC Adaptive Explanations

In this section, we will show examples of interactions with the SAIC agent illustrating how it supports schema-based cognitive tasks and manages the dialogue. Using the examples, the behaviour of the agent can be described in terms of how it activates relevant prior knowledge, chooses an appropriate *learning mode*, (i.e. accretion, tuning, restructuring or creation), and selects suitable explanation strategies.

To illustrate the interaction with the SAIC agent, we will show an example of how the agent explains the new concept `space shuttle`. The following sub-sections will explain each of the game in the example.

Diagnose game
SAIC “Do you know what the word space shuttle means?”
{“Yes”, “No”}

Student “Yes”

SAIC “You think you know what a space shuttle is. Good”

Activation game
SAIC “A space shuttle is a kind of: ...”,
{“Fruit”, “Animal”, “Vehicle”.
“I don’t know”, “I don’t understand”}

Student “Vehicle”

SAIC “Brilliant. Space shuttle is a kind of vehicle. Examples of vehicles are: car, bus and bicycle.”
SAIC “Like aeroplane, space shuttle has wings.”
{“ Yes”, “ No”.
“I don’t know”, “I don’t understand”}

Student “Yes”

Tuning game
SAIC “The destination of an aeroplane is an airport but the destination of a space shuttle is the moon.”
{“Continue”}

6.6.1 Example of a diagnose game

At the beginning of an explanation, the agent wants to check whether the student knows something about the new concept `space shuttle`, though the student cognitive model shows that there are facts recorded about the schema. The diagnosis is required to confirm the student’s knowledge of the fact. In the example, the student cognitive model indicates that the schema `space shuttle` has been created in the student cognitive model. An excerpt from the student cognitive model is given below:

```
space-shuttle(isa, vehicle, 3).
space-shuttle(function, transports-astronauts-to-moon, 0).
space-shuttle(process, blasts-into-the-outer-space, 2).
space-shuttle(can, fly, 3).
space-shuttle(has, wing, 4).
```

The SAIC agent diagnoses the student's understanding of `space shuttle` by asking the student about the schema. On the basis of the affirmative answer `Yes` given by the student, the agent confirms that the schema is created.

```
SAIC "Do you know what the word space shuttle means?"
//The agent probes the schema.
{"Yes", "No"}

Student "Yes"

SAIC "You think you know what a space shuttle is. Good"
//Confirmation that the schema is created.
```

This example shows that the student answer `Yes` to indicate that he knows something about the new concept but needs explanation. If the student answers `No`, he explicitly states that he does not know what it means. The agent repeats the statement in a full sentence and praises the student for having confidence in his knowledge. In addition to the praise, there is a variety of praises generated randomly by the agent, such as `Very good`, `Brilliant` and `You're clever`.

6.6.2 Example of an activation game

The student has explicitly stated that he understands what the word `space shuttle` means. Starting from this confirmation, the SAIC agent wants to activate some schematic knowledge about the `space shuttle`. The domain model contains the following facts about `space shuttle`:

```
space-shuttle(isa, vehicle).
space-shuttle(has, wings).
space-shuttle(destination, moon).
```

The student cognitive model contains these facts about relevant concrete concepts:

```
apple(isa, fruit, 4).
cat(isa, animal, 3).
car(isa, vehicle, 5).
bus(isa, vehicle, 4).
bicycle(isa, vehicle, 3).
```

The student cognitive model shows that the student knows some categories or superclasses of schemas, i.e. `fruit`, `animal` and `vehicle`. One of the activation strategies is “Informing the *isa* category and its instance”, see Section 3.4.3, Chapter 3. The agent uses the query described in Section 4.3.3, Chapter 4, to search for a list of superclasses in the student model and finds fruit, animal and vehicle that are used to generate the multiple-choice answer.

```
SAIC “A space shuttle is a kind of: ...”,
{“Fruit”, “Animal”, “Vehicle”.
“I don’t know”, “I don’t understand”}
// Previous knowledge is activated to probe the relevant
group

Student “Vehicle” //The right category is confirmed.

SAIC “Brilliant. Space shuttle is a kind of vehicle.
Examples of vehicles are: car, bus and bicycle.” //
Confirm-praise and activate relevant schemas. Car, bus,
and bicycle are chosen as the student cognitive
model indicates that they are the most familiar vehicles
to this child.

SAIC “Space shuttle has wings.” // The agent activates
the structure of space shuttle from the previous
knowledge.
{“ Yes”, “ No”.
“I don’t know”, “I don’t understand”}

Student “Yes”
```

The example shows that the student answers correctly the questions, i.e. `space shuttle` is a kind of `vehicle`. The SAIC agent praises the student for the correct answer. Then, the agent informs the student about instances of `vehicle` that are familiar to the student: `car, bus` and `bicycle`. In addition, the agent also activates a property a `space shuttle` that is familiar to the student, i.e. `has wing`, and asks the student to confirm his knowledge about the schematic knowledge. In this example, the student answers `Yes` that indicates that he knows about the schematic knowledge of ‘`space shuttle, has, wings`’.

At this stage of explanation, the SAIC agent has helped the student to activate the schema `space shuttle` from his LTM into his STM, see Section 5.3, Chapter 5. In addition, relevant prior knowledge and schematic knowledge of the schema are also activated, i.e. `car, bus, bicycle` and `has wings`.

The student can select one of the suggested answers, or inform either he does not know the answer or does not understand the question. For example, the SAIC agent asks ‘`Space shuttle is a ....’,` and provides a list of possible answers: `fruit, vehicle` and `animal` (based on the prior knowledge of the student).

- If the student answers the question correctly, the SAIC agent will praise the student, i.e. `SAIC “Good. Space shuttle is a vehicle”.` In the example, the student’s answer is correct if it equals to `vehicle` as stated in the domain model, i.e. `space-shuttle.isa: vehicle.`
- If the student selects an incorrect answer, the agent will utter a polite remark stating that the given answer is not correct. For example: `SAIC “I don’t think space shuttle is an animal”.` The student’s answer is considered as incorrect if it does not equal to `vehicle`.
- If the student informs the agent that he does not know the answer, the agent will inform example/s of `fruit` and of `vehicle`, and then asks the student to think whether he knows the answer after the explanation.

```
SAIC “Ok. I’ll help you”.

SAIC “You know that apple is a fruit”.

SAIC “You also know that cat is an animal”.

//Space shuttle is...?

SAIC “Do you know the answer now?”

SAIC “Space shuttle is a ....’
```

- If the student informs that he does not understand the question, the agent will explain what the question means by giving examples, and then suggests the student to think about the answer to the question.

```
SAIC “Ok. I’ll help you”.

SAIC “I’m asking you a space shuttle is a kind of what”.

SAIC “For example, a cat is a kind of animal”.

SAIC “An apple is a kind of fruit”.

//Space shuttle is...?

SAIC “Do you understand the questions now?”

SAIC “Space shuttle is a ....’
```

6.6.3 Example of a tuning game

The SAIC agent finds that the student does not know schematic knowledge about `space shuttle`, e.g. `space-shuttle(destination, moon)`. However, the student cognitive model shows that the student believes these facts:

```
aeroplane(isa, vehicle, 3).
aeroplane(has, wings, 4).
aeroplane(destination, airport, 2).
```

The student cognitive model shows that the `destination` property of `space shuttle` is found not in the child's cognitive model but he has the property for the frame aeroplane. In this case, tuning can be used to explain space shuttle, i.e. using the activation strategy 'contrasting the new concept with schemas under the same isa category'. Hence, the tuning explanation mode is chosen. A contrasting explanation strategy is initiated that triggers several dialogue games (episodes), as shown in the following dialogue extract.

```
SAIC "The destination of an aeroplane is an airport but
the destination of a space shuttle is the moon."
// Tuning space shuttle by contrasting with aeroplane.
{"Continue"}
```

To perform the tuning strategy, the SAIC agent compares the schematic knowledge in both the student and domain models and finds out that the cognitive model does not have the frame for `space_shuttle.destination = moon`. It queries the cognitive model to find a schema under a `vehicle` superclass that has a `destination` property. A schema `aeroplane` has the same property but with different value. The conditions satisfy the requirement for a tuning game, provided that `aeroplane` is activated in the STM.

6.7 Conclusions

This chapter has described in detail how the SAIC agent prototype was implemented to demonstrate the proposed SAIC architecture of a schema theory-based computational approach to support children's conceptual understanding. We have shown how the SAIC agent interacts with students to generate adaptive explanations of new concepts based on the principles of schema theory and human teachers' explanation strategies. The tasks carried out to build the main components of the SAIC agent as an extension to a conventional multimedia system were described in detail.

We have presented how the Astronomy domain '*Going to the Moon'*, which is based on a graded book, suitable for the age group. We have presented the domain as a container of new concepts to be learned by the students. The process of incorporating a student's schematic knowledge of the new concepts was also shown in this chapter.

We have also described how the domain '*Going to the Moon'* was converted into a multimedia educational system. The authoring of the multimedia system was discussed in terms of its textual and graphical media elements and their sequence. How students can navigate in the multimedia educational system has been illustrated with sample screenshots.

We have also described in detail how we developed a SAIC agent based on the architecture in terms of its interface, domain model, student model and dialogue. The integration of the SAIC agent into the multimedia system was elaborated. Finally we have provided a sample of generated explanations to illustrate how the SAIC agent infers how to explain based on conditions.

The developed SAIC agent has been shown as capable of simulating the help provided by human teachers when they explain new concepts to students. The next logical step is to empirically evaluate the SAIC agent to measure its effectiveness in supporting students' conceptual understanding and its usefulness for teaching and learning new concepts. The empirical evaluation of the SAIC agent will be discussed in the next chapter.

Chapter 7

Evaluation of the SAIC Pedagogical Agent

7.1. Introduction

In the previous chapter, we discussed the development of a SAIC agent prototype as an implementation of our proposed ITS design architecture. The subsequent integration of the prototype into a multimedia educational system demonstrated the agent's applicability to existing educational systems. An empirical evaluation of the agent has been conducted to assess the effectiveness and usefulness of the agent, which can validate its design architecture.

In this chapter, we will present an evaluation study of the SAIC agent used with primary school children to support reading sessions. The data collected will be examined to assess the effectiveness and usefulness of the agent in supporting children's conceptual understanding. We will focus on the theoretical claims outlined in Chapter 2 and will search for evidence to validate the SAIC design principles derived from these claims. To address the research hypotheses discussed in Chapter 1, we need to answer the following questions:

- Is educational multimedia software with the SAIC agent more effective at explaining new concepts to children than educational multimedia software without the agent?
- Do the children who use the SAIC agent to learn new concepts improve their conceptual understanding and what types of improvements are indicated?
- What do teachers and children feel about the use of the SAIC agent to help explain new concepts and provide support to traditional classroom reading sessions?

We have conducted both formative and summative evaluation and will analyse the collected data to answer the questions above.

The chapter is structured as follows: Section 7.2 provides an overview of Tutoring System evaluations. Section 7.3 outlines the aims of the SAIC evaluation. Section 7.4 and Section 7.5 present the formative and summative evaluation of the SAIC agent, respectively. We present the results of the analysis in Section 7.6 and discuss the findings in Section 7.7. Finally, Section 7.8 provides a summary of the study.

7.2. An Overview of Tutoring System Evaluations

Evaluation is an important stage in the development process of tutoring systems. In our case, the effectiveness of our pedagogical agent must be measured to demonstrate its educational impact. We will present here an overview of the evaluation processes relevant to ITSs that we considered when deciding how the SAIC agent might be evaluated.

Many researchers, for example, Heller (1991), Shute and Regian (1993), Mark and Greer (1993), Iqbal et al, (1999) and Ainsworth (2003, 2005), have stressed the importance of conducting ITS evaluation. ITSs based on learning theories are often validated in empirical studies with prototypes, see Anderson et al. (1995), Luckin and du Boulay (1999), Akhras and Self (2000). The design of our agent has been informed by the principles of schema theory (see Section 2.3 in Chapter 2) and so the empirical evaluation will look for evidence that validates both the design and the design principles in the context of schema theory.

Some successful tutoring systems, for instance the LISP Tutor (Anderson & Reiser, 1985), PAT (Koedinger et al., 1997), ECOLAB (Luckin & du Boulay, 1999), and SQL Tutor (Mitrovic, 1998), have been evaluated by measuring *student performance*. Students are reported to have performed better after using the systems. Adopting this approach, the performance of children using the SAIC agent prototype will be measured to look for an improvement in the children's conceptual understanding resulting from the support provided by the agent. Our evaluation will differ from the evaluations of the systems listed above because we will measure performance by assessing children's schematic knowledge. Measurements will be made before and after interaction with the SAIC agent. Aist (2001) uses *experimentally-defined measures* to test students' vocabulary; this decision was validated by the National Reading Panel (2000). In the SAIC evaluation, we will adopt a similar approach. The measurement of the students' schematic knowledge will take place after the teachers' class assessment in accordance with the stated aims of our experimental study, see Section 7.3.

ITSs comparable to our system, for example PAT (Koedinger et al., 1997), ECOLAB (Luckin & du Boulay, 1999) and Reading Tutor (Aist, 2001), have been evaluated in *real classroom settings* to show the practicality and usefulness of the systems. Real life studies with children are challenging. It would be easier to carry out experiments in a computer laboratory, where the variables could more easily be controlled, but the evidence from the literature suggests that evaluation studies carried out in real settings give the most useful feedback on the impact of the ITSs. We will conduct the evaluation of the SAIC agent in real classroom settings and collect data to establish how children interact with the agent and to what extent this influences their schematic knowledge.

7.3. Aims of the Evaluation

The aim of the evaluation is to describe and quantify the impact of the SAIC agent in terms of its support of children's conceptual understanding. The questions we wish to answer are listed in Section 7.1. We also aim to validate the SAIC agent and its design principles.

We will conduct an empirical study to collect data using the prototype that implements the agent within a learning system (see Chapter 6). The data will be analysed to answer the evaluative questions and provide decisive feedback relating to the SAIC design. The evaluation of the SAIC pedagogical agent will take place in two developmental stages: formative and summative. The formative stage takes place while the system is still under development. The summative evaluation, for the fully developed system uses real students in real classroom settings.

The main objective of the study is to test whether *interactions with the SAIC agent incorporated in a learning system improves children's conceptual understanding.*

7.3.1. Validating the SAIC design principles

A validation of ITS design will show its reliability. Mertz (1997) argues that if an architecture is validated by human data then any system that follows the architecture inherits that validation. Following this argument, it is necessary to validate the SAIC design by providing evidence derived from an analysis of human data.

SAIC is a computational approach to the support of children's conceptual understanding. Baker (2000) suggests that a computational model should only be implemented to the extent that it can be validated. In line with this argument, Ravenscroft and Pilkington (2000), stress the importance of validating a system from both educational and computational perspectives.

Educational validation of SAIC necessarily requires an assessment of the ability of the agent to support children's conceptual understanding. The assumption is that if the learning gain of the students who interact with SAIC is improved then we can consider the design as educationally validated. On the other hand, **computational validation** looks at the effectiveness of its reasoning to generate dynamic and adaptive explanations.

By analysing the empirical data, we can evaluate and validate the SAIC design principles presented in Chapter 2.

7.4. Formative Evaluation of the SAIC Agent Prototype

The formative evaluation of SAIC is focused on the usability of the system, its revision and improvement.

7.4.1. Goals of the formative evaluation

During the formative development stage, the system may not be robust. The design and behaviour of the system must be revised and improved (Mark & Greer, 1993) before being used by children in real settings.

The questions addressed in the SAIC formative evaluation are:

- *The usability of the template-based interface*: Is the template-based approach to communication adequate for explaining new concepts? Can the students understand the explanations provided by the SAIC agent's interface? Can students follow the reasoning of the agent?
- *The selection of explanation strategies*: Is the mechanism for selecting explanation strategies appropriate?
- *The maintenance of the student cognitive model*: Does the mechanism used to maintain the student cognitive model allow the agent to explain new concepts? Does the cognitive model adequately represent the prior knowledge of the student?
- *The schema activation and modification*: Is the mechanism for activating and modifying relevant schema appropriate?

7.4.2. Experimental design of the formative evaluation

Participants

Two primary school teachers, four parents and six students used the system and commented on its features and faults. The teacher participants had several years of experience teaching children at the CHALCS school (see Section 7.5). The parent participants had one or more children studying at primary schools in Leeds and usually helped them to learn new lessons at home. The student participants were potential users of the system, being in the target age category (7 to 11 years old), and having hands-on experience of computers both at home and at school.

Formative evaluation method

The parent and teacher participants were given an explanation of the aims of the SAIC agent together with a brief introduction to the approach of the system to explaining new concepts and an outline of the functional specifications of the system. The six student participants had the role of the agent explained to them and were then asked to interact with the agent to find out how the system could be improved.

The experimenter recorded the parents' and teachers' comments and also used a video camera to capture the students' interactions with the agent. Each student spent about 30 minutes to read the lesson and interact with the SAIC agent. Throughout the sessions with the tutoring system, the participants were asked to express their opinions about the system's behaviour and this helped us to clarify some design aspects of the system.

7.4.3. Results of the formative evaluation

In general, the parent and teacher participants found the SAIC agent an appropriate tool for helping students to understand new concepts. The student participants learned to communicate with the agent and understood its role within a reasonable time, i.e. after a five-minute demonstration by the experimenter, which suggested the system had adequate learnability (Dix et al, 2003). Analysis of the videotaped student-agent interaction showed that the students were engaged with the agent for most of the time. A written post-test was held immediately after the interactions and indicated elements of improvement in the students' conceptual understanding[12].

A number of problems with the current prototype implementation were observed by the teacher and parent participants during the student-agent interaction and brought to our attention.

The problems, and steps taken to improve the system, were as follows:

- *Distraction by other computer programs.* The existence of other icons on the computer screen distracted the students. Moreover, the student participants could run other computer programs via the Windows operating system while interacting with the SAIC agent. Both the teacher and parent participants regarded the problem as a distraction to the students. To solve this problem, the tutoring system was resized to a full screen to ensure that only the SAIC system was visible during a learning session.
- *Readability*. Some students had difficulty reading texts *on screen* for a long period of time. A normal reading session in traditional classrooms lasts 20 to 30 minutes but the students found it uncomfortable to read the texts on screen for such a long time. This comment led to two design revisions.

---

[12] Note, that a thorough analysis of the improvement of conceptual understanding was not performed, as the studies were conducted at the formative stage and in laboratory conditions. The improvement of conceptual understanding was analysed during the summative evaluation, described in the next section.

Firstly, the texts of the lesson were shortened so that the students could finish reading the lesson and interacting with the SAIC agent in about 20 minutes and, secondly, we used bigger fonts and pictures.

- *Modifying some utterances*. The teacher participants considered some of the templates used by the SAIC to be cumbersome and suggested that these should be replaced with common computer messages. For example, the template 'Would you like to…" was changed to a 'Yes' / 'No' computer message. Although the new messages sounded less polite they were more familiar to the children who all had some experience with online communication. The teacher participants pointed out that some generated explanations were lengthy for a text-based interaction and should be shortened and rephrased. To address these comments, we asked English native speakers to validate and improve the speech act templates used by the SAIC agent.
- *Unanticipated user inputs.* There were occasions when the children pressed keys on the keyboard, for example `Enter`, `Delete` and `Alt`, instead of pressing mouse buttons. This highlighted a need for improved error prevention (Dix et al., 2003) and led to addition of programming code that forced the system to ignore error messages caused by the erroneous key presses.

The observations of the system while it was being used by the children, together with the suggestions provided by teacher and parent participants, led to an improved, more robust prototype ready for the summative evaluation stage. This stage is discussed in the following section.

7.5. Summative Evaluation of the SAIC Agent

Several improvements during the formative stage resulted in a robust and stable prototype ready for a summative evaluation of the educational impact of the pedagogical agent. An experimental study was conducted to collect and analyse evaluation data in order to measure the effectiveness of the SAIC agent.

We approached real students who attended literacy classes at the Chapeltown and Harehills Assisted Learning Computer School (CHALCS). CHALCS is a community-based organisation providing after school activities for children. Most of the children come from minority groups. At CHALCS, computers are widely used to assist learning, and communication activities are encouraged. In this respect, the CHALCS staff, who fully supported the study and integrated the agent into their classroom activities, considered the SAIC idea favourably.

In order to become familiar with the learning environment and to identify the best deployment settings, the author spent one year as a teaching volunteer at CHALCS. This enabled a deeper understanding of the teaching process and the users' needs and was vital for the successful deployment of the SAIC agent in the real settings. A reading session at CHALCS normally consisted of a teacher and a group of several children, see Appendix-B for the usual format of a CHALCS reading session. The teacher managed the reading session and guided the children how to understand new concepts in the lesson. The SAIC agent would play the role of the teacher: to explain the new concepts when asked by the children. A guided reading session at CHALCS consisted of the following steps:

- Selecting a book for the reading session.
- Reading the book.
- Asking the students to explain new concepts in the book.
- Guiding the students how to better understand the concepts.

These steps were considered in the summative evaluation of the agent: the role of guiding the students was played by the SAIC agent.

To fully assess the usefulness of the pedagogical agent in supporting children's conceptual understanding, we incorporated the pedagogical agent into an educational multimedia software system called '*Going to the Moon'* (presented in Chapter 6) and introduced it into a real learning experience at CHALCS. The integration enabled us to investigate the computational and educational benefits of the design approach.

The summative evaluation was conducted to address the evaluative questions (discussed in Section 7.1) and to provide evidence to validate the design principles (revisited in Section 7.3).

7.5.1. Summative evaluation method

Experimental design

An experimental study was performed with 32 students and 5 teachers from CHALCS. The students were divided into *three ability groups* (low, middle and high) based on their reading and writing skills, as assessed by the teachers. See Appendix-C for guidelines for the teachers who participated in the study. The main learning activities in the after-school classes at CHALCS are reading, writing and mathematics. There was a good fit between the normal reading activities at CHALCS, which usually consist of pre-reading, reading session and book review, and our intended study. Children from several literacy classes took part in the reading session.

We followed *control group design* (Ainsworth, 2003). The children were equally divided into two groups according to their reading and writing abilities as assessed by the teachers: 32 children, 16 were assigned into a control group and 16 into an experimental group. See Appendix-D for the distribution between the groups. The *control group* used '*Going to the Moon'* as a traditional multimedia system without the SAIC agent, while *the experimental group* used the system with the agent.

