

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Cognitive Trait Model for Persistent and Fine-Tuned Student Modelling In Adaptive Virtual Learning Environments

A thesis presented in partial fulfilment of the
requirements for the degree of

Master of Information Science

Department of Information System
Massey University
Palmerston North,
New Zealand

Taiyu Lin
2003

Abstract

The increasing need for individualised instructional in both academic and corporate training environment encourages the emergence and popularity of adaptivity in virtual learning environments (VLEs). Adaptivity can be applied in VLEs as adaptivity content presentation, which generates the learning content adaptively to suit the particular learner's aptitude, and as adaptive navigational control, which dynamically modifies the structure of the virtual learning environment presented to the learner in order to prevent overloading the learner's cognitive load.

Techniques for both adaptive content presentation and adaptive navigational control need to be integrated in a conceptual framework so their benefits can be synthesised to obtain a synergic result. Exploration space control (ESC) theory attempts to adjust the learning space, called exploration space, to allow the learners to reach an adequate amount of information that their cognitive load is not overloaded. Multiple presentation (MR) approach provides guidelines for the selection of multimedia objects for both the learning content presentation and as navigational links.

ESC is further formalised by including the consideration of individual learner's cognitive traits, which are the cognitive characteristics and abilities the learner relevant in the process of learning. Cognitive traits selected in the formalisation include working memory capacity, inductive reasoning skill, associative learning skill, and information processing speed. The formalisation attempts to formulate a guideline on how the learning content and navigational space should be adjusted in order to support a learner with a particular set of cognitive traits.

However, in order to support the provision of adaptivity, the learners and their activities in the VLEs need to be profiled; the profiling process is called student modelling. Student models nowadays can be categorised into state models, and process models. State models record learners' progress as states (e.g. learned, not learned), whereas a process model represents the learners in term of both the knowledge they learned in the domain, and the inference procedures they used for completing a process (task). State models and process models are both competence-

based, and they do not provide the information of an individual's cognitive abilities required by the formalisation of exploration space control. A new approach of student modelling is required, and this approach is called cognitive trait model (CTM).

The basis of CTM lies in the field of cognitive science. The process for the creation of CTM includes the following subtasks. The cognitive trait under inquiry is studied in order to find its indicative signs (e.g. sign A indicates high working memory capacity). The signs are called the manifests of the cognitive trait. Manifests are always in pairs, i.e. if manifest A indicates high working memory capacity, A's inverse, B, would indicate low working memory capacity. The manifests are then translated into implementation patterns which are observable patterns in the records of learner-system interaction. Implementation patterns are regarded as machine-recognisable manifests. The manifests are used to create nodes in a neural network like structure called individualised temperament network (ITN). Every node in the ITN has its weight that conditions and is conditioned by the overall result of the execution of ITN. The output of the ITN's execution is used to update the CTM.

A formative evaluation was carried out for a prototype created in this work. The positive results of the evaluation show the educational potential of the CTM approach. The current CTM only cater for the working memory capacity, in the future research more cognitive traits will be studied and included into the CTM.

Table of Content

CHAPTER 1: INTRODUCTION	1
1.1: Custom Made Education.....	1
1.2: Thesis Layout	2
CHAPTER 2: ADAPTIVE HYPERMEDIA SYSTEMS	7
2.1: Introduction	7
2.2: Multimedia and Learning.....	7
2.3: Hypermedia.....	9
2.4: Adaptive Hypermedia	10
2.5: Summary	11
CHAPTER 3: ADAPTIVE NAVIGATION	13
3.1: Introduction	13
3.2: Navigational Support	13
3.3: Techniques for Adaptive Navigation	14
3.4: Summary	16
CHAPTER 4: ADAPTIVE CONTENT PRESENTATION.....	17
4.1: Introduction	17
4.2: Adaptive Content.....	17
4.3: Techniques of Adaptive Content Presentation	18
4.4: Summary	21
CHAPTER 5: INTEGRATED THEORIES OF ADAPTIVITY	22
5.1: Introduction	22
5.2: Multiple Representation Approach	22
5.2.1: Multimedia Object Selection	23
5.2.2: Navigational Object Selection	24
5.2.3: Integration of Multimedia Object	26
5.3: Exploration Space Control	28

5.4: Summary	30
CHAPTER 6: EXPLORATION SPACE CONTROL FORMALISATION	31
6.1: Introduction	31
6.2: The Definition and Selection of Cognitive Traits	32
6.3: Exploration Space Control Elements.....	35
6.4: Working Memory Capacity	38
6.4.1: Formalising with consideration of working memory	39
6.5: Inductive Reasoning Skill	41
6.5.1: Formalising with consideration of inductive reasoning skill.....	42
6.6: Information Processing Speed.....	43
6.6.1: Formalising with consideration of information processing speed	43
6.7: Associative Learning Skill.....	45
6.7.1: Formalising with consideration of associative learning skill.....	45
6.8: Summary	47
CHAPTER 7: STUDENT MODEL	49
7.1: Introduction	49
7.2: Student Modelling.....	49
7.3: Application of Student Model.....	50
7.4: Modelling Competence State	52
7.4.1: Overlay Student Model.....	53
7.4.2: Differential Student Model	54
7.4.3: Perturbation Student Model.....	54
7.4.4: Constraint-based Student Model.....	55
7.5: Modelling the Process.....	55
7.6: Student Model Diagnosis.....	58
7.6.1: Stereotypical Modelling.....	58
7.6.2: Coverage Measure	59
7.6.3: Performance Measure	60
7.6.4: Problem-solving State Tracing	60
7.6.5: Path Finding.....	61
7.6.6: Skill Tracing	61
7.6.7: Case-based Reasoning	61
7.6.8: Expert System	62
7.7: Constructivism and Student Modelling.....	62

7.8: Summary	64
CHAPTER 8: COGNITIVE TRAIT MODEL	65
8.1: Introduction	65
8.2: New Perspective for Student Modelling	65
8.3: New Approach for Student Modelling.....	66
8.4: Architecture For Incorporating Cognitive Trait Model	68
8.5: Summary	71
CHAPTER 9: LEARNING OBJECTS AND RELATION-BASED BROWSING PATTERN ANALYSIS	72
9.1: Introduction	72
9.2: Learning Objects	72
9.3: Navigational Pattern and Relation-Based Analysis.....	73
9.3.1: Content-less and Content-based Navigational Pattern Analysis.....	73
9.3.2: Relation-based Browsing Pattern Analysis.....	74
9.4: Learning Object Relations	75
9.5: Summary	80
CHAPTER 10: TRAIT ANALYSIS	82
10.1: Introduction	82
10.2: Trait Analyser	82
10.2.1: Structure of Trait Analyser	82
10.3: Analysis of Working Memory	84
10.4: Summary	88
CHAPTER 11: MANIFESTS AND IMPLEMENTATION PATTERNS	89
11.1: Introduction	89
11.2: Manifests.....	89
11.3: Implementation Patterns	91
11.3.1: Implementation Pattern for Navigational Linearity.....	92
11.3.2: Implementation Pattern for Reverse Navigation	94
11.3.3: Implementation Pattern for Excursions	97
11.3.4: Implementation Pattern for Simultaneous Tasks.....	98

11.3.5: Implementation Pattern for Retrieval of Information from Long-Term Memory	100
11.3.6: Implementation Pattern for Long Sequence of Calculation or Procedures	101
11.3.7: Other Manifests not Suitable for Implementation	102
11.4: Summary	103
CHAPTER 12: INDIVIDUALISED TEMPERAMENT NETWORK	104
12.1: Introduction	104
12.2: Individualised Temperament Network.....	104
12.3: Simulation for Selecting Appropriate Gradient Constant.....	108
12.4: Summary	116
CHAPTER 13: PROTOTYPE WEB-BASED TUTORIAL WITH COGNITIVE TRAIT MODEL	117
13.1: Introduction	117
13.2: Structure of the Domain.....	117
13.3: Description of the Tutorial.....	118
13.4: Technical Details of the Prototype	121
13.4.1: The Interface Module.....	121
13.4.2: The Student Behaviour History	122
13.4.3: The Trait Analyser	122
13.4.4: Student Performance Model	123
13.5: Example of Trait Analysis	123
13.5.1: Navigational Linearity	124
13.5.2: Reverse Navigation.....	125
13.5.3: Excursions.....	125
13.6: Formative Evaluation of the Prototype	128
13.6.1: Participants of the Evaluation	128
13.6.2: Evaluation Questionnaire and Summary	128
13.6.3: Discussion of the Evaluation	132
13.7: Summary	133
CHAPTER 14: DISCUSSION AND FUTURE WORK	134
14.1: Discussion	134
14.2: Future Improvement	135
REFERENCES	138

APPENDIX 1: STATISTICAL RESULT OF THE SIMULATION	154
APPENDIX 2: EVALUATION OF COGNITIVE TRAIT MODEL.....	155
Brief Introduction to Concepts Used in This Evaluation	155
Introduction to the Prototype	155
Structure of the Domain (Optional for Evaluators to Fully Read this Section)	156
Brief Description of the Tutorial	157
Technical Details of the Prototype.....	160
The Interface Module.....	160
The Student Behaviour History	161
The Trait Analyser	161
Student Performance Model	162
Example of Trait Analysis (Optional for Evaluator).....	162
Navigational Linearity (Optional for Evaluators).....	163
Reverse Navigation (Optional for Evaluators)	164
Excursions (Optional for Evaluators)	164
Evaluation Questions - Cognitive Trait Model	167

Chapter 1: Introduction

In the increasing heterogeneous student population in both the academic and industrial training environment, the need of customised instructional design is more and more obvious. There are a variety of factors, including the past experience, cognitive abilities, personal preference and so forth, that influence the knowledge transfer, acquisition, and construction during the learning process.

The ideal learning environment, as described by Glibert and Han (1999), consists of many (or more exactly “infinite”) instructors, each having their unique teaching styles, available for every learners to choose, and the learners could choose the instructor that perfectly matches their own learning styles. Forbus & Feltovich (2001) even added that the instructors or assistants should be at the every learner’s elbow whenever learners are ready to learn, and for however long that takes.

Cronbach and Snow’s (1989) Aptitude-Treatment Interaction theory indicates that the instructional strategies (the treatments) are effective for different individuals because of their specific abilities. ATI therefore suggested that the best learning results when the instruction matches the learner’s aptitude.

1.1: Custom Made Education

Computer-based educational systems try to overcome the difficulties of providing the “ideal” learning environment that arise due to real-life limitations, by the means of providing customisable learning environments. Customisation, a prominent idea in

modern marketing, applies perfectly to education as well with the assistance of the computer-based technologies. In the virtual learning environments, customisation could either be within users' control (adaptable), or achieved without users' intervention (adaptive). Nonetheless, the generalised term "adaptivity" supersedes both of them and is used to cover the idea in the following discussion. Systems equipped with adaptivity have been proven more effective and efficient than traditional non-adaptive systems (De Bra, 1999). Examples of the systems built with the idea of adaptivity include ELM-ART-II, CALAT, InterBook, AST, Medtec, DCG, 2L670, and C-Book (as quoted by Brusilovsky, 1998).

Currently, there are many educational system design techniques and methods for supporting learning in adaptive virtual learning environments (VLEs). For example, adaptive hypermedia systems (AHSs) support the navigation that the learners can follow, and tailor information to be presented (Boyle & Encarnacion, 1993) according individual's need. VLEs that employ problem sequencing allow the learners to develop their understanding in a granular manner (Halff, 1998), and some simulation-based VLEs allow the users to specify and control simulation parameters thus making it easier to interpret the simulation results (e.g. Eliot & Woolf, 1995).

1.2: Thesis Layout

Adaptive hypermedia systems (AHSs) will be discussed next in this thesis. Many of the current educational VLEs are implemented as AHSs due to the widespread use of Internet as a medium of learning. The educational value of multimedia, and the

evolutional background of AHSs will be briefly covered to provide an historical perspective of adaptive VLEs.

Most of the techniques used in AHSs can be categorized into either adaptive navigational support or adaptive content presentation (Brusilovsky, 1998). Adaptive navigational support works at the link level, and is usually implemented as visual cues attached to links to express suggestions. Whereas, the adaptive content presentation works at the (learning) content level, and tailor the presentation of information according to learners' competence level, preference, goal, and so on. Many existing adaptive techniques will be introduced in this chapter to provide an overall understanding on how those techniques can be integrated into VLEs and their potential contribution to the curriculum.

However, all these adaptive techniques have their strengths and weaknesses, and they do not identify with each other as a means to support learning in an adaptive environment. There exists the necessity of a framework to provide the integration and re-organisation of the existing methods and to provide suggestive guidelines on how to utilise adaptive techniques to achieve the best results. Multiple Representation (MR) approach provides integrating frameworks on how the multimedia objects should be incorporated into adaptive VLEs (Kinshuk et al., 1999). Whereas, Exploration Space Control (ESC) supplies the theoretical background on guiding the implementation of adaptive techniques, using different control levels (e.g. difficulty levels, size of exploration space, and so on) to support individualised exploratory learning experiences (Kashihara et al., 2000).

Exploration Space Control Formalisation, ESCF (Lin, 2002), is an attempt to bring the benefits of ESC theory and MR approach one step closer to educational system developers. Although both ESC and MR can be applied in a wide-range of contexts, ESCF provides an exemplary tactical plan. The benefits, and the procedures of ESCF will be discussed straight after the introduction of MR and ESC.

One of the most important components in any adaptive VLEs is the student model. Student model makes available the information necessary for adaptation to the VLE. Attributes of students are collected, analysed, and stored in student models. Most of the student models focus on students' competence levels (hence called Competence Models), and they could be implemented as state models (Brusilovsky, 1994; Staff, 2001; Urban-Lurain, 1996; Stankov, 1996) or process model (El-Sheikh, & Sticklen, 1998). However, there is a growing consensus in the student (user) modelling community that modelling competence alone does not provide sufficient adaptive ability to a VLE that is needed in an increasingly heterogeneous demography of the learners (Lovett et al., 2000; Barnard, 1993; Green, 1994; Tweedie & Barnard, 1992). Therefore, a new approach of student modelling is introduced in this thesis; it is called the Cognitive Trait Model (CTM). CTM concentrates on those student attributes (cognitive traits) that are persistent for a long duration, and consistent for a variety of the tasks. The role of CTM is not to replace existing competence models but to supplement in order to provide fine-grained adaptivity. Both state model and process model, and their implementation will be examined in detail before the discussion of CTM in order to (1) provide an overview of student modelling, and (2) show the necessity for the new approach of CTM.

Once the theoretical background of CTM has been covered, the focus is then turned to the implementation. Chapter 9 talks about a rising concept (paradigm) for the construction of on-line learning materials in a manageable, scalable and reusable manner. This new concept is named “learning object”. The Learning Technology Standard Committee’s (LTSC) standard learning object Metadata version 1 (LOM, 2002) provides a guideline for creating learning objects, and ensures compatibility of learning objects. Learning object is used as the building block for an exemplary prototype that demonstrates how the CTM could be integrated in VLEs. Relation is one of the metadata of learning objects defined in the LOM v1 standard. A relation-based browsing pattern analysis is developed in this thesis to cater for the need of analysing learners’ actions and inferring their cognitive capacities. Learning objects and relation-based browsing pattern analysis is discussed in detail in chapter 9.

Chapter 10 further explains the function and structure of one of CTM’s components, the Trait Analyser. One of the cognitive trait, working memory capacity, that is important in many aspects of the learning process, is selected as an example to carry out the analysis and is also included in chapter 10. The result of the trait analysis is summarised into units called manifests, and manifests need to be further translated into implementation patterns for implementation reason. Manifests and implementation patterns are presented in chapter 11.

Individualised Temperament Network is an important component of the Trait Analyser. It resembles the structure of an artificial neural network, and uses a method called retrapoling to adjust itself in order to represent the cognitive ability of the learner. Individualised Temperament Network is discussed in chapter 12.

Chapter 14 presents a simple prototypical web-based tutorial of recursion programming that incorporates a CTM with limited feature. Finally, a summary is given along with the discussion of further improvement possibilities of CTM to conclude this thesis.

Chapter 2: Adaptive Hypermedia Systems

2.1: Introduction

Multimedia has been used in education for years. The educational content presented through radio and television used various media for representation of the content. The rapid advancements in computers expand the value of multimedia to a new dimension. The ability to present information with interactivity sets the new landmark for education. Hypermedia is the combination of the hypertext and multimedia. Furthermore, the adaptive hypermedia systems remove the limitation that only the highly-motivated students can achieve well in computer-based learning environments (Brusilovsky, 1998).

2.2: Multimedia and Learning

Multimedia literally means the use of more than one media element. It could be the combination of text, sound, graphics, animation, video or any other media elements that may appear in the future. The recent advancements in computer technology have made it possible to integrate multimedia in computers for educational purposes.

The use of multimedia in learning environment aims to help the learners. This assumption is supported in the literature reviewed by Najjar (1996): “*Meta-analyses (Bosco, 1986; Fletcher, 1989, 1990; Khalili & Shashaani, 1994; Kulik, Bangert, & Williams, 1983; Kulik, Kulik, & Bangert-Drowns, 1985; Kulik, Kulik, & Cohen, 1980; Kulik, Kulik, & Schwalb, 1986; Schmidt, Weinstein, Niemic, & Walberg, 1985)*

examined over 200 studies that compared learning information that was presented in a traditional classroom lecture to learning the same information presented via computer-based multimedia instruction. ... Over this wide range of students and topics, the meta-analyses found that learning was higher when the information was presented via computer-based multimedia systems than traditional classroom lectures". Another important finding by Najjar was that the learning time is relatively shorter by multimedia instruction than by traditional methods. Research also shows that people remember only 20 percent of what they see, 30 percent of what they hear. When they see it and hear it, they remember 50 percent. When they see it, hear it, and interact with it, they remember 80 percent (Coorough, 1998).

Coorough (1998) added another important point that *"some people are very visual. They learn or are inspired by reading, seeing, or visualizing. Others are quite auditory and learn best by listening. Finally, there are kinaesthetic learners who learn by doing"*.

Symbol systems theory, developed by Salomon (1977), intended to explain the effects of multimedia on learning. It stated that, *"symbol systems of media affect the acquisition of knowledge in a number of ways. First, they highlight different aspects of the content. Second, they vary with respect to the ease of recoding. Third, specific coding elements can save the learner from difficult mental elaborations by overtly supplanting or short-circuiting specific elaboration. Fourth, symbol systems differ with respect to how much processing they demand or allow. Fifth, symbol systems differ with respect to the kinds of mental processes they call on for recoding and*

elaboration. Thus, symbol systems partly determine who will acquire how much knowledge from what kinds of messages" (Salomon, 1977).

2.3: Hypermedia

The combination of multimedia and the hyperlinking ability to navigate freely instantiates the creation of hypermedia. Multimedia objects, often used to denote linkage to another web pages, can be more expressive and meaningful than just plain text. Hypermedia, in a sense, subsumes the meaning of hypertext because text is also a kind of media.

Symbols are often more intuitively recognisable than a textual caption; the most common example would be the previous and next arrow in an Internet browser. Geology Lab (Andris & Stueber, 1994), a hypermedia system teaching introductory geology uses "iris open" and "iris close" to represented the hierarchically deeper (more specific), and higher (more general) location of the pages in a topic.

Information presented in a hypermedia system has a node-like structure and the navigation does not need to be linear (Eklund & Zeiliger, 1996). Jacobs (1992) explained that *"the principal attraction of hypermedia is that it lends itself to naturally non-sequential educational approaches, since it encourages the free association characteristics of human thought"*.

Najjar (1996) also gave following reasons that make hypermedia learning more effective:

1. Instructional method: Computer-based instruction forces the instructional designer to better organize and structure the learning material.
2. Interactivity: Computer-based multimedia instruction tends to be more interactive than just traditional classroom teaching.
3. Control of learning pace: The learners have greater degree of control on the pace of learning.

Knowledge structure in hypermedia is divided into internal and external structure by Eklund & Zeiliger (1996), who described the internal structure as the representation of the content through the sequencing of the nodes and links, whereas the external structure in a hypermedia system (HMS) is the expert knowledge being embedded so that the “*HMS is capable of dynamically altering content or links to suit the needs of individual user*”. The external structure includes some sort of knowledge representation scheme and the student model to achieve its individualised tailoring ability.

In information retrieval systems, hypermedia provides the opportunity of incidental learning; user can explore the information space and discover information that could never be requested in a formal query (Brusilovsky, 1996).

2.4: Adaptive Hypermedia

Hypermedia systems are inherently information systems with visual navigational aids that can be used to move through a hyper-linked information space. There are hypermedia systems that have the ability to adapt their content according to the users' characteristics, and thus they are defined as externally structured. These systems are

generally the combination of hypermedia systems (hyper-link, multimedia), the intelligent tutoring systems (adaptivity), the integration of student-centred learning (hypermedia, browsing, self-directed), and the tutor-centred learning (the intelligent tutor, directive, guidance) (Jonassen & Wang, 1990; Costa Pereria et al., 1991).

Examples of adaptive hypermedia systems include the ANATOM-TUTOR (Beaumont, 1994), the Lisp tutor, ELM-ART II (Weber & Specht, 1997), and InterSim (Kinshuk et al., 1999).

The adaptive characteristics improve the usability of the hypermedia system. Without adaptivity, the hypermedia systems would present the same material, with the same set of links, to all learners. It could work well if the intended user group has the same/similar learning characteristics, but it is rarely the case in real learning environment. Without adaptivity, hypermedia systems will mimic the traditional classroom, with one-to-many interaction.

Adaptive hypermedia systems, with the ability to change the content or structure of the links according to the student's needs, provide similar situation as there are many instructors available for each individual student, *“the learner's chances of doing well in this classroom would appear to be significantly better than in a classroom with one instructor because each learner would adapt to the instructor(s) that would facilitate his/her learning style”* (Glibert & Han, 1999).

2.5: Summary

The application of multimedia in education has been proven to be a success (Coorough, 1998). The right type of media facilitates adequate knowledge acquisition. The combination of multimedia and hypertext forms the hypermedia. Hypermedia inherits the strengths of multimedia and at the same time, supports human learning with its non-linear presentation possibilities. Inclusion of adaptivity makes the hypermedia systems more usable and suitable for a wider range of users. Brusilovsky (1998) further distinguished two categories of features that could be adapted in adaptive hypermedia systems: content adaptation and navigational adaptation. These two adaptation features will be discussed subsequently in the next two chapters of this thesis.

Chapter 3: Adaptive Navigation

3.1: Introduction

Navigational adaptation focuses on building a sensible navigational path for every individual student. It supports the learning process by providing a customised learning space for each learner. This chapter of the thesis talks about navigational support, and the techniques used for navigational support.

3.2: Navigational Support

Navigational support in computer-based learning environments has been studied intensively because of its significant effect on the student learning process (for example, Brusilovsky, 1996; Specht et al., 1997; Eklund & Zeiliger, 1996). Adaptive navigation plays a supportive role and provides suggestions on how to go through the learning materials in a way that could suit the student's learning style. This is often accomplished by providing guidance on the available paths in a manner that is both pedagogically sound and customised to each individual student.

The navigational support works at the link level. The available links determine where the student is able to go, and thus the reachable information space. By adding adaptivity into navigational support, the information space can be tailored to each student, and the extent of this tailoring is mainly judged by the system's knowledge about the current states of the student. This mechanism prevents the student from

getting lost in the information space and helps the student to get the most out of the domain content available.

3.3: Techniques for Adaptive Navigation

Navigational adaptivity is often presented as visual cues (Andris & Stueber, 1994), and can be implemented as direct guidance, adaptive hiding or re-ordering of links, link annotation, or map adaptation (Eklund & Zeiliger, 1996).

A simple annotation could be highlighting the previously visited link(s). Annotation, generally, is to augment a link with some form of hint/comment that tells about the link's destination. Annotation could be in textual form (Zao et al., 1993) or as visual cues (Brusilovsky, 1994). Examples like ELM-ART (Brusilovsky, 1996) and Interbook (Brusilovsky & Schwarz, 1998) use coloured dots and arrows as annotations. ISIS-Tutor (Brusilovsky & Pesin, 1998) colours the anchor text, and AHA (De Bra & Calvi, 1997; De Bra & Calvi, 1998) even allows the students to choose the colour scheme by themselves.

Muan et al. (2001) brought the annotation accuracy one step further by declaring a technique to annotate a link with a percentage of effort spent, with regard to the student's current reading speed, which is determined by a pre-test when the student first logs in. It attempts to solve the problem of determining whether a particular node is read with attention and serious effort, or just simply clicked. This is, however, still an assumption that the pre-tested speed is the actual speed in reading for the students.

Direct guidance works out the best next node for the student, based on the information available in the student model. It also involves the inference and consultation of the expert model, which contains the pedagogical information and rules. This technique aims at helping the student to find an optimal path through the hyperspace of learning material. The selected node is recommended to the student but rather than strictly forced, students are generally allowed to ignore the recommendation if they want to. This mechanism can be very helpful when the system detects an un-met prerequisite and recommends the necessary preparation to be done. Web-based systems like ELM-ART-II (Weber & Specht, 1997), CALAT (Nakabayashi et al., 1997), InterBook (Brusilovsky et al., 1997), AST (Specht et al., 1997), Medtec (Eliot, Neiman & Lamar, 1997), and DCG (Vassileva, 1997) uses direct guidance in various forms.

Link re-ordering sorts the available links in the order of relevance according to the values in student model. It works like a filter. The most relevant link is displayed on top while the links with lower relevancy are displayed on the bottom or they are hidden. The hiding takes away those links that are less useful (or more precisely, less useful by the system expert's point of view), and thus reduces the amount and complexity of the information contained in the page. "*The cognitive load of student required for the re-ordered page is less than if it is not, and speed up the learning process*" (De Bra et al., 1999). This technique is useful in information retrieval systems and in goal oriented educational systems. Link sorting is used in Hypadapter (Hohl et al., 1996).

Some hypermedia systems use graphic representations of the link structure. The graph is also called a map. Links are part of the graph and are called thumbnails. The whole map (or a subset of it, for example a chapter) shows an overall view of the links/paths and how they are related to each other. This is good in the learning process to review and reflect on the content learned. Such a map can also be subject to adaptation (Mukherjea, 2000; Benford et al., 2000).

De Bra (1999) added two more navigational support techniques, they are: link disabling, and link removal. Link disabling can be either by annotation or hiding. System greys out the inappropriate links if the link annotation technique is used, or makes the link invisible while link hiding is chosen to be the main technique used. Link removal removes inappropriate links or anchors. Link removal can be easily applied in a list, but inadequate in anchors inside textual paragraphs. ISIS-Tutor (Brusilovsky & Pesin, 1998) uses link removal.

3.4: Summary

Navigational adaptation provides learning support by manipulating the available links. It could take the form of link disabling, re-ordering, removal, annotation, direct guidance, or a map showing the current location of the page within the entire navigational structure. Another type of adaptation concerns the adjustment of information content to suit to individual learners and is called content or presentational adaptation. Content adaptation will be discussed in the next chapter of thesis.

Chapter 4: Adaptive Content Presentation

4.1: Introduction

Navigational adaptation without content would not be of any good use. Adaptive content presentation emphasizes the point that the presented learning content has to be tailored according to the student's needs as well. This chapter talks about the adaptivity of content, its educational value, and existing techniques employed for content adaptation.

4.2: Adaptive Content

"There is considerable evidence that different people learn best in different ways" (Kay & Kummerfeld, 1994). This means that it is very unlikely that one presentation style could suit to all students. The writing style of an author may be accepted by one reader but not another. Humours can be very pleasant to those who understand them but may irritate the readers who just do not grasp them. Information can be presented in different abstract levels and has varied effects in a group of students with diverse characteristics. As described by Kay and Kummerfeld (1994), programmers tend to like very terse and extremely abstract presentations of new information unlike non-programmers.

A student with prior knowledge of any other programming language can most probably learn C with less effort than those who are completely new to the programming field. However, the background knowledge can also hinder the acquisition of new knowledge in some cases. Therefore, correct identification of the

student's prior knowledge is essential to help the system identifying the likely misconceptions of the student and assists him/her to get over those misconceptions during the learning process.

For example, an important technique used to support students while going through the domain content is to provide glossary links. Students can click on the link if they are not sure of the definition of a term. However, there is often just one definition and the students have to click on the links every time they encounter unfamiliar terms. This not only adds extra effort to read, the additional links also add the complexity to the hyperspace. In order to facilitate the students' learning process, it would be better that students can tell the system (or system learns from student's behaviour) the style of the definition they prefer, and the system can then work out when and where the definition should appear automatically for them according to learners' preferences. This could be applied to the level of abstraction of information, examples, granularity, the anchor text (hotwords) in paragraphs, and the media type representing the information. To fully utilise the advantages of adaptive presentation, the systems needs to (Pascoe & Sallis, 1998):

- provide supplementary material for background areas to those students whose background knowledge is below what is presumed by the instructor,
- provide different styles of documentation for the definitions/concepts, and
- give answer to the exercises using clearly identifiable supplementary material, so that the students can be directed to clarify their misunderstandings.

4.3: Techniques of Adaptive Content Presentation

Adaptive content presentation works at the content level; how the content is presented. The information contained in a node, often in a web page, can be adapted to various details, difficulty, and media usage, and is aimed “*to meet users with different needs, background knowledge, interaction style and cognitive characteristics*” (Kobsa et al., 1994; Recker et al., 1995).

Adaptive presentation in a page is often the manipulation of text fragments. The manipulation is aimed at “*providing prerequisite, additional or comparative explanations*” (De Bra et al., 1999). Additional information can be added for the user, in a specific state, identified as some missing prerequisite knowledge, or as special interests. The additional explanations could also be in the form of comparison with previously known knowledge. De Bra (1999) pointed out that there are two techniques used to provide such explanations:

1. Conditional inclusion of fragments: Depending on the student model, the system determines which text fragment should be displayed.
2. Stretch text: For each information fragment, there is a short and visible “place holder”. The adaptive hypermedia system decides which place-holder should be stretched (displayed) and which should be shrunk (hide, only place holder visible). The student can choose to stretch or shrink place-holder, and the system records the action for adaptation of the display of subsequent pages. Stretch text is used in MetaDoc (Boyle & Encarnacion, 1994).

Hothi and Hall (1998) introduced another technique where the text fragments, evaluated as less useful, are greyed out instead of removed or shrunk. Two processes are essential in Hothi’s technique:

1. Providing explanation variant: The same information can be presented and structured differently. The level of difficulty, other related concepts, the length of the presentation, and media type can be adjusted according to the values in the student model. The scope of the variant can be only for a single page, or as a guidance through different pages. The latter one is essentially adaptive navigational control rather than just adaptive presentation. Explanation variant is used in Anatom-Tutor (Beaumont, 1994), and Hypadapter (Hohl et al., 1996).
2. Re-ordering information: The display order of information fragments can be altered depending on the student model. For example, some students may prefer to see examples before definition, or some would like to see the learning objective before going into details of the chapter. Sorting the order of information display places the most relevant information at the top while the least one at the bottom. Information sorting is used in Hypadapter (Hohl et al., 1996).

The most common content adaptation techniques include conditional inclusion of text fragment and stretch-text. Hothi and Hall (1998) suggested greying out the text fragments instead of removal of them; he also explained additional content adaptation techniques like explanation variant, and information re-ordering. And the experiments carried out on content adaptation show that it helps the memorisation of information, and improves the overall comprehension. Content adaptation certainly possesses great potential in the educational field.

4.4: Summary

Most of the research efforts on adaptivity, no matter in ITSs or AHMs, focused on the adaptive techniques and their demonstration of usefulness by field studies. However, an important issue had been neglected in most of the practical works discussed so far. Without a theoretical background on how should the adaptivity facilitate the students' cognitive process during learning, combinational use of those techniques creates severe problems. All these techniques have their strengths and weaknesses, and they do not identify with each other as a means to support learning in an adaptive environment (Kashihara et al., 2000). Therefore, the next chapter will talk about the Multiple Representation approach, an approach for content adaptation using multimedia objects, and the Exploration Space Control, a theory that integrates both adaptive content and navigation.

Chapter 5: Integrated Theories of Adaptivity

5.1: Introduction

Multiple Representation (MR) approach is an extension of traditional content adaptation that focuses on the use of multimedia objects. With its inclusion of both textual and non-textual content adaptation, multiple representation approach could provide a solid foundation for modern and future content adaptation systems (Kinshuk et al., 1999). Exploration Space Control (ESC) tries to integrate and re-organise existing adaptive techniques and to offer the best result of adaptivity by combining both the adaptive content and adaptive navigational support (Kashihara et al., 2000).

5.2: Multiple Representation Approach

All the adaptive presentation discussed so far is mainly the manipulation of textual fragments (adaptive text presentation). With the increasing popularity of multimedia as a learning support, there is definitely a need that adaptive presentation covers the application of multimedia as well.

Kinshuk et al. (1999) proposed a method to incorporate multimedia objects in adaptively presented domain content. It is called Multiple Representation (MR) approach.

By providing adequate content presentation for individual cognitive differences and by guiding the multimedia objects selection, navigational objects selection, and

integration of multimedia object, MR approach ensures the suitability of the domain presentation to different students' needs.

5.2.1: Multimedia Object Selection

In the discussion of MR approach, those suggestions are made for the selection of multimedia objects:

- *Task specificity and learner's competence*: The selection of multimedia object should depend on the characteristic of the task at hand. The selection should also consider the learner's competence level and appropriate tasks should be assigned accordingly (table 5.1).

<i>Domain competence level</i>	<i>Task</i>	<i>Examples of multimedia objects</i>
Novice in both in knowledge and skills	Direct instruction for knowledge	Text, pictures, audio, animations
Intermediate in knowledge, novice in skills	Direct instruction for skills with little exploration possibilities	Animations, videos, textual links, sensitive parts in static pictures
Intermediate in both knowledge and skills (Ready for problem solving)	Learning by problem solving for both skills and knowledge	Pictorial VR (e.g. asking correct position of a part in structure), Flowchart (e.g. asking a decision point)
Expert in knowledge, intermediate in skills	Advance exploration Possibilities	Flowcharts, user controlled animations, simulations
Intermediate in knowledge, expert in skills	Advance active observations	User controlled animations, advance pictorial VR

Expert in both knowledge and skills	Practice required for achieving mastery	Advance user controlled animations, advance simulations
--	--	---

Table 5.1: Multimedia objects selection for cognitive apprenticeship framework (Kinshuk et al.,1999)

- *Reference and revisit of already learned content:* MR approach recommends revisits of learned content in different context, as it enforces the links between concepts, and enhances the build-up of mental model.
- *Use of multi-sensory channels:* “The reception by learners enhances if the representation of domain content involves all relevant sensory channels” (Kinshuk et al., 1999). Appropriate selection of sensory channel also should not be overlooked.
- *Content-based selection of multimedia objects:* The representation should use the most appropriate objects in a particular context if there are more than one multimedia objects available.

5.2.2: Navigational Object Selection

Six types of navigational links are identified in MR approach:

- Direct successor links – They are links that lead to the successive domain concept/unit after the learning task in current concept/unit is fulfilled.
- Parallel concept links – They are links that lead to the analogous concepts that are related to the domain concepts. Comparison and contrast are available through those links for these who wish to increase their knowledge in width.
- Fine-grained links – They are links that lead to detailed explanation of the current concept. They provide and preserve finer granularity of domain

concepts, more accurate and more specific information to those who really need in-depth understanding.

- Glossary links – They are links that lead to the definitions of domain concepts/terms.
- Excursion links – They are links that lead to related learning that is usually outside of current context. They are sources where learners can find other related knowledge of their interests, and also create the possibility for accidental learning to take place (Reisberg, 1997).
- Problem links – They are links that lead to problems/exercises of the unit.

MR approach suggests the use of both interaction objects (push buttons, radio boxes) - for transfer from part of the system to another, and interactive objects (anchored text, pictures) - for system recommended contextual transfers. The result of activating a link should match the student's expectation by the provided interface (including the type of link, and type of navigational object). Therefore, MR approach recommends that the use of navigational objects should consider its contextual implication and be intuitive/explicit in terms of the results obtained after activating the navigation (Table 5.2).