Participants

All participants were students of the CHALCS primary school and ranged in age from seven to eleven years old. All of the children regularly participated in the literacy classes at CHALCS, where they learned to read and write with the help of human tutors and computers. All of them had necessary computer skills, for example using mouse, typing, using a web browser and playing computer games. More than 50 students initially participated in the experimental study but because of absence, time and schedule constraints only 32 of them completed both the pre-test and post-test. The students participated in the SAIC study on a voluntary basis, which was agreed to in advance by the school administrator.

Procedure

The experiment can be divided into four main stages: pre-test, learning sessions with the SAIC agent, post-test and interviews. The procedure of the experiment is as follows:

- *Checking the experimental settings before the pre-test*. To know the best experimental settings for the agent, we had three meetings with the teachers a few weeks before the pre-test started. We asked them what to prepare before a reading session, how to conduct the session and how to evaluate the students (see Appendix B). During the meetings, we asked them to comment on the materials prepared for the experiments, e.g. the list of new concepts, guidelines for teachers, questions for student group interview, and questionnaires for interview (see appendices A, C, I, respectively). The teachers verified that the materials were suitable for the reading sessions. The experimenter also attended the reading sessions conducted by the teachers to observe how they interacted with their students. As a consequence, we had insights of how to employ the SAIC agent in the environment.

- *Conducting the pre-test.* To measure the students' schematic knowledge before the reading session, we conducted a pre-test with the students. The students' schematic knowledge was assessed using a *word association* technique (Jonassen, 1987; Heylingen, 2001; Preece, 1976; Steyvers & Tenenbaum, 2005). Appendix-E contains samples of students' answers to the pre-test. Zakaluk, Samuels and Taylor (1988) stress that prior knowledge is positively linked with reading comprehension and methods of measuring prior knowledge are important. They suggest word association as a useful technique for evaluating topic familiarity. Therefore, we adopted this technique to measure students' schematic knowledge.

- *Constructing the student models.* To inform the SAIC agent about the students' relevant prior knowledge, we used the *pre-test scripts*. To initialise the student cognitive models (by hand) we analysed all the explanations in pre-test scripts. We classified each explanation into *function, structure* and *process* (see Chapter 2, Section 2.3). The teachers were then asked to check the students' schematic knowledge and make necessary changes in order to refine and validate the student cognitive model that had been built by the experimenter. The refined models were used to initialise the prior knowledge components of the student models used by the SAIC agent. When the student models were ready, we installed the tutoring system 'Going to the Moon' into the computers of the school. We asked the teachers to interact with the agent for two sessions in order to confirm the adequacy and accuracy of the student models.

- *Engaging in the reading sessions.* To enable interaction with the agent to learn new concepts, we allocated a computer to each student. The students in the experimental group (with SAIC) were briefed that there was an agent that would help them if they needed more explanation, see Appendix-F for the briefing text. A 3-minute demo was given to show the students how to call and interact with the agent. The experimenter observed the classroom to ensure that there were no technical problems with the computers.

- *Conducting the post-test.* To test students' knowledge after interacting with the SAIC agent, the students were asked to answer a *post-test* that was in the same format as the pre-test and tested the same concepts. The test was conducted immediately after the reading session, see Appendix G for samples of post-test transcripts.

- *Interviewing both the students and teachers.* To get students' and teachers' opinions about learning and teaching new concepts with the help of the SAIC agent, we conducted an interview both with the students and the teachers one week after the reading sessions. The students' opinions about the SAIC agent were gathered during group interviews, which were conducted by the teachers, see Appendix I for the questions asked during the student interview. We recorded their answers on audiotapes and also observed their behaviours to record their reactions during the group interview. Eventually, we interviewed the teachers to find out what they felt about the likely impact of SAIC on their practices, see Appendix K for the questions during the teacher interview.

Materials for pre-test and post-test

The test materials drew on material from a graded astronomy book suitable for children aged between seven and eleven years old. The book by Potter (2000) is called '*Space Mission'* and is described in more detail in Chapter 6. It contains both concrete and abstract objects and ideas to be learned by the students.

In the pre-test, we selected eighteen concepts listed in the glossary section of the book and asked the students to explain these concepts using pencil and paper. In accordance with usual CHALCS practice, this exercise was carried as a pre-reading activity. In the post-test, the participants explained the same concepts presented to them in the pre-test.

Data

Both quantitative and qualitative data were collected from the experimental study. The quantitative data were in the form of pre-test transcripts, post-test transcripts, and log files that recorded the interaction between the child and the SAIC agent. The qualitative data were obtained from semi-structured group interviews with the students together with the interviews of the teachers carried out at the end of the study. A summary of the collected data is presented in Table 7.1.

Table 7.1 The description of the data collected in the experimental study

Pre-test scripts	Paper and pencil tests. Students' explanations of concepts are used to initialise the student cognitive models. The scripts were written in about 30 minutes by the 32 students who participated in both the pre-test and post-test.
Post-test scripts	Students' explanations of the concepts after using the system '*Going to the Moon*'. The scripts were written by the 32 students during the post-test.
Log files	The interactions of 16 students in the experimental group who were using SAIC. These interactions were automatically recorded by the system.
Students' interview	4 semi-structured group interviews conducted in the classrooms with the help of the teachers. These were recorded on audio tapes
Teachers' interview	4 individual interviews with the teachers who took part in the experimental study

Examples of the collected data are provided in appendices G, H, J and L, respectively.

7.5.2. Data analysis

We analysed the students' explanations of the domain concepts in the pre-test and post-test to identify and classify the improvement of domain concept explanations. There were 576 explanations to be analysed (18 domain concepts x 32 students), see Appendix-G for examples of student explanations.

The following steps were taken to analyse the explanations:

1- Deciding how to measure improvement of the explanations, i.e. whether to use national examination as a measure or to have an experiment-defined measure. Because of the specific aims of the evaluation, we employed an experiment-defined measure, i.e. improvement of a student's schematic knowledge caused by his interaction with the SAIC agent.

2- Identifying schematic knowledge from each student explanation (function, structure and process). We measured schematic knowledge improvement based on the principles of schema theory (see Chapter 2, Section 2.3.5), i.e. in terms of *relating new concepts to existing prior knowledge.*

3- Assessing explanation by comparing a pre-test explanation with its post-test explanation. The teachers were asked to specify *which* concepts were explained in a better way in the post-test and *why* the explanation was considered better.

4- Verifying the teachers' assessment by checking the consensus between the teachers regarding their assessment of the explanations. Initially, there was disagreement with the assessment of a few concepts but it was quickly resolved after the teachers justified why an explanation was considered improved or showed no improvement.

5- Deriving the rules or improvement criteria of the assessment. The experimenter conducted two discussions with each teacher to get their opinion why a post-test explanation was considered better than its pre-test explanation. There was a close correspondence between the criteria and the principles of schema theory discussed in Chapter 2, Section 2.4.1.

Applying the criteria used by the teachers, we have a means to measure students' schematic improvement. Based on the criteria, we classify the students' improved explanations into three categories. The criteria, their description and examples are discussed below.

Explanation improvement type 1: 'More specific'

An explanation was categorised as 'More specific' if the schematic knowledge in the post-test was different from the schematic knowledge in the pre-test and if the student was able to relate everyday concept to a domain concept.

For example, the *function* of *astronaut* in the pre-test was explained as:

`'Astronaut goes into the space.'`

In the post-test, the *function* was explained as:

`'Astronaut goes to the moon. He wants to walk on the moon'.`

Using frames, the explanations can be represented as [astronaut.function: goes_into_space] and [astronaut.function: goes_to_the_moon], respectively. In the lesson '*Going to the Moon'*, an astronaut was described as a person who goes to the moon, which is in outer space. The post-test explanation indicates that the student has a more focused schema, i.e. from a general everyday concept to a domain-specific one. Therefore, it is considered more specific.

Explanation improvement type 2: 'More elaborate'

An explanation was categorised as 'More elaborate' if the post-test schema included the pre-test schema and the student was able to relate a domain concept to its schematic knowledge (function, structure or process).

For example, the *function* of *astronaut* in the pre-test was explained as:

`'Astronaut wants to go to the moon.'`

In the post-test, the *function* was explained as:

`'Astronaut wants to go to the moon to explore the moon.'`

Using frames, the pre-test explanation can be represented as [astronaut.function:goes_to_the_moon] and post-test explanation as [astronaut.function: goes_to_the_moon] and [astronaut.explore: moon]. In this case, the student managed to explain the function of astronaut in a more elaborate manner, i.e. by providing more details about the schema. Therefore, the post-test explanation was considered to be more elaborate.

Explanation improvement type 3: 'More complete'

An explanation was categorised as more complete if the post-test schema included the pre-test schema and the student was able to relate a domain concept to other domain concepts/s.

For example, the f*unction* of *astronaut* in the pre-test was explained as: `'Astronaut wants to go to the moon.'`

In the post-test, the *function* was explained as:

`'Astronaut wants to go to the moon using a space shuttle.'`

Using frames, the pre-test explanation can be represented as [astronaut.function:goes_to_the_moon] and post-test explanation as [astronaut.function: goes_to_the_moon] and [astronaut.vehicle: space_shuttle]. In the lesson, space shuttle is a main domain concept considered necessary to the understanding of '*Going to the Moon*'. In this case, the student managed to relate the domain concept astronaut to space shuttle in his explanation. Hence, the student was able to make a complex explanation. Therefore, the post-test explanation was considered more complete.

7.6. Results of the Analysis

Using the improvement criteria described in Section 7.5.2 above, we can analyse an explanation by comparing a student's pre-test explanation with his post-test explanation. As a result of the analysis, we have identified that 74 concepts had been explained in an improved way in the post-test when compared with the answer given by the same student in the pre-test. Refer to Appendix-M for the results. This section presents the results of the analysis by answering the questions identified in Section 7.1.

7.6.1. The effectiveness of the SAIC agent

The main objective of analysing the effectiveness of the SAIC agent is to answer the research question:

> *1- Is educational multimedia software with the SAIC agent more effective at explaining new concepts than educational multimedia software without the agent?*

The analysis of the SAIC educational impact was conducted in two stages. In the first stage the control group was compared with the experimental group. In the second stage, the students were compared across ability groups.

Educational impact on the group: control and experimental

Figure 7.1 shows that there was only one child from the experimental group (with SAIC) who did not make any improvement as opposed to 4 children from the control group (without SAIC) who did not make any improved explanation. In addition, some children from the experimental group improved their knowledge of 5, 6 and 7 concepts and this was not observed within the control group. This suggests that SAIC is effective in terms of both *reducing* the number of students with no improvement, and *enabling* the students to have more improvements.

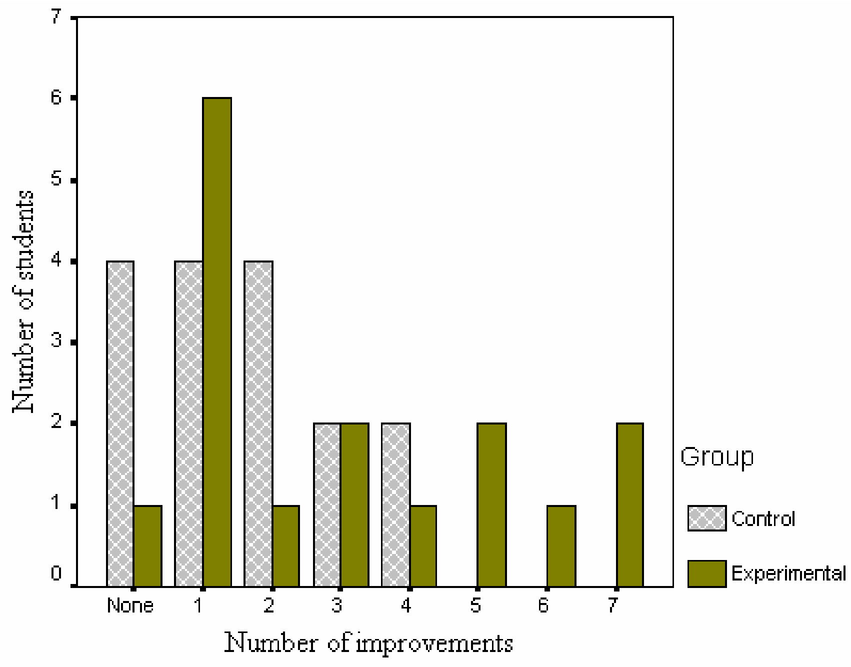


Figure 7.1. The number of improvements made by the students in the control and experimental groups. Each improvement indicates better schematic knowledge of one domain concept

A Mann-Whitney test of the improvement numbers of the two groups, however, does not indicate a statistically significant difference ($U = 87.000$, $N_1 = 16$, $N_2 = 16$, $p = 0.128$). However, both groups (control and experimental) contained children of low, middle and high ability and, since reading and writing ability might have a bearing on the result, we need to analyse the learning gain across the three ability groups.

Educational impact on the ability groups: low, middle and high

To further analyse conceptual changes due to the SAIC agent, we classified the students of the experimental group into three categories according to their reading and writing skills (low, middle and high). The results are presented in Table 7.2. A comparison of learning gain by the different abilities of both the control and experimental groups is presented in Table 7.4 in Section 7.6.3.

Table 7.2 The number of improvements for each ability of the experimental group. The numbers indicate how many students have demonstrated improvement for 0, 1, 3, 4, 5, 6 and 7 concepts, respectively

Ability group	**Number of improvements**								**Total**
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
Low	1	4	0	0	0	0	0	0	5
Middle	0	2	1	1	0	1	0	1	6
High	0	0	1	1	1	1	1	1	5
Total	1	6	2	2	1	2	1	2	16

A one-way ANOVA test showed there was a statistically significant effect of ability group on schematic improvement in the children who use SAIC ($F_{(2,13)}$ = 7.276, p = .008). This shows that the SAIC agent is differently effective for the different ability groups at supporting their conceptual understanding; 0 and 1 improvement for the low group, 2 to 7 improvements for the middle group, and 3 to 7 improvements for the high group. The result suggests that the agent reduces the number of 0 improvement for the low group students and enables the student in the middle and high groups to have 5, 6 and 7 improvements. This indicates that the agent is more beneficial to the students of the middle and high ability groups, who were characterised by better reading and writing skills than the low group. Therefore, the result is sensible because the interaction with the agent required more reading and the explanation required more writing from the students.

The small learning gain by the students in the experimental group was to be expected because they interacted with the agent only for about 20 to 30 minutes in one reading session. As highlighted by Rumelhart (1980), learning new concepts may take days, weeks or even longer.

7.6.2. The interactions with the SAIC agent and the explanation of the domain concepts

This analysis is required to answer the question:

2- Do the students who use the SAIC agent to learn new concepts improve their conceptual understanding?

We have analysed the improved explanations to examine which concepts were explained by the students in the experimental group, the frequency of the concepts in the explanations, and whether the students learned these concepts by triggering the SAIC agent. The triggers were identified from the log files. A trigger represents an interaction with the agent during which the students followed the guidance provided by the agent.

A Pearson correlation test showed there was statistically significant positive correlation between the number of triggers and the number of improved explanations made by the experimental group (r = 0.831, p = .000). This shows that the more the experimental group triggered the agent to learn a concept the more improved explanations the group made on the concept. The result suggests that the adaptive explanations generated by the agent and the explanation strategies it employed helped the experimental group to understand new concepts better.

Table 7.3 shows that the students triggered the agent 88 times to ask about the domain concepts and made 48 improvements. Domain concepts with a high frequency of improved explanations (`astronaut`, `computer`, `oxygen`, `earth` and `moon`) also have high number of triggers (14, 6, 8, 8, and 11 respectively). On the other hand, domain concepts with a low frequency of improved explanation, e.g. `interviewing`, `satellite`, `space`, `swimming` and `taking off`, tend to have a low number of triggers.

The close connection between the agent triggers and the explanation improvements suggests that the agent is effective at supporting the students' conceptual understanding and promoting schema-based cognitive tasks. Note that the relationship between the triggers and improved explanations was considered on a *group basis*, i.e. by the whole experimental group instead of on an individual basis. This is because an interaction with one concept can help a student explain other concepts and this supports the idea of the SAIC agent acting to promote schema-based cognitive tasks in the students.

Table 7.3 The domain concepts, number of triggers and number of improvements

Domain concepts	No. of triggers	No. of improvements
1- Astronaut	14	6
2- Computer	6	7
3- Oxygen	8	5
4- Earth	8	5
5- Moon	11	4
6- Telescope	6	3
7- Training	2	3
8- Space centre	5	3
9- Gravity	6	3
10- Floating	5	2
11- Star	3	2
12- Planet	2	2
13- Space shuttle	5	2
14- Interviewing	0	1
15- Satellite	3	1
16- Space	0	1
17- Swimming	3	0
18- Taking off	1	0
Total	88	48

The analysis of individual student models also shows that the update by the agent relates to the post-test explanation: 37 out of 48 (77%) concepts in the improved explanations were explained by the SAIC agent, see Appendix M for the analysis of the explanations. This implies that students' knowledge of a concept was updated (as indicated by their student models) because of their interaction with the SAIC agent and the change in their knowledge was reflected in their post-test explanation. This shows a relatively high accuracy of the agent's representation of the students' dynamic schematic knowledge. The correspondence between the update of the student model and the improved explanations can be attributed to the effectiveness of the explanation generated by the agent. However, in several cases (23%) students did explain new concepts in an improved ways without asking for explanation from the agent. The improvement might has been caused by a *transfer of learning*, i.e. the students learned the explanation skill and were able to apply it to the new concepts. Because of the high accuracy of the agent's estimate of the students' knowledge, as assessed by the teachers (see Section 7.5.2), we argue that the SAIC agent's model of the students' schematic knowledge was accurate and sufficient.

7.6.3. Types of improvement made by the participants

This analysis is conducted to answer the question:

3- *What type of improvement can be indicated?*

Three types of improvement were identified in Section 7.5.2. These were 'More specific', 'More elaborate' and 'More complete'. The analysis of the students' responses in terms of the types of improvement took place in two stages. We first compared the types of improvement observed in the control and experimental groups and then looked at the types of improvements identified across the ability groups.

Types of improvement made by the control and experimental groups

Table 7.4 shows the types of improvement made by the control and experimental groups. The students in the experimental group made more improved explanations in terms improvement types 'More specific' (11) and 'More complete' (19), whereas the students in the control group made only 2 improvements for the type 'More specific' and only 4 improvements for the type 'More complete'. These differences would suggest that students who interacted with the SAIC agent could explain domain concepts in a more specific and complete way when compared to the students who do not interact with the pedagogical agent. Therefore, the improvement of types 'More specific' and 'More complete' can be attributed to the students' interaction with the agent.

Following the teachers' improvement criteria ('More specific', 'More elaborate' and 'More specific') as discussed in Section 7.5.2, a 'More specific' explanation indicates that the students manages to relate their existing prior knowledge or already acquired schemas, which are concrete and general, to the domain concepts presented in the lesson and, consequently, explains domain concepts in a way more specific to the lesson. This can be related to the *schema activation role* of the agent and its dialogue strategies, such as 'Informing the isa category and its instance', see Table 3.4 in Chapter 3 for the SAIC tutoring strategies.

A 'More complete' explanation indicates that the students managed to relate a domain concepts to other domain concept/s presented in the lesson. The significant improvement of type 'More complete' can be related to the way schematic knowledge is represented in the domain and student models, i.e. in terms of *function, structure* and *process*, as discussed in Chapter 2, in which domain concepts can be linked and explained to students accordingly. This can be related to the *schema modification role* of the agent and its dialogue strategies, such as 'Comparing the new concept with its *isa* instances' and 'Adding a new property to the new concept which has just been created'.

Table 7.4 The number of improvement type made by the control and experimental groups

Group	Types of improvement			
	More specific	More elaborate	More complete	Total
Control	2	20	4	26
Experimental	11	18	19	48

The table also shows that students in the control group have made more improved explanation of type 'More elaborate' (20) than the experimental group (18). This may indicate that the students in the control group can explain domain concepts in a more elaborate manner, i.e. providing more details to a schematic knowledge, and may be due to the fact that the students in the control group did not interact with the SAIC agent and therefore had more time to read the lesson. In addition, some students in the control group had more time to re-read the lesson compared to the students in the experimental group, who were engaged in the interaction with the agent. Therefore, the improvement of type 'More elaborated' might be attributed to a difference in the time available for the students to read the text of the lesson.

A 'More elaborate' explanation (as discussed in Section 7.5.2) indicates that the students managed to explain a concept in terms of its schematic knowledge (function, structure and process) as provided by the lesson without explicit link to their prior knowledge or to other domain concepts. Without the help from the agent, it means the students explained domain concepts based on their reading and had to discover by themselves the link between the domain concepts and their prior knowledge or other domain concepts. Therefore, a high number (as shown in Table 7.4) of type 'More elaborate' improvements can be related to the reading of the lesson, in which they had more time to read the details of the concepts.

The results suggest that the students in the experimental group have a deeper schematic knowledge ('More specific' and 'More complete') than the students in the control group, and that the resultant knowledge was caused by the guidance of the agent.

A Kruskal-Wallis test of the improvement numbers of the two groups indicates a statistically significant difference (H = 9.325, N_1 = 16, N_2 = 16, p = .002) for the improvement type 'More complete', which was defined as the ability to expand the understanding of a concept by relating it to other domain concepts. This shows that the SAIC agent is effective at helping the experimental group students to make more complete explanations.

Types of improvement made by the ability groups

A further analysis of the improvement types can show what educational impact the agent has on each ability group. Because of the small number (16) of the students in the experimental group that was further divided into 3 ability groups, i.e. low, middle, and high, we have identified and classified the improvements made by each ability group to compare the learning gain of the control and experimental groups according to their abilities.

Low ability:

Table 7.5 shows that the students in the low ability category of the experimental group exhibited more improvement of type 'More complete'. This finding is consistent with the result of the Kruskal-Wallis test (H = 9.325, N_1 = 16, N_2 = 16, p = .002) conducted to compare the control and experimental groups.