<i>Examples</i>	<i>Recommended uses</i>
Textual links from main text (<i>Interactive object</i>)	<ul style="list-style-type: none"> * transfer to successor unit * transfer to excursion * transfer to glossary pop-up
Textual links from messages (<i>Interactive object</i>)	<ul style="list-style-type: none"> * transfer to successor unit * transfer to excursion * transfer to problems

Sensitive parts of static pictures (image maps) (<i>Interactive object</i>)	* transfer to fine grained unit
Push buttons (<i>Interaction objects</i>)	* transfer to successor unit * transfer to another learning unit on learner's explicit request * transfer to another aspect of same learning unit * transfer from message (e. g., arrow button in message)
Pop-up menu items (<i>Interaction objects</i>)	* transfer to another learning unit on learner's explicit request

Table 5.2: Types of multimedia objects as navigational links and their uses (Kinshuk et al.,1999)

5.2.3: Integration of Multimedia Object

The dual coding theory tries to balance the importance of non-verbal information processing to the verbal one, which is traditionally the mainstream medium. Paivio (1986), the originator of the dual coding theory, states that the "*human cognition is unique in that it has become specialized for dealing simultaneously with language and with nonverbal objects and events. Moreover, the language system is peculiar in that it deals directly with linguistic input and output (in the form of speech or writing) while at the same time serving a symbolic function with respect to nonverbal objects, events, and behaviours. Any representational theory must accommodate this dual functionality*".

The use of various multimedia objects to present a domain concept in a supplementary manner to the textual content is often the practice of multimedia-based educational systems. However, just the collection of multimedia objects does not

guarantee improved learning (Rogers et al., 1995). Therefore, the MR approach suggested the following rules while the combination of multimedia object occurs.

They are:

- There should be no more than one active multimedia object at a time on one visible screen, with the exception of a comparative study, because processing such information demands high cognitive load.
- The integration of multimedia objects should be complementary, and synchronised. There should not be any duplicate information presented in more than one multimedia objects.
- Presence of any decision intensive objects (e.g. flow chart) should avoid any other multimedia objects because of their high cognitive-load demand.
- Any multimedia objects, which are not distinguishable in the first sight, such as a static picture and a user-controlled animation with same appearance, should not be presented together to avoid confusion.
- Integration of dynamic objects and static objects using the same sensory channel should be avoided (e.g. force the students to observe an animation and read text at same time).

MR approach provides a guideline on how the presentational adaptation can be achieved in the perspective that the learning material can be categorised into multimedia objects. Exploration Space Control (Kashihara et al., 2000) examines how to integrate both adaptive content and adaptive navigation to achieve the potential of adaptation.

5.3: *Exploration Space Control*

Exploration is a self-directed process, and is an effective way of learning (Carroll et al., 1985). Exploratory learning, which involves searching the information, navigation through the learning space, understanding of domain-related conceptual knowledge and acquiring skills, often require high cognitive effort on the part of the learner.

Kashihara et al. (2000) proposed a theory to support exploratory learning in multimedia-based educational systems by allowing students to explore/acquire domain concepts and skills with as less cognitive load as possible.

ESC attempts to limit the learning space, called exploration space, to control the students' cognitive load at an adequate level depending on their learning approaches. ESC tries to facilitate adequate learning for a whole spectrum of learning competence by adopting two extreme approaches - active learning support and step-by-step learning support.

Active learning support offers the initial learning space as wide as possible using only those restrictions that are required to protect the students from cognitive overload, and adding or removing restrictions according to students' progress. This approach fits the students with better competence level and with more familiarity and experiences of the domain; for example those who come back for further education after working in an area related to the domain.

The step-by-step learning support adopts the opposite approach from active learning support by starting with as restricted information space as possible and then gradually

enlarging it. This approach provides a sense of comfort and security for those who are completely new to the domain area or do not have active learning styles. By the combination of these two approaches, “*ESC can be employed to facilitate proper learning for all types of learners*” (Kashihara et al., 2000).

The main concept of ESC is control; to control the available learning space by adaptive navigation, and to control the presentation of the learning content by adaptive presentation. The most important reason for the control is to try to give the student the most suitable instructional style to achieve the best results. Several types of control are proposed by ESC at different levels:

1. Embedded information: Information sources (e.g. different text fragment) are selected according to individual student needs and used by the process of scaffolding to create the information space.
2. Limiting information resources: The selection of information resources has great effect on what the student is going to see and interact with. Multimedia resources that suit the student’s learning style and integrated with consideration of student’s cognitive ability yield the best result.
3. Limiting exploration paths: ESC supports the use of navigational adaptation to offer the suitable paths in the learning space. This will ensure that the students are not discouraged by the complexity of the entire learning space and do not receive the material that is not appropriate for them at a certain stage in the learning process.
4. Limiting information to be presented: When the information includes many media resources and/or is available in large amounts, care has to be taken on how to present it in order to achieve the best effect. More information does not

mean better understanding; sometimes it is a burden on the students. For example, textual explanation that matches the abstractness and granularity currently appropriate for a student may produce better learning result than flashy multimedia presentation. Therefore, all information resources are to be filtered and only the necessary ones to be presented.

5.4: Summary

Multiple representation approach gives a thorough prescription on how to use media resources wisely in instructional design, while Exploration Space Control offers a framework that integrates the techniques for both navigational adaptation and content adaptation. There exists a need of a formal guideline on how to produce educational system that provides the facility of adaptivity, and at the same time adheres to the principles of human cognition based of the framework of ESC. In the next section, an attempt to address this issue will be elaborated in detail.

Chapter 6: Exploration Space Control Formalisation

6.1: Introduction

After the discussion of all the benefits that are available by the adaptive theories and techniques, there are still some difficulties identified in the research of this field. When the information space is large and the users are from different background or with different goals, the risk of unproductive wandering in the hyperspace is relatively high, and this is named as “problem of disorientation” by Eklund & Zeiliger (1996). Hypermedia based systems allow students to learn at their own pace (Laurillard, 1993) but *“some direction may be necessary for hypermedia to be an effective educational tool”* (Eklund & Zeiliger, 1996). As the learners are granted more and more control in adaptive systems, it becomes debatable whether this is really beneficial to the learners. In fact, *“there is a growing body of empirical evidences to suggest that learners tend to make poor decisions in learner controlled systems.... Students become lost, skip important content, choose not to answer questions, look for visually stimulating rather than informative material, and use the navigational features unwisely”* (Eklund & Zeiliger, 1996).

There is a growing urgency that research is required to ensure that all the benefits and advantages of the adaptive theories and techniques really come alive. It could be a difficult task for courseware developers to search the best theory or techniques to use as each of them have their strengths and weaknesses (Kashihara et al., 2000).

Exploration Space Control Formalisation (ESCF) is an attempt to provide a formalised plan on how to utilise the ESC guidelines to assist learning in a technology-based learning environment (Lin, 2002). The aim of ESCF is to provide a welding mould where courseware content can be just cast in and the benefits of adaptivity could be ensured. ESCF tries to formalise the level of control (size of exploration space, difficulty level, and so on) to suit each student's learning abilities. Intensive research work on human cognition is carried out to identify the characteristics of each of the selected learning abilities, and how could the control level be adapted to suit the different requirements caused by each individual's differences. The differences in learning abilities between each individual student has to be categorised and measured in order to be useful to the VLE when making pedagogic decisions. In the ESCF, the categorisation is based on students' cognitive trait. The definition of and the rational for selecting certain cognitive traits are provided next.

6.2: The Definition and Selection of Cognitive Traits

Certain human learning abilities are derived from the aggregation of a group of lower-level abilities. Such aggregation of lower-level abilities could have structured relationship (such as tree-structure) resulting into higher-level abilities. Cognitive-load is a good example.

Cognitive load is an important factor in human learning, and its implication in learning environment has received intense discussion (Kashihara et al., 2000; Kinshuk et al., 1999). Students with higher cognitive load is said to learn faster and have better

comprehension. The speed of learning is also related to how fast human brain process the in-coming information (information processing speed), the ability to integrate the new knowledge into existing one (associative learning skill), and the ability to work out the underlying rules of the presented information (inductive reasoning skill). The comprehension is also closely linked with associative and inductive learning skills, working memory and also the ability to reflect on what was learnt and determination of whether the previous perceptions are correct or not (reflectivity). Therefore, the function and ability of cognitive-load can be broken into even subtler elements.

Besides, with higher cognitive-load, someone can easily reduce the load on his/her working memory by constructing a learning strategy, which represents higher-order rules and can generate other rules/data from it. For example, some people remember the series on number (such as telephone numbers) by remembering the sequence of the positions of the number-pad, while someone good in music could remember it as a series of pitches of sound; many others will probably have to look up the phone book and dial the number many times just to remember them.

It is obvious that the cognitive load consists of the function of information processing ability (Brown & Hirst, 1983), the inductive and associative learning skills (William, 2001; Authur, 1994), and determines the load of the working memory and thus its efficiency (Scandura, 1973; Anderson, 1983). Therefore, it could be said that the cognitive-load is in the higher level of the ability hierarchy than information processing speed, inductive reasoning skill, associative reasoning skill, and working memory capacity. Also, for the purpose of this discussion, there is a need to think of the learning abilities, as elementary and basic as possible, in order to identify clear

matches and relationships of the learning abilities to the curriculum designs, but there must be clear understanding of how elementary is acceptable. *“There are simply too many levels of abstraction between the operations of human cognition and the level of phenomena that modern neuroscience can address. Reducing cognitive science to neuroscience makes about as much sense as reducing neuroscience to quantum mechanics”* (Forbus & Feltovich, 2001).

Therefore, the definition of cognitive trait (CT) in this context is the lowest-level element in the ability hierarchy, and is relevant for categorising and measuring students' learning abilities in hypermedia-based learning environment.

The cognitivist view of learning, which posits that knowledge is represented as mental schema and learning is the process of reorganising the schema, is adopted here and the term, “trait”, denotes the characteristics of the abilities that are relatively persistent in time, and consistent across domains.

Unlike the knowledge or expertise, which are more domain-specific and evolving, cognitive traits are relatively stable and portable over different tasks. Thus, the student model, based on cognitive traits, obtained in one domain can provide more reliable prediction of the performance the same student will have in another domain. Differences of cognitive traits (CTs) cause differences in quality, efficiency, and overall performance on the tasks carried out. Four CTs were selected for the discussion: working memory capacity, inductive reasoning ability, information processing speed, and associative learning skill.

Before starting the discussion on Exploration Space Control Formalisation, another term needs further explanation. It is called the Exploration Space Control Element.

6.3: Exploration Space Control Elements

The aim of Exploration Space Control is to provide guidance to the instructional designers who wish to have their instructional systems equipped with adaptive and customisation functionalities.

Exploration space control elements (ESCEs) are those gauges on both the learning content and navigational paths that could be modified to create different versions of the same materials to suit different needs. Therefore, the selection of the Exploration Space Control Elements (ESCEs) has to cover both the content and navigation. Table 6.1 attempts to list some examples of these gauges.

Category	Element
Path	Number
	Relevance
Content	Amount (detail)
	Concreteness
	Structureness
Info Resource	Number

Table 6.1: Exploration space control elements

The selection of ESCEs is purely research-based, and the aim of the formalisation process is to create an exemplary framework that could be tested to determine the

correctness and usefulness of the theory. It is necessary to examine the list of ESCEs and modify it according to the nature of different projects or domains.

There are two elements related to the path, the number of the paths and the relevance of the paths. Since most of the modern adaptive VLEs are built by hypermedia technology, paths are typically presented as the series of links. Therefore the term link and path will be used interchangeably in the discussion.

There are six types of navigational paths identified by Kinshuk et al. (1999) as listed in table 6.2.







<i>Types of link</i>	<i>Relevance to the domain</i>
1. Direct successor link	
2. Fine grained link	
3. Glossary link	
4. Problem link	
5. Parallel concept link	
6. Excursion link	

Table 6.2: Types of link and their relevance to the domain

Links with higher relevancy are those that are closely related to the domain (such as direct successor, fine grained links), while the links with lower relevancy are to the concepts that are more of supplementary nature (such as excursion links).

The category of path (table 6.1) covers both the number of paths and their relevance. The “number of paths” represents the amount of links to be presented to the students

and the “relevance of paths” guides how relevant the linked targets should be and the proportions of various types of links (table 6.2).

In table 6.1, there are three ESCEs for the category of content, the amount/detail, the concreteness and structureness. Amount/detail of content decides how detailed the knowledge presentation is. The detail-ness affects the volume of the presentation; therefore, both of them are classified as only one ESCE.

The concreteness determines the abstract level of the information. The more concrete a piece of information is, the more fundamentals and examples it should include. Abstraction refers to the opposite of the concreteness.

The structureness of information indicates the ordering, arrangement, and sequencing of the information presented as a series/set of concepts. Concepts may or may not be directly or indirectly related to each other with certain sequence. The elaboration theory (Reigeluth, 1992) suggested a structured approach by stating that the instruction should be organized in increasing order of complexity for optimal learning. The more structured the information is the more orderly fashion it could be presented in. Structured information helps the learner in building up the mental model, but provides less freedom for free navigation.

Another category in table 6.1 is information resource. Information could be presented by different media, such as text, audio, visual, and so on. The effect of each media for presenting information is different. *“Audio is good to stimulate imagination, video clips for action information, text to convey details whereas diagrams are good to*

convey ideas” (Kinshuk et al., 1999). Besides, each media type has different impact on each individual’s learning. If there is more than one media (e.g. a textual explanation and a chart), it could make the information more impressive and easier for future recall. The greater the number of information resources, the more choices the system has for selection to suit the students’ characteristics.

6.4: *Working Memory Capacity*

Working memory is the cognitive system that allows us to keep active a limited amount of information (roughly, 7 ± 2 items) for a brief period of time (Miller, 1956), to temporarily store the outcomes of intermediate computations when solving a problem, and to perform further computations on these temporary outcomes (Baddely, 1986).

The research on working memory (Kearsley, 2001; Scandura, 1973; Anderson, 1983; Huai, 2000 as quoted by Kommers, 2000; Case, 1995; Salthouse & Babcock, 1991; Byrne, 1996) shows the fact that the speed of learning, the memorisation of learned concepts, effectiveness of skill acquisition and many other learning abilities are all affected by the capacity of working memory which is mainly comprised of two components; the limited storage system, and the central execution unit carrying out the cognitive operation efforts.

There are already many educational system design guidelines, written by human-computer interaction (HCI) experts, devoted to address the relationship between the storage aspect of the working memory and good interface designs. The effort to

facilitate learning with regard to working memory should mainly be focused on the instructional design to assist the synchronised operation with the central execution unit by assisting the formation of higher order rules, build up of the mental model and of course not to overload the storage system of the working memory.

6.4.1: Formalising with consideration of working memory

When the working memory capacity of the learner is low:

The number of paths and the amount of information should decrease to protect the learners from getting lost in the vast amount the information, from overloading the working memory with complex hyperspace structure (Kashihara et al., 2000), and allow more time for learner to review essential content if necessary. The relevance of the information should increase so that the learners get the important information. The concreteness of the information should increase so that the learners can grasp the fundamental rules first and use them to generate higher-order rules as suggested by the structured learning theory (Scandura, 1973). The structure of the information should stay unchanged. The increase of structure-ness can facilitate the building of mental model and assist future recall of the learned information. But as Huai (2000 as quoted by Kommers, 2000) indicated the versatile learners tend to have smaller short-term memory (storage aspect of working memory) than serial learners, and the increase of structure-ness limits their navigational freedom, which is the primary way they learn. So basically, the net effect cancels out and the structure of the information is recommended to stay unchanged. The number of the information resources should increase, so that the learners could choose the media resources that work best along

their cognitive styles and gain deeper understanding on the subject domain (Aptitude Treatment: Cronbach and Snow, 1989; Level of Processing: Craik & Lockhart, 1972).

When the working memory capacity of the learner is high:

The number of paths and the amount of information should increase and the relevance of the information should decrease to enlarge the exploration and domain space of the learning process so that more knowledge is available to the learners who process more higher-orders rules which *“account for creative behaviour (unanticipated outcomes) as well as the ability to solve complex problems by making it possible to generate (learn) new rules”* (Kearsley, 2001). The concreteness of the information should decrease to avoid boredom for the learners resulting from too many similar examples. The structure of the information and the number of information resources stay unchanged because there are no direct and apparent benefits associated.

Table 6.3 summarises the discussion of working memory capacity into a table.

	<i>Path</i>		<i>Content</i>			<i>Info Resource</i>
Level	Number	Relevance	Amount	Concreteness	Structure	Number
Low	-	+	-	+	\	+
High	\+	\-	+	-	\	\

Table 6.3: Working Memory Formalisation

The symbols used in the formalisation table (table 6.3 – 6.6) have the following meanings:

"+" → should increase

"-" → should decrease

"\" → should slightly increase (recommend only), or should increase

"\" → should slightly decrease (recommend only), or should decrease

For example, table 6.3 suggested that the number of path (navigational link) should decrease while the learner's working memory is identified low.

6.5: Inductive Reasoning Skill

Induction is to figure out the rules/theories/principles from observed instances of event. William (2001) described that the "*inductive reasoning works the other way (from deduction), moving from specific observations to broader generalizations and theories*". It is a bottom-up approach and has an open-ended and exploratory nature.

The research on inductive reasoning skill (Heit, 2000; Authur, 1994; Bower & Hilgard, 1981; William, 2001) shows that the higher the inductive reasoning ability, the easier it is to build up the mental model of the information learned. Mental model, also called cognitive structure, "*provides meaning and organization to experiences and allows the individual to go beyond the information given*" (Kearsley, 1998). From the constructivist's point of view, the learner's selection and transformation of information, constructs, hypotheses, and the decision process, all rely on the mental model (Bruner, 1973). For students who possess better inductive reasoning skill, it is easier for them to recognize a previously known pattern, generalize higher-order rules and, as a result, the load on working memory is reduced, and the learning process is more efficient. For its educational value, there is a need to specify means to support

those with lower inductive reasoning skill and to maximize the learning for those who are already good at induction.