This shows that the low ability students can, as a result of interaction with the SAIC agent, explain new concepts in a more complete way than the low ability students from the control group. That is, the students explain concepts by relating them to other domain concepts. This would suggest that the explanation strategies employed by the agent are effective in explaining new concepts to low ability students in terms of the 'More complete' explanation type. However, there is no difference for the 'More specific' and 'More elaborate' types.

Table 7.5 The number of improvement type made by students in each ability group

Ability group	**More specific**		**More elaborate**		**More complete**	
	Control	**Experim.**	**Control**	**Experim.**	**Control**	**Experim.**
Low	0	0	1	1	1	3
Middle	0	5	8	8	3	6
High	2	6	11	9	0	10

Middle ability:

The middle ability students of the experimental group made 5 improvements of the type 'More specific' and 6 of the type 'More complete', whereas the middle ability students of the control group made 0 and 3 improvements, respectively. The number of improvements for the type 'More complete' made by the middle ability group corresponds with the result of the Kruskal-Wallis test conducted to compare the types of improvement made by the control and experimental groups, as discussed earlier.

This indicates that the SAIC agent helped to the middle ability students of the experimental group to make an explanation improvement for the types 'More specific' and 'More complete'. This would suggest that the SAIC agent helps middle ability children to understand new concepts in a more specific and complete manner. However, the middle ability students of both the control and experimental groups perform equally well in terms of the explanation type 'More elaborate'.

High ability:

The high ability students in the experimental group made 6 improvements of type 'More specific' and 10 of the type 'More complete', whereas the high ability students of the control group made 2 and 0 improvement, respectively. The number of improvements for the type 'More complete' made by the high ability group corresponds with the result of the Kruskal-Wallis test conducted to compare the control and experimental groups, which shows that students who interacted with the SAIC agent made significantly more improvement for the type 'More complete' than the students without the agent.

This indicates that the SAIC agent helped high ability students to make improvements of the explanation type for the types 'More complete' rather than 'More specific' and 'More elaborate'. However, the high ability students of the control group exhibited more improvements in terms of 'More elaborate' (11) than the high ability students of the experimental group (9). This may be because that the high ability end of the control group have used the longer reading time available to them to their advantage and can, as a result, make more elaborate explanations.

The results suggest that the support provided by the agent can help the students in the three ability categories (low, middle and high) of the experimental group to write better explanations in terms of the explanation types 'More specific' and 'More complete'. Though the Kruskal-Wallis test does not show statistically significant difference between the control and experimental groups in terms of 'More specific', Table 7.5 shows that there is a *close association* between the student ability and the improvement type ‘More specific’: the higher the writing and reading abilities of the students in the experimental group the more improvement the students will make in terms of the type ‘More specific’.

These findings can be attributed to the agent’s dialogue and dialogue strategies. On the other hand, the students who learn new concepts without the agent’s support can explain the concepts in a more elaborate manner and this can be attributed to more time available for the students to read the texts. Therefore, we can argue that the SAIC agent is useful to students in terms of making their explanations *more specific and complete*, especially for middle and high ability children.

7.6.4. The usefulness of the SAIC agent: teacher and student interview

The aim of this sub-section is to present the opinions of the teachers and students about the usefulness of the SAIC agent in supporting the student to learn new concepts. We have interviewed the teachers and students to collect qualitative data and analysed them in order to examine the teachers’ and students’ opinions. See appendices I, J, K and L for the questions asked during the interviews and samples of the interview transcripts.

The SAIC agent was developed to simulate how human teachers guide students’ reasoning as the students work towards an understanding of new concepts. The teachers’ and students’ feedback about the agent can be used to inform the SAIC design approach. In particular, the teachers’ and students’ acceptance of the SAIC agent might indicate the feasibility of future use of SAIC in literacy classes.

In the next section we briefly describe the interviews, and with the help of extracts from the transcripts, we discuss the opinions of the teachers and students.

Teachers’ opinions

The qualitative data obtained from the interviews were analysed to answer this question:

4- *What do the teachers feel about the use of the SAIC agent to help explain new concepts and to support traditional classroom reading sessions?*

The main purpose of the teacher interview was to examine the teachers’ opinions about the usefulness of the help provided by the SAIC agent. Four teachers were interviewed individually after the post-test.

Do teachers consider it feasible to integrate with SAIC in their teaching practice?

The teacher participants, who were all fairly familiar with educational software, said that the SAIC agent was helpful. Example extracts are shown below. They viewed the SAIC agent, which they referred to as 'the software', as a helpful learning aid that they could use in their reading classes. The teachers did not mention any difficulty with integrating the SAIC agent into their usual classroom practice. However, they regarded the SAIC agent as a 'new type of software' or just 'the software' and none referred to it as a computer tutor or a pedagogical agent. This shows that one of SAIC's strengths is its integration within a multimedia system. The quality and the relevance of the material in the multimedia system is as important as the interaction with SAIC.

> Teacher J said: `"We need to have that kind of software in our computers. It will help both the teachers and students."`

> Teacher S said: `"It is possible to integrate the software. We have a variety of software in our computers, in addition to using books. The software is a helpful learning material."`

All teachers said they would use the SAIC agent integrated into an appropriate multimedia system if it was provided for their class. All of them voiced their concerns about the time and effort to put the information about their students (the student models) and the information about the domain (the domain knowledge) into the computers. During each semester of reading sessions the students get through a large quantity of learning materials and, as end users of computer software, the teachers have insufficient resources to convert the learning materials into educational multimedia systems. Nor do they have the expertise to then integrate SAIC into those systems. Despite these potential problems, SAIC achieved a high degree of acceptance by this group of classroom teachers and this can be ascribed to the perceived benefits of SAIC's guiding behaviour resulting from its dialogue with the students.

How would teachers use SAIC?

The teachers pointed out that the SAIC agent was a useful software tool that a student could use to learn new concepts. They envisaged that the SAIC agent could be used individually by a student or as part of class activities. They considered potential applications of the agent in reading classes and suggested that the agent would be useful for both teachers and students.

> Teacher S said: `"as interactive whiteboard, class displays and discussions with students."`

> Teacher B said: `“as a problem solver when my students don’t know what a new word means.”`
>
> Teacher A said: `“It is a software tool. My student can activate it to learn when I’m busy with other students.”`

Teachers’ opinion about the study

The teachers felt that SAIC offered a solution to some problems they regularly faced in CHALCS literacy classes: they found themselves repeatedly explaining the same concepts to different students and, every time a student asked a question, the teacher had to tailor the explanation to each individual student - as articulated by Teacher J in the extract below. It is impractical to achieve a ratio of one teacher per student in organisation like CHALCS but the use of the SAIC agent in reading classes would mean every student had the opportunity to ask whenever he needed an explanation of a new word in a lesson. The teachers also liked the way they have the opportunity to confirm the student cognitive models within SAIC because this makes the student’s knowledge explicit to the teachers.

> Teacher J said: `“You’ve addressed a real problem faced by the teachers and students in reading classes.”`
>
> Teacher B said: `“It’s a great effort to study what my students understand and what they don’t. We’ll have a better idea how to teach them.”`

Negative aspects of the study

The teachers were also questioned about negative aspects of the study and asked to suggest improvements for SAIC. They thought that the SAIC agent would be better if it used spoken language when interacting with students pointing out that text-based interaction might increase the students’ cognitive load.

> Teacher A said: `“My students will have to read all the explanations in addition to the new lesson.”`

This points at difficulties which children with low ability skills in reading and writing can face when using the agent. It sheds some light about the fairly low improvement from the low ability group. Speech technology would be used to make the agent more effective for these children. The development of children’s speech recognition technology was beyond the scope of this thesis. Other researchers are conducting work along these lines, for example (Beck, Mostow & Bey, 2004; Chi et el., 1994).

Students' opinions

The analysis was conducted to answer this question:

5- *What do the students feel about the use of the SAIC agent to help explain new concepts and provide support to traditional classroom reading sessions?*

The main purpose of the student group interview was to examine the students' opinions about the usefulness of the help by the SAIC agent. The student group interview was conducted by their teachers and took place soon after the post-test. In the interview, the class teachers asked the questions and the students had to raise their hands to answer to prevent simultaneous answers from several students. The experimenter observed the semi-structured group interviews to count the number of students who answered a question via spoken answers or hand-raises, and also recorded the answers on tapes. Hence, the students' opinions were based on the questions asked during the group interview (see Appendix-I).

Did the children enjoy interacting with SAIC and what did they enjoy most?

Almost all students said they enjoyed using the SAIC agent.

Table 7.6 Why the students enjoyed interacting with the SAIC agent

No. of students	Reasons
8	They could interact with the agent
3	The agent helped them understand the lesson
2	The educational software included pictures and colours
1	The children could ask questions

The reasons given by the students to explain why they enjoyed using the SAIC agent indicate that the students found the agent useful. The agent motivated the students to learn by helping them to find answers and explain new concepts.

Table 7.7 What the students enjoyed most when interacting with the SAIC agent

No. of students	Factors
6	The answers given by the agent
3	The whole lesson with the computer
1	That the lesson included pictures
1	That they could ask questions

All students thought that if they had a computer helper like SAIC they would be able to understand new words when they read text. The teachers who conducted the interview explained to the class that teachers did not normally give direct answers to any student but tried to help students find the answers themselves and that the SAIC agent behaved similarly but almost all students said they wanted answers or definitions of new concepts from the SAIC agent. This request for direct answers points at the need to help children understand how important reasoning processes are to learning. Although teachers acknowledged the benefits of schema-based interactions, the children still considered the agent as a source for providing answers rather than as a guide to reasoning. This points at the need to incorporate some meta-cognitive dialogue games, as examined by (Mendez, du Boulay & Luckin, 2005).

Negative experiences with SAIC

There were seven students who expressed some negative feelings they had about SAIC. Their answers are summarised in Table 7.8.

Table 7.8 Reasons why some students were unhappy when interacting with the SAIC agent

No. of students	Reason
3	There was too much reading
3	There was no sound
1	The agent was slow

The first comment may have resulted from the way the agent uses texts to guide the reasoning of the students instead of giving a direct answer. The second and third comments point towards future improvements of the agent.

Most of the students generally found the agent helpful and interactive and would recommend it to a friend. Their opinions provide evidence that a pedagogical agent design based on the SAIC approach has the potential to be used by students in literacy classes. The comments also pointed at the need to make the interaction with the agent more robust and natural by adding appropriate speech techniques.

7.7. Discussion: Validating the SAIC Design Principles

Each of the theoretical claims and associated design principles revisited in Section 7.3 is discussed in turn. The evidence in this chapter consists of the quantitative analysis of changes in the children's schematic knowledge and qualitative data of teachers' and students' opinions about the usefulness of the SAIC agent. The SAIC design principles are supported by the results of the analysis presented in Section 7.6.

7.7.1. Design Principle 1: During the interaction with the child, the agent has to activate prior knowledge and use it to introduce new information

The theoretical basis of Design Principle 1 is that the activation of relevant prior knowledge is required if the child is to interpret new concepts in terms of existing schemas. As a consequence of this design principle, the SAIC agent has a student model component that contains prior knowledge represented in the form of schemas. To generate adaptive explanations, the SAIC agent employs activation strategies that simulate those of human tutors.

An analysis of the frequency of improved explanations in the post-tests (Figure 7.1) shows that the students whose relevant prior knowledge had been activated by the agent gave fewer explanations that showed no improvement when compared to the pre-test than the students whose relevant prior knowledge had not been activated. In addition, the students with activated prior knowledge through their interactions with the agent achieved a greater number of improvements than the control group. These results indicate that schema activation performed by the agent plays an important role in helping children to make improved explanations of new concepts.

An investigation of the types of improvement made by the students who interacted with the SAIC agent is also interesting. It reveals that it is necessary to activate student's prior knowledge. Table 7.3 shows that interaction with the agent, which supports schema activation, helped the students make more specific and more complete explanations. This suggests that the agent has managed to activate *appropriate* schemas to be employed in understanding new concepts. The nature of the explanation improvement, i.e. 'More specific' and 'More complete', can be attributed to the SAIC's ability to infer and select the most suitable prior knowledge based on the domain and student knowledge and on the answers provided by the students during the interaction.

The opinions of the teachers and students also provide evidence to validate Design Principle 1. The teachers, all of whom had several years of teaching experience, agreed that the way the agent tailored its explanations to students' prior knowledge when explaining new concepts was consistent with their established teaching practice. Therefore, the agent's schema activation was not considered to be a new teaching technique but instead seen as a simulation of a teacher's normal classroom activity in the way it related new concepts to existing knowledge. The students, however, were not aware of the schema activation process because they viewed the agent as a tool providing answers rather than guidance.

In summary, the analysis above shows that the SAIC agent does activate, use and build upon students' prior knowledge and thus Design Principle 1 is validated. It also points at the need to incorporate schema-based strategies with strategies that improve motivation and promote meta-cognition, which refer to future improvement of SAIC.

7.7.2. Design Principle 2: The agent should promote the cognitive tasks proposed by schema theory

The theoretical basis of the second design principle is that modifying the activated schema performs the interpretation of a new concept. In order to implement Design Principle 2, the SAIC agent performs schema modification (accretion, tuning, restructuring and creation) using dialogue strategies.

An analysis of the frequency of improved explanations in the post-tests, shown in Figure 7.1 and confirmed by Table 7.3, shows that the students who interacted with the agent offered fewer explanations that showed no improvement over their pre-test explanations than the students who had not interacted with the agent. In addition, the students who had interacted with the agent achieved a greater number of improvements than the control group. The greater number of improvements can be attributed to the schema-based cognitive tasks promoted by the agent. The agent decides how the child might best understand new concepts and guides the child's reasoning accordingly. Hence, we argue that the improvement of the student's explanations was caused by the guidance of the agent.

The types of improvement made by the students who interacted with the SAIC agent (shown in Table 7.3) show the effect of schema modification supported by the agent. Explanation improvement of the type 'More specific' can be credited to the modification strategies of the agent (see Chapter 3, Section 3.4.3). The agent explains new concepts 'property' and 'property value' that are considered necessary to the understanding of the lesson content as defined by the agent's domain model. Likewise, 'More complete' explanations, i.e. the students' ability to create a more complex explanation by explaining a new concept in terms of another domain concept can be attributed to the agent's strategies when performing schema modification. Tuning and restructuring, for example, involve comparing and contrasting a new concept with other schema. This suggests that the students adopted SAIC's approach to the explanation of new concepts.

The log files of the student-SAIC agent interactions show that the students interacted 88 times with the SAIC agent, see Table 7.9. The agent *explicitly* initiated activation episodes (88 in total) during each interaction to activate relevant prior knowledge. Dialogue meta-episodes (introductory, diagnose, summarizing and confirmation) were also initiated in each interaction. The SAIC agent initiated 19 accretion, 42 tuning, 16 restructuring and 11 creation dialogue episodes. The schema-based episodes were initiated based on student's prior knowledge and answers to SAIC agent's questions during the interaction.

Table 7.9 Dialogue episodes used by the SAIC agent and their frequency in the interaction

Dialogue episodes	Frequency
Activation	88
Accretion	19
Tuning	42
Restructuring	16
Creation	11
Introductory	88
Diagnose	88
Summarizing and confirmation	88

This table shows that the SAIC agent conducted a combination of dialogue episodes with the students during the interactions. The following table shows the number of invocations for each dialogue strategy. It is shown in Table 7.10 that the SAIC agent used a combination of dialogue strategies for each dialogue episode. Thus, the improvement of explanations for the students who interacted with the SAIC agent can be related to the dialogue strategies used by the agent. This improvement also suggests the effectiveness of the tutoring strategies.

Table 7.10 A summary of tutoring strategies used by the SAIC agent to perform the dialogue episodes and the frequency of invocation for each strategy

Dialogue episode	**Tutoring strategies**	**Frequency**
Activation	Asking property value of a schema.	26
	Asking name of the schema with a property value.	41
	Showing picture of a schema.	8
	Informing the *isa* category and its instance.	13
Modification	Inform a new property of the domain concept.	30
	Showing a picture of the domain concept.	4
	Informing a wrong property value of the domain concept.	7
	Informing and asking about the *isa* and instance of the domain concept.	12
	Comparing and contrasting the domain concept with a schema	35

The interviews with the teachers showed that the teachers recognised the role of the agent in guiding reasoning so that the student reaches an answer and the teachers were also able to visualise how they might integrate the agent into their practice. Therefore, the role of the agent as schema modifier is acceptable to teachers. Although the agent was capable of performing schema modification adaptively, the role of the agent was never identified by the teachers as being that of a 'pedagogical agent' or a 'computer tutor'. Nor was any comparison made between SAIC and any of the other roles a human tutor might perform, e.g. social (Johnson et al., 2004), affective (Heffernan & Croteau, 2004; Mendez, du Boulay & Luckin, 2005) and parental (Toppings, 2001). However, this should not invalidate the design principle because the agent was specifically designed to support schema-based cognitive tasks, i.e. schema activation and modification. The SAIC framework can be combined with other frameworks that address the additional aspects, as pointed out above.

In summary, the evidence of the learning gained by students above shows that the SAIC agent does promote the cognitive tasks proposed by schema theory and thus Design Principle 2 is validated.

7.7.3. Design Principle 3: The agent should provide concrete examples and should avoid the use of abstract concepts

The theoretical basis of Design Principle 3 is that children's reasoning is based on concrete objects and ideas. The student model of the SAIC agent consists of a representation of the concrete knowledge the student had before using the agent together with any domain concepts the student already has acquired before starting to use SAIC. The agent provides concrete examples to explain abstract concepts. For example, the *isa, function, structure* and *process* of a space shuttle are explained using the concrete schemas: `bicycle`, `car`, `bus` and `aeroplane`. Thus, the students can see how a new concept is related to what they have learned.

The decisions taken by the agent as it supported a child's reasoning resulted in the child being given guidance based on schema activation and modification. The student found this useful. We have shown that SAIC improves explanations by promoting a concrete reasoning style. This establishes that Design Principle 3 is valid and should be taken into account when designing tutoring systems for young students.

7.8. Summary: Analyses, Findings and Limitations of the Study

This section provides a summary of the analysis we carried out, our findings and the limitations of the study.

7.8.1. Analyses

We did several analyses to identify general and specific improvements in the students' explanation that indicated conceptual change caused by the interaction with the SAIC agent. The types of analysis and their descriptions are given below:

A- Improvement of explanations by the control and experimental groups

The analysis used experimentally-defined criteria based on the teachers' previous teaching. By considering the goals of the agent and the aims of the evaluation, the teachers were able to assess the improvement made by the students and explain the rules of their assessment. As a result, the schematic improvement of the students' knowledge, as shown in their explanations, could be assessed in terms of the explanation improvement types 'More specific', 'More elaborate' and 'More complete'. If we assume that the assessment expertise of an experienced human tutor is a golden standard to follow, we can be reasonably sure that the teachers' judgements of the students' explanations were appropriate and accurate. Even so, this analysis relied on the subjective opinion of the teachers in the study so to fully validate the SAIC approach, further analysis was undertaken.

B- Group improvement according to reading and writing ability

This type of analysis was conducted to examine the effectiveness of the agent for students of different abilities: low, middle and high. The basis of the analysis is that the same agent had been available to all the students, regardless of ability. Therefore, all the students should have had relevant schemas activated and been guided as to how, according to their individual student models, to perform schema activation and modification. All students received the same type of instruction and hence, we can examine what type of learning is gained by the students across the three ability groupings.

C- Interaction versus improvement

This analysis was designed to check whether interactions with the SAIC agent could help students learn new concepts. We wanted to find out if there was a causal relationship between the help provided by the agent and the improvement in the students' conceptual understanding. We assumed that once a student had acquired the schema-based cognitive skill and knew how to activate schemas and how to perform schema modification, that he would be able to apply this skill and understand any new concept. If this is the case, an interaction that leads to a student learning one new concept will help the student understand other concepts in the lesson. Therefore, improvement is considered due to the application of a skill promoted by the agent.

D- Type of improvement

Following the results of the analyses discussed above, we analysed what type of improvement the control and experimental groups made. In addition, we looked at the ability groups in terms of the experimentally-defined criteria 'More specific', 'More elaborate' and 'More complete'. The objective of this analysis was to examine the type of schematic knowledge caused by the interaction with the agent.

The difference in schematic knowledge gained by the experimental group can be attributed to the agent. The schematic knowledge gained by each ability level within the group shows the educational impact of the agent on students of different reading and writing abilities.

E- Teacher interview

The objective of this qualitative analysis was to establish the teacher participants' view of the usefulness of the agent. Their opinions gave insights into future possibilities for the adoption and integration of the agent into existing teaching practice.

F- Student interview

The agent was intended to be used by students in reading classes and therefore, qualitative data collected in real classroom reading settings was invaluable in showing us how the agent helps the students learn new concepts.

7.8.2. Findings

The effectiveness of the SAIC design approach has been demonstrated by the prototype. The evaluation confirms the instructional effectiveness of SAIC and this validates its design. Our findings can be summarised as follows:

Firstly, we have found that after students have spent just one session interacting with the SAIC agent, more of their explanations show improvement between the pre-test and the post-test. In addition, the students achieve a greater number of improved explanations with the help of the agent than without the help of the agent. Luckin and du Boulay (1999) found that students who interact with a tutor (human or computer) make better explanations because of the help provided by the tutor, and we agree that this might be attributed to a Zone of Proximal Development (Vygotsky, 1978) in which a student's learning is supported by a more able learning partner. On this basis, we argue that the SAIC agent has the features necessary to improve children's conceptual understanding. This capability highlights the necessity to provide this agent to students in reading classes.

Secondly, we have found that the SAIC agent does help the students understand new concepts in more specific and complete ways. The greatest effects are seen in the middle and high ability groups. This points to their deeper understanding of the domain concepts, and can be attributed to the reasoning and decision-making performed by the agent while performing schema-based cognitive tasks. It is assumed that a deeper understanding requires more thinking from the students' side. This indicates that the SAIC agent has managed to promote schema-based cognitive tasks in the students in a relatively short time. Though, in general, the students in the experimental group did not make big conceptual changes compared to the students in the control group, they had a deeper understanding because of the support provided by the agent. This finding is in agreement with the Rumelhart's (1980) emphasis that learning may take days, weeks, months or even longer. To address this issue we employed experimentally-defined measure, discussed in Section 7.2. Hence, we can argue that SAIC agent is effective in supporting children's conceptual understanding.