6.5.1: Formalising with consideration of inductive reasoning skill

When the learner's inductive reasoning skill is poor:

The number of paths should increase to give the learner more opportunity to observe and thus promotes their induction. The relevance of paths should decrease and the concreteness of the information should increase so the learner can have more diverse observations to promote induction. The amount of information should increase to give detailed and step-by-step explanation to the learners, so that they can see the rules/theories easier. The structure of the information should increase so that it is easier for the learner to build up the mental model and see the sequential relationship of the topics and relationship of concepts. The number of information resources does not need to change because there are no direct and apparent benefits associated.

When the learner's inductive reasoning skill is good:

The number of paths and the amount of the information should decrease to speed up the learning process. The concreteness of the information should decrease to avoid the boredom resulted from too many similar examples. The structure of the information, the number of information resources, and the relevance of paths do not need to change because there are no direct and apparent benefits associated.

Table 6.4 summarises the discussion of inductive reasoning skill.

	<i>Path</i>		<i>Content</i>			<i>Info Resource</i>
Level	Number	Relevance	Amount	Concreteness	Structure	Number
Poor	\+	\-	+	+	+	\
Good	\-	\	-	\-	\	\

Table 6.4: Inductive Reasoning Skill Formalisation

6.6: Information Processing Speed

Information processing speed determines how fast the learners acquire the information correctly. Textual information is used to be the main source of information in the educational systems before the start of the multimedia. The information processing speed was greatly determined by the reading speed and it still true nowadays to a certain extent as text still takes up the majority of the literature (Raiford, & Dudley, 2000; Brown & Hirst, 1983).

Instructional designs of educational systems should take into account the consideration of learner's reading speed. They should be able to provide adequate support in the form of adaptivity by providing hints to slow readers and to facilitate the maximum intake of knowledge for fast readers.

6.6.1: Formalising with consideration of information processing speed

When the learner's information processing speed is slow:

The number of the paths and the amount of information should decrease, and the relevance of the paths should increase so that only the fundamental/important points are presented to the learner to enable him/her to complete the course on time. The structure of the information should increase to facilitate and speed up the learning process. The concreteness of the information and the number of information resources does not need to change because there are no direct and apparent benefits associated

When the learner's information processing speed is fast:

The number of paths, and the amount of information should increase; the relevance of the paths should decrease in order to enlarge the information space and hence the available knowledge and provide in-depth and comprehensive insight into the subject matters. The concreteness of the information, the structure of the information, and the number of the information resources does not need to change because there are no direct and apparent benefits associated.

The discussion of information processing speed is summarised in table 6.5.

	<i>Path</i>		<i>Content</i>			<i>Info Resource</i>
Level	Number	Relevance	Amount	Concreteness	Structure	Number
Slow	-	+	-	\	\+	\
Fast	+	-	+	\	\	\

Table 6.5: Information Processing Speed Formalisation

6.7: Associative Learning Skill

Associative learning skill denotes the skill to link existing knowledge to the new information in order to make sense of it and thus the information can become new knowledge. Quinton (2001) pointed out *“we do not learn information as discrete, isolated facts, but instead integrate new information with knowledge we already possess. Our best learning occurs when new material is readily connected with what is often ‘complex and multiple link of association’”*.

“Cognitive psychology suggests that a mental model consist of two major components: knowledge structure and the processes of using this knowledge” (Merrill, 2000).

Inductive reasoning skill aids to figure out the pattern of the correct operations on how to use the knowledge structure whereas the associative learning skill assists the build-up and storage of the knowledge structure itself.

In order to assist the association processes during the student’s learning, the instruction needs to assist the recall (remembering) of learned information, clearly shows the relationships of concepts (new to existing), and facilitates the formation of new and creative association/insight by providing sufficient information of related domain area.

6.7.1: Formalising with consideration of associative learning skill

When the learner’s associative learning skill is poor:

The number of paths, structure of the information should increase. More hints and information could help the learner to associate one concept to another. The number of information resources should increase to stimulate the forming of associations. Different information resources (media) could provide different magnitude of understanding of the same concept. The relevance of the paths should increase to prevent the learner from getting lost in the less-relevant information and create too many useless associations. The amount of information and the concreteness of the information do not need to change because there are no direct and apparent benefits associated.

When the learner's associative learning skill is good:

The number of paths and the structure of the information should decrease so learners can navigate more freely and hence enhance the learning speed, and stimulate more associations. The relevance of the paths should decrease to enlarge the information space and hence the available knowledge. The amount of information, the concreteness of information, and the number of information resources do not need to change because there are no direct and apparent benefits associated.

Table 6.6 summarises the discussion of associative learning skill.

	<i>Path</i>		<i>Content</i>			<i>Info Resource</i>
Level	Number	Relevance	Amount	Concreteness	Structure	Number
Poor	+	+	\	\	+	+
Good	\-	-	\	\	\-	\

Table 6.6: Associative Learning Skill Formalisation

6.8: Summary

ESC theory inherits the strength of both adaptive navigation and adaptive content presentation. It is designed to be used in a technology-based learning environment and therefore it encompasses the advantages offered by multimedia, hypermedia and adaptive hypermedia. On the other hand, human's cognitive ability has equal emphasis in the ESC theory, which is the basis for determining different levels of control in exploration process. The overall aim is to make the learning a satisfactory experience for students who all have different cognitive abilities.

The motivation for ESC Formalisation came from the question of how to really make good use of the Exploration Space Control theory to produce good educational material that could suit the diverse needs of students in the modern learning environment. Formalisation is a process that substantiates the theory in terms of its pragmatic value. The discussion has demonstrated how the formalisation was carried out and clearly shows its value and its educational potential from the perspective of cognitive science. But an important issue has deliberately left un-addressed, and only until it is properly addressed, the work of the Formalisation can be pragmatic. This issue is about student modelling.

Student model, as regarded as an essential component in any adaptive systems, has attracted intensive research focuses in the last few decades (Smith, 1998; El-Sheikh, 1997; Martin, 1999). But to the authors' knowledge, none of the available student modelling approach is capable of delivering the necessary information required for

fine-grained adaptivity such as required by ESC Formalisation in terms of scope, facilities, and structure. Therefore, a new approach of student modelling will be introduced in the next chapter, and it is called the Cognitive Trait Model (CTM). Before the discussion of CTM, a review of the current-status of the development of student models will be briefly covered in order to provide the insight and the appreciation that a new approach is urgently in demand.

Chapter 7: Student Model

7.1: Introduction

During the past few decades, many important ideas have been presented by the user modelling (UM) community. UM in educational context is called student modelling. However, it is far beyond the scope of this section to come up with a complete overview of all the research endeavours. Instead, the aim of this section is to outline some of the main characteristics of UM that are of interest for understanding, referencing, or developing applications of UM in VLEs.

7.2: Student Modelling

Student modelling is the process of creating a representation of student in the VLE, and this representation is often named as the student model. Student models contain the information about individual students. It is an essential component in any adaptive virtual learning environment (Han et al., 2001). Student modelling also includes the process of collection, to analysis, and the representation of students' knowledge on the domain and tries to infer how they reason during problem solving. The modelling process is sometimes called diagnosis as it constantly checks and records students' progress. El-Sheikh & Sticklen (1998) called the student model as *"one of the most difficult modules of an ITS to develop"*.

Paiva et al. (1995) defined student models as *"...representations of some characteristics and attitudes of the learners, which are useful for achieving the*

adequate and individualised interaction established between computational environments and students". Attributes of students relevant to the learning process are recorded, constantly updated, and continuously accumulated. The VLEs can then use this accumulated knowledge of student as the basis of adaptation to student needs (El-Sheikh, 1997).

7.3: Application of Student Model

In general, the common uses of a student model include the decision about the advancement to a new topic or different level, offering advice and guidance to the learner, generating problems to access the student's knowledge level, and providing explanations or feedback to the learners (El-Sheikh & Sticklen, 1998).

Hume (1995) listed out, in the description of CIRCSIM-Tutor, that the student model can be used to:

- assist the instructional planner to create the tutoring plan;
- help the discourse generator to determine the nature of CST's contributions to the dialogue;
- providing statistics for future research;
- validate planning and student modelling rules;
- store the student model on a disk for future analysis and use in a subsequent tutoring session; and
- initialise the student model from a disk.

The student model in CIRCSIM-Tutor is used to guide the planning of the tutoring dialogue, switching tutoring protocols, and, in large, adjusting the curriculum (Zhou & Evens, 1999).

The attributes of the learners enable the virtual learning environment to adapt itself to the individual learners. Webb et al. (2001) classified the student attributes into the following types:

- about the cognitive processes that underlie the student's actions;
- about the differences between the student's skills and expert skills;
- about the student's behavioural patterns or preferences; and
- about the student's characteristics.

Han et al. (2001) described that the VLE should record the pre-knowledge of the students before learning, the post-knowledge of the students after learning, and the change of the cognitive state during the learning process. Cognitive states represent students' beliefs in a particular moment, which could either be correct or buggy.

Stauffer (1996) raised an issue regarding to student modelling in procedural domains (Goldstein, 1982; Burton, 1982). Stauffer (1996) divided procedural knowledge into three categories (levels):

- Effective Knowledge - includes the formulas, which enable effective problem solving.
- Principle Knowledge - includes the definition of which the effective knowledge is derived from.

- Heuristic Knowledge - which implies what knowledge to apply to a particular state.

In procedural domains, students tend to learn only the skills for solving problems, and some basic knowledge that is needed to solve those problems. There is a tendency to not learn as much about why the problem can be solved as about how to solve them. Problem solving skill can be obtained by merely repetitively solving similar type of problems without knowing why. The novice-expert shift is the transition from the possession of only the effective knowledge to the inclusion of all three of them (effective, principle, and heuristic). Note that the principle knowledge can otherwise be seemed as a kind of declarative knowledge, which reveals that understanding the concepts or the principles is also an essential step in learning procedural domains. Student models, trying to represent students' understanding of the domain, can be used to model both, of the procedural and the declarative knowledge.

While, the separation of the declarative and procedural knowledge aims to differentiate the type of student knowledge in terms of the structure, the definition of the state and process model tries to distinguish the method by which the knowledge is modelled. A state model contains only state information about the students, where as a process model can simulate the students' problem-solving process (El-Sheikh, & Sticklen, 1998). Both of state models and process models will be discussed next.

7.4: Modelling Competence State

Students' domain competence, which is identified as the most important feature in the existing systems, has to be constantly updated to reflect the progress in the students'

understanding. This is often accomplished by recording the nodes/ concepts visited by the students and the result of the learning from some form of assessment. Updating or diagnosis is an important task to maintain the currency and correctness of the student model. The following is a list of state models:

- overlay student model
- differential student model
- perturbation student model
- constraint-based student model

7.4.1: Overlay Student Model

The student's knowledge at any point is assumed to be a subset of expert's knowledge in an overlay student model. The differences between the student's and the expert's set of knowledge are believed to be the student's lack of skills and knowledge, and the instructional objective is to eliminate these differences as much as possible (Brusilovsky, 1994; Staff, 2001; Urban-Lurain, 1996; Stankov, 1996). Data in the overlay model can either be binary (e.g. Mastered, or Not-Mastered) or probabilistic (e.g. any fraction within 0-1). Overlay model has been used in GUIDON (Clancey, 1983), which is a tutoring system to teach the diagnosis of bacterial infections, and the SQL-Tutor (Mitrovic & Olhsson, 1999), which combines the overlay and the constraint-based model. An issue with the overlay model is that there is no mechanism to differentiate between the knowledge the student has not yet grasped and the knowledge the student has not yet been exposed, and this has implications for the tutoring strategies (Smith, 1998).

7.4.2: Differential Student Model

This model is a more structured variant of the overlay model. Differential model separates the entire domain knowledge into learned by the student and not-learned by the student (Staff, 2001; Urban-Lurain, 1996; Smith, 1998). The WEST system (Burton & Brown, 1982) employed the differential student model.

7.4.3: Perturbation Student Model

This model retains the information of what the student has known, and what the student has known incorrectly (misconceptions or errors in an error model). The error model can be created by enumerative or generative process. The enumerative process lists all possible bugs usually via an analysis of the problem domain and the errors that students make (Smith, 1998). Virvou et al. (2000) ITS, teaching the passive voice of the English grammar to Greek students, is an example of perturbation student model with an enumerative error model. It includes the errors that are caused by the interference of the mother tongue and other foreign languages that the students are familiar with. The inclusion of misconceptions in student models provides the excellent facility and resource to expand the explanation during the feedback to the students. However, misconceptions predefined during system development increases the time-span needed for implementation, and causes computational problems as *“the search space involved in constructing and maintaining the student model is greatly expanded, requiring the use of heuristics to prune the search space”* (El-Sheikh, 1997). On the other hand, the generative approach attempts to generate bugs from an

underlying cognitive theory and avoids both the problems mentioned for the enumerative approach.

7.4.4: Constraint-based Student Model

Martin's (1999) SQL-Tutor adopted constraint-based model (CBM), which represent both domain and student's knowledge as a set of constraints. Constraints represent basic rules/concepts of the domain. The student model of SQL-Tutor represents a set of constraints that the student has violated (misunderstood), and therefore can be thought as an implicit bug library (Martin, 1999). This bug library is different from the error model of the perturbation model; the bugs in CBM are defined as the failure to give the right solutions to the question and nothing else, whereas the error model can contain information about errors that are not presented in the domain knowledge model. The CBM only keeps track of the incorrect knowledge via recording the amount of constraints violated, but not the current coverage of the student's correct knowledge, which is redundant in this model, and therefore gains a great improvement in the storage utilization. Error patterns described by Hume (1995) in the CIRCSIM-Tutor serve the similar role as (violated) constraints described by Martin (1999), and the difference between them is only nominal.

7.5: Modelling the Process

Process models are oriented to model the problem-solving process the students undertake. A process model represents the students in term of both the knowledge they learned in the domain, and inference procedures. *“Such a model would be an*

executable process model, and could thus predict what the learner will do next, as well as work backwards from learner behavior to generate explanations” (El-Sheikh, 1997).

El-Sheikh & Sticklen (1998) described a particular type of a process model called the function-based model or functional model for short. The aim of the functional model is to make use of the known functionalities inside a learning environment to 1) make causal inferences, and 2) provide a reasoning algorithm that can be used to simulate the learning environment for given starting conditions. Each function is subdivided further into a set of behaviours. The entire learning environment can be represented as a hierarchy of functions and behaviours. Causal inferences about students’ current competences can be made by the set of functions that they had gone through. For example, if the start condition is called *a*, and the finish state of a particular task is called *b*, and let *C* represent all the sub-functions and behaviours of the function *f* where:

$$f(a) = b$$

It could be inferred that the set of behaviours, *C*, had been carried out given the function *f* was performed successfully. By this inference mechanism, the entire learning environment can be simulated explicitly, and the predictions of both the students’ and the tutor’s (module) actions can be made.

In the perspective of learning theory, the state modelling approach originates from the theory of behaviourism, which treats the student as a black box capable of producing

a certain response when a specific stimulus is given. The instructional goal is to improve the student response in terms of competence or mastery level by the process called conditioning (Mayo, 2001). Testing through questions-and-answers is an effective tool to measure the student performance in the behaviourist instruction. However, the cognitivism claims that learning is not merely the direct transfer of external information into internal knowledge, but is a process of reorganisation of the outside information into an internal representation (mental model) of the domain (Reisberg, 1997). And the process model is aimed at assisting both the students and tutors to form, understand, and better utilise the mental model in order to optimise the effects of learning and teaching.

Both the state model and process model need to be constantly updated to reflect beliefs of the student, no matter whether those beliefs are related to declarative or procedural knowledge. As described by Mayo (2001), the short-term representations of the student model are often very specific, e.g. “the student has given the right answer to question 11”, whereas the long-term representations typically contain inferred beliefs. This inference is also called student model diagnosis. Along the line of student model diagnosis, a distinction can be made about enumerative modelling and generative modelling.

For enumerative modelling, the system developers analyse the model and determine possible errors students can make or are prone to make (Smith, 1998). An error can either be a primitive error, or a composite error, which is the combination of primitive errors. DEBUGGY (Burton, 1982) used enumerative technique and catered both primitive and composite errors. Enumerative modelling techniques suffer from costly

computational requirements while matching the errors, and resource-intensive development cycle. An alternative to enumerative modelling is generative modelling. For generative modelling, the system uses an expert's model, which is a cognitive model, to detect students' errors. No errors or bugs need to be predefined because they can be deducted from the cognitive model. Errors are regarded as failed extrapolation of the concepts learned, and if the general form of the extrapolation errors can be found, then the majority of the errors can be explained (Mayo, 2001).

7.6: Student Model Diagnosis

Diagnosis is carried out to update the student model, and resolve temporal inconsistencies. The goal of the diagnosis is to enable students to find, overcome, and correct mistakes they made (Stankov, 1996). The tasks required for diagnosis include collecting information about students and analyse/ measure their performances.

Kobsa (1994) categorized three main sources where the information about students can be obtained:

- **Stereotypes:** The stereotypes categorise users into predefined categories.
- **User supplied preferences:** These are requested explicitly from students.
- **Analysis of user actions:** The actions carried out by users are used to infer the user's preferences, competency level, and so on.

7.6.1: Stereotypical Modelling

A stereotype model is a simple student model that places students into predefined stereotypical groups (Brusilovsky, 1994). This simple stereotype model represents a

student's knowledge in a topic-value pair. The value describes only a particular student attribute and is usually the mastery level (novice, intermediate, expert) for the corresponding topic/concept. Stereotypical model is a very coarse-grained approach and researchers have even questioned its validity. Webb et al. (2001) claimed, by modelling the community (stereotype) in commercial context, that “*very substantial increases in purchases are claimed for systems that recommend products to users of retail web sites using models based on purchases by other users*”, whereas academic research has been focused on modelling individuals only (Webb et al. 2001). A more recent trend of using stereotypical model is called default model. When a new student is encountered, there is no historical data that the system could infer on, and the default model is therefore created by the demographic or user-supplied data to cater for this situation. A number of studies (Tang et al., 2001; Pitkow, n.d.) characterise users according to their navigational strategies. The characterisation is described as follow:

- a. **Serendipitous browser:** Users who avoid the repetition of long invocation sequences.
- b. **General purpose browser:** Users who navigate in a probabilistic way. “*They (users) have roughly a one in four chance of repeating a more complex navigation sequence*” (Eklund & Zeiliger, 1996).
- o **Searcher:** Users who infrequently perform the same short navigation sequences, but do perform long navigational sequences often.