Thirdly, we found that both the teachers and students accept the SAIC agent. This indicates that the agent's simulation of supporting schema activation and modification is in agreement with the teachers' existing teaching practice, i.e. in reading sessions. The teachers even envisaged how they might integrate the agent into reading sessions. The students also accept the agent. Though the students were interested in 'what' answer is given by the agent, the agent can get them to see 'how' to get the answer by themselves though following its guidance.

In short, the findings illustrate the ability and potential of the approach in supporting children's conceptual understanding, which is the main goal of the agent.

7.8.3. Limitations of the study

The evaluation of the SAIC agent presented above is characterized by certain limitations: a small number of students involved in the study, a single interaction with the agent and a text-based communication medium. However, these limitations can be overcome by a deep analysis of the collected quantitative and qualitative data.

The first limitation of the study is the small number of students involved. At the initial stage of the study, more than 50 students expressed an interest in participating but some did not come either to the pre-test or post-test. As a result, we finally had only 32 students who had participated in both tests and whose explanations before and after the reading sessions could therefore be compared. The issue of non-attendance was beyond our control and is considered to be a normal problem by teachers at CHALCS.

The second limitation is that the study is based on the analysis of a single interaction session with the SAIC agent that lasted for just 20 to 30 minutes. This length of time is considered normal for a reading session with primary schoolchildren. The post-test was conducted one-week after the pre-test. In contrast, the organization of the tests took up a great deal of time. It took two months to obtain the parents' permission, ensure suitable security measures were in place for the children and rearrange the teachers' schedules.

The third limitation is that the agent used a text-based natural language that was reduced to the minimal subset required to perform the schema-based cognitive tasks. In a normal classroom, human teachers use rich spoken language and body language and these help the students to learn. However, because of technological constraints in human speech recognition research, the text-based medium of communication is considered sufficient for the study.

Our intention was to measure the effectiveness of a SAIC agent designed based on established schema theory and a WoZ empirical study. A full study would involve a large number of students and multiple learning sessions with the SAIC agent and might allow for a better evaluation of effectiveness of the agent. This is left for future work. However, it is important to point out that the study validates the design principles of the agent. We envision pedagogical agents that follow the SAIC approach would be effective at supporting children to learn new concepts regardless of the medium of communication employed by the agents, e.g. textual or spoken. We also expect that with speech technology the effectiveness of the agent for children with low reading skills will be greater.

7.9. Conclusions

We have discussed the results of the evaluation of the SAIC pedagogical agent; a prototype developed to demonstrate the computational approach of supporting children's conceptual understanding. The analysis of the evaluation data was performed to measure the effectiveness and usefulness of the SAIC agent and this has validated the agent's design architecture.

At the initial stage of the development, we conducted a formative evaluation of the system to assess the behaviour of its components and its working as functional pedagogical agent. Problems have been detected with the implementation and improvements have been made in its subsequent versions.

Using the prototype as an implementation of the design architecture informed by the theoretical principles of schema theory we have shown that a schema theory-based pedagogical agent can effectively support children's conceptual understanding. The prototype has demonstrated the agent's support of the cognitive process can help the students understand new concepts. The manipulation of the relevant schemas through the simulation of the support has given us an opportunity to examine the role of prior knowledge in conceptual understanding.

The results show that the students have improved their schematic knowledge after interacting with the agent to learn new concepts. Therefore, we have argued that the SAIC agent is both effective and useful in supporting children's conceptual understanding, and the SAIC design principles have been validated by the results.

The experimental study with the agent shows possible problems related to the agent, for example the effort required to prepare the domain and student models as pointed by the teachers, which may need a lot of human resources and funding. The study also shows potential improvements of the SAIC agent, e.g. increasing the speed of the agent's utterances generation and adding a spoken dialogue capability to the agent.

Chapter 8

Conclusion

This work has presented a design of an Intelligent Tutoring System that follows a learning theory to simulate the support to children provided by human teachers. The research has proposed a new approach to the design of an interactive pedagogical agent capable of generating adaptive explanations tailored to the needs of individual students. The basis of the computational approach adopted is a set of design principles derived from the claims of schema theory. The theory explains how people, learners in our case, gain knowledge of new concepts. We have defined schema-based teaching strategies found to be effective in supporting the learning of students as they undertake guided reading activities. This thesis has examined a design architecture called SAIC which was based on theoretical foundations from schema theory and an empirical study with human teachers. The architecture was demonstrated with a pedagogical agent with whom learners can interact to get explanations of new concepts they encounter in a reading exercise with a multimedia textbook. The main contribution of this research is the design architecture containing components that form effective explanatory tutoring systems, which has been validated in a prototype and evaluated in real classroom settings.

This chapter will present a summary of the work described in the thesis and will sketch out some future research directions. Section 8.1 will briefly present the results of our work and Section 8.2 will highlight the contribution to the corresponding research areas. Finally, Section 8.3 will point at future work.

8.1 Summary

We have presented a schema theory-based computational approach to the support of children's conceptual understanding. Our computational framework interactively guides children by generating adaptive explanations so that they can understand new concepts. In order to define an effective framework, we have investigated the principles of schema theory and examined the dialogue strategies that could be used to support schema-based cognitive tasks. The resulting design, called *Schema Activation and Interpersonal Communication* (SAIC), is based on the precise description of the following main components:

- *Representation of schematic knowledge*. We have shown how to represent domain and student schematic knowledge to provide information needed by a pedagogical agent for the explanation generation process.
- *Student modelling*. We have described how to update a schema-based student model to reflect changes caused by instructional events.
- *Mechanisms for supporting schema activation and modification*. We have formally described the schema-based processes following the Information Processing Model. This has enabled us to have relatively precise definitions of what changes are needed in the STM and LTM, and to plan the behaviour of the pedagogical agent.
- *Mechanism for planning dialogue and generating the agent's utterances*. We have shown how to plan the one-to-one dialogue between the agent and the student using schema-based patterns and template-based utterances. The dialogue management approach enables the agent to guide the interaction in order to support the child to perform appropriate schema-based cognitive tasks.

Our goal was to design a robust computer simulation of an interactive explanation process in real learning situations. To achieve this goal, we constructed formal, precise descriptions of learning theory claims. To demonstrate the validity of our design approach, we developed a prototype of a SAIC pedagogical agent and applied it to an astronomy domain presented as a multimedia educational system '*Going to the Moon'*.

The SAIC pedagogical agent demonstrates the main idea of schema theory, namely, the use of prior knowledge to help with the interpretation of new concepts. Specifically, the agent shows how domain and student schematic models can be used to generate interactive explanations tailored to the child's previous knowledge. The agent has been used as an implementation of the approach and has demonstrated its application in learning environments.

8.1.1 Adequacy of the SAIC approach

We conducted an experimental study using the SAIC agent in real classroom settings to test the validity of the approach and to examine the effectiveness of the agent in supporting children to learn new concepts. The evaluation study confirmed that the framework could simulate the support provided by human teachers and, if appropriately integrated in a learning environment, can be effective at explaining new concepts to children.

The utilisation of the SAIC agent in the experimental study helped us to identify the benefits of the computational approach in terms of improving students' schematic knowledge. The results showed that the students who used SAIC had more specific, elaborate and complete schematic knowledge after interacting with the agent. The improvement of schematic knowledge was based on definitions of the teachers who best know the learning settings and the development of the children they teach. Thus, we can consider that the evaluation findings provided evidence to validate the design principles of the agent.

The study also allowed us to examine what support the agent provided to promote schema-based reasoning in the children and how the children benefited from this. It was found that the students in the high ability group had higher learning gains compared to the students in the middle and low ability groups. The high ability group interacted more with the SAIC pedagogical agent and managed to explain more new concepts in a better way. Moreover, both teachers and student participants in the study could see the usefulness of the approach in their teaching and learning of new concepts. Thus, we argue that designing a pedagogical agent around the principles of schema theory is a fruitful approach to support children's learning of new concepts in computer-based learning environments. There are two main benefits: firstly, children's reasoning is guided so that they know how to understand new concepts and, secondly, the reasoning skills children need to prepare them to be autonomous are promoted.

The communication approach adopted by the SAIC agent has the potential to provide student-adaptive support for reading sessions by interacting with each individual child and tailoring the explanations to the problems experienced by that child. Although this activity is acknowledged as very important, it is often time-consuming and demanding for human teachers and may not be effectively integrated in the classroom teaching.

The experimental study identified issues relating to the implementation of the approach in real settings and specified the problems with the implementation. As a result, we identified several potential improvements we could make to the architecture of the agent, which will be discussed in Section 8.3.

8.1.2 Generality of the SAIC approach

Although the SAIC pedagogical agent has been demonstrated in a particular domain - introductory Astronomy – suitable for reading sessions with children and included in the UK curriculum for 7-11 years old students, the representation of schematic knowledge and the reasoning processes of the agent are not specific to any particular domain and it is expected that the pedagogical agent will be applicable to any domain that can be represented as frames. The agent should be able to employ knowledge from other systems which use frame-based representation such as Frame Logic (Kifer, Lausen & Wu, 1995), DAML+OIL (Horrocks et al., 2001), and RDFS (Brickley & Guha, 2002) as discussed in Section 4.3.2 because the pedagogical agent does not have to change its inference and decision making mechanisms when it utilises domain and student knowledge (both represented as frames). It has to be noted though that the agent will use only schematic knowledge that represents the *function, structure* and *process* of a concept because the reasoning of the agent focuses on improving a student's schematic knowledge. There may be other types of knowledge (e.g. procedural knowledge or event scripts, as discussed in Chapter 4) that can be present in the domain and student models but will not be taken into account if the agent architecture proposed in this thesis is followed. In order to consider the other types of knowledge, the agent will have to be extended with appropriate knowledge inference mechanisms and tutoring strategies.

The template-based communication approach of the SAIC agent allows both the agent and the student to actively contribute to the dialogue. This form of communication was based on an empirical study with teachers (see Chapter 3) and was later confirmed as appropriate in an evaluation study with other teachers (discussed in Chapter 7). The communication style is close to both the natural language used by students and teachers in normal classrooms and the way children engage in communication activities over the Internet. The use of a natural language ensures that the students will not have to learn a new interaction method to interact with the agent.

Although we have experimented with tutoring strategies that use templates combining text with graphics, the interaction method can be extended to include traditional menu-based interactions or to support children as they read aloud using a spoken language in an approach similar to that adopted by Beck, Mostow and Bey (2004). The SAIC interaction mode, which is based on the use of speech acts explicitly to specify the intention of an action, can be improved. For example, SAIC could be extended to include a free-style interaction language, similar to (Rosé & VanLehn, 2005; VanLehn et al., 2005). This, however, will require further research in natural language understanding and generation that is beyond the scope of this study.

The explanation strategies used by the SAIC agent are fairly general and could be applied to other pedagogical agents that support conceptual understanding because the strategies are appropriate to any chat-based communication between an agent and a child. In human-human interactions there are other issues to consider such as social and emotional factors but these are outside the scope of this research.

However, the strategies can be extended to incorporate more dialogue episodes, if the need arises, without changing the reasoning and decision-making mechanisms of the agent.

The SAIC formalisation can be used to provide a precise description of how to design a pedagogical agent based on a learning theory. In addition, it can be used by teachers to inform them about what to teach and by instructional designers who want to know how to design interactive instructional materials such as books and educational multimedia systems.

8.2 Contributions of This Work

This section highlights the contributions of this work and positions the research within relevant areas.

8.2.1 Contributions to AIED research

During our investigations, we followed a major AI in Education assumption that effective computer tutors are adaptive and dynamic in their teaching, see for example the discussion in Self (1999). We, therefore, hypothesised that computer-based explanations tailored to individual students would be more effective than generic explanations aimed at all students. We assumed that pedagogical agents used in educational settings should follow the nature of the teaching process and therefore should incorporate adaptivity, knowledge representation and reasoning. The implementation of this approach to the design of a pedagogical agent resulted in a pedagogical agent that promoted reasoning skills and the agent could raise the children's awareness of the processes that lead to conceptual understanding. The implementation of the agent used AI methods to provide the necessary knowledge representation and reasoning capabilities. The SAIC approach and the pedagogical agent that demonstrates the approach have contributed to research in AIED, in general, and to the interactive pedagogical agents stream, in particular.

The SAIC approach extends the idea of software scaffolding (Luckin, 1998): the SAIC pedagogical agent alleviates the difficulties faced by children when they are learning new concepts and, in addition to providing guidance, encourages explicit thinking about how knowledge is constructed. This helps the children to become active and autonomous readers. The work presented in this thesis focuses on supporting schema-based cognitive tasks and is the first computational attempt to design ITS based on schema theory. In this line, the work presented here contributes to ITS design approaches that are based on learning theories (as proposed by du

Boulay and Luckin, 1999), and gives supporting evidence that this is a fruitful design methodology.

8.2.2 Contributions to Education

In this work we have analysed schema theory and identified three main teaching principles based on the theory, namely, the activation of relevant prior knowledge, the performance of schema modification and the need to support children's reasoning characteristics when carrying out schema-based cognitive tasks. We have derived computational design principles based on these learning theory principles. In addition, we have identified a computational method for assessing children's schematic knowledge that defines the types of schematic knowledge the children have before and after an interaction. The analysis contributes to research on Education by describing a method for defining the children's prior knowledge explicitly and by formalising a method of schema-based teaching so that it can be implemented in computer-based tutors. In addition, the description of the support for schema-based cognitive tasks given in this thesis can be used by instructional designers or authors of multimedia educational materials for children.

8.2.3 Contributions to knowledge-based systems

The SAIC agent utilises frames to represent the knowledge of a teaching domain and the student's prior knowledge. This thesis demonstrated the feasibility of frames as the representation formalism in Intelligent Tutoring Systems and has proposed some original mechanisms for using frames to design pedagogical agents. In addition, we have defined a precise way the reasoning processes children employ when learning new concepts, which can be used to inform the design of intelligent learning environments. In this respect, the work presented in this thesis contributes to research on knowledge-based systems, in general, and to applying reasoning methods for planning user-adaptive behaviour, in specific.

8.3 Future work

In this section we discuss possible applications of the research and outline some ideas for further improvements.

8.3.1 Possible applications of the SAIC approach

Application in Malaysian primary schools – to help children

The SAIC pedagogical agent has been integrated into multimedia educational systems and evaluated in a UK primary school. The next logical step is to integrate the agent into educational software developed for Malaysian primary schools. The Malaysian government sponsored this research and will expect a nationwide implementation of our design approach. The application will require the representation of the new concepts to teach in the existing educational software, tuning the prior knowledge stereotypes to comply with the Malaysian educational system, and integrating the software into appropriate multimedia systems used in Malaysia, for example in Malaysia's *Smart School* project (1999) and, more specifically, in science educational software in Malaysia Smart School (Halim et al. (2005). Since the integration of the agent is easy (by providing a hyperlink to invoke the agent), it is realistic to expect that the SAIC approach can be applied in Malaysian primary schools.

The integration will offer new applications of our framework beyond the reading classes where the agent has been evaluated. Examples include science, mathematics and language classes. For example, Malaysian primary schoolchildren are required to learn new concepts in English grammar and, as part of their lessons, may be asked by their teachers to make inferences about the *future tense* based on their prior knowledge of the *present tense* and the *past tense,* as addressed in the blueprint of the Malaysia's *Smart School* project (1999). The difficulties faced by children learning English grammar are widely acknowledged by Malaysian teachers and the government (Hashim & Ramlan, 2004) and the agent could be applied to address these problems.

In order to apply the SAIC agent with more students in Malaysian primary schools, we expect the following steps are essential:

1- Adding a user access mechanism. The teaching strategies of the agent remain the same but the agent needs to have a mechanism to assign a student model to the right student. The access to the agent can be based on user names and passwords.

2- Putting the SAIC agent on the Web to deliver an updated SAIC agent to schools at various geographical locations. This will enable a nationwide application of the SAIC agent.

3- Cooperating with teachers in various schools and asking their students to interact with the SAIC agent. There is a need to inform the teachers about the roles of the agent and how to benefit from it.

4- Collecting the interaction data and store it automatically into a server. A dedicated server to store all the interaction data will be required.

5- Collecting the students' explanation and assessing it based on the improvement criteria, as defined in Section 7.5.2 in Chapter 7.

6- Improving the SAIC agent based on the analysis of the explanation, if necessary. New research findings in related fields such as Human Computer Interaction, Natural Language Processing and Education may be used to enhance the agent. This iterative process will ensure that the agent is always up-to-date.

Another possible application of the agent is as an explanation provider for new concepts that are domain-independent: the use of the agent as a stand-alone application that is not integrated in an educational system. Dictionaries for student, for example (Merriam-Webster Student Dictionary, wordcentral.com), are normally domain-independent and provide meaning without adapting the explanation to the prior knowledge of the children, i.e. in forms of same, generic meanings to *all* students regardless of their prior knowledge. To address this issue, the SAIC agent could be used by children to construct meaning of new concepts based on their prior knowledge. Although no domain is specified, the application of the SAIC agent will help the children relate their prior knowledge to new concepts.

Application in primary schools – to help teachers

The SAIC pedagogical agent has been developed based on schema theory, which is an established learning theory, and presented as implementation of the schema-based ITS design principles. Malaysian teachers, for example, who study a variety of learning theories in short courses (eight to ten months), could use the agent as complementary material, or as a case study, to learn how to explain new concepts to children by adapting to the prior knowledge of children (e.g. for students who are trained to become primary school teachers).

The role of parents in assisting children to acquire reading skills is stressed by researchers, for example Topping (2001) and Brooks (2002): parents might also use the SAIC pedagogical agent to help their children at home. Parents could use the agent on home computers and learn some teaching skills, especially those related to teaching difficult concepts to their children. Through the use of the agent, parents may realise the difficulties faced by their children (discussed in Chapter 2), and provide the necessary support such as encouraging their children to interact with the SAIC agent.

Studying children's learning strategies to understand new concepts

The use of the SAIC agent in primary schools will allow us to study children's learning strategies in order to understand new concepts in a computer-based learning environment. As argued by du Boulay and Luckin (2001), if we observe how children use an agent to learn new concepts, we can derive teaching principles and improve the design of the agent accordingly.

Log files would provide information about the details of the help requested by the children, the time taken for the interaction and the level of improvement in the children's schematic knowledge. This information would allow an analysis of the learning behaviours. The issue of educational data mining has been addressed by researchers, for example, Kosba (2005), Merceron and Yacef (2005) and Mostow et al. (2005). Instead of observing children in classrooms, which is time consuming and requires substantial effort and resources, the use of computer to record interaction data would ensure that researchers had reliable and ample data to analyse. This potential of the SAIC agent to capture children's learning behaviours and derive learning strategies is a promising application.

Studying children's motivational and emotional aspects while learning new concepts

The SAIC pedagogical agent acts as a helper to children during reading comprehension exercises. The agent aims to simulate certain aspects of the help offered by experienced human teachers, namely, supporting schema-based cognitive tasks using appropriate dialogue strategies. However, human teachers are also assumed to adapt their explanation to the motivational (Heffernan & Croteau, 2004; Mendez et al., 2005; Qu, Wang & Johnson, 2005; Wang et al., 2005) and emotional aspects of the children (Poel et al., 2004; Rotaru & Litman, 2005). These aspects have not been addressed in the design of the SAIC agent.

The deployment of the agent in real educational settings, similar to AMBRE-AWP (Nogry, Jean-Daubias & Duclosson, 2004), will enable us to study the motivational and emotional aspects of the children while learning new concepts in computer-based learning environment. Unlike human teachers, the agent is always available and willing to offer help. The nature of the assistance provided by the agent may motivate the children to learn more with the help of the agent. Similar to NIMIS project (Cooper & Brna, 2002), we can analyse video recorded sessions to examine the children's engagement and motivation when working with an interactive pedagogical agent. Following the evaluation methodology of Ms.Lindquist Tutor (Heffernan & Croteau, 2004), in which the tutor is networked, we can collect online interaction data of the SAIC agent-student interaction to examine motivational and emotional aspects.

8.3.2 Potential improvements of the SAIC approach

More enhanced schematic knowledge representation

The SAIC pedagogical agent generates adaptive explanations based on the knowledge represented in its knowledge base and the information supplied by a student during the interaction. The schema-based reasoning of the agent and its decision-making would be improved if the schematic knowledge representation of the domain were to be enhanced.

The schematic knowledge represented in the domain model and stereotype-based student model is not restricted to any specific domain or student. A range of domains that can be represented as frames would be suitable to be employed in the SAIC architecture. Thus, domain ontologies of various domains could be developed for children, and plugged into the architecture to enable explanations of more domain concepts. This suggests there is a need for suitable tools to create ontologies that reflects children's reasoning characteristics. Specifically, the key challenge is to create *a suitable ontology for each domain* that defines the concepts to be learned and the relationships among the concepts; in a schematic format that corresponds with the idea of children's concrete prior knowledge.

General ontology editors have been developed to create ontologies, for example Ontolingua (Ontolingua, 2001), OilEd (OilEd, 2001) and Protégé (Protégé, 2006). However, the ontologies created using these tools are not adapted to children's reasoning characteristics; the tools do not assume any difference between concrete and abstract concepts and, hence, the resultant ontologies may not be suitable for children. This discussion points to the necessity to develop a schema-based tool that can create ontologies suitable for children. In addition, as an ontology-aware ITS, it is necessary for the SAIC agent to be able to use existing ontologies, which implies that the agent must have an ability to convert the format of existing ontologies (e.g. OWL, RDF and XML Schema) into a schema-based ontology suitable for children. This points to the need to develop a converter to convert the ontology formats into a schema-based ontology.

Improved student modelling mechanism

Adaptive explanation requires an improved method for the modelling of students' current cognitive states in order to represent the changes of the students' schematic knowledge resulting from the SAIC agent's instructional events. This thesis presents these changes in terms of a simplified information processing perspective. Advances in cognition research may lead to improved student modelling, for example how mental tasks performed by brain are related to instruction (Anderson & Gluck, 2000) and modelling cognitive states in an interaction (Akhras, 2004). Knowledge from cognitive science may inform how to model complex changes of student's cognitive states that occur in LTM and STM during a reading process.