7.6.2: Coverage Measure

This is one of the simple method of diagnosis. It measures, in the entire learning space, how much the student has covered, by recording which node or page had been visited by the student. The results or the effects of those visits are not taken into account.

7.6.3: Performance Measure

This is similar to the coverage measure, but the only difference is that the system measures what was learned, by means of giving tests to the students. RIDE (Munro & Pizzini, 1996) used this technique. *“Although this is a straight forward method of measuring a student’s performance it gives reasonable clues about what type and how much information the learner will need”* (Smith, 1998).

7.6.4: Problem-solving State Tracing

This technique compares the problem solving steps of the student to that of the expert’s answer (Stankov, 1996). The student usually does not need to follow every step strictly, alternatives and tolerances can be specified. Systems employing this technique need to have sophisticated problem-solving models, and those problem-solving model are very domain-dependant, thus it is usually used within domains where the problems are well-structured, and there exist strictly followed de facto standard procedures for problem solving. LISP-Tutor (Anderson & Reiser, 1985), and WEST (Burton & Brown, 1982; Fischer, 2001) are examples of systems that utilise the problem-solving state tracing technique.

7.6.5: Path Finding

It is an extension of the problem-solving state tracing. Path finding includes algorithms to find the transition from one state to another. The transition, once found, can be used to predict student actions (Stankov, 1996). Path finding is technically complex as it requires the system to model the student's cognitive process of the problem solving.

7.6.6: Skill Tracing

Skill tracing is also called knowledge tracing (Han et al. 2001). This technique is often used in domains where the procedural (know-how) knowledge dominates. In those domains, procedure skill acquisition is the main focus of learning and the skills in such domains are often called rules. The entire domain can be modelled by a set of rules. As opposed to problem-solving state tracing, which has to model all the states in a problem-solving episode, skill tracing only records the usage of skills demonstrated during the problem-solving process (Smith, 1998). Probabilities can also be used to represent the student's understanding on each rule (Han et al. 2001). Skill tracing has been implemented in the WEST tutoring system (Burton & Brown, 1982).

7.6.7: Case-based Reasoning

Case-based reasoning (CBR) (Han et al. 2001) is used to perform both student modelling and machine learning. CBR stores two main types of information:

information of the students, and information of the cases of problem solving. When a student starts a new session, the system creates a new case. The system can then find similar cases by looking up cases solved successfully by other students with similar attributes to the current one. Those similar cases can be used to predict the student's behaviour and also the result. Interventions or guidance can be planned. In the circumstances where the strategy used by the current student is different from that of the similar cases, and the final answer is proved to be consistent with the expert's, then this case is regarded as an instance of machine learning.

7.6.8: Expert System

Many systems employ separate expert system to perform the diagnosis. The expert system is often independent from the tutoring system. GUIDON (Clancey, 1983) used the MYCIN expert system (Shortliffe, 1976). GUIDON checked the student's actions from that of the actions MYCIN would take, and intervened only when GUIDON found that the student's actions were sub-optimal or leading to an incorrect outcome.

7.7: Constructivism and Student Modelling

By the description of the function of student model, it seems that student modelling is in conflict with constructivist point of view about learning. Constructivist theory postulates that knowledge is "formed subjectively" rather than "transferred" from the student's previous experience. Knowledge is asserted to be existent only inside human mind (Duffy, & Jonassen, 1991) and the existence is unique to the very individual. Furthermore, the instructor should use a variety of material including raw data and

primary sources (Brook & Brook, 1993). And *“constructivism says that domain knowledge should not be decomposed and presented to the student in parts; instead, learning should take place in realistic settings with all the ambiguities and extraneous details that entails”* (Mayo, 2001), so the students would be able to better apply their knowledge in real-world context. Lebow (1993 as quoted by Tam, 2000) summarized, *“Traditional educational technology values of replicability, reliability, communication, and control contrast sharply with the seven primary constructivist values of collaboration, personal autonomy, generativity, reflectivity, active engagement, personal relevance, and pluralism”*.

Based on the argument that the cognitive abilities of every student are unique, it becomes futile to represent the student in any predefined structure. Does it mean that the efforts on student modelling are pointless? Does or can student modelling or adaptivity fit into the constructivist's paradigm of learning? Mayo (2001) gave an alternative view by pointing out following statements:

1. Novice students often lack the meta-cognitive skill required to explore in a constructivist learning environment. Getting lost in the learning environment is often the cause of disengagement or lost of motivation (Bouillion & Gomez, 2001; Kashiara et al., 2000).
2. New students to a domain require direct guidance as they have no previous experiences to build-on. Advocates of transitory model suggest that the students should initially be taught in objectivist approach before going to the intermediate-level where the constructivist approach could be applied.
3. The adaptivity can be directed to support constructivist principles.

In constructivism, good problems are those that are able to stimulate the exploration and reflection necessary for knowledge construction (Brooks & Brooks, 1993). Efficient exploration is one of the key skills required to be able to actively engage in a constructivist learning-environment. When the inability to demonstrate exploratory skill is detected, the student model ought to advice the tutoring module to take responsive actions to support the skill. However, none of those student models mentioned above are designed to support this new demand.

7.8: Summary

Modelling of individual differences in cognitive processing is one of the areas where the full potential of student modelling has not yet been achieved (Lovett et al., 2000). This idea may be novel in its wider ambitions, but several HCI researchers have noted its potential (Barnard, 1993; Green, 1994; Tweedie & Barnard, 1992). In the next section of this thesis, a new approach of student modelling will be introduced. This approach involves the cognitive traits (CTs) discussed above, and is thereof named Cognitive Trait Model.

Chapter 8: Cognitive Trait Model

8.1: Introduction

It is the increasing need to model student's cognitive abilities/ traits that give rise to a completely new approach of student modelling: the Cognitive Trait Model (CTM). The ultimate aim of this new approach is to provide fine-tuned system adaptivity to support the cognitive processes of students during learning.

It has to be clearly understood that the purpose of CTM is not to replace performance-based student models but to complement. Student performance models (state models and process models) record dynamic student-domain specific data whereas CTM stores those student attributes, which could be multi-dimensional or stochastic, are relatively persistent over time and across different domains. These attributes, therefore, make the learning environments incorporating CTM have better possibility to provide adaptivity by fine tuning student models.

8.2: New Perspective for Student Modelling

The ultimate goal of Cognitive Trait Model (CTM) is to have a student model that can be persistent over a long period of time and consistent across a variety of domains, thus the CTM is perfectly suitable for those students who aim to proceed on life-long learning. It is not essential for CTM to be able to predict the student behaviour at the very first time or first few times the student use the system, but instead, what the students need is a student model that can grow to know them very well over time.

This changes the idea of the student model that is just a database sitting on the server and is full of numbers for only a particular task. The CTM offers the role of learning companion which can be consulted by and interacted with different learning environments about a particular student. The CTM can still be valid after a long period of time due to the more or less persistent nature of cognitive traits of human

beings. When a student encounters a new learning environment, the learning environment can directly use the CTM of the particular student, and doesn't need to "re-learn the student" from scratch again. The CTM can be saved to removable media and access every time the student starts up a learning session. In this sense, the CTM is like a learning companion, and can be analogous to those health records that can be carried by people in smart cards.

In order to enable the implementation of CTM, a dedicated mechanism is developed, which is able to gradually increase the CTM's understanding towards the students. This mechanism is called "Individualised Temperament Network".

8.3: New Approach for Student Modelling

CTM could enable the learning environments to provide fine-grained adaptivity that take each individual student's cognitive abilities and resources into account. Traits are relatively stable over a long-period of time and persistent across different domains. Ideally, such a modelling effort would be able to predict individuals' performance in a new task with no new free parameters, presumably after deriving an estimate of each individual's processing parameter from previous modelling of other tasks.

The new approach to model cognitive traits raises a novel issue, i.e. how can cognitive traits be modelled? This issue can be clarified by comparing the CTM approach to other student modelling approaches mentioned in previous chapters.

Most of the modelling approaches mentioned in previous chapters try to represent:

- the students' competence state (conceptual and procedural knowledge) by recording how much content has been accessed, by the student's navigational path, and how much has been learned by the student by analysis the solutions to questions (state models: Smith, 1998; Staff, 2001; Urban-Lurain, 1996; Burton & Brown, 1982, skill tracing: Han et al. 2001),
- the students' understanding of the procedure for solving a task by comparing the problem-solving scenario with the expert's model (constraint based mode:

Martin's 1999; Problem-solving state tracing: Stankov, 1996; Anderson & Reiser, 1985), and

- the students' approach of the procedures for solving a task, by analysing the processes/steps that they have taken (process model: El-Sheikh, 1997).

While learning object is defined as "*any entity, digital or non-digital, that may be used for learning, education or training*" (LTSC IEEE, 2002), pages, concepts, questions, and topics can all be learning objects. Then each student model can be generalized as a representation of how the results of operations (read, answer a question) were carried out on learning objects, and which learning objects had been operated. On the other hand, in order to model cognitive traits, a level of abstraction has to be abandoned, students' behaviours have to be recorded in a granularity level descriptive enough so that the necessary aspects of the student attributes can be inferred by the learning environment.

There are many existing systems that not only record student's competence state, but also their behaviours for different purposes. CRISIM-Tutor (Zhou, & Evens, 1999) records student's behaviours in a relatively detailed manner, such as student reply history, student solution record, and tutorial history in order to provide fine-tuned adaptivity, feedback, and future instructional design references. The Geology Lab Simulation (Andris, & Stueber, 1994) records the sequence of the screen the student visited in every session (for linearity and reversibility), the tasks performed can be derived from the sequence, and the time for every visit is also recorded in the hope that student's learning styles can be derived from the data collected. It is not hard to see the trend that more detailed adaptivity is required at this stage of student modelling development by collecting interaction data in a fine-grained manner. Keates and Robinson (1997) used the Model Human Processor, which is a mechanism for timing an action performed taking into account of perceptual, cognitive processing, and motor response time, to develop and calibrate user models based on motion impaired users' performance parameters.

In discussion of cognitive science and human computer interaction, Barnard and May (1993) clearly pointed out the vital role cognitive theory should have in system design, they said: "*where successive experience of design can lead to the development of craft*

skills that can only be communicated by general 'rules of thumb' or guidelines, cognitive ergonomics attempts to develop principles that are applicable in specific design contexts, and which make specific design prescriptions. ... The contribution of cognitive science to HCI research is to provide the conceptual background against which the engineering principles can be understood. Engineering practice (showing that the principles work) is one form of validation; scientific explanation (showing why the principles work) is another. Together engineering and science produce a synthesis to support design". In the same vein, the work of CTM is derived from the principles of human cognition (Baddely, 1986; Salthouse et al., 1989; Daneman, & Carpenter, 1980; Richards-Ward, 1996; Miller, 1956), especially from the work about trait analysis.

8.4: Architecture For Incorporating Cognitive Trait Model

In order to maintain the logical modularity, a multi-component architecture of student model is required (figure 8.1). The student performance model can be either a state model or a process model depending on the nature of the domain (process model is more suitable for procedural knowledge as it models the process of a problem solving procedure).

The interface module (figure 8.1) is the interface to the student. In web-based systems, the interfaces module is generally implemented inside the web browser. Due to the state-less nature of the HTTP protocol used by the web browsers to communicate to the servers, recording student actions at a very fine-grained level becomes a technical difficulty. However, a notation of learning object (LOM, 2002) is used to organise the contents of the web pages. "*Learning objects are elements of a new type of computer-based instruction grounded in the object-oriented paradigm of computer science*" (Wiley, 2000). By this definition, all digital instructional elements can be thought of as learning objects. If learning objects are the fundamental units for recording students' actions, and if pages are prepared and organised into learning objects, then it indeed makes no difference between recording student behaviours/actions in closed system and web-based systems. The concepts and definition of learning objects will be discussed in greater details in later chapters. All

student actions are recorded in the student behaviour history as the interaction between the student and learning objects.

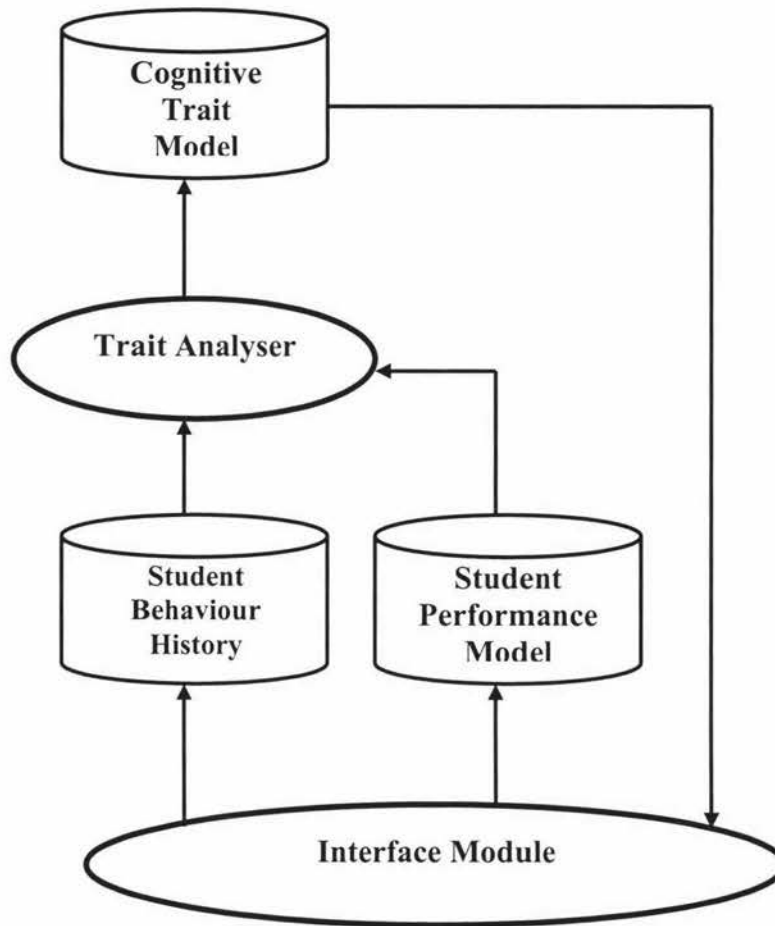


Figure 8.1: Architecture for Incorporating Cognitive Trait Model

Student performance model records student-domain specific data, which can either be student states (state models) or the skills that the student has acquired so far (process models). Some of the data in the performance model may be required during the process of trait analysis. For example, in order to check whether an excursion that a student has taken during a learning session has any effect on the performance of the student, the performance model needs to be consulted for the learning result (assessment result). The definition of a session needs to be given here which is:

A session is a continuous interaction between the system and the learner, and the length of the session is defined by its boundary.

In web-based systems, the boundary of session starts from a successful login and ends with an intentional logout or automatic forced logout after a certain length of inactivity.

While excursion can be a rich source to expand learning experiences and scopes, it could also distract students away from what they should be focusing on. Trait analyser takes the logged information in the behaviour history and sometimes also from the student performance model, and performs the analysis. Details about trait analysis and the structure of the trait analyser are discussed in later chapters.

Systems with similar granularity levels include the READY (Jameson et al., 1999) that tries to model users' time limitation and distraction from the system-user dialog, and Meyer's (1997) that attempts to find out the age group differences in WWW navigation, log all navigational movement between and within pages, and asks participants to think aloud while searching.

Jameson et al.'s (1999) approach on collecting student behaviour seems to be appropriate to the level of granularity needed for CTM. They suggested:

- incremental use of sparse evidence;
- integration of unreliable evidence from a diverse set of observations (in particular, concerning causes and symptoms of resource limitations); and
- explicit reasoning about the ways in which the user's resource limitations change during interaction.

The cognitive trait model (CTM) represents those selected traits that are (regarded) relevant to the domain. Possible traits could be working memory capacity, inductive reasoning skill, associative learning skill, and information processing speed. Selection of traits has to take into consideration following issues:

- The nature of the domain: When the purpose of teaching procedural skills dominates in a domain, it is probably a good idea to model students' inductive reasoning skill and working memory capacity. This is because in those domains, students tend to learn only the skills for solving problems and some

basic knowledge needed to solve those problems. There is the tendency to not learn as much about why the problem can be solved (Stauffer, 1996). Therefore, the ability to infer the principle from experiences of problem solving becomes an essential skill in acquisition and assimilation of procedural knowledge. Consequently, learning a procedure usually involves remembering a series of steps, and working memory capacity differences among individuals can cause large variance in the learning results.

- The audience of the VLE: Since the audiences of the learning system may vary in their ages, it is essential to take the working memory capacity into account in designing strategies for adaptation. Working memory decline is an observed phenomenon in older adults (Salthouse, 1989).
- The nature of the curriculum: If the domain teaching style is deductive (as most of the early education for small children is), the effort to model students' inductive reasoning skill is redundant whereas the need for associative learning skill increases.

8.5: *Summary*

The ultimate aim of the Cognitive Trait Modelling approach is to provide fine-tuned system adaptivity to support the cognitive processes of students during learning. It has to be clearly understood that the purpose of CTM is not to replace performance-based student models but to complement them. The CTM offers the role of a learning companion for the learner's life-long learning.

Chapter 9: Learning Objects and Relation-Based Browsing Pattern Analysis

9.1: Introduction

For web-based systems, recording students' interaction with the learning environment is not as easy as in a closed system. A method to solve this problem is by organising learning materials into learning objects and recording the students' interaction with learning objects. Furthermore, organisation by the concepts of learning objects ease the effort of reusing the learning objects in the future. In the first half of this chapter, the concept of learning objects is introduced and explained.

A dedicated browsing pattern analysis is designed in order to interpret the behaviours students exhibit during their interactions with the learning environment. Incorporating the concepts of and the definition of learning objects, a relation-based browsing pattern analysis strategy, which is capable performing domain-independent analysis, is made possible. The second half of this chapter discusses the relation-based browsing pattern analysis and compare it with other commonly used browsing pattern analysis strategies.