The initialisation process of the student model would be improved with the help of experts in education and knowledge engineering. This will define what types of knowledge children already have and what should be represented in the system to help children to learn a domain. Techniques from AI, e.g. machine learning, could be employed to initialise and update the student model (Aïmeur et. al., 2002; Beck & Woolf, 2000).

Better inference of the relevant prior knowledge and learning mode

The quality of the explanation provided by the SAIC agent depends on the relevance of the activated prior knowledge to the selected new concepts. The selection of these concepts is determined by *isa*, *part-of* relationships and the familiarity of the students with the prior knowledge. If the agent had a better inference mechanism of what prior knowledge to be activated, for example using logical inference techniques (Baldoni, Baroglio & Patti, 2003) and probabilistic inference techniques (Conati & Maclaren, 2004; Mayo & Mitrovic, 2000; Murray & VanLehn, 2000), it might have more relevant prior knowledge to be activated and used in the explanation.

The quality of the explanation also depends on the learning mode that is most suitable for the explanation of new concepts. This indicates that there is a need for a better inference mechanism for the selection of the most suitable learning mode in a specific learning condition and, moreover, that there is a need for a decision-making mechanism that can select which schematic knowledge to explain to a specific student in order to facilitate the reasoning of the student. Similar to selecting suitable prior knowledge for activation, logic and probabilistic techniques may be used to select a suitable learning mode.

More robust dialogue management and utterance generation technique

To conduct an engaging dialogue with children, the SAIC agent needs a more robust dialogue management system that caters for possible student-SAIC agent interaction styles, such as typed and touch-screen input, in a variety of reading contexts, e.g.

interacting with the agent and getting some help from peers, teachers or parents. The current implementation of the approach only accepts textual input and conducts one-to-one interaction. Moreover, in primary schools, teachers may adapt their dialogue according to motivational and emotional states of the children. Researchers in education, e.g. Cooper (2003), have emphasised that these factors have a great influence on children's learning, and should be taken into account. Advances in computer-based learning may give insights to provide a more robust dialogue management to the SAIC agent.

The SAIC agent is currently using structured dialogue to communicate with children and the intention of each speech act is explicitly defined. This may limit the creativity of the children to express their ideas in different ways. The representation of dialogue episodes as dialogue games could be extended to include more speech acts and more templates to inject some variety into the SAIC utterances. This might help the children's language development. Moreover, the generation of the utterances could be presented in an audio format and children would learn new concepts by listening to the explanations, for example, Reading Tutor uses digitised human speech to read menu options to students to help children with reading comprehension in *reading aloud* classes (Beck, Mostow & Bey, 2004) and Why/Autotutor uses synthesised speech to engage students in interactions (Jackson et al., 2004). Advances in computer science and computational linguistics may help the agent reliably parse spoken input. Similar to the Adele tutor (Johnson et al., 2003), the communication of the SAIC approach could be enriched using an animated pedagogical agent.

Expanding the SAIC agent

The SAIC pedagogical agent presented in this thesis shows the implementation of the SAIC design architecture with a limited number of students. The existing system needs to be expanded if the number of student increases, e.g. when applied at a nationwide level, to support more potential users of the system. In addition to considering improvement to the main components of the SAIC agent as suggested earlier in this section, we need to perform some changes to the agent in order to expand it; especially its medium of instruction, medium of delivery, maintenance of the agent and hardware requirements.

Students may come from various ethnic backgrounds who speak different languages, e.g. Malay and Chinese languages. Therefore, it is impractical to expect all users to understand English and use it as the medium of instruction. To enable more students to interact the agent, the medium of instruction of the SAIC agent should be adapted to the students. This can be direct translation of the strategies, which is currently in English, into other potential languages of the students. We also may need to consult experts in a language to ensure the suitability of the words and sentences used to articulate the SAIC strategies. This implies that the agent needs to have a mechanism to provide options to students about their language preference.

The current system is manually installed on every computer at CHALCS for evaluation purposes as discussed in the previous chapter; the experimenter went to the school to install the system. When the number of student increases, the SAIC agent should be hosted on a web server so that students of various geographical locations can download the agent; e.g. in a similar way to the WoZ system presented in Chapter 3, Section 3.3.2. The use of Internet will make the delivery of the agent to hundreds or thousands of students relatively easy. In addition, using Internet technology we can automatically update the SAIC agent without the need of user interference. This can be done by hosting the agent on a web site using *File Transfer Protocol* technology where the agent can be easily downloaded and set up, e.g. by specifying the name of a folder for the SAIC agent and placing an icon on the computer desktop to call the agent. Thus, it is reasonable to expect students or teachers with minimum level of computer knowledge to be able to perform the installation process.

Maintenance of the SAIC agent, especially its user model, may become a crucial issue when the number of users increases. When the users use computers at different locations to learn with the help of the agent, it is important to ensure that the agent is using the most updated student model. This can be done by dedicating a server for all the students models, connected to the students every time they use the SAIC agent. The access to the server can be based on user name and password. In addition to that, any technical errors with the user model or problems with user models can be addressed from a centralised location.

Application of the SAIC agent to many users assumes that their computers have an Internet connection. Expansion of the SAIC agent implies that hardware requirements concern mainly the server-side effort to host the SAIC agent, manage user interaction data and update the user models. A *Macromedia MultiUser Server* application that can reliably manage 1,000 users simultaneously should be sufficient for this purpose. *A dedicated Web server*: to enable delivery of the agent to many users at different geographical locations, we need to put the agent on the web. This requires a dedicated server to host the agent. A possible alternative is to host the agent on a reliable commercial server. *A fast computer and Internet connection*: to record user interactions in log files and updates of user models, a fast computer is needed. A Pentium IV computer with a broadband Internet connection may be sufficient for this purpose. *Hard disks* with large storage capacity are also needed to record and perform back up of the data. A 25-minute session normally requires 0.2MB of diskspace for saving its log file. Thus, 10GB of diskspace should be allocated for 1,000 users who may interact with the agent in 50 sessions in a year, i.e. an average of six sessions per month. To ensure the safety of the interaction data, another 10GB of diskspace will be required to backup the data in a server. Using *Windows Task Scheduler*, the backup can be performed automatically on an hourly or daily basis.

This project has helped the author get a deep understanding of how to design effective computer tutors and we believe that many of the aspects presented in the thesis can be implemented and applied in real educational systems. We have a strong intention to follow this work in larger projects in Malaysia and to benefit the advancements of e-learning systems there.

References

Aïmeur, E., Brassard, G., Dufort, H. & Gambs, S. (2002). CLARISSE: A machine Learning Tool to Initialize Student Models. *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pp. 718-728.

Ainsworth, S. (2003). Evaluation methods for learning environment. *Workshop held in conjunction with the 11th International Conference on Artificial Intelligence in Education (AIED'2003), Sydney, Australia.*

Ainsworth, S. (2005). Evaluation methods for learning environment. *Workshop held in conjunction with the 12th International Conference on Artificial Intelligence in Education (AIED'2005), Amsterdam, Netherlands.*

Ainsworth, S., Williams, B.C & Wood, D.J. (2003). Comparing the learning effectiveness of REDEEM and CBT. In U. Hoppe, F Verdejo. & J. Kay (Eds.) *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, 11, pp 123-130.

Aist, G. (2000). *Helping children learn vocabulary during computer-assisted oral reading*. PhD. Thesis. School of Computer Science, Carnegie Mellon University. Available online: http://www.lti.cs.cmu.edu/Research/Thesis/GregoryAist00.pdf

Aist, G. (2001). Towards automatic glossarisation: automatically constructing and administering vocabulary assistance factoids and multiple-choice assessment. *International Journal of Artificial Intelligence in Education*, 12, pp. 212-231.

Aist, G. & Mostow, J. (2000). Improving story choice in a Reading Tutor that listens. *Intelligent Tutoring Systems 2000*, pp. 645.

Ajideh, P. (2003). Schema theory-based pre-reading tasks: A neglected essential in the ESL reading class. *The reading matrix*, 3(1), pp. 1-14.

Akhras, F. N. (2004). From Modelling Teaching Strategies to Modelling Affordances. *Proceedings of the ITS'2004 Workshop on Modelling Human Teaching Tactics and Strategies, in conjunction with the 7th International Conference on Intelligent Tutoring Systems*, pp. 61-70.

Akhras, F. N. & Self, J. A. (2000). System intelligence in constructivist learning. *International Journal of Artificial Intelligence in Education*, 11, pp. 344-376.

Albacete P.L., & VanLehn, K. (2000). The conceptual helper: An intelligent tutoring system for teaching fundamental physics concepts. In G. Gauthier, C. Frasson & K. VanLehn (Eds), *Intelligent Tutoring Systems: 5th International Conference*. Berlin: Springer (Lecture Notes in Computer Science, Vol. 1839), pp. 564-573.

Aleven, V. & Koedinger, K. (2000). Limitations of student control: do students know when they need help? *Intelligent Tutoring Systems 2000*, pp. 292-303.

Aleven, V., Popescu, O., Koedinger, K.R. (2001). Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. *International Conference on Artificial Intelligence in Education*, pp. 246-255.

Aleven, V., Roll, I., McLaren, B., Ryu, E. J. & Koedinger, K. (2005). An architecture to combine meta-cognitive and cognitive tutoring: Pilot testing the Help Tutor, *International Journal of Artificial Intelligence in Education*, pp. 17-24.

Aleven, V., Stahl, E., Schworm, S., Fischer, F. & Wallace, R. (2003). Help seeking and help design in interactive learning environments, *Review of Educational Research*, 73(3), pp. 277-320.

Alpert, S.R., Singley, M.K., & Fairweather, P.G. (1999). Deploying Intelligent Tutors on the Web: An Architecture and an Example, *International Journal of Artificial Intelligence in Education*, 10, pp. 183-197.

Anderson, J. R. (1983). *The architecture of cognition*. Harvard University Press.

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Lawrence Erlbaum.

Anderson, J. R., Boyle, C. F., & Yost, G. (1985). The geometry tutor. *Proceedings of IJCAI*, pp. 1-7.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). The cognitive tutors: lessons learned*.* *Journal of the Learning Sciences*, 4(2), pp. 167–207.

Anderson, J. R., & Gluck, K. A. (2000). *What role do cognitive architectures play in intelligent tutoring systems?* Carnegie Mellon University.

Anderson, J. R. & Reiser, B. (1985). The LISP tutor. *Byte*, 10, 159-175.

Anderson, R.C. (1984). Role of the reader's schema in comprehension, learning, and memory. *Learning to Read in American Schools.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Anderson, R.C., & Pearson, P.D. (1984). A schema-theoretic view of basic reading processes. In P.D. Pearson (Ed.), *Handbook of reading research*. New York: Longman.

Anderson R. C., & Pearson, P. D. (1988). A Schema-theoretic View of Basic Processes in Reading Comprehension. in Patricia L. Carrell, et al (Eds.). *Interactive Approaches to Second Language Reading.* N.Y.: Cambridge University Press, pp. 37-55.

Andriessen, J., & Sandberg. J. (1999). Where is education heading and how about AI? *International Journal of Artificial Intelligence in Education*, 10, pp. 130-150.

Anthony, L., Corbett, A., Wagner, A., Stevens, S., & Koedinger, K. (2004). Student Question-Asking Patterns in an Intelligent Algebra Tutor. *7th International Conference on Intelligent Tutoring Systems (ITS 2004),* Maceió, Alagoas, Brazil.

Arbib, M. A. (1989). *The Metaphorical Brain 2: Neural Networks and Beyond*. New York: Wiley.

Atkinson, S. (1990). *Book of Space*. Grisewood and Dempsey.

Ausubel, D. P. (1967). *Learning theory and classroom practice*. Toronto, Canada: The Ontario Institute for Studies in Education.

Baker, M. (2000). The roles of models in Artificial Intelligence and Education research: a prospective view. *International Journal of Artificial Intelligence in Education*, 11, pp. 122-143.

Baldoni, M., Baroglio, C., & Patti, V. (2003). Applying Logic Inference Techniques for Gaining Flexibility and Adaptivity in Tutoring Systems. *HCII03,* pp. 517-521.

Bartlett, F. C. (1958). *Thinking*. New York: Basic Books.

Barunger, J., & Lewis, J. (1997). *Building a knowledge based in reading*. Newark, DE: International Reading Association.

Batliner, A., Fischer, K., Huber R., Spilker, J. & Nöth, E. (2003). How to find trouble in communication. *Speech Communication*, 40, pp.117-143.

Beck, J. E. (2000). What (and how) are we trying to learn? *Tutoring Systems that Learn workshop. The Fifth International Conference on Intelligent Tutoring Systems.*

Beck, J. E., Mostow, J., & Bey, J. (2004). Can Automated Questions Scaffold Children's Reading Comprehension? *Intelligent Tutoring Systems 2004*, pp. 478–490.

Beck, J. E. & Sison, J. (2004). Using knowledge tracing to measure student reading proficiencies. in *Proceedings of International Conference on Intelligent Tutoring Systems,* pp. 624-634.

Beck, J. E. & Woolf, B, P. (2000). High-Level Student Modeling with Machine Learning. *Intelligent Tutoring Systems 2000*, pp. 584–593.

Bernsen, N.O, Dybkjær, H. & Dybkjær, L. (1996). Principles for the design of cooperative spoken human-machine dialogue. *Proceedings of the International Conference on Spoken Language Processing (ICSLP) '96*, pp. 729-732.

Biddulph, J. (2002). *The Guided Reading Approach Theory and Research*. Learning Media Limited, New Zealand. Available online: http://www.thebiddulphgroup.net.nz/Portals/0/GuidedReading.pdf

Bigus, J. P., & Bigus, J. (1997). *Constructing Intelligent Agents with Java: A Programmer's Guide to Smarter Applications*, John Wiley & Sons, Inc., New York, NY.

Black, J. B. & Bower, G. H. (1980). Episodes as chunks in narrative memory. *Journal of Verbal Learning and Verbal Behavior,* 18, pp. 309-318.

Blessing, S. B. (1997). A programming by demonstration authoring tool for model-tracing tutors. *International Journal of Artificial Intelligence in Education*, 8, pp. 233-261.

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(6), pp. 4-16.

Bos, C. S., & Vaughn, S. (1998). *Strategies for teaching students with learning and behavior problems* (4th ed.). Needham Heights, MA: Allyn & Bacon.

Bransford, J. (1979). *Human cognition: Learning, understanding, and remembering*. Belmont, CA: Wadsworth Publishing Company.

Bransford, J. (1985). Schema activation and schema acquisition. In H. Singer & R. B. Ruddell (Eds.), *Theoretical models and processes of reading*, 3rd ed. Newark, DE: International Reading Association, pp. 385-397.

Bratko, I., (1997). *Prolog Programming for Artificial Intelligence*. Addison-Wesley. Second edition.

Braunger, J., & Lewis, J. (1998). *Building a knowledge base in reading (2nd ed.).* Newark: International Reading Association.

Brickley, D., & Guha, R.V. (2002). *RDF Vocabulary Description Language 1.0: RDF Schema*. W3Cworking. Draft 12. November 2002. W3C.

Brna, P. & Cooper, B. (2002). Supporting Young Children Learning to Write Stories Together in a 'Classroom of the Future'. *International Journal of Continuing Engineering Education and Lifelong Learning,* 12(5/6), pp. 485-503.

Brooks, G. (2002). *What works for children with literacy difficulties? The effectiveness of intervention schemes*. Research Report RR380. London: Department for Education & Skills. Available online: http://www.dfes.gov.uk/research/data/ uploadfiles/RR380.pdf

Brown, J.S. & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2.

Bruner, J., 1996. *Toward a Theory of Instruction*, Harvard University Press, Cambridge, MA.

Brusilovsky, P. (1996). Adaptive Hypermedia, an attempt to analyse and generalize. *Multimedia, Hypermedia, and Virtual Reality, Lecture Notes in Computer Science*, Vol. 1077: Springer-Verlag.

Brusilovsky, P. (2003). Adaptive web-based educational systems. *Workshop held in conjunction with the 11th International Conference on Artificial Intelligence in Education (AIED'2003), Sydney, Australia.*

Bull, S. (2004). Supporting learning with open learner models. *Proc. 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education*, Athens.

Bull, S. & Broady, E. (1997). Spontaneous Peer Tutoring from Sharing Student Models in B. du Boulay & R. Mizoguchi (eds), *Artificial Intelligence in Education*, IOS Press, Amsterdam, 1997, 143 - 150.

Bull, S. & McKay, M. (2004). An Open Learner Model for Children and Teachers: Inspecting Knowledge Level of Individuals and Peers. *Proceedings of ITS 2004*, Springer-Verlag, Berlin Heidelberg.

Bull, S., Pain, H., & Brna, P. (1995). Mr. Collins: A collaboratively constructed, inspectable student model for intelligent computer assisted language learning. *Instructional Science*, 23, pp. 65-87.

Burton, M., Brna, P. & Pilkington, R. (2000). Clarissa: A Laboratory for the Modelling of Collaboration. *International Journal of Artificial Intelligence in Education*, 11, pp. 79-105.

Carr, B. & Goldstein, I.P. (1977). *Overlays: A theory of modeling for computer aided instruction*. Technical Report AI Memo 406, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

CHALCS (2002). *Chapteltown & Harehills Assisted Learning Computer School*. Annual Performance Report/Results for Sponsors, Patrons & Parents.

Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: a practical guide. *The Journal of the Learning Sciences*, 6, pp. 271-315.

Chi, M.T.H., DeLeeuw, N., Chiu, M.-H., LaVancher, C. (1994). Eliciting Self-explanations Improves Understanding. *Cognitive Science*, 18, pp. 439–477.

Chi, M.T.H., Siler, S.A., Jeong, H., Yamauchi, T., & Hausmann, R.G. (2001). Learning from human tutoring. *Cognitive Science*, 25, pp. 471–533.

Chou, C. (2000). Schema Theory-based instructional design of asynchronous Web-based language courses. In S. S. C. Young, J. Greer, H. Maurer, & Y. S. Chee (Eds.), *Proceedings of .International Conference on Computers in Education/International Conference on Computer-Assisted Instruction*, pp. 569-575.

Ciaramita, M, & Johnson, M. (2000). Explaining away ambiguity: Learning verb selectional preference with Bayesian networks. *COLING 2000*, pp. 187-193.

Collins, A., Neville, P. & Bielaczyc, K. (2000). The Role of Different Media in Designing Learning Environments. *International Journal of Artificial Intelligence in Education,* 11, pp. 144-162.

Conati, C. & Maclaren, H. (2004). Evaluating a Probabilistic Model of Student Affect. *Intelligent Tutoring Systems 2004,* pp. 55-66.

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

Cooper, B. (2003). Care – Making the Affective Leap: More Than a Concerned Interest in a Learner's Cognitive Abilities. *International Journal of Artificial Intelligence in Education*, pp. 3-9.

Cooper, B. & Brna, P. (2002). Supporting high quality interaction and motivation in the classroom using ICT: the social and emotional learning and engagement in the NIMIS project, *Education, Communication & Information*, 2(1/2), pp. 113-138.

D'Andrade, R. G. (1992) Schemas and motivation. In R. G. D'Andrade and C. Strauss (Eds.) *Human Motives and Cultural Models*. Cambridge: Cambridge University Press.

Dahlback, N., Jonsson, A. & Ahrenberg, L. (1993). Wizard of OZ studies - why and how. *Proceedings Intelligent User Interfaces '93*, pp. 193–200.

Deempter, K. V., Krahmer, E. & Theune, M. (2005). Real vs. template-based natural language generation: a false opposition. *Computational Linguistics*, 31(1), pp.15-24.

Derry, S. J. (1996). Cognitive schema theory in the constructivist debate. *Educational psychologist,* 31(3), pp. 163-174.

Dimitrova, V. (2003). STyLE-OLM: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, 13, pp. 35–78.

Dimitrova, V., Self, J., & Brna, P. (2001). Applying Interactive Open Learner Models to Learning Technical Terminology. *Proceedings of the 8th UM 2001*, Springer, pp. 148-157.

Dix, A., Finlay, J., Abowd, G. D. & Beale, R. (2003). *Human-Computer Interaction (3rd ed.)*. Prentice-Hall.

Dochy, F. (1992). *Assessment of prior knowledge as a determinant for future learning*. Ultrecht, London: Lemma BV/Kingsley Publishers.

du Boulay, B. & Luckin, R. (2001). Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education*, 12, pp. 235-256.

Dye, G. A. (2000). Graphic organizers to the rescue! *Teaching exceptional children*, 32 (3), pp. 72-76.

Eisenberg, A.N. (1999). *An educational program for paper sculpture: a case study in the design of software to enhance children's spatial reasoning.* PhD thesis. Department of Computer Science, University of Colarado.

Ell, F. (2001). *Strategies and thinking about number in children aged 9-11 years*. Technical report 17, October 2001. University of Auckland.

Ericsson, M. (1996). *Commenting tools as design support - a Wizard-of-02 study*. Licentiate thesis, Linkoping Studies in Science and Technology. No. 576. Linkopings universitet, Sweden.

Evens, M., Brandle, S., Chang, R.C., Freedman, R., Glass, M., Lee, Y.H., Shim, L.S., Woo, C.W., Zhang, Y., Zhou, Y., Michael, J.A., & Rovick, A.A. (2001). CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue. In F. Moore (ed.), *Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference,* pp. 16-23.

Ferraris C., Brunier P., & Martel C. (2000). Constructing collaborative pedagogical situations in classrooms: a scenario and role based approach, *CSCL'2002*, Boulder, Colorado, USA.

Fielding, L.G., & Pearson, P.D. (1994). Synthesis of research: Reading comprehension: What works. *Educational Leadership*, 51(5), pp. 62–67.

Fomichov, V.A. & Fomichova, O.S. (2000). The social responsibility of computer science specialists for the creative potential of the young generation. *International Journal of Artificial Intelligence in Education. Special Issue on AIED 2010*, pp. 208-219.

Freedman, R. (1999). Atlas: A Plan Manager for Mixed-Initiative, Multimodal Dialogue. *AAA '99 Workshop on Mixed-Initiative Intelligence*, pp. 107-114.