9.2: Learning Objects

“Learning objects” is a popular term used in the context of technology-based learning. *“Learning objects are elements of a new type of computer-based instruction grounded in the object-oriented paradigm of computer science”* (Wiley, 2000). As its name indicates, the main purpose of the learning object concept is reusability, and standards like the Learning Object Metadata (LOM, 2002) from the Learning Technology Standards Committee (LTSC) facilitate compatibility and interoperability of materials.

The advantages of learning objects over traditional instructional media (such as books or video tapes) are described by Wiley (2000) as follows:

- Learning objects are digital entities deliverable over the Internet, which implies that they are accessible for many people simultaneously at any time.
- Users of learning object can collaborate and be benefited immediately from new versions.

Learning objects are used as the building blocks for an exemplary prototype that demonstrates how the CTM could be integrated in VLEs. Relation is one of the metadata of learning objects defined in the LOM v1 standard. A relation-based browsing pattern analysis is developed in this thesis to cater for the need of analysing learners' actions and inferring their cognitive capacities, and it is discussed next.

9.3: Navigational Pattern and Relation-Based Analysis

The navigational behaviour of the users is seen by many researchers as the trace for understanding aspects of the users (Sun & Ching, 1995; Mullier, 1999), and hence a valuable resource for the construction of student models. There are two most dominant navigational pattern analysis methods at the time of this writing; they are content-less navigational pattern analysis and content-based navigational pattern analysis.

9.3.1: Content-less and Content-based Navigational Pattern Analysis

Mullier (1999) took the approach of content-less browsing pattern analysis. In his analysis, every learning object is treated as a node in the hyperspace, and every node is treated equally regardless of its content. The focus is on the browsing behaviours, and certain browsing pattern may indicate certain type of browsing strategy, which can tell the learning environment about what the students are actually doing. Those browsing strategies include activities such as scanning, browsing, searching, exploring, and wandering Mullier (1999). The most important advantage of content-less browsing analysis is its domain-independence.

At first glance, many of the manifests described in the previous chapter seem to exhibit similar characteristic as that of the content-less browsing pattern such as “constantly reverse navigation”, or “non-linear navigational pattern”. However, for the purpose of cognitive trait modelling, content-less browsing analysis would not be sufficiently accurate enough. For example, while determining a student’s working memory capacity, it can not be creditable that “constant revisit” implies low working memory capacity (LWMC) without knowing that this pattern (constant reverse) may be the result of a student trying to look up the standard deviation table while doing statistical questions. Thus, the problem of inaccuracy, which results from content-less navigational pattern analysis approach’s domain-independence, renders this analysis approach not suitable for the requirement of CTM.

The opposite approach to content-less browsing analysis is called content-based browsing analysis (Mullier, 1999). It is the primary analysis method for content-based student models (Mitrovic & Olhsson, 1999; Staff, 2001; Urban-Lurain, 1996; Smith, 1998). The semantic information of every node is recorded by the learning environment and by recording the student’s progress based on the semantic information; the learning environment then makes inferences about the students (their interests, skill levels, etc.), and stores those inferences in the student model. Mullier (1999) pointed out that the most serious drawback of the content-based approach is that it is totally domain-dependant. The similar effort of analysis has to be carried out for every new topic or curriculum.

The approach of this work can not be content-less as it is too inaccurate, nor can it be content-based as the purpose of the Cognitive Trait Model is for a model that can be valid across multiple domains. A completely different perspective is needed on the analytical method required by CTM. It is not the content that should be of focus, but it is the relationships among learning objects.

9.3.2: Relation-based Browsing Pattern Analysis

Relation-based approach for browsing behaviour analysis is a field that is not yet extensively explored at the time of this writing. It is devised based on the need of the analysis required for CTM. Other potential applications of this approach could be possible but not yet researched. Please note that the existence of relation-based approach is not intended nor could it possibly completely replace any existing browsing pattern analysis method. It is analogous to that of the CTM that is not to replace any content-based student models, but only to supplement them in order to provide in-time assistant to learners based on a cognitive measure. The analysis method is discussed in length in the following chapters, and the rest of this chapter covers the fundamental building blocks of relation-based analysis; namely the learning object relations.

9.4: Learning Object Relations

A learning object in a learning context may relate to other learning objects in the same context or outside the current learning context. In hypermedia learning environment, relationships of learning objects could be instantiated as links and they provide means of navigation in the learning environment. If all learning objects are thought to be nodes in the hyperspace of learning, then their relations are the connections from node to node.

In every learning environment, learning objects relate to learning objects differently. For example, learning object B1 may be related to learning object B by an “IsPartOf” relation, whereas learning object B may be related to each one of learning object B1 and B2 by a “HasPart” relation. There are many types of relations among learning objects, and they can be put into categories such as IsPartOf and HasPart. LOM v1 (2002) has defined 12 different types of categories based on Dublin Core. LOM v1’s relations will be the basis of this work, and will be discussed in detail later in this chapter.

Categorisation of relations enables relation-based analysis to be domain-independent, which is the most important advantage of content-less browsing pattern analysis over content-based analysis. In addition, relation-based analysis still contains some

semantic information of learning objects which makes it possible to perform analysis that combines the strengths of content-base and content-less analysis.

By LOM v1 (2002)'s definition, for any single relation to exist, it must be directional and binary (learning object A to learning object B), as illustrated in figure 9.1.

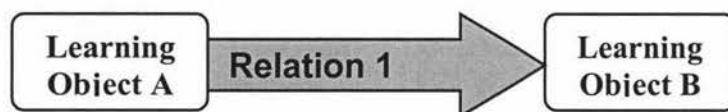


Figure 9.1: Direction of relation

However, every learning object can have more than one relations (figure 9.2).

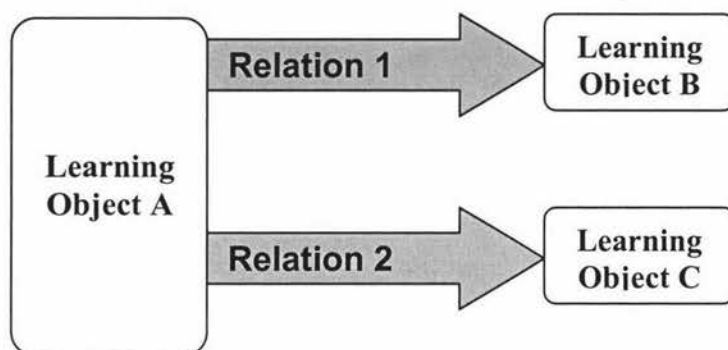


Figure 9.2: Multiplicity of relations

There are studies on adaptive multimedia that have already identified and categorised different types of links used in both the Internet and web-based learning systems. Kinshuk et al. (1999) listed seven type/categories of links used in web-based learning environment, they are:

- Direct successor links – They are links that lead to the successive domain concept/unit after the learning task in current concept/unit is fulfilled.
- Parallel concept links – They are links that lead to the analogous concepts that are related to the domain concepts. Comparison and contrast are available through those links for those who wish to increase their knowledge in width.
- Fine-grained links – They are links that lead to detailed explanation of the current concept. They provide and preserve finer granularity of domain concepts, more accurate and more specific information to those who really need in-depth understandings.
- Glossary links – They are links that lead to the definitions of domain concepts/terms.

- Excursion links – They are links that lead to related learning that is usually outside current context. They are sources that learners can find other related knowledge of their interests, and also create the possibility for accidental learning to take place (Reisberg, 1997).
- Problem links – They are links that lead to problems/exercises of the unit.

Relations are abstractions of the links and they can also be categorised. As stated above, the LOM v1's definition is the basis for relations that are used in this work. It is the intension to be as “conforming” as possible to LOM's standard and maximise semantic interoperability, so that there will be only extensions of the LOM's relations, not replacement of them. In the description of learning object's relations, LOM v1 (2002) proposed that relations are those that a learning object may have towards other learning objects:

- IsPartOf: is part of
- HasPart: has part
- IsVersionOf: is version of
- HasVersion: has version
- IsFormatOf: is format of
- HasFormat: has format
- References: references
- IsReferencedBy: is referenced by
- IsBasedOn: is base on
- IsBasisFor: is basis for
- Requires: requires
- IsRequiredBy: is required by

IsPartOf and HasPart define learning object's hierarchical relationship. IsVersionOf and HasVersion describe that one learning object is an instantiation of another one. IsFormatOf and HasFormat are used to specify the formative information that a learning object contains of another one. Referential relationships are catered by References and IsReferencedBy. IsBasedOn and IsBasisFor relations describe that a learning object is fundamental to another and finally, Requires and IsRequiredBy point out the prerequisite relationships.

These relations are based on Dublin Core (LOM, 2002) and provide a thorough coverage of relationships between data. A notable similarity exists between the LOM's relations and the links described in MR approach. The comparison is discussed below.

The direct successor link can be expressed by the *IsBasedOn* and *IsBasisFor* relations because the succeeding is based on its successor. The *IsPartOf* and *HasPart* relations' hierarchical nature best suit the definition of the fine-grained link. Basically, the glossary link can be substituted by the *References*, and the *IsReferencedBy* relations. Any other learning objects particularly designed for referential purposes can use those relations, and therefore glossary links are only a subset of *References* and *IsReferencedBy* relations.

The other three type of links left un-addressed by those of LOM's relations are parallel concept link, problem link, and excursion link. These three types of link are not covered or can be translated in LOM's terminology and are discussed in detail below.

Parallel concept links lead to those analogous concepts that are related to the current concepts in focus. Parallel concept links occur frequently in comparative studies and in explanations of a theory or principle's applications. An example of parallel concept could be the cause of raining and the formation of dew on the window. This type of link (relation) between learning objects occurs frequently in educational context because the ability to create analogies and to be able to utilise theories for practical purpose are highly valued. Therefore, one relation is named here "Parallels" to be an extension of current LOM's set of relations.

Problems provide evaluation for the learning, and a problem link points to an evaluation unit of the learning topic. An evaluation unit is expected to be a learning object in this discussion and will be called an evaluative learning object. Therefore, there are two new types of relations required for this purpose, and they are named: *Evaluates*, and *EvaluatedBy*.

One evaluative learning object can provide evaluation for multiple learning objects. For example (figure 9.3), the evaluative learning object for Topic A provides/is expected to provide evaluation for all learning objects (concepts/ skills) in Topic A. So there could be physically only one evaluative learning object for Topic A, but this evaluative learning object provides evaluations for all the learning objects that conceptually belong to Topic A.

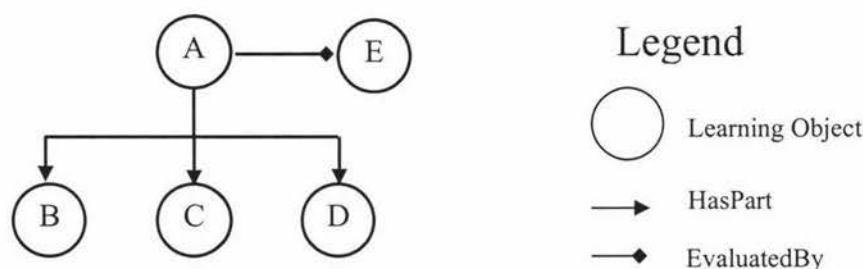


Figure 9.3: Evaluative Learning Object

In figure 9.3, learning object A has three parts, i.e. learning object B, C, D, and an evaluation, i.e. learning object E. In figure 9.3, even though there is only A that E evaluates explicitly, but implicitly E also evaluates B, C, and D.

The result of the evaluation can be recorded directly as a property of evaluative learning objects or it could be interfaced by a module to a content-based student model. What is discussed here is only the logical relationship; the implementation details are left for the system designers.

Excursion links occur frequently in an Internet-based systems. Excursion links lead to other learning objects, which are usually outside the current learning context, but are related to the current learning object. Resiberg (1997) pointed out that excursion links provide chances for accidental learning to take place, and therefore may lead to a much wider learning experience. This type of relation is rare in traditional learning media (books, or videotapes), which are usually closed and focused, and this maybe the reason that it is not considered as necessary in Dublin Core's specification and subsequently the LOM. For the reason that Wiley (2000) deliberately defined learning object must be Internet-based, there is obvious need to include this new relation in order to provide more descriptive relations of learning objects. The new relations are

named here as *ExcursionTo*, and *IsExcursionOf*. After the extension, the new list of relations is:

- *IsPartOf*: is part of
- *HasPart*: has part
- *IsVersionOf*: is version of
- *HasVersion*: has version
- *IsFormatOf*: is format of
- *HasFormat*: has format
- *References*: references
- *IsReferencedBy*: is referenced by
- *IsBasedOn*: is based on
- *IsBasisFor*: is basis for
- *Requires*: requires
- *IsRequiredBy*: is required by
- *Parallels*: parallels
- *ExcursionTo*: excurses to
- *IsExcursionOf*: is excursion of
- *Evaluates*
- *EvaluatedBy*

9.5: *Summary*

The concept of learning object is used to organise the learning materials in a learning environment and thus provide a method to structure the domain and make the reuse of learning materials easier.

The relation-based navigational analysis has the benefit of domain-independence that is similar to that of in content-less browsing pattern analysis. But it has more semantic information than what the content-less browsing pattern analysis could provide and thus is free of the inaccuracy problem embedded in the content-less browsing pattern analysis. And because of its independence, once an analysis policy is created, it is

possible to use the policy to a new domain, but more research will be required to verify this possibility.

The LOM's (2002) notation of relation is used, but the list is expanded in this chapter. The expansion strictly follows the suggestion of the Learning Technology Standards Committee (LTSC) and the resulted list is conforming to the LOM standard and therefore the interoperability is preserved.

Chapter 10: Trait Analysis

10.1: Introduction

This chapter expands and elaborates the Trait Analyser in terms of its functions and structure. A very important cognitive trait, working memory, is selected in this thesis and analysed in great details. This analysis exemplifies the analysis process that could be carried out for each individual cognitive traits discussed in the chapter of Exploration Space Control Formalisation, and a prototype of a CTM is built based on the trait, working memory, analysed in this chapter. The prototype is discussed in chapter thirteen.

10.2: Trait Analyser

The trait analyser is the main component to determine the input into the CTM from analysis of student behaviours. The trait analyser has to carry out the task of cognitive analysis of a list of actions. The cognitive analysis has to be backed up by thorough research on the theories of human cognition. One of the cognitive ability, working memory, is accounted as a necessary requirement for a vast number of human activities (Henry, 2001; Calvo, 2001), and it also takes a vital role in the process of human learning no matter whether from the perspective of behaviourism (Tarpay, & Mayer, 1978; Wickelgren, 1977), cognitivism (Tarpay, & Mayer, 1978), or constructivism (Cofer et al., 1975). A detailed research on working memory for its definition, characteristics, and the effect on learning will be presented next along with an exemplary analysis to illustrate the procedure and the task of the trait analyser.

10.2.1: Structure of Trait Analyser

Trait Analyser (figure 10.1) takes inputs from the Student Behaviour History, perform the analysis, and updates the Cognitive Trait Model according to the result of the

analysis. Trait Analyser can further be divided into sub-components, and the analysis process can be broken into sub-processes.

Pattern Detector examines the record of the student's actions to find out patterns that give clues about the student's cognitive ability. Those signs are called manifests and they are abstract description of implementation patterns that are pattern observable in terms of learning objects and their relations. Manifests and implementation patterns are discussed in details in the next chapter.

The results of the Pattern Detector is taken as inputs of the Individualised Temperament Networks (ITNs). An ITN is an artificial neural network like component that adjusts itself according to the result of the pattern detector. Each cognitive trait is presented as one ITN. Multiple number of cognitive traits form a cluster of networks each is independent to others. ITN is discussed in details in chapter twelve.

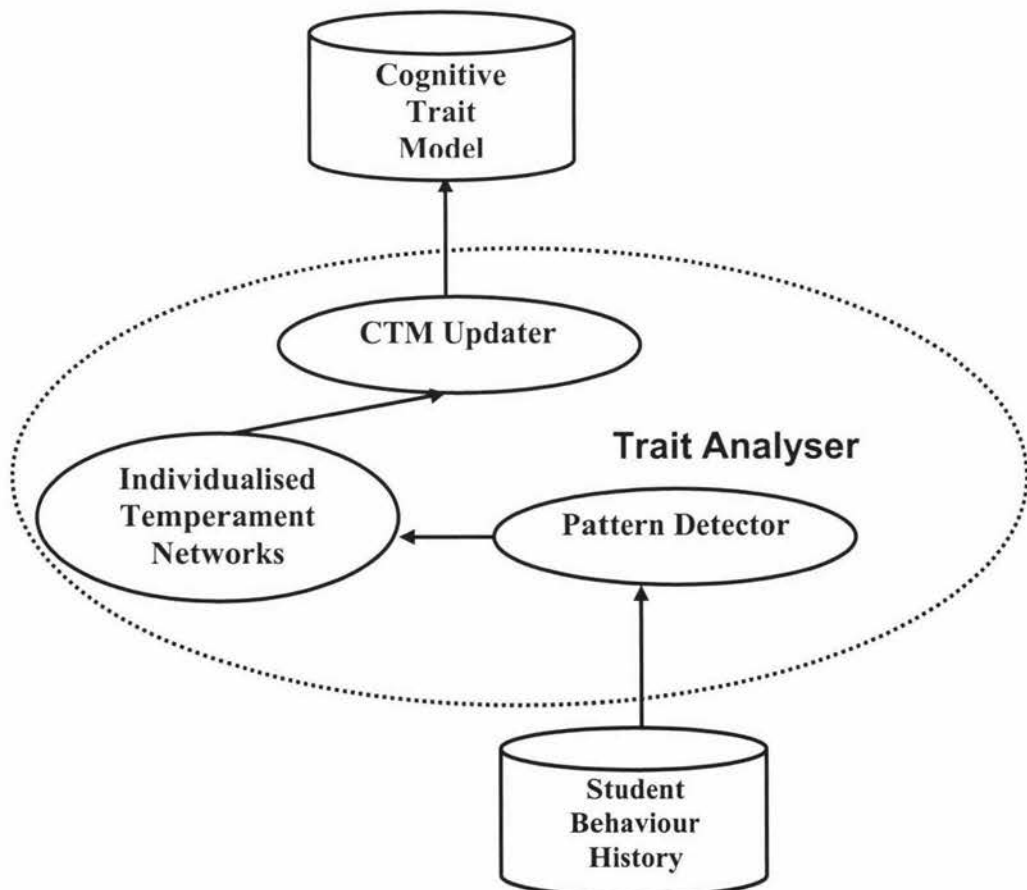


Figure 10.1: Structure of Trait Analyser

The CTM Updater makes updates to the CTM for the ITNs. The main reason for this component is to achieve the separation of data and the program logic.

In this thesis, one cognitive trait, working memory, is used to create a prototype of CTM. The analysis of working memory is required as the theoretical background of the Trait Analyser, and is discussed next.

10.3: Analysis of Working Memory

Working memory is also referred as short-term memory. It denotes the memory capable of transient preservation of information and is functionally different from the memory that stores historical information (the long-term memory). Richards-Ward (1996) named it the Short-Term Store (STS) to emphasis its role of temporally storage of recently perceived information. STS allows us to keep active a limited amount of information (roughly 7 ± 2 items) for a brief period of time (Miller, 1956).

The term working memory refers to the same construct as the STS does, in terms of the capacity of transient storage. In addition, it also acknowledges that cognitive processes take place in working memory. Cognitive scientist Baddely (1986) assumed that the major function of the working memory is to temporarily store the outcomes of intermediate computations when solving a problem, and allow operations of further computations on these temporary outcomes.

Baddeley (1992) tried to study and understand the working memory by decomposing it into components. The structure of working memory was described as a control-slave system comprised of the Central Executive (controlling component), Phonological Loop (slave component for verbal information), and a Visual-Spatial Sketch-Pad (slave component for graphical information). The Central Executive takes the role to monitor and control the output of the two slave-systems and to select what is relevant for potential processing (Richards-Ward, 1996). Metaphors used for working memory include "blackboard of the mind" (Reddy, 1980); "mental sketch-pad" (Baddeley, 1986), and "on-line memory" (Goldman-Rakic, 1987). Although these metaphors capture the essential idea of the transient storage capacity of working

memory, they are still suggestively emphasizing the role of the storage functionality of the working memory. The Central Executive which is conceptualised as very active and responsible for the selection, initiation, and termination of processing routines (e.g., selecting, encoding, storing, and retrieving) is still the least understood aspect of the working memory (French, n.d.).

As Baddeley (1986, 1992) defined working memory structurally, others defined it as a process (Salthouse et al., 1989; Daneman, & Carpenter, 1980). Salthouse et al. (1989) proposed that working memory consist of (1) a storage capacity sensitive to the number of items presented, and (2) an operational capacity sensitive to the number of operations performed on items. Salthouse et al. (1989) found that young adults have higher operational capacity than older adults especially among the highly capable participants. Little or no difference was reported on the storage capacity across the age differences. Therefore, they concluded, at that time, that it was the operational capacity that causes age-related working memory decline. A further study of the operational efficiency of working memory showed that it was not the operational capacity (number of operations allowed) that contributed the most to the efficiency of working memory, but it actually was the speed of execution (e.g. comparison speed) that determined the performance of the overall system of working memory (Salthouse & Babcock, 1991). Even though, these two different points of view do not agree on a common structure of working memory, they both agree that working memory consists of both storage and operational sub-systems (Richards-Ward, 1996).

In addition to the storage capacity, Atkinson and Shiffrin (1968) defined working memory functionally as the gateway allowing information to be transferred to the long-term memory. This definition stresses on the ability to channel the incoming isolated information (as received by our senses) to the semantically networked structure in the long-term memory. This involves a great degree of cognitive efforts such as interpretation, translation, association, memorisation, and so on. This functionality is comparable to the central execution unit mentioned above and essentially it transforms and transfers the messages from the short-term storage system into the long-term one. The transformation process invokes the formation of rules (data with operational application) from pure data (in the form of incoming

messages) and the transfer process filters which rules and data are to be stored for long-term and which are to be discarded.

Several studies have shown that age-related performance of young children and old adults compared with young adults can be characterized by the inability to retain information in working memory while simultaneously processing other information (Case, 1995; Salthouse & Babcock, 1991). Deficiencies in working memory capacity result in different performances in a variety of tasks. Examples of tasks affected could include natural language use (comprehension, production, etc), recognition of declarative memory, skill acquisition and so on (Byrne, 1996).

An empirical study by Huai (2000 as quoted by Kommers, 2000) showed that students with holistic learning style also have a significantly smaller short-term working memory (quoted by Kommers, 2000) but have remarkably higher learning effect in the long run, whereas the students with serial learning style (highly capable to follow and remember sequentially fixed information) have better short-term working memory capacity but poorer learning result in the long run. This point shows the intricate relationship between humans' inbuilt abilities and how different learning styles are adopted to circumvent any deficiencies in those abilities. The navigational strategies adopted by serial learners are linear whereas holists sometimes do better by jumping directly to complex concepts (Felder, 1988).

The discussion so far has covered aspects of working memory from the perspective of its definition and structure. Here is the summary of the properties of working memory discussed so far, and their implications for cognitive analysis:

- Working memory temporarily stores material for recall for only a few seconds (Atkinson, & Shiffrin, 1968; Byrne, 1996). Constantly revisiting learned materials at very short intervals indicates sign of low working memory capacity.
- As new items enter into working memory, other items become harder to access, and the cognitive system becomes less efficient (Byrne, 1996; Baddeley, 1986). This is called the displacement or interference. Less tolerance of displacement or interference indicates sign of low working

memory capacity. Tolerance level can be demonstrated as the ability to learn other side-information, without been led astray from the main theme. Side information is usually represented by excursions via excursion links (Kinshuk et al., 1999), tips, warning, cases, and so forth.

- Working memory has a limited capacity for storage (Miller, 1956; Salthouse et al., 1989). Constant short-distant reversed navigation indicates sign for low working memory.
- Working memory has a limited capacity for operation (Miller, 1956; Salthouse et al., 1989). Ability to perform tasks simultaneously indicates sign of high working memory capacity.
- Unlike the long-term memory, in which interference is ascribed the main cause of retrieval failure (Reisberg, 1997), simple execution speed (e.g. comparison) greatly determines the overall performance of working memory due to the characteristic of rapid decay (Salthouse, & Babcock, 1991; Richards-Ward, 1996). Higher comparison speed implies higher working memory capacity. The amount of time spent on comparison tasks provides indications for working memory capacity.
- Working memory is the intermediate storage to and from long-term memory (Atkinson, & Shiffrin, 1968). The ability and speed to retrieve knowledge from the long-term memory provides signs for working memory capacity.
- During a long sequence of calculation or procedures, frequently missing steps or lost components indicates signs for low working memory (Campbell, Charness, 1990).
- Working memory capacity varies with age (Salthouse, & Babcock, 1991; Kausler, 1991). Younger adults indicate signs of higher working memory.
- For those with higher working memory capacity, it is likely that they follow the curriculum sequentially (Huai, 2000 as quoted by Kommers, 2000; Felder, 1988). Therefore, there will be less trans-state violation. Trans-state violation means that the student is doing something that s/he should have finished (done) in other previous state(s).
- For those with lower working memory capacity, it is likely that they follow the curriculum non-linearly (Huai, 2000 as quoted by Kommers, 2000; Felder, 1988). Therefore, there will be more trans-state violation.

- Inferential ability is dependent on working memory capacity (Masson & Miller, 1983, quoted by Richards-Ward, 1996). Inferential ability is the ability to understand high-demand text or complex concepts. The inferential ability takes the role of bridging the gap between the necessary semantics and thus it is called bridging inferential (Calvo, 2001). High number of inter-navigation activities between the current concept and its sub-concepts provide indication of lack of bridging inferential ability and thus lower working memory capacity.

10.4: Summary

In order to cater for the required qualitative analysis, the trait analyser adopts the strategy that makes incremental use of sparse evidence, integration of unreliable evidence from a diverse set of observations, and explicit reasoning about the ways in which the user's resource limitations change during interaction (as suggested by Jameson et al., 1990). The analyser can then perform inferences on students' cognitive traits and store the results in the CTM. The CTM can assist instructional decisions the VLE has to make in conjunction with any performance-based models.

Working memory is an important cognitive resource in many aspects of the learning process (cognitive, retrieval from memory, and so on). By the analysis of the working memory, several implications are identified that could be used to indicate one's working memory capacity. The list is not exhaustive, but provides a ground that the trait analyser and hence the CTM could be built on.

Those implications found as the result of the analysis are given the name "manifests", as they provide signs for the capacity of one's working memory. Manifests have to be translated a step further into expression that could be implemented, and those expressions are called implementation patterns. Manifests and implementation patterns of working memory capacity are discussed next.

Chapter 11: Manifests and Implementation Patterns

11.1: Introduction

The results of the theoretical work described in the previous chapter is translated into implementable expressions which can be divided into two levels: manifests and implementation patterns. Manifests of the capacity of working memory are discussed first, and then the translation of manifests into implementation patterns follows.

11.2: Manifests

Each of the point mentioned in the result of the analysis of working memory gives sign and indication about the student's working memory capacity. However, in order to enable a learning environment to make judgement about the current student's working memory capacity, each of the point has to be translated into more definite terms, and they are called manifests in this discussion. A manifest is a defined student's behaviour pattern or attribute, observable during the student's learning process. A manifest can be a single observable student action (e.g. comparison speed) or it can be a complex pattern comprised of a long sequence of observable student actions (e.g. frequently revisiting learned materials). A manifest can also be a student's attribute (e.g. age). The implications of analysis of working memory can be translated into the following manifests:

For low working memory capacity (LWMC):

1. In long sequence of calculation or procedures, frequently missing steps or lost components manifests LWMC
2. Non-linear navigational pattern manifests LWMC
3. Unable to absorb side information while still remain progressing in the main track manifests LWMC.
4. Unable to perform tasks simultaneously manifests LWMC.
5. Low comparison speed manifests LWMC.

6. Unable to retrieve information effectively from long-term memory manifests LWMC.
7. For adults, older age manifests LWMC.
8. Constant reverse navigation manifests LWMC
9. Frequently revisits to already learned materials manifests LWMC
10. Unable to comprehend highly demanding text or concepts manifests LWMC.

For high working memory capacity (HWMC):

1. Performing long sequence of calculation or procedures, without missing steps or lost components manifests HWMC.
2. Linear navigational pattern manifests HWMC.
3. Being able to learn side information while still remain progressing in the main track manifests HWMC.
4. Being able to perform tasks simultaneously manifests HWMC.
5. High comparison speed manifests HWMC.
6. Being able to retrieve information from long-term memory efficiently manifests HWMC.
7. For adults, younger age manifests HWMC.
8. Rare (or none) reverse navigation manifests HWMC.
9. Infrequent (or none) revisits to already learned material manifest HWMC.
10. Being able to comprehend highly demanding text or concepts manifests HWMC.

Table 11.1 summarises these manifests:

Low Working Memory Capacity	High Working Memory Capacity
non-linear navigational pattern	linear navigational pattern
constantly reverse navigation	rare (or none) reverse navigation
frequently revisit learned materials	infrequent (or none) revisit learned material
unable to absorb side information while still remain progressing in the main tract manifests	able to learn side information while still remain progressing in the main tract
unable to perform tasks simultaneously	able to perform tasks simultaneously

low comparison speed	high comparison speed
unable to retrieve information effectively from long-term memory	able to retrieve information from long-term memory effectively
in long sequence of calculation or procedural, frequently missing steps or lost components	performing long sequence of calculation or procedural, without missing steps or lost components
older age	younger age
unable to comprehend highly demanding text or concepts	able to comprehend highly demanding text or concepts

Table 11.1: Manifest of Working Memory Capacity

Each of the manifests provides only indications to the working memory capacity. There could be contradictions among the manifest observed from the same student. For example, an adult student may have young age (manifests HPMC) and may still be using constantly reversed navigation (manifests LPMC). This is perfectly acceptable because an attribute, such as working memory, can be decomposed further into a group of sub-components or could be studied (looked at) from a number of different perspectives. Each perspective has its focus and method of explaining why this particular attributes work in a certain way. For example, Atkinson and Shiffrin (1968) defined working memory functionally, whereas Baddeley (1986, 1992) defined it structurally.

This work is not aimed to identify which perspective is valid and which should be abandoned, but rather, offer an approach devised to incorporate different perspectives. The approach is called Individualised Temperament Network (ITN) which provides a means for “natural selection” of the aspects/theories discussed above that is suitable for each particular student. ITN will be discussed in next chapter.

11.3: Implementation Patterns

The manifests discussed previously are the extraction of the results of analysis of working memory capacity. They provide a very general guideline for analysing learners’ behaviours in order to infer their working memory capacity. However,

sentences like “frequent revisits to already learned materials manifests low working memory capacity” are too general to be useful in terms of implementation. Therefore, more detailed account of those manifests is considered in this section.

Each of the manifests listed in table 11.1, except age, involves a pattern of student behaviour. For example, frequent reverse navigation is a manifest for LWMC, and is itself a pattern detectable in the log of the learner’s interaction with the system. However, in order to bring those patterns a step closer to implementation level, they have to be translated in terms of learning objects, properties, and relations of learning objects. Those detailed accounts are called the implementation patterns. Each of the implementation patterns for manifests in table 11.1 is discussed next.

11.3.1: Implementation Pattern for Navigational Linearity

Linear navigation implies that the learner is traversing the learning space in a way that was originally designed to be the main route. In many learning environments, it is the next button that guides the learner on the intended route designed by the curriculum designers. In terms of link type, it is the direct successor link, and in terms of relations, it belongs to the *IsBasisFor* relation, and *IsBasedOn* indicates the relation with opposite direction to the direct successor link. The linearity can be represented as follows:

In a single session, for all learning objects in a set of all visited learning objects, the implementation pattern of the linearity can be represented by the following equation:

$$LM = (No(B) / ISBASISFOR(V)) / (No(E) / EXCURSIONTO(V))$$

Where

LM is the measure of linearity

W is the set of all visited learning objects.

V is a learning object in the set of **W**.

No() returns the number of items in a set

ISBASISFOR (V) returns the set of learning objects that V has *IsBasisFor* relations with.

B is the subset of **ISBASISFOR(V)** where for every item **bi** in **B**, (**bi** $\not\subset$ **W**)

$\not\subset$ means not in the set of.

EXCURSIONTO(V) returns the set of learning object that **V** has ExcursionTo relations with.

E is the subset of **EXCURSIONTO(V)** where for every item **ei** in **E**, (**ei** \subset **W**)

\subset means in the set of.

In order to explain the linearity of a navigational path, consider the following illustrations (figure 11.1)

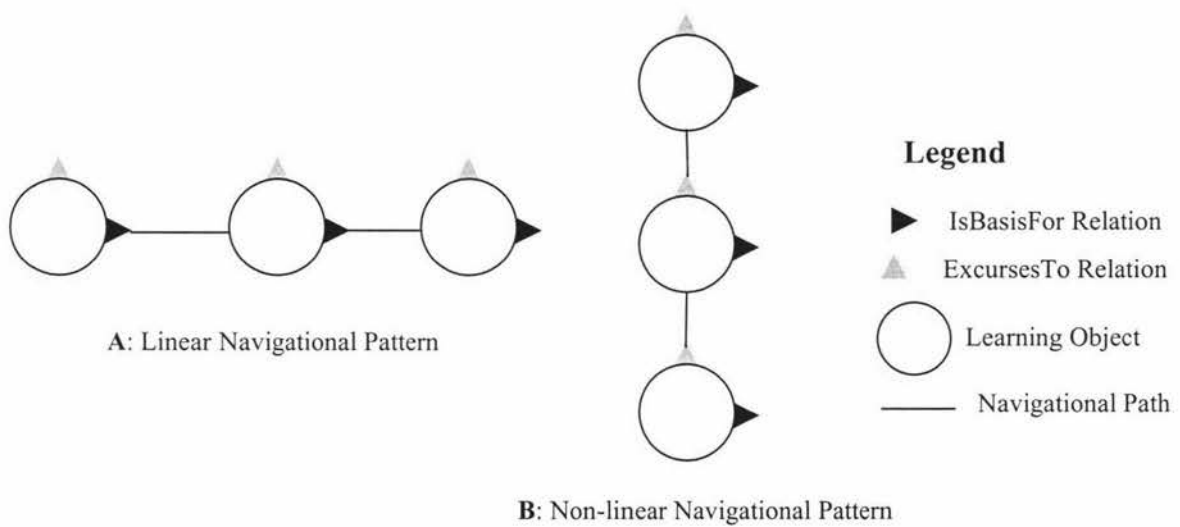


Figure 11.1: Linearity of Navigational Pattern

Figure 11.1.A shows a completely linear navigational pattern, where only the IsBasisFor relations (direct successor links) are involved, and none of the excursion links are travelled. Figure 11.1.B shows a completely non-linear navigational pattern, where none of the IsBasisFor relation is present but only the ExcursionTo relations are present

However in reality, it is quite possible to have a mixture of linear and non-linear navigational patterns in any session of learning interaction. Therefore, the linearity measure is designed for, as its name suggests, the purpose of measuring linearity of navigation.

In the equation of linearity measure, which is

$$LM = (No(B) / ISBASISFOR(V)) / (No(E) / EXCURSIONTO(V))$$

the **No(B) / ISBASISFOR(V)** accounts for the ratio of IsBasisFor relations visited in the session. It represents the ratio of learning objects (a set) that V has IsBasisFor relations with (**No(B)**) and not visited in a session (where for every item **bi** in **B**, (**bi** $\not\subset$ **W**), $\not\subset$ means not in the set of), to the total number of learning object that V has IsBasisFor relations with (**ISBASISFOR(V)**). Whereas (**No(E) / EXCURSIONTO(V)**) is for the ratio of ExcursionTo relations. It represents the ratio of learning objects (a set) that V has ExcursionTo relations with (**No(E)**) and visited in a session (where for every item **ei** in **E**, (**ei** \subset **W**), \subset means in the set of), to the total number of learning object that V has ExcursionTo relations with (**EXCURSIONTO(V)**). A ratio of the ratio of the qualified IsBasisFor relations to the ratio of qualified ExcursionTo relations is again taken for the final value of the linearity measure.