Gagne, R. & Glaser, R. (1987). Foundations in Learning Research. In R. Gagne (Ed.) *Instructional Technology: Foundations*. Hillsdale, NJ: Erlbaum.

Garland, A., Lesh, N.B. & Rich, C., (2003). Responding to and Recovering from Mistakes During Collaboration, *International Joint Conference on Artificial Intelligence (IJCA 2003).*

Georgeff, M. P. & Ingrand, F. F. (1989). Decision-Making in an Embedded Reasoning System. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI '89),* pp. 972–978.

Ginzburg, J. (1994). An update semantics for dialogue. In H. Bunt, editor, *Proceedings of the International Workshop on Computational Semantics,* pp. 111–120.

Gluck, K. A, Anderson J. R. & Douglass. S. A. (2000) Broader Bandwidth in Student Modeling: What if ITS Were "Eye"TS? *Intelligent Tutoring Systems*, pp. 504-513.

Goodman, B., Soller, A., Linton, F., & Gaimari, R. (1998). Encouraging Student Reflection and Articulation using a Learning Companion. *International Journal of Artificial Intelligence in Education (9).*

Graesser, A. C., Lu, S., Jackson, G.T., Mitchell, H., Ventura, M., Olney, A., & Louwerse, M.M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioural Research Methods, Instruments, and Computers*, 36, pp. 180-193.

Graesser, A. C., Person, N. K., Harter, D., & TRG (2000). Teaching tactics in AutoTutor. *Proceedings of the workshop on modeling human teaching tactics and strategies at the Intelligent Tutoring Systems 2000 conference*, pp. 49-57.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9, pp. 495-522.

Greer, J.E & McCalla, G. (1994). Student Modeling: The Key to Individualized Knowledge-Based Instruction. Springer-Verlag. Berlin Germany.

Gross, R. D. (1992). *Psychology the science of mind and behaviour*. Hodder & Stoughton, pp. 741 – 762.

Gross, P., & Gross, M. (2002). *Macromedia Director 8.5 Shockwave Studio for 3D*. Macromedia Press.

Grosz, B. & Sidner, C., (1990). Plans for Discourse. *Intentions in Communication*. MIT Press.

Grow, G.O. (1996). Serving the Strategic Reader: Reader Response Theory and Its Implications for the Teaching of Writing. *Journalism and Mass Communication.*

Halim, A. H., Zain, M. Z. M., Luan, W. S., & Atan, H. (2005). The Taxonomical Analysis of Science Educational Software in Malaysian Smart Schools. *Malaysian Online Journal of Instructional Technology (MOJIT),* pp.106-113.

Hartley, D., & Mitrovic, A. (2002). Supporting learning by opening the student model. In: S. Cerri, G. Gouarderes & F. Paraguacu (eds.) Proc. 6th *Int. Conf on Intelligent Tutoring Systems ITS 2002,* Biarritz, France, LCNS 2363, pp. 453-462.

Hashim, F. & Ramlan, Z. (2004). Language Literacy Impacting on Mathematical Literacy. *The Etj Journal*, pp. 11-13.

Heffernan, N. T. & Croteau, E. (2004) Web-Based Evaluations Showing Differential Learning for Tutorial Strategies Employed by the Ms. Lindquist Tutor. *Proceedings of 7th Annual Intelligent Tutoring Systems Conference*, pp. 491-500.

Heffernan, N.T & Koedinger, K.R (2002). An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. *Intelligent Tutoring System 2002*, pp. 596-608.

Heift, T., & Nicholson, D. (2001). Web delivery of adaptive and interactive Language Tutoring. *International Journal of Artificial Intelligence in Education*, 12, pp. 310-324.

Heller R. S. (1991). Evaluating software: A review of the options. *Computers & Education*, 17(4), pp. 285-291.

Henk, W. (1993). New Directions in Reading Assessment. *Reading and Writing Quarterly*, vol. 9: pp. 103-120.

Henze, N. & Nejdl, W. (2001). Adaptation in Open Corpus Hypermedia. *International Journal of Artificial Intelligence in Education,* 12.

Heylinghen, F. (2001). Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs. *Kybernetes*, 30, pp. 691-722.

Hill, I. (1983). Natural language versus computer language. In M. Sime & M. Coombs (Eds.) *Designing for Human-Computer Communication*. Academic Press.

Holland, D. & N. Quinn (Eds.) (1987). *Cultural Models in Language and Thought*. Cambridge: Cambridge University Press.

Holt, P., Dubs, S., Jones, M., Greer, J.E. (1994). The State of Student Modelling. *Student Modeling: the Key to Individualized Knowledge-based Instruction*. Berlin: Springer-Verlag, pp. 3-35.

Horrocks, I., Harmelen, F. N., Schneider, P. P.I., Horrocks, F. van Harmelen & Patel-Schneider, P. (2001). *DAML+OIL. DAML Program*.

Hsieh, P.Y., Halff, H. M. & Redfield, C.L. Four easy pieces: development systems for knowledge-based generative instruction. *International Journal of Artificial Intelligence in Education,* 10, pp. 1-45.

Hunter, I.M.L. (1964). *Memory*. Pelican.

Hutchins, E. (1994). *Cognition in the wild*. Cambridge, MA. MIT Press.

Iqbal A., Oppermann R., Patel A. & Kinshuk (1999). A Classification of Evaluation Methods for Intelligent Tutoring Systems. In U. Arend, E. Eberleh & K. Pitschke (Eds.) *Software Ergonomie '99 - Design von Informationswelten*, B. G. Teubner Stuttgart, Leipzig, pp. 169-18.

Jackson, G. T., Ventura, M., Chewle, P., Graesser, A., & the Tutoring Research Group (2004). The Impact of Why/AutoTutor on Learning and Retention of Conceptual Physics, *Intelligent Tutoring Systems*, pp. 501-510.

Jitendra, A. (2002). Teaching students math problem-solving through graphic representations. *Teaching exceptional children*, 34, pp. 34-38.

Johnson, W.L. (1998). Pedagogical agents. *Proceedings of ICCE ’98, the Sixth International Conference on Computers in Education*.

Johnson, W. L. (1990). Understanding and debugging novice programs. *Artificial Intelligence*. 42, pp. 51-97.

Johnson, W.L., Paola Rizzo, P., Hyokeong Lee, H., Ning Wang, N., & Shaw, E. (2004). *Modeling Motivational and Social Aspects of Tutorial Dialog, ITS’04. 7th International Conference on Intelligent Tutoring Systems.*

Johnson, W.L., Shaw, E., Marshal, A., LaBore, C. (2003). Evolution of User Interaction: The Case of Agent Adele. *In Proceedings of the 2003 International Conference on Intelligent User Interfaces,* pp. 93-100.

Jonassen, D. H. (1987). Assessing cognitive structure: verifying a method using pattern notes. *Journal of Research and Development in Education*, 20 (3), pp. 1-13.

Kalyuga, S (2003). Rapid Assessment of Learners' Knowledge in Adaptive Learning Environments. *Artificial Intelligence In Education*, pp. 167-174.

Kamp, H. & Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic, and Discourse Representation Theory*. Dordrecht: Kluwer.

Kass, R., & Finin, T. (1988). Modeling the user in natural language systems. *Computational Linguistics*, 14, pp. 5-22.

Katz, S., O'Donnell, G. & Kay, H. (2000). An approach to analyzing the role and structure of reflective dialogue. *International Journal of Artificial Intelligence in Education,* 11(3), pp. 320-343.

Kay, J. (2000). Stereotypes, Student Models and Scrutability. *Intelligent Tutoring Systems 2000*, pp.19-30.

Kellog, R., T. (1995). *Cognitive Psychology*. Sage.

Kifer, M., Lausen, G, & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages, *Journal of the ACM*, 42(4), pp. 741-843.

Kneser, C., Pilkington, R. M. & Treasure-Jones, T. (2001). The tutor's role: An investigation of the power of exchange structure analysis to identify different roles in CMC seminars. *International Journal of Artificial Intelligence in Education*, 12, pp. 63-84.

Knowles, M. (1984). *The Adult Learner: A Neglected Species* (3rd Edition). Houston, TX: Gulf Publishing.

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

Kosba, E., (2005). *Generating Computer-based Advice in Web-based Distance Education Environments*. PhD Thesis, University of Leeds, Leeds. UK.

Kosba, E., Dimitrova, V. & Boyle, R. (2004). Using fuzzy techniques to model students in web-based learning environments. *International Journal on Artificial Intelligence Tools*, 13, pp. 279-298.

Le, T. V. (1993). *Techniques of Prolog programming, with implementation of logical negation and quantified goals*. John Wiley, New York.

Lesgold, A. M., Lajoie, S. P., Bunzo, M., & Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In Larkin, J. & Chabay, R. (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared issues and complementary approaches*, pp. 201-238.

Lester, J., Voerman, J., Towns, S., & Callaway, C. (1997). Cosmo: A Life-like Animated Pedagogical Agent with Deictic Believability. *IJCAI'97 Workshop on Animated Interface Agents: Making Them Intelligent*, pp. 61-69.

Levin, J., & Moore, J. (1977). Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science*, 1 (4), pp. 395-420.

Lochbaum, K.E. (1998). A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4).

Luckin, R., (1998). *ECOLAB: Explorations in the Zone of Proximal Development*. PhD Thesis, University of Sussex, Sussex. UK.

Luckin, R. (1998). Ecolab: Exploring the construction of a Learning Assistant. In B. P. Goettl & H. M. Halff, C. L. Redfield & V. J. Shute (Eds.), *Intelligent Tutoring Systems: 4th International Conference, ITS'98*, 1452, pp. 304 - 313.

Luckin, R. & du Boulay, B. (1999). Ecolab: The development and evaluation of a Vygotskian design framework. *International Journal of Artificial Intelligence in Education,* 10(2), pp. 198–220.

Luckin, R., & Hammerton, L. (2002). Getting to know me: helping learners understand their own learning needs through metacognitive scaffolding. In S. A. Cerri & G. Gouarderes & F. Paranguaca (Eds.), *Intelligent Tutoring Systems* (pp. 759-771). Berlin: Springer-Verlag.

Luckin, R., Plowman, L., Laurillard, D., Stratfold, M., Taylor, J. & Corben, S. (2001). Narrative Evolution: Learning From Students' Talk About Species Variation. *International Journal of Artificial Intelligence in Education,* 12, pp. 100-123.

Mahoney, T. (2001*). Astronomy the Study of the Universe*. Snapping-Turtle Guide.

Mark, M.A. & Greer, J.E. (1993). Evaluation methodologies for intelligent tutoring systems, *International Journal of Artificial Intelligence in Education*, 4, pp. 129-153.

Marshall, S. P. (1995). *Schemas in problem-solving*. London: Cambridge University Press. Organizational Psychology, San Diego, CA.

Martin, B. (1999). Constraint-Based Modeling: Representing Student Knowledge. *New Zealand Journal of Computing,* 7 (2), pp 30-38.

McCalla, G. (2000). The Fragmentation of Culture, Learning, Teaching and Technology: Implications for the Artificial Intelligence in Education Research Agenda in 2010. *International Journal of Artificial Intelligence in Education*, 11, pp. 177-196.

McRoy. S. W., Channarukul. S. & Ali, S. S. (2003). An Augmented Template-Based Approach to Text Realization. *Natural Language Engineering*, 9(2), pp. 1–40.

Melis, E., Andrès, E., Büdenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). ActiveMath: A Generic and Adaptive Web-Based Learning. Environment. *International Journal of Artificial Intelligence in Education,* 12, pp. 385-407.

Mendez, G. R, du Boulay, B. & Luckin, R. (2005). “Be Bold and take a challenge”: Could motivational strategies improve help-seeking? *Artificial Intelligence in Education*, pp. 459 – 466.

Merceron, A. & K. Yacef (2005). Educational Data Mining: a Case Study. *Artificial Intelligence in Education,* pp. 467-474.

Mertz, J.S. (1997). Using A Simulated Student for Instructional Design. *International Journal of Artificial Intelligence in Education*, 8, pp. 116-141.

Miller, G.A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, pp. 81-97.

Milosavljevic, M. (1997). Augmenting the user’s knowledge via comparison. *User modeling: Proceedings of the Sixth International Conference, UM97,* pp.119-130.

Minsky, M. (1975). A framework for representing knowledge. *The psychology of computer vision.* New York: McGraw-Hill, pp. 211-277.

Minsky, M. (1981). A Framework for Representing Knowledge. In J. Haugeland (ed.), *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pp. 95-128.

Mitrovic, A. (1998). A knowledge-based teaching system for SQL. In T. Ottmann & I. Tomek (Eds.), *Proceedings of ED-MEDIA ’98*, pp. 1027-1032.

Mitrovic, A., & Ohlsson, S. (1999) Evaluation of a Constraint-based Tutor for a Database Language. *International Journal on Artificial Intelligence in Education*, 10, pp. 238-256.

Mizoguchi, R. & Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal on Artificial Intelligence in Education,* 11, pp.1-12.

Montessori, M. (1973). *From Childhood to Adolescence*. New York: Schocken Books.

Mosenthal, J., Lipson, M., Mekkelsen, J., Russ, B. & Sortino, S. (2001*). Elementary schools where students succeed in reading.* The Northeast and Islands Regional Educational Laboratory. The Education Alliance. Brown University.

Mostow, J., Beck, J., Cuneo, A., Gouvea, E., & Heiner, C. (2005). A Generic Tool to Browse Tutor-Student Interactions: Time Will Tell! Proceedings of the 12th *International Conference on Artificial Intelligence in Education*, pp. 884-886.

Muirden, J. (1987*). Junior Illustrated Encyclopedia The Universe*. Grisewood and Dempsey.

Murray, R. C., & VanLehn, K. (2000). DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions. *Intelligent Tutoring Systems 2000*, pp. 153-162.

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. *International Journal of Artificial Intelligence in Education*, 10, pp. 98-129.

National Center to Improve the Tools for Educators (1996). *Reading: The first chapter in education.* U.S. Department of Education's Learning to Read, Reading to Learn campaign. University of Oregon.

National Institute of Child Health and Human Development, Report of the National Reading Panel. (2000). *Teaching Children to Read: An Evidence-Based Assessment of the Scientific Research Literature on Reading and Its Implications for Reading Instruction:* Reports of the Subgroups (NIH Publication No. 00-4754). Washington, DC: U.S. Government Printing Office. Available online from http://www.nationalreadingpanel.org

Nishida, H. (1999). A Cognitive Approach to Intercultural Communication Based on SchemaTheory. *International Journal of Intercultural Relations*, 23 (5), pp. 753 - 777.

Nkambou, R. Gauthier, G. & Frasson, C. (1996). CREAM-TOOLS: An Authoring Environment for Curriculum and Course Building in an Intelligent Tutoring System. *3rd International Conference on Computer Aided Learning and Instruction in Science Engineering.* Springer Verlag, 1108, pp. 186-194.

Nogry, S., Jean-Daubias, S., & Duclosson, N. (2004). ITS Evaluation in Classroom: The Case of AMBRE-AWP. *Intelligent Tutoring Systems 2004*, pp. 511–520.

Norman, D. (1998). *The Design of Everyday Things*. New York. Doubleday.

Norris, S. & Phillips, L. (1987). Explanations of reading comprehension: Schema theory and critical thinking theory. *Teachers College Record*, 89, 2, pp. 281-306.

Novak, J. & Gowin, D. (1984). *Learning how to learn*. New York: Cambridge University Press.

Oates, J. M. & Grayson, A. (2004). *Cognitive and language development in children*. Oxford. Blackwell.

OilED. http://oiled.man.ac.uk/

Ontolingua. http://www-ksl-svc.stanford.edu/

Paiva A., Costa M., Chaves R., Piedade M., Mourão D., Sobral D., Höök K., Andersson G., Bullock A. (2003). SenToy: an Affective Sympathetic Interface, *International Journal of Human-Computer Studies,* 59, pp. 227-235.

Papert, (1993). *The children's machinge: Rethinking school in the age of the computer*. New York: Basic Books.

Pelachaud, C., Badler, N. & Steedman, M. (1996). Generating Facial Expressions for Speech. *Cognitive Science*, 20(1), pp. 1-46.

Person, N.K., Graesser, A.C., Bautista, L., Mathews, E., & the Tutoring Reasearch Group. (2001). Evaluating student learning gains in two versions of AutoTutor. In J.. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.) *Artificial intelligence in education: AI-ED in the wired and wireless future*, pp. 286-293.

PET (2004*). Preliminary English Test. Examination Report*. University of Cambridge.
http://www.cambridgeesol.org/support/dloads/pet/ pet_rep_dec04.pdf

Piaget, J. (1977). *Epistemology and Psychology of Functions*. Dordrecht: D. Reidel Publishing Company.

Piaget, J. & Inhelder, B. (1969). *The Psychology of the Child*. NY: Basic Books.

Pilkington, R.M. (1999). *Analyzing Educational Discourse: The DISCOUNT Scheme*. Version 3, January, CBL Technical Report No. 99/2, ISBN: 1 901418 022.

Poel, A., Heylen, D., & Nijholt, A. (2004). Emotion Based Agent Architectures for Tutoring Systems The INES Architecture. *Cybernetics and Systems. Workshop on Affective Computational Entities* (ACE 2004), pp. 663-667.

Poesio, M., & Traum, D. (1997). Representing Conversation Acts in a Unified Semantic/Pragmatic Framework, in *working notes of AAAI Fall Symposium on Communicative Action in Humans and Machines*.

Poesio, M. & Traum, D. (1998). Towards an Axiomatization of Dialogue Acts, In J. Hulstijn & A. Nijholt (eds.), *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues*, pp. 207-222.

Potter, M. (2000). *Elementary A Space Mission.* Dorling Kindersley. London.

Preece, F. F. W. (1976) Mapping cognitive structure: A comparison of methods. *Journal of Educational Psychology*, pp. 68, 1–8.

Pressley, M. (1998). *Reading instruction that works: The case for balanced teaching*. New York: Guilford.

Price, E. & Driscoll, M. (1997). An Inquiry into the spontaneous transfer of Problem-solving skill. *Contemporary Educational Psychology,* 22, pp. 472–494.

Protégé (2006). http://protege.stanford.edu/

Pulman, S. (1996). Conversational games, belief revision and bayesian networks. In *CLIN VII: Proceedings of 7th Computational Linguistics*, pp. 1–24.

Pulman, S. (1999). Relating dialogue games to information state, in J. Terken & M. Swerts, eds, *Proceedings of the ESCA Workshop on dialogue and prosody*, pp.17-24.

Qu, L., Wang, N, & Johnson, W. L. (2005). Using Learner Focus of Attention to Detect Learner Motivation Factors. *User Modelling*, pp. 70-73.

Rapp, S. & Strube, M. (2002). An iterative data collection approach for multimodal dialogue systems. In: *Proceedings of the Second International Conference on Language Resources and Evaluation LREC 2002*, pp. 661-665.

Ratnaparkhi, A. (1998). *Maximum entropy models for natural language ambiguity resolution.* PhD thesis. Computer and Information Science. University of Pennsylvania.

Ravenscroft, A. & Pilkington, R. M. (2000). Investigation by Design: Developing Dialogue Models to Support Reasoning and Conceptual Change. *International Journal of Artificial Intelligence in Education,* 11, pp. 273-298.

Rich E. (1979). User Modeling via Stereotypes. *International Journal of Cognitive Science,* 3, pp. 329-354.

Rich, E. (1989). Stereotypes and user modeling. In: Kobsa, A., Wahlster, W.: *User models in dialog systems*. Springer-Verlag, pp. 35-51.

Rich, C., & Sidner, C., (1998). Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8, pp. 315 - 350.

Risko, V. & Alvarez, M. (1986). An investigation of poor readers' use of a thematic strategy to comprehend text. *Reading Research Quarterly*, 21, pp. 298-316.

Rizzo, P., Lee, H., Shaw, E., Johnson, W.L., Wang, N. & Mayer, R.E. (2005). A Semi-Automated Wizard of Oz Interface for Modelling Tutorial Strategies. *User Modeling 2005*, pp. 174-178.

Rizzo, P., Shaw, E., & Johnson, W.L. (2002). An agent that helps children to author rhetorically-structured digital puppet presentations. *Intelligent Tutoring Systems 2002*, pp. 903-912.

Robertson, J. & Wiemer-Hastings, P. (2002). Feedback on children's stories via multiple agents. *Intelligent Tutoring Systems 2002*, pp. 923-932.

Rosé C., & VanLehn, K. (2005). An Evaluation of a Hybrid Language Understanding Approach for Robust Selection of Tutoring Goals. *International Journal of Artificial Intelligence in Education*, 15, pp. 325-355.

Rosé, C., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive Conceptual Tutoring in Atlas-Andes. *In Proceeding of AI in Education 2001*.

Rotaru, M. & Litman, D. (2005). Using Word-level Pitch Features to Better Predict Student Emotions during Spoken Tutoring Dialogues. In *Proceedings 9th European Conference on Speech Communication and Technology (Interspeech-2005/Eurospeech).*

Rumelhart, D. (1980). Schemata: The building blocks of cognition. In Spiro, R.J., Bruce, B.C & Brewer, W.C. (Eds*.), Theoretical issues in reading comprehension*, Hillsdale, NJ: Erlbaum, pp. 33-58.

Rumelhart, D. & Norman, D. (1978). Accretion, tuning and restructuring: Three modes of learning. In. J.W. Cotton & R. Klatzky (eds.), *Semantic Factors in Cognition*. Hillsdale, NJ: Erlbaum.

Rumelhart, D. & Norman, D. (1981). Analogical processes in learning. In J.R. Anderson (ed.), *Cognitive Skills and their Acquisition*. Hillsdale, NJ: Erlbaum.

Salber, D., Coutaz, J. (1993) Applying the Wizard of Oz Technique to the Study of Multimodal Systems. In: *EWHCI'93*, Springer-Verlag, Berlin, pp. 219-230.

Sallaberry, C., Nodenot, T., Laforcade, P. & Marquesuzaa, C. (2004). Modelling Scenarios of Cooperation Promoting Human Learning: Added Value of the CPM Profile. *Intelligent Tutoring Systems 2004. 7th Conference on Intelligent Tutoring Systems*.

Salomon, G. (1993) No distribution without individual's cognition: a dynamic interactional view. In Salomon (Ed.) *Distributed Cognitions*, pp. 111-138.