The reason of taking the ratio instead of the total number is that: there is no fixed number of IsBasisFor or ExcursionTo relations in any learning objects, and the linearity measure could be easily biased because one of the type of relations (IsBasisFor or ExcursionTo) can just outnumber the other type in a single learning object.

Exception could arise if there is no excursion in a unit. In this case, this implementation pattern is automatically deactivated.

11.3.2: Implementation Pattern for Reverse Navigation

The two manifests, reverse navigation and revisit already learned materials, have some degree of similarity, and manifests from the LWMC group are used for both of them in this discussion.

One of the navigational patterns that manifest LWMC is “constant reverse navigation” (Figure 11.2.A), which indicates that a learner goes back to learned material often. In many systems, it may be most obviously represented by the “Back” button. In a general sense, the course of frequent reverse navigation is caused by an insufficient capacity of the working memory to hold on that material that has just been accessed.

Another navigational pattern “frequent revisits to already learned material” (Figure 11.2.B), may seem different from the pattern shown in Figure 11.2.A. But the pattern “frequent revisits to already learned material” could be a generalisation of the pattern “frequent reverse navigation”. This is explained further in the discussion of the implementation patterns.

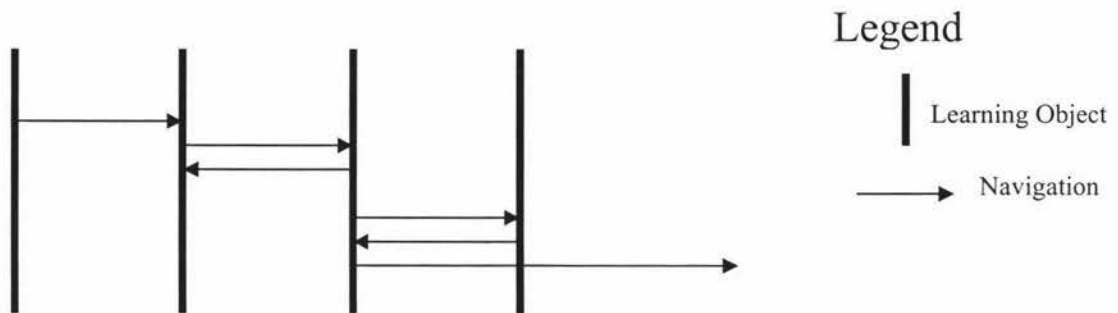


Figure 11.2.A. Constant Reverse Navigation

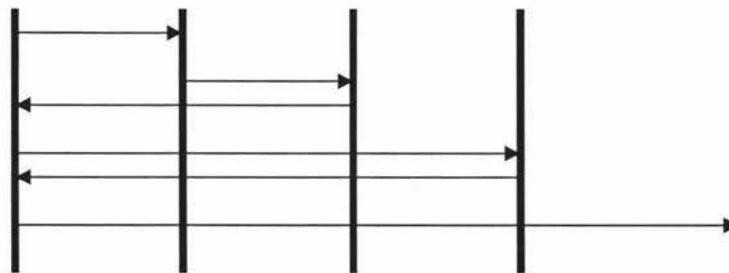


Figure 11.2.B. Frequently revisit learned material

In terms of working memory capacity, the causes for these two patterns are qualitatively the same. This is another reason that these two different patterns are treated as a single pattern “reverse navigation” in this work.

It must be noted that for the detection of this pattern, learning object has to be understood than just treated as an equal node to every other node. For example, if a learning object is designed as a reference (which could legitimately be a learning object by learning object’s definition), then it is perfectly sensible that a learner comes

back to this particular learning object often. That is a very serious limitation of content-less browsing pattern analysis.

Implementation pattern for reverse navigation can be represented as follows:

*In a session of learner's interaction, there exist a certain number of ordered pairs (R) that represents a navigational action and at least one of the items in the pair is visited. $R = (\text{fromLearningObject}, \text{toLearningObject})$, where **fromLearningObject** represents the source of the navigational action and **toLearningObject** the destination.*

The phrase “a certain number of R ” will be discussed in more details below. This implementation pattern looks at each navigational action in a session. It could be easily understood conceptually if all visited learning objects are arranged in a sequential manner according to the time stamp on them.

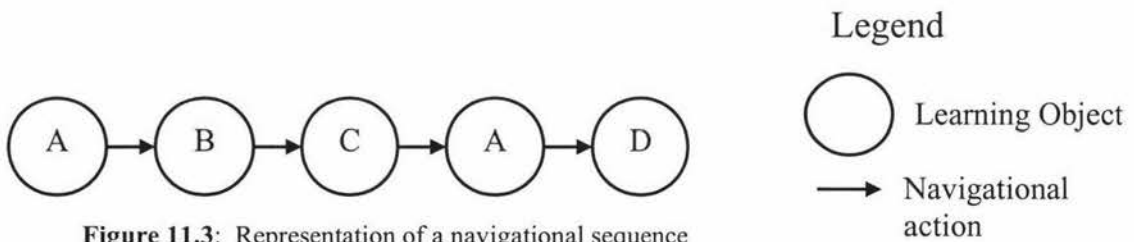


Figure 11.3: Representation of a navigational sequence

In figure 11.3, there are the following ordered pair of navigation: (A, B), (B, C), (C, A), (A, D). Also, there is a pair that qualifies the description of R , that is (C, A) in which A is already visited.

However, how many R will be needed for it to become “frequent”? Assuming that,

each learning object represents a concept in a learning topic, and each learning object is presented in a single web page. ... (Assumption 11.1)

Then, Assumption 11.1 allows us to claim that,

each page will only be viewed once, ... (Statement 11.1)

because of the following conditions:

- *CTM is only designed to be used in an adaptive learning system. ...(**Condition 11.1**)*
- *In an adaptive learning system, the information presented in a page should have already been tailored to the interacting learner. ...(**Condition 11.2**)*

Therefore, it could be concluded that if any of the reverse navigation, **R**, should occur, the working memory capacity of the learner is actually lower than what the system believes, and the belief of the system needs to be changed.

11.3.3: Implementation Pattern for Excursions

Learners can take excursions to obtain side information during a course of learning. Excursions bring less-relevant information to the student, and therefore broaden the learning experience, and create opportunities for accidental learning. But excursions also bring distractions to the learner. Any distraction would affect the processing and working of the working memory, which has limited capacity by its nature (Resiberg, 1997).

The implementation pattern for side information is therefore straightforward:

*For every learning object (v_i), if v_i has any **ExcursionTo** relation, and if any of its **ExcursionTo** relations are activated, and the result of that unit is equal to fail.*

Where

V is the set of all visited learning objects in a session.

v_i is a learning object in V.

In order to detect this implementation pattern, every activation of **ExcursionTo** relation is recorded. And if the learner fails this learning unit in which at least one of the **ExcursionTo** relations in one of the learning object belongs to this unit is recorded, this pattern is caught.

This is because, in an adaptive learning environment, the main course of learning is adapted (presentation and navigation) to the learner, but the excursions are not. It is quite often that the excursions bring the learners to other web sites for which the curriculum designer has not control of. Therefore, it can be assumed that if the learner has not been to any excursions at all during a course, s/he should be able to pass the course since this course is adapted to his/her individual ability. The distractions from the excursions are accounted for the failure of the learner whose working memory capacity is just not enough to handle the distractions.

By the recommendations of the formalisation of Exploration Space Control (Lin, 2002), for those who possess lower working memory capacity, excursion links should be avoided. It is not sensible to present too much information to the learner who cannot handle it.

11.3.4: Implementation Pattern for Simultaneous Tasks

The previous analysis indicates that being able to perform simultaneous tasks manifests higher working memory capacity, and vice versa for lower working memory capacity. Simultaneous tasks have to compete for the limited resource of working memory; therefore it provides indications for working memory's capacity.

Simultaneous tasks in a curriculum do not need to be deliberately designed, but they could happen due to learner's wandering away from the current task, performing something else, and then coming back to the current task later. For example, Student S is half way through reading Paragraph P when he encounters Vocabulary V that S needs to go to the Internet and find out the definition and other related information of V. At the time S is searching and reading information about V, S has to try to relate and fit what is found into the context of P in order to understand V and P. This simple scenario is an example of simultaneous tasks. During the action of relating V to P, S may need to rehearse V constantly, and review briefly P in order to see the contextual meaning of V. Therefore, the nature of the pattern of simultaneous tasks could, but not limited to, be accidental.

However, one has to be clear about the definition of tasks. In a learning space constructed by learning objects, tasks can be defined by learning objects:

A task is an interaction between a learner and one or more learning objects. And a task has its boundary.

Boundary is the important structural element that gives tasks their forms. In order to impose structures into learning objects, hierarchical information of learning object has to be used. In figure 11.4, assuming learning object A and E are different topics (topic A, E), the navigation from B to C is considered to be within the same task (learning A), but the navigation from C to F is considered to be inter-task due to the reason that C and F are not in the same topic. In this example, it is clear that hierarchical boundary can be used to determine tasks' boundary.

But hierarchical boundary is not the only limit, more research on learner's perception of what is a task is required, and this is beyond the scope of current project. Further research possibilities in this area are discussed in the latter section of this thesis under the section of the future improvement and research opportunities.

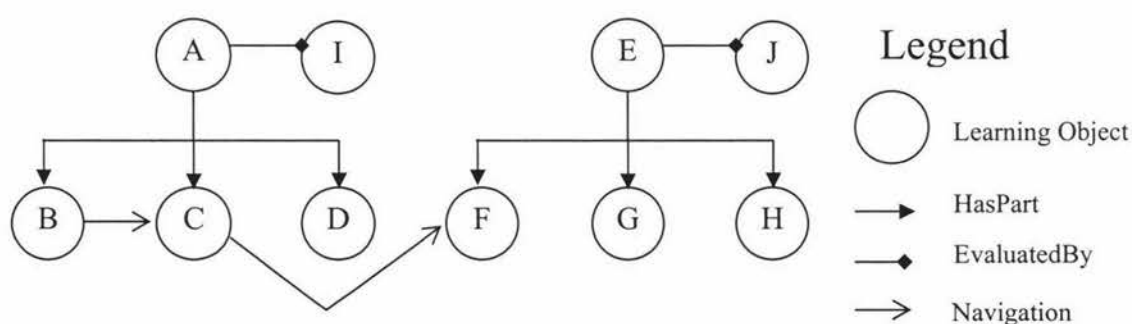


Figure 11.4: Boundary of Tasks

Please note that in figure 11.4, if the learner did not come back to topic A afterwards, it has to be assumed that the learner has wandered off topic A and therefore no simultaneous tasks were detected. It could be the case that the learner finds topic A to be not useful to him/her at this moment and decides to proceed on topic E first. So a complete pattern for simultaneous tasks in Figure 11.4 could be (B, C, F, G, H, J, D, I). In that pattern, learner jumps from topic A to topic E, finishes topic E (including the evaluation), and then comes back to topic A, and finishes topic A (including the

evaluation). If the learner fails at least one of the evaluations (either I or J), it indicates that his/her working memory capacity could not afford to do both tasks (learning topic A and learning topic E) simultaneously. And if the learner passes both evaluations, it indicates the learner has high working memory capacity. Without completing both topics A and E, this pattern is not activated.

The implementation pattern for simultaneous tasks could therefore be represented as follows:

For LWMC:

For **V** and **U**, **Overlaps (V, U)** = true, and **Pass (V)** = false or **Pass (U)** = false.

For HWMC:

For **V** and **U**, **Overlaps (V, U)** = true, and **Pass (V)** = true or **Pass (U)** = true.

Where

V and **U** are two sets of learning objects, and each is a separate task.

Overlaps (Set1, Set2) returns true if items in Set1 overlap temporally with items in Set2.

Pass (learning object) return true if the learner passes evaluation of learning object, returns false if failed, or returns Null for not evaluated.

11.3.5: Implementation Pattern for Retrieval of Information from Long-Term Memory

It has been discussed that working memory capacity is one of the factors that affect the retrieval of information from long-term memory (LTM). Forgetting may be the general term used for the failure of information retrieval from LTM. Many learning theories address the issue of forgetting, for example, according to Ausubel's (1963) subsumption theory, forgetting occurs because certain details get integrated and lose their individual identity, and Guthrie's (1938) Contiguity theory said that forgetting is due to the interference; stimuli is associated with new responses. However, the focus here is the method of how to identify this forgetting phenomena.

The previous discussion on manifests concluded that “being able to retrieve information from LTM effectively manifests HPMC”, and “being unable to retrieve information from LTM effectively manifests LPMC”, and this relates to being able to recall and being unable to recall learned information respectively. However, it is only meaningful to detect the pattern for “being unable to retrieve information from LTM effectively” because the learner is expected to recall what s/he had already learned.

The manifest pattern could be expressed as follows:

For LPMC:

For all vi in V , if vi is visited in a previous session, and $Pass(vi) = \text{true}$, and $Reference(vi)=\text{false}$

Where

V is the set of all visited learning object

vi is a learning object in V

Pass (learning object) returns true if the learner passes evaluation of learning object, return false if failed, or return Null for not evaluated.

Reference (learning object) returns true if learning object is a referential learning object, returns false if not.

This manifest pattern checks that if learning object A has been visited in any previous session, and A is learned (by **Pass(vi)=true**), and it has been visited again indicating that the learner is unable to recall and has to look up A . Please note that vi must not be a referential learning object for this pattern to be detected; looking up referential learning object does not indicate LPMC.

11.3.6: Implementation Pattern for Long Sequence of Calculation or Procedures

In long sequence of calculation or procedures, frequently missing steps or components manifests LWMC, and no missing steps or components manifests HWMC. In order to detect this pattern, a very different approach is required, and it involves insights from the process model.

The discussion of student model involves explanation and elaboration of process-models. Process-models are usually used in a closed system where the primary focus of learning is a particular set of procedures. The process models are capable of closely observing each of the learner's action/step and making adaptive assistances. Thus, if the CTM is used to supplement a process-model, each learning object would have to be considered as a procedure or skill. Since the process-model would certainly be capable of detecting such a pattern (missing step), it can directly input this information into the Individualised Temperament Network.

11.3.7: Other Manifests not Suitable for Implementation

Three more manifests are left without implementation patterns. They are age, comparison speed, and able/unable to comprehend highly demanding text/concepts. They are all measures of the capacity of working memory, however, they are not implemented at this work due to various reasons explained below.

Although age affects the capacity of working memory and is easy to detect if a learner profile is present, but unlike other patterns that are reoccurring and contribute to the formation of the Individualised Temperament Network (ITN), age can only provide a once-off influence on the overall score. It is not of any use in the current structure of the CTM, and therefore not included in the list of implementation patterns.

Comparison speed, although related to working memory, is more suitable as a pattern for information processing speed, not working memory capacity. And in order to measure speed (learning or comparison), it requires another set of metadata about what is the average/expected duration of learning for all learning objects. More

research is required to determine whether the benefit is worthy for the overhead incurred.

In order to be able to detect the ability to comprehend highly demanding text or difficult concepts, each learning object would need to be rated. The rating process requires a systematic categorisation and is beyond the scope of this work.

11.4: Summary

Each of the manifests discussed in this chapter provides indications to learner's working memory capacity. The manifests provide very general guidelines for analysing learners' behaviours in order to infer their working memory capacity. Each of the manifests, except age, involves a pattern of student behaviour and those patterns are called the implementation patterns.

In this chapter implementation patterns for navigational linearity, reverse navigation, excursions, simultaneous task, long sequence of calculation or procedures, and retrieval of information from long-term memory are discussed. Many of the methods used to translate manifests into implementation patterns could be inaccurate, and they are indeed areas for future research work.

Chapter 12: Individualised Temperament Network

12.1: Introduction

Individual Temperament Network (ITN) is designed in this work due to the unique nature and requirements of the Cognitive Trait Model (CTM). The main idea of Individualised Temperament Network is to let the student's behaviours determine which theory or method works for him/herself the best. Retrapolling is a process for updating the ITN and hence CTM. An important factor for retrapolling is the gradient constant which determines how rapidly the values in ITN changes. However, a major drawback of novel hypermedia systems is the difficulty in obtaining empirical data concerning their usefulness or otherwise (Mullier, 1999). Therefore, a simulation is created for searching the appropriate value of the gradient constant.

12.2: Individualised Temperament Network

ITN inherits the networked structure of an artificial neural network (ANN) (Tebo, 1994). Each node in ITN represents a manifest, and processes an associated weight. The most obvious difference between ITN and ANN is the training and the feedback mechanism of the network.

Before each ANN can be applied to a particular job (e.g. predicting market behaviour), it has to be trained for this job. The training gives feedback to the network, and the network learns during the training (Mitchell, 1997). After large amount of training (data), the ANN can usually perform some tasks quite accurately with a specified uncertainty. However, while the promoted idea of adaptive learning environment is individualization, it is not tolerable that the environment can only be correctly adaptive to 90% of the student populations, and to 90% of every student population. A set percentage (e.g. 90%) is good enough for some areas (e.g. gambling prediction), but essentially it is against the meaning of adaptation.

Therefore, the concept of ITN proposes a mechanism that can adapt itself to every single student. Each of the manifests represents a node in ITN. Each node has an associated variable (as opposed to constant in ANN), called weight. The amount of the weight determines the node's influential power over the overall output. The output is a spectrum ranged from -1 to 1, while -1 means the lowest end of working memory capacity and 1 the highest end. The output value can be situated anywhere between these two ends. The total number of weights is processed through a sigmoid unit (Morton, 2002), and the output of it is a value ranged from -1 to +1. This is accomplished by a sigmoid function (Eq 12.1):

$$f(x) = \frac{2}{1+e^{-2xk}} - 1$$

Eq. 12.1: Sigmoid Function

When x is negative, the $f(x)$ will be in between -1 and 0, and when x is positive, $f(x)$ will be in between 0 and 1.

Let T denotes the entire network, L for the group of LWMC, and H for the group of HWMC. L is a set of negative real numbers (representing low working memory), whereas H positive (representing high working memory). The total initial weight of the entire network, Ta , has to be zero, i.e. the total initial weight for all the LWMC manifests (La) plus the total initial weight for all the HWMC manifests (Ha) equals zero, illustrated below by Eq 12.2:

$$Ta = La + Ha = 0$$

Eq. 12.2: Weight initialisation

La is equally shared by all the manifests belonging to the LWMC