Schank, R.C. (1982). *Reading and Understanding*. Hillsdale, NJ: Erlbaum.

Schank, R.C. & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Assoc.

Schulze, K. G., Shelby, R. N., Treacy, D. J., Wintersgill, M. C., VanLehn, K., & Gertner, A. (2000). Andes: An intelligent tutor for classical physics, *The Journal of Electronic Publishing*, University of Michigan Press, Ann Arbor.

Schwartz, N. H., Ellsworth, L., Graham, L. & Knight, B. (1998). Accessing prior knowledge to remember text: A comparison of advance organizers and maps. *Contemporary educational psychology*, 23, pp. 65-89.

Seeds, E. M. (2005). *A Philosophy of Introductory Astronomy Teaching.* Department of Physics and Astronomy. Franklin & Marshall College. http://www.fandm.edu/x6458.xml

Self, J. A. (1990). Bypassing the Intractable Problem of Student Modeling. *Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*. Norwood: Ablex, pp. 107-123.

Self, J. A. (1994). Formal approaches to student modelling. In J.E. Greer & G.I. McCalla (eds.), *Student Modelling: the Key to Individualized Knowledge-Based Instruction,* Berlin: Springer-Verlag.

Self, J. A. (1994). The role of student models in learning environments. Trans. of the Institute of Electronics, *Information and Communication Engineers* E77-D, pp. 3–8.

Self, J. A. (1999). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence in Education,* 10, pp. 350-364.

Shute, V. J. & Regian J. W. (1993). Principles for evaluating intelligent tutoring systems. *Journal of Artificial Intelligence and Education*, 4 (2/3), pp. 245-271.

Sison R. & Shimura M. (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education* 9, pp.128-158.

Slavin, R. E. (1991). *Educational psychology*(3rd ed.). Needham Heights, MA: Allyn &Bacon.

Sleeman, D., Kelly, A. E., Martinak, R., Ward, R. D., & Moore, J. L. (1989). Studies of diagnosis and remediation with high school algebra students. *Cognitive Science*, 13, pp. 551- 568.

Sleeman, D.H., Stacey, M.K., Edwards, P. & Gray, N.A.B. (1989). An Architecture for Theory-Driven Scientific Discovery. In *Proceedings of the Fourth European Working Session on Learning,* pp.11-23.

Smart School (1999). http://www.ppk.kpm.my/smartschool/

Stern, M., Beck, J., & Woolf, B.P. (1996). Adaptation of problem presentation and feedback in an intelligent mathematics tutor. In C. Frasson, G. Gauthier & A. Lesgold (Eds.), *Intelligent tutoring systems*, pp. 603-613.

Steyvers, M., & Tenenbaum, J. (2005). The Large Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science,* 29(1), pp. 41-78.

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

Sweller, J. & Chandler, P. (1994). Why Some Material is Difficult to Learn. *Cognition and Instruction,* 4, pp. 295-312.

Tambe, M. (1997). Towards Flexible Teamwork, *Journal of Artificial Intelligence Research,* 7, pp. 83-124.

Tchetagui, J. M. P., & Nkambou R. (2002). Hierarchical Representation and Evaluation of the Students in an Intelligent Tutoring System. In Cerri, S.A., Gouarderes, G., & Paraguacu, F. (eds.): *Sixth International Conference on Intelligent Tutoring Systems (ITS-2002).* Lecture Notes in Computer Science, Vol. 2363. Springer-Verlag, Berlin, pp. 708-717.

Tedesco, P. A (2003). MArCo: Building an Artificial Conflict Mediator to Support Group Planning Interactions. *International Journal of Artificial Intelligence in Education,* 13, pp. 117-155.

Topping, K. J. (2001) *Peer and parent assisted learning in reading, writing, spelling and thinking skills*: Spotlight No. 82. Edinburgh: Scottish Council for Research in Education.

Towne, D. M. (1997). Approximate Reasoning Techniques for Intelligent Diagnostic Instruction. *International Journal of Artificial Intelligence in Education*, 8, pp. 262-283.

Tsiriga, V. & Virvou, M. (2003) Modelling the Student to Individualise Tutoring in a Web-Based ICALL. *International Journal of Continuing Engineering Education and Life-long learning,* 13, pp.350-365.

UCLES (2001). *University of Cambridge Local Examinations Syndicate (UCLES) report on the Preliminary English Test project with Realschule pupils in Baden* Württemberg.
http://www.cambridge-exams.de/news/ Baden_Wuerttemberg.pdf.

VanLehn., K. (1988). *Architectures for intelligence*. The Twenty-second Carnegie Mellon Symposium on Cognition. Hillsdale, N.J.: L. Erlbaum Associates.

VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence in Education,* pp.147-204.

Vassileva, J. (1997). TOBIE: An Implementation of a Domain- Independent ITS-Architecture in the Domain of Symbolic Integration, in Kommers P. (Ed.) *New Media and Telematic Technologies for Education in Eastern European Countries,* Twente University Press, pp. 241-258.

Virvou, M. (2000). Involving effectively teachers and students in the life cycle of an intelligent tutoring system. *Educational Technology & Society* 3(3), pp. 511-521.

Virvou, M., & Mundridou, M (2001). Adding an Instructor Modelling Component to the Architecture of ITS Authoring Tools. *International Journal of Artificial Intelligence in Education,* 12, pp. 185-211.

Vygotsky, L. S. (1978). *Mind and society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.

Wang, N., Johnson, W. L., Rizzo, P., Shaw, E. & Mayer, R. E. (2005). Experimental evaluation of polite interaction tactics for pedagogical agents.

Proceedings of the 10th international conference on Intelligent user interfaces (IUI), pp. 12-19.

Wang, W.C. & Chan, T.W. (2000). CAROL5: An Agent-Oriented Programming Language for Developing Social Learning Systems. *International Journal of Artificial Intelligence in Education*, 11, pp. 1-32.

Weber, G. & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education,* 12, pp. 351-384.

Weibelzahl, S. & Weber, G. (2002). Adapting to prior knowledge of learners. *Proceedings of the International Conference on Adaptive Hypermedia AH200*, pp. 448-451.

Weiskamp, K., & Hengl, T. (1988). *Artificial Intelligence Programming with Turbo Prolog.* John Wiley & Sons Inc.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

White, M. & Caldwell, T. (1998). EXEMPLARS: A Practical, Extensible Framework for Dynamic Text Generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation,* pp. 266-275.

Wilensky, R., Chin, D.N., Luria, M., Martin, J. & Mayfield, J. (2000). The UNIX Consultant Project. In *Artificial Intelligence Review* Vol. 14, No. 1-2, pp. 43-88.

Wilkins, D. (1988). *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan-Kaufmann: San Mateo.

Wong, W.K. & Chan, T.W. (1997). A Multimedia Authoring System for Crafting Topic Hierarchy, Learning Strategies, and Intelligent Models. *International Journal of Artificial Intelligence in Education,* 8, pp. 71-96.

Wood, D. (2001). Scaffolding, contingent tutoring and computer-supported learning. *International Journal of Artificial Intelligence in Education*, 12, pg. 280-292.

Woolf, B. P. (2003). *Building intelligent tutoring systems*. University of Massachusetts. Amherst, MA.

Yamada, K., R. Mizoguchi, et al. (1993) Model of Utterance and Its Use in Cooperative Response Generation, *Proc. of East-West International Conference on Human-Computer, Interaction (EWHCI '93),* pp. 252-264.

Yang, Y., Okamoto, M. & Ishida, T. (2000). Applying wizard of oz method to learning interface agent. In: *IEICE Workshop on Software Agent and its Applications (SAA2000)*, pp. 223-230.

Yen, J., Yin, J., Ioerger, T. R., Miller, M. S., Xu, D. & Volz, R.A. (2001). CAST: Collaborative Agents for Simulating Teamwork, *Proc. Of the 17th International Joint Conference on Artificial Intelligence* (IJCAI'2001), pp. 1135-1142.

Young, S. S. (2003). *Guided reading implementation*. Case study of implementation Pinellas County, Florida.

Zakaluk, B. L., & Samuels, S. J. (1988). Toward a new approach to predicting text comprehensibility. In B. L. Zakaluk & Samuels, S. J. (Eds.). *Readability, Its Past, Present, and Future*, Newark, DE: International Reading Association.

Zapata-Rivera, J.D. & Greer, J. (2000). Inspecting and Visualizing Distributed Bayesian Student Models. *Proceedings of the 5th International Conference on Intelligent Tutoring Systems, Intelligent Tutoring Systems 2000*, pp. 544-553.

Zapata-Rivera, J.D. & Greer, J. (2003). Analysing student reflection in The Learning Game. *Artificial Intelligence in Education*, pp. 288-298.

Zinn, C., Moore, J. D., & Core, M. (2002). A 3-tier Planning Architecture for Managing Tutorial Dialogue*, Intelligent Tutoring Systems 2002,* pp. 574-584.

Zouaq, A., Frason, C. & Rouane, K. (2000). The explanation agent. *Intelligent Tutoring Systems 2000*, pp. 554-563.

Appendix-A

List of the new concepts and examples of their schematic knowledge

This appendix presents the new concepts and example of their schematic knowledge. The following schematic knowledge is based on several discussions with CHALCS teachers.

```
astronaut(isa, profession).
astronaut(function, goes-to-the-moon).
astronaut(process, pilots-space-shuttle).
astronaut(is, brave).
astronaut(can, walk-on-moon).
astronaut(has, uniform).
astronaut(workplace, space-centre).
astronaut(vehicle, space-shuttle).

moon(isa, planet).
moon(function, rotates-around-the-earth).
moon(is, visible-at-night).
moon(shape, round).
moon(surface, rocky).

space-shuttle(isa, vehicle).
space-shuttle(function,transports-people-in-outer-space).
space-shuttle(process, blasts-into-outer-space).
space-shuttle(picture, yes).
space-shuttle(is, very-fast).
space-shuttle(has, wings).
space-shuttle(can, travel-in-outer-space).
space-shuttle(can, land-on-the-moon).
space-shuttle(engine, rocket).
space-shuttle(driver, pilot).
space-shuttle(destination, moon).

earth(isa, planet).
earth(function, moves-around-the-sun).
earth(is, the-planet-on-which-we-live).
earth(is, third-planet-from-the-sun).
earth(is, round).
earth(has, air).
earth(has, oxygen).

telescope(isa, image-magnifier).
telescope(function,magnifies-images-of-distant-objects).
telescope(process, uses-prism).
telescope(is, expensive).

star(isa, planet).
star(function, blinks-at-night).
star(is, visible-at-night).

planet(isa, sky-object).
planet(function, rotates-around-the-sun).

space(isa, place-outside-world).
space(has, planets).

space-centre(isa, learning-centre).
space-centre(function,prepares-astronaut-for-space-
exploration).
space-centre(process, trains-astronaut).
space-centre(student, astronauts).
space-centre(has, space-scientists).

training(isa, activity).
training(function, improves-skill).
training(process, corrects-mistakes).

interviewing(isa, discussion).
interviewing(function, gets-information).
interviewing(process, asks-questions).

gravity(isa, force-of-attraction).
gravity(function, gives-weight-to-object).
gravity(process, attracts-objects-around-earth).
gravity(attraction, less-on-the-moon).

swimming(isa, activity).
swimming(function, makes-astronaut-healthy).
swimming(function, floats-on-water).
swimming(place, swimming-pool).

taking-off(isa, activity).
taking-off(function, sends-space-shuttle-into-outer-space).
taking-off(place, space-centre).

computer(isa, electronic-device).
computer(function,controls-the-space-shuttle's-landing).
computer(process,processes-data-from-the-moon's-surface).
computer(has, operating-system).
computer(has, processing-unit).

satellite(isa, scientific-equipment).
satellite(function, sends-signal-back-to-earth)
satellite(can, orbits-around-the-earth).
satellite(location, is-in-the-outer-space).

floating(is, activity).
floating(place, in-the-air).

oxygen(isa, gas).
oxygen(function, helps-people-breath).
oxygen(is, colourless).
oxygen(is, odourless).

landing(isa, activity).
landing(function,brings-space-shuttle-onto-the-moon's-
surface).
landing(place, on-the-moon's-surface).
landing(is, controlled-by-computer).

space-suit(isa, uniform).
space-suit(function, protects-astronaut-from-heat).
space-suit(has, supply-of-oxygen).
```

Appendix B

The usual format of a CHALCS reading session

The following picture shows the teacher E.W. demonstrating how she explains important concepts in a story to her students. Based on her experience teaching the students, she considers some words in the story as important for children to understand in order to understand the whole story and therefore they must be explained.

Figure B.1 A reading session at CHALCS

The explanation process:

1- The teacher selects a book which she thinks suitable for the students. The factors which determine the selection:
   - Is the book interesting? – She asks for their opinion. Everybody agrees to read a book 'Tell me a story'.
   - Is the book suitable to their age level? – The book is written for children age 9 to 10 years old. It is therefore considered suitable.
   - Have the students read the book? – She asks them whether they have read it in the previous lessons.

2- The students read the book
   - They take turns reading the book; a paragraph for each student or two paragraphs if they are short.
   - She makes sure that all students have an equal opportunity to read the paragraphs during the reading session.

3- Asking students whether they understand a certain word
    - When a student finishes reading a sentence in a paragraph and if the sentence contains a word which requires explanation, she asks the student to stop reading for a while and asks the class if they know what that particular word means.

4- Students explain the word
    - Students raise their hand to explain.
    - To ensure that everybody involved in the reading process, she makes sure that everybody have the opportunity to answer.

5- The teacher gives her explanation
    - Explanation is based on the context; a folk story about the inseparable relationship between humans and animals.
    - If the explanation is correct, the teacher confirms it and praises the student for his or her brilliant answer.
    - If the explanation is partially correct, the teacher confirms it and adds additional explanation.
    - If the explanation is wrong, the teacher informs him/or her about it, thanks the student for his/her explanation, and ask other students if they think they know how to explain the word.

6- The students continue reading the lesson
    - The reading session ends when the students finish reading the story or the teacher has to stop it when runs out of time.

Appendix C

Guidelines for teachers

This appendix provides the guidelines for teachers who participated in the evaluation study and samples of pre-test transcripts.

Description: the teachers should have a clear idea of how the experiment will be conducted. The experimenter will explain the types of help required from the teacher at each evaluation stage: the pre-test, training session, reading session, book review and interview. This appendix provides the scripts prepared the evaluation stages.

Objective: to explain the purpose of each evaluation stage to the teachers and how the experiment will be conducted so that they know how to help.

Script:

1- Pre-test. To be conducted one week before the reading session.

The purpose of this stage is to *assess* the background knowledge of each student. We can call this stage the 'General knowledge test' which is equivalent to brainstorming and concept mapping sessions you normally have in the classroom. You should encourage them to inform students what they know about the important words. At this stage, we want to construct the background knowledge which may help them understand the key concepts in the lesson. The 'General knowledge test' will take about 20-30 minutes.

For research purposes, I need their answers to inform the pedagogical agent how to adjust its explanation to suit the background knowledge of each student.

2- Training session

We will explain to the students how to interact with the pedagogical agent to get more explanation about the new concepts which are highlighted. In the training session, we want to ensure that the pedagogical agent will appear to help every time a student clicks on a new concept.

We can view the interaction with the pedagogical agent as the interaction with a human tutor who is always ready to explain whenever the student asks. Because the interaction with a pedagogical agent is something new to the children, we need to train them how to do it.

3- Reading session.

Reading from a screen is similar to reading an ordinary book. The exception is that to turn to the next or previous page, the students have to click on the next or previous buttons respectively.

The difference is that the children are not only reading the book but also interacting with the pedagogical agent. In real life, the students need to interact with teachers to get explanation but using the new approach they interact with a pedagogical agent. However, the pedagogical agent is not going to replace human teachers.

4- Book review.

In the book review the students are required to write their opinion about the book, the name of the author and a summary of the book. In addition to that, they are required to give their explanation of the same important words they have seen in the 'General knowledge test'.

The purpose of the book review is to know what they have learned from the lesson and how the pedagogical agent has helped them with the important words.

5- Interview session with the children.

The experimenter wants to know what the children feel when interacting with the pedagogical agent to get an explanation. The interview will be audio recorded to capture what they say. They also may want to tell more about the important concepts in the lesson.

6- Interview session with the teachers.

Teachers are encouraged to give their opinions and suggestions about the pedagogical agent. This information may inform the experimenter about how to improve the pedagogical agent to better suit their teaching practice.

Appendix D

Student allocation into control and experimental groups

This appendix provides information about the students assigned to control and experimental groups. This information is used to compare between the two groups during the SAIC agent evaluation. The following tables also show the allocation of the students according to their reading and writing ability – low, medium and high. The allocation is based on continuous assessment by class teachers in a literacy class.

All students were in CHALCS literacy classes where they learned to read and write with help of teachers.

- Table D.1 shows allocation of students based on class teachers.
- Table D.2 shows details of the allocation.

Table D.1 Teachers and the number of their students who completed both the pre and post-tests

Teacher	No of students
J	4
B	14
S	14

Description of the allocation is as follows:

1- Teacher J. Only 4 out of 8 students in his class participated in both pre and post-tests. This was due to absence and late attendance.

2- Teacher B. 14 out of 18 students in his class participated in both pre and post-tests.

3- Teacher S. 14 out of 18 students in her class participated in both pre and post-tests.

Table D.2 Details of the student allocation

	Group	Ability	Student_ID	Teacher
1	Control	Low	Student_16	B
2	Control	Low	Student_18	B
3	Control	Low	Student_19	S
4	Control	Low	Student_31	S
5	Control	Low	Student_32	S
6	Control	Middle	Student_4	J
7	Control	Middle	Student_9	B
8	Control	Middle	Student_10	B
9	Control	Middle	Student_12	B
10	Control	Middle	Student_20	S
11	Control	Middle	Student_29	S
12	Control	High	Student_3	J
13	Control	High	Student_8	B
14	Control	High	Student_11	B
15	Control	High	Student_21	S
16	Control	High	Student_26	S
17	Experimental	Low	Student_5	B
18	Experimental	Low	Student_6	B
19	Experimental	Low	Student_24	S
20	Experimental	Low	Student_27	S
21	Experimental	Low	Student_30	S
22	Experimental	Middle	Student_2	J
23	Experimental	Middle	Student_7	B
24	Experimental	Middle	Student_13	B
25	Experimental	Middle	Student_15	B
26	Experimental	Middle	Student_22	S
27	Experimental	Middle	Student_23	S
28	Experimental	High	Student_1	J
29	Experimental	High	Student_14	B
30	Experimental	High	Student_17	B
31	Experimental	High	Student_25	S
32	Experimental	High	Student_28	S

Appendix E
Samples of pre-test transcripts

This appendix provides 3 samples of pre-test transcripts written by 3 students of different abilities who interacted with the SAIC agent. The post-test transcripts of the students are presented in Appendix G.

Student_1 - experimental group, high ability

1- Astronaut: A man that goes out into space.

2- Moon: The moon reflects to the sun so the moon could shine

3- Earth: Lots of people live on earth

4- Star: A star is something that shine. It is made of billions of shiny stuff

5- Planet: A planet is like earth. There is lots of different planets in the universe.

6- Space: If you go out to space without suitable cloth you will die

7- Space centre: A space centre is when a astronaut goes out to space. He could tell the space centre what he's doing.

8- Gravity: If we didn't have gravity we will be floating around the world.

9- Oxygen: We need oxygen to breath in our lungs

10- Space shuttle:

11- Telescope: You could look through a telescope to look at the stars.

12- Satellite:

13- Computer: Computer is a game that you could play game on.

14- Training: When you are playing a football match you need to train.

15- Interviewing: When you talk to someone

16- Swimming: You could go swimming every time

17- Taking off:

18- Floating: When something is floating on water.

Student_2 - experimental group, middle ability.

1- Astronaut: An astronaut is a person that been on the planet walks and jumps around.

2- Moon: Moon is white and comes out at night.

3- Earth: The earth is a planet. We live on the earth and there are lots of people who live on the earth.

4- Star: The stars come out at the night and they glow about.

5- Planet: The planet is in the sky and there are twelve planets.

6- Space: Space is far far away and space stays night.

7- Space centre: Space centre is where all space shuttles be.

8- Gravity: Gravity is when you be on the planet and you walk properly.

9- Oxygen: Oxygen has come from the rainforest trees and people chop trees off then we cannot breathe.

10- Space shuttle: Space shuttle is when you blast of in the sky

11- Telescope: you look through it and you see the planets

12- Satellite: Satellite is when you put it on your t.v. works

13- Computer: A computer is something you revise on.

14- Training: Training is when you run or warm yourself up.

15- Interviewing: Interviewing is when you make a chat and you record it on tape.

16- Swimming: Swimming is when you be in the water.

17- Taking off: Taking off means you take off your bed.

18- Floating: Floating is when you stay in the water and don't drown.

Student_6 - experimental group, low ability.

1- Astronaut: Astronaut is someone who travels to the moon. They explore outer space.

2- Moon: The moon is not a light source. The sun reflects onto the moon to give us light. Neil Amstrong was the first person to travel on the moon.

3- Earth: The earth is a planet and we live on it. The earth has lots of countries in it.

4- Star: The stars is something that comes out at night. They are all different sizes.

5- Planet: A planet is something that is in outer space like mars, venus, mercury, Saturn, Uranus, Neptune and so on.

6- Space: Space is something where lots of astronauts visit and it doesn't have any gravity.

7- Space centre: Space centre is something to do with space.

8- Gravity: Gravity is something when you can float up and everything is light just like in a swimming pool.

9- Oxygen: Oxygen is something we have in our bodies to make us fit.

10- Space shuttle: A space shuttle is something that flies up into space.

11- Telescope: A telescope is a bit like a magnifying glass and you can have a look at the stars. Scientists mostly use it.

12- Satellite: The satellite is in space and that makes your television work and the telephones.

13- Computer: A computer is an electrical thing and it's got an Internet on it and you can find stuff on it.

14- Training: When you train that means you want to be something so you have to be good at it.

15- Interviewing: When you interview someone you want to find out about them.

16- Swimming: When you swim you're in a big tube of water.

17- Taking off: If you take off that's like you're going somewhere on a plane.

18- Floating: If you float you lay on top of something.

Appendix F

Post-test briefing scripts

Description: the briefing is required to introduce the SAIC pedagogical agent to the children and to explain how to interact with the agent to ask for help. It will take approximately 5 minutes. The briefing is provided just before the reading session.

Objective: to inform the children the new way to get help when they have problem in understanding new concepts presented in a lesson.

Script:

Good morning. Today you are reading an interesting book on screen. Before you start reading the book, I want to show you how to interact with the computer tutor to get an explanation about the important words in this lesson.

Ok. As you can see on the screen, some words are underlined. It is very important for you to understand these words well in order to understand the lesson.

What to do if you want explanation about the important words? What you have to do is to click on one of the words. Now the computer tutor appears to help you.

The computer tutor will ask you questions to help you understand the word. You have to answer the questions. After asking some questions, the computer tutor will disappear.

Normally you ask your teacher when you need explanation. Today you have to ask the computer tutor. You should ask the computer tutor as many times as you like because it is there to help you. After reading you are required to write what the words mean.

Appendix G
Samples of post-test transcripts

This appendix provides 3 samples of post-test transcripts written by 3 students of different abilities who interacted with the SAIC agent.

Student_1 - experimental group, high ability

1- Astronaut: A person that goes to space.

2- Moon: The moon has no oxygen so you can't breath if you go without any space suit you will die.

3- Earth: Lots of people live on earth. You could breath on earth.

4- Star: The star shine in the night.

5- Planet: There are lots of planets in the galaxy.

6- Space: You can float in space.

7- Space centre: When someone goes into space they could speak to someone on earth.

8- Gravity: Gravity keeps people of the ground.

9- Oxygen: Oxygen helps you breathe if you cannot have oxygen you will die.

10- Space shuttle: When you are in the shuttle you can see the earth.

11- Telescope: You could look through the telescope.

12- Satellite: It is like a camera.

13- Computer: When you are in space the people on the earth tell you where you are going.

14- Training: You need to train before you go out to space.

15- Interviewing: The people interview you before you can go out to space.

16- Swimming: The people who are going out to space have to go underwater for a long time because it is like no gravity underwater.

17- Taking off: It is when the astronauts are taking off into space.

18- Floating: When you float in space it means that there is no weight on you.

Student_2 - experimental group, middle ability.

1- Astronaut: Astronauts are when they go onto the moon.

2- Moon: The moon is in the sky and it is white colour.

3- Earth: The earth is where people live.

4- Star: Stars are little and they are on the moon.

5- Planet: Planets are like an earth and there are 12 different kinds of planets.

6- Space: When you float.

7- Space centre: When you go next to the people to take off in a space shuttle.

8- Gravity: When you're on the earth you can float but when you wear gravity boots then you can't float.

9- Oxygen: Oxygen makes you breathe.

10- Space shuttle: Space shuttle is when you take off and go to space.

11- Telescope: Telescope is when you look through there you could look at everything in the sky.

12- Satellite: Satellite is for your t.v.

13- Computer: Computer is for working on.

14- Training: Training is when you warm up or something.

15- Interviewing: Interviewing is when you go to a meeting.

16- Swimming: Swimming when you go into the water.

17- Taking off: Taking off is like a space shuttle takes off.

18- Floating: When you float in space.

Student_6 - experimental group, low ability.

1- Astronaut: An astronaut is someone who travels to the moon. They travel to the moon by space-craft and they travel to the moon to explore.

2- Moon: The moon is a big, round rock that reflects from the sun. It reflects light so we can see in the dark.

3- Earth: The earth is a planet with around it and it has big and small countries and we live on earth.

4- Star: The stars is one of our sun. It also can give a little bit of light. Most people can follow the stars if they get lost.

5- Planet:

6- Space:

7- Space centre:

8- Gravity:

9- Oxygen:

10- Space shuttle:

11- Telescope:

12- Satellite:

13- Computer:

14- Training:

15- Interviewing:

16- Swimming:

17- Taking off:

18- Floating:

Appendix H

Samples of log files

This appendix provides samples of the log files recorded in the post-test.

Samples of the log files

```
--- NEW CONCEPT = computer
--- Decide an introductory strategy: ‘Informing what the student will learn’.
1 Good. You clicked on an important word.
2 Computer is an important word in this lesson.

--- Decide a diagnose strategy: ‘Asking whether the student knows properties of the new concepts’.
3 Do you know what the word computer means?
4 You think you know what it means. Very good!

--- Decide an activation strategy: ‘Asking a property of a schema’
5 A computer is a kind of .....
6 You're clever. A computer is a kind of electronic device.
7 What does a computer do?
8 You're clever! A computer controls the space shuttle's landing.
9 A computer has operating system. Yes or no?
10 You're clever! A computer has operating system.
11 A computer controls the space shuttle's landing. How?
12 Brilliant! A computer processes data from the moon's surface.

---Decide explanation mode – accretion.
---Decide an accretion strategy: ‘Informing the isa of the new concept’
13 A computer has an operating system. Yes or no?
14 You're clever! A computer has operating system.
15 Computer is a kind of electronic device.
16 Examples of electronic devices are: mobile phone and
telephone.
17 Very good. You know that a computer is a kind of  electronic
device.

---Decide a summarizing strategy: ‘Confirming the knowledge of the student’
18 Good! You have learned about computers.
19 I think you have understood the word computer now.
```

--- NEW CONCEPT = astronaut.

----Decide an introductory strategy: ‘Informing what the student will learn’

```
1 Good. You clicked on an important word.
2 Astronaut is an important word in this lesson.
```

----Decide a diagnose strategy: ‘Asking whether the student is familiar with the new concept’.

```
3 Do you know what the word astronaut means?
4 You think you know what it means. Good!
```

----Decide an activation strategy: ‘Asking property value of a schema’

```
5 An astronaut is a kind of .....
6 You're clever. An astronaut is a kind of profession.
7 What does an astronaut do?
8 You're clever! An astronaut goes to the moon.
9 An astronaut has uniform. Yes or no?
10 You're clever! An astronaut has uniform.
11 An astronaut goes to the moon. How?
12 I don't think an astronaut catches thieves.
```

----Decide explanation strategy – tuning.

----Decide a tuning strategy: ‘Informing the *isa* of the new concept’

```
13 An astronaut is a kind of profession.
14 Examples of professions are: teacher, doctor and policeman.
```

----Decide a summarizing strategy: ‘Informing the properties of the new concept and confirming the knowledge of the student’.

```
15 Very good. You know that astronaut is a kind of profession.
16 An astronaut has uniform.
17 Good! You have learned about astronauts.
```

Appendix I

Questions for student group interview

Introduction:

It is not easy for young children to understand some important concepts in a new lesson. Can we use the computer to help them?

Objective:

To know students' feelings when they learn new concepts:

- If they have to explain the concepts without help from the teacher or the computer.

- If they have to explain the concepts with help from the computer.

What is needed from the teacher:

To ask the following questions with the intent to know what they feel when they learn new concepts on computer.

1- Have you enjoyed interacting with the computer to learn? Why?

2- What did you enjoy most when interacting with the computer?

3- If help is given by the computer, do you think you will understand the words?

4- How can the computer help you?

5- If help is not given by the computer, do you think you will understand the new words in the lesson? Why?

6- What made you unhappy when interacting with the computer?

7- Would you recommend the lesson to your friend?

Appendix-J

Transcripts for student group interview

This appendix provides the transcripts to the questions in Appendix-I. 4 groups answered the questions.

Group 1:

1- Have you enjoyed interacting with the computer to learn? Why?
"Yes – the computer gave me answers".
"Yes – I can ask and the computer give me the answer".
"Yes – it's colourful".
"Yes – it's colourful and has picture astronaut and moon, as well".
"No – I hate reading. Lots of reading. I want to see more pictures".
"No – I want to read my book".

2- If help is given by the computer, do you think you will understand the words?
"Yes".
"No".
"Not sure".

3- How can computer help you?
"Give me answers".
"Tell me what satellite means".
"Tell me the answer when I click".
"Computer must give me the answer, ... quick and easy".

4- If help is not given by the computer, do you think you will understand the new words in the lesson? Why?
"No. I don't know".
"No. Space centre is difficult. I don't know what it means".
"No. Satellite and gravity are too difficult. Don't know how to explain".
"Yes. Dictionary can help me".
"Yes. I can ask the teacher".

5- What made you unhappy when interacting with the computer?
"I clicked but it did not speak".

6- Would you recommend the lesson to your friend?
"Yes."

Group 2:

1- Have you enjoyed interacting with the computer to learn? Why?
"Yes – the computer is interactive".
"Yes – I like to go to space when I grow up".
"Yes – astronaut is a scientist. I like reading about astronaut going to the moon".

"Yes – it gives me answer when I click".
"Yes – I can click and computer tell me what a space shuttle does".
"No – teacher is better than computer...teacher can tell me when I don't know".

2- If help is given by the computer, do you think you will understand the words?
"Yes".
"No".

3- How can computer help you?
"Give me the answer".
"Tell me what to do".
"I want to have more pictures".

4- If help is not given by the computer, do you think you will understand the new words in the lesson? Why?
"No – because I can't ask".
"No – the lesson is not easy".

5- What made you unhappy when interacting with the computer?
"I have to read. I want answer from the computer".
"The program is slow. I want it fast, CBB is fast". // refers to a website.

6- Would you recommend the lesson to your friend?
"Yes".

Group 3:

1- Have you enjoyed interacting with the computer to learn? Why?
"Yes – I can click on gravity and the face appears, and it tells me the answer". // refers to the SAIC agent
"Yes – the computer helps me. I could understand planet, astronaut and stars". // refers to the new concepts
"Yes – I got answer".
"No – I had to read more".
"No – I don't want to ask the computer".

2- If help is given by the computer, do you think you will understand the words?
"Yes".

3- How can computer help you?
"Help me understand astronomy".
"Tell me how astronauts go to space".
"Listen to me and then give me answer".

4- If help is not given by the computer, do you think you will understand the new words in the lesson? Why?
"Yes – I can ask my friends".
"Yes – I can guess what it means".
"No – computer should tell me".

5- What made you unhappy when interacting with the computer?
"I don't want to read. It looks nice but I don't want to read. I want an answer from the computer". // refers to the SAIC agent

6- Would you recommend the lesson to your friend?
"Yes".

Group 4:

1- Have you enjoyed interacting with the computer to learn? Why?
"Yes – I like interacting with computer".
"Yes – computer told me how astronaut went to the moon".
"Yes – computer gave me answer. I can ask computer and teacher as well".
"Yes – but I don't have that program at home".
"No – astronaut is nice. But I already understand what he does in space centre". // refers to the lesson
"No – I enjoy reading my book. My book also has pictures, astronaut, moon, Venus and rocket". // refers to new concepts in the lesson

2- If help is given by the computer, do you think you will understand the words?
"Yes. I can click on the lesson and computer tells me".
"Yes. The program asks me and I know what to answer".
"No. Computer did not help me". // refers to no quick definition was given by the SAIC agent

3- How can computer help you?
"Computer can speak to me and tell me what space shuttle does".
"Computer can tell me quickly and does not ask me".
"I have to read. I want answer from the computer".

4- If help is not given by the computer, do you think you will understand the new words in the lesson? Why?
"Yes – I can guess what planet means".
"Yes – I can type www and use dictionary". // refers to an online dictionary
"No. I will not understand".

5- What made you unhappy when interacting with the computer?
"My parents can tell me better".

6- Would you recommend the lesson to your friend?
"Yes".

Appendix K

Questions for teacher interview

This appendix provides a list of questions to ask the teachers who participated in the evaluation study.

The questions:

1- Do you think it is possible to integrate the computer tutor into your teaching practice?

2- Would you integrate the computer tutor if it was provided?

3- How would you use the computer tutor?

4- What did you like about this study?

5- What did you not like in the study?

Appendix-L

Transcripts for the teacher interview

This appendix provides transcripts for the teacher interview. 4 teachers answered the questions presented in Appendix-K. They had participated in several discussions during the various stages of the agent development and tested it during the formative evaluation of the agent.

Teacher A:

1- Do you think it is possible to integrate the computer tutor into your teaching practice?

"I believe it is possible to integrate the software. I can't see any problem with that. As you know, we teachers need such software to help the students."

2- Would you integrate the computer tutor if it was provided?

"Certainly but I'm not sure how information about my students can be coded into the SAIC sofware."

3- How would you use the computer tutor?

"It is better than explanation from a dictionary because the software considers the prior knowledge of the student as I always do. I'll recommend my students to refer to explanation given by the SAIC sofware."

4- What did you like about this study?

"Your great effort to understand my students, I mean to to understand why it is so difficult for them to understand the lessons. We are here to help them but, as you can see, we don't have time to answer all their questions. You also recommended a wonderful solution; explanation generated by your software"

5- What did you not like in the study?

"My students will have to read all the explanations in addition to the lesson. I think it will be more interesting if you programme it to speak"

Teacher B:

1- Do you think it is possible to integrate the computer tutor into your teaching practice?

“sure – in my reading classes.”

2- Would you integrate the computer tutor if it was provided?

“My students enjoyed interacting with the SAIC software. I would integrate it.”

3- How would you use the computer tutor?

“I will use it as a problem solver when my students don’t know what a new word means. Better they interact with the SAIC software than continue reading without understanding.”

4- What did you like about this study?

“It’s a great effort to study what my students understand and what they don’t. We’ll have a better idea how to teach them.”

5- What did you not like in the study?

“My concern is the technical part of the software, I do not think teachers have the expertise to put all the information about CHALCS students and lessons we teach into the computer. The multimedia software is great.”

Teacher J:

1- Do you think it is possible to integrate the computer tutor into your teaching practice?

“We need to have that kind of software in our computers. It will help both the teachers and students.”

6- Would you integrate the computer tutor if it was provided?

“Yes, it will be an interactive session in my reading classes.”

7- How would you use the computer tutor?

“As a means to provide links between what they learned and what what they need to learn in new lessons.”

8- What did you like about this study?

“You’ve addressed a real problem faced by the teachers and students in reading classes. As has been emphasized by the software, my students must think how to explain. The software is also very attractive.”

9- What did you not like in the study?

“The difficult process of putting information into the SAIC software. If we’re going to use it in our classes, we certainly need a way to do it faster.”

Teacher S:

1- Do you think it is possible to integrate the computer tutor into your teaching practice?

“It is possible to integrate the software. We have a variety of software in our computers, in addition to using books. The software is a helpful learning material.”

2- Would you integrate the computer tutor if it was provided?

“Yes. I think all CHALCS teachers would.”

3- How would you use the computer tutor?

“We can use it as interactive whiteboard, class displays and discussions with students.”

4- What did you like about this study?

“The multimedia software is interactive and colourful.”

5- What did you not like in the study?

“The SAIC software should have voice. My students should also be guided how to pronouce the new words. I’m sure you know online dictionaries that have buttons, students can click and listen.”

Appendix M

Analysis of the student explanations

* The letters C and E after the student ID refers to control and experimental, respectively.

Student	Concept	Pre-test	Post-test	Schematic	Improve
Student_1.E	1- Moon	Reflects sunlight to shine	Use space suit to breath on moon	Process	Specific
	2- Earth	Has people living on it	Has people living on it. Can breath	Structure	Elaborate
	3- Space centre	Astronaut tell space centre	Astronaut can speak to someone on earth	Structure	Elaborate
	4- Computer	To play game	To tell astronaut where to go	Function	Specific
	5- Training	Train to play football	Train before going to the moon	Function	Complete
	6- Interviwing	Talk to someone	Before you can go to space	Function	Specific
	7- Floating	Float on water	Float in space, no weight	Function	Elaborate
Student_2.E	1- Astronaut	Has been on planet, jump around	Go to the moon	Function	Specific
	2- Planet	In the sky, 12 planets	Like an earth, 12 planets	isa	Specific
	3- Space-centre	Where all the space shuttles be	Where you go to take off in space shuttle	Structure	Complete
	4- Oxygen	From rainforest, chop tree, cannot breath	Make people breath	Function	Specific
	5- Space-shuttle	Blast of in the sky	Take off and go to sky	Function	Elaborate
	6- Telescope	Look through it and see planet	Look through it and see everything in sky	Function	Elaborate
	7- Floating	Stay in water, not drown	Float in space	Function	Complete
Student_3.C	1- Astronaut	Goes to space	Goes to moon	Function	Elaborate
	2- Earth	Big planet, we live on it	Earth moves around the sun	Function	Specific
	3- Star	A small dot in sky, white	A kind of sky object	isa	Elaborate
	4- Gravity	Makes us stand up, not floating	Gives weight to object	Function	Elaborate
Student_4.C	1- Astronaut	Wear space suit in sky	Wear space suit, travel in space in space shuttle	Process	Elaborate
	2- Star	In the sky	Spread out in the sky, can see at night	Structure	Elaborate
	3- Space	Can see moon in space	Where all the planets are, cannot breath	Structure	Complete
Student_5.E	1- Astronaut	Discovers things on planet	A person that explores moon and space	Function	Complete
Student_6.E	1- Astronaut	Someone who travels to the moon, explore space	Explores moon using spacecraft	Structure	Complete

Student	Concept	Pre-test	Post-test	Schematic	Improve
Student_7.E	1- Astronaut	Travel to space and discover things	Travel by space shuttle to moon and finds things out	Function	Complete
	2- Star	Lots of star in the sky	Space, in the sky, twinkles, gives us light	Structure	Elaborate
	3- Space	In the sky	In the sky and has planets	Structure	Complete
	4- Telescope	To look through	To look through to look at stars	Function	Elaborate
	5- Computer	Is an electronic tv	Is an electronic thing to work on	isa	Elaborate
Student_8.C	1- Computer	Is a machine	Is a smart brainy machine	isa	Elaborate
	2- Training	Try to do something	Try to do something like football	Function	Elaborate
	3- Interviewing	Interview football player	Interview at space centre	Function	Elaborate
	4- Space centre	A place having space teacher, costume, men	A place to do interview agout space and other things about space	isa	Elaborate
Student_9.C	1- Gravity	Keep us on the floor, will float without it	Keep you down, make body heavy	Function	Elaborate
	2- Space shuttle	Take astronaut into space	Take astronaut into space because astronaut want to go into space	Function	Complete
Student_10.C	1- Astronaut	Goes to space in a rocket	Goes into space to find things out	Function	Elaborate
	2- Earth	A ball, blue, green, turn around	Is a planet, people live on it	isa	Elaborate
Student_11.C	1- Oxygen	Oxygen from plant	Astronaut uses space suit to have oxygen to breath	Process	Elaborate
	2- Training	When people train and become stronger	When people practise to do something, for example to be doctor or any job	Function	Elaborate
	3- Floating	Like something light flot on top of water	When people fly and like you float on top of water	Function	Elaborate
Student_12.C	1- Space	Above earth	Black and dark, where astronaut is	Structure	Complete
	2- Telescope	To discover things far away	To make things bigger and you can see things from far away	Function	Elaborate
Student_13.E	1- Space shuttle	Have people inside to go to space	Have people, have engine to go to space	Structure	Elaborate
	2- Satellite	Get signal to watch tv	Get signal to see what happens in space	Function	Complete

Student	Concept	Pre-test	Post-test	Schematic	Improve
Student_14.E	1- Planet	Hotter, colder than earth	In space, have planets	Structure	Elaborate
	2- Space centre	For scientists to figure out what happens in space	People go to be astronaut and do training	Function	Elaborate
	3- Gravity	Make people float without it	Make you stay on floor not floating	Function	Complete
	4- Telescope	To look into space	To look at stars in space	Function	Elaborate
	5- Computer	To play game, give information	In space centre, connect people in space	Function	Specific
	6- Training	Learning to be better	Train in space centre to be astronaut	Function	Specific
Student_15.E	1- Moon	Appears at night	Visited by astronaut	Function	Specific
	2- Gravity	Without it you will float	What keeps you on the ground	Function	Elaborate
	3- Oxygen	We need oxygen	We breath in oxygen	Function	Elaborate
Student_16.C	0				
Student_17.E	1- Earth	Where we live	The planet where we live on	isa	Complete
	2- Gravity	To walk on	To keep people on the ground, not floating	Function	Complete
	3- Oxygen	We breath in oxygen, earth has oxygen	Astronaut use space suit, it has oxygen to breath	Process	Complete
	4- Computer	Help people do many things, job, learn	Help people do research on space	Function	Elaborate
	5- Training	Train using computer and other things	To do something, have to do a test	Process	Elaborate
Student_18.C	1- Astronaut	Who finds out about earth and other planet	Who travels to the moon	Function	Elaborate
Student_19.C	1- Space centre	Interview astronaut	People go to space centre for an interview	Function	Complete
Student_20.C	1- Astronaut	In space	Into space, on the moon	Structure	Elaborate
Student_21.C	1- Moon	White, shiny, comes out at night	Sphere shape, white	Structure	Specific
	2- Earth	Sphere shape	Blue, sphere shape, have countries	Structure	Elaborate
Student_22.E	1- Computer	To play game	Can see astronaut	Function	Specific
Student_23.E	1- Astronaut	Goes in space	Floats in space, walks on the moon	Function	Complete
Student_24.E	1- Earth	In space	Big, round	Structure	Elaborate

Student	Concept	Pre-test	Post-test	Schematic	Improve
Student_25.E	1- Moon	White, sometimes full-moon	Beautiful, white, have astronauts walk on it	Structure	Complete
	2- Star	Glows in the sky	Comes out at night, look closer if we see using telescope	Function	Complete
	3- Oxygen	We breathe in, could not live without it	Helps us breath, from plant, in space no plant, so astronaut uses mask	Function	Complete
Student_26.C	0				
Student_27.E	1- Earth	Round	Far from the moon	Structure	Complete
Student_28.E	1- Astronaut	Goes to space using rocket	Goes to space, goes to moon wearing special costume	Process	Complete
	2- Moon	High up in sky	Have astronaut walking on it	Structure	Specific
	3- Earth	Small	Can see earth from space	Structure	Complete
	4- Oxygen	Use oxygen underwater	Use oxygen in space to make you breath	Function	Elaborate
Student_29.C	1- Star	Produces light	Produces light in the sky	Function	Elaborate
Student_30.E	0				
Student_31.C	0				
Student_32.C	0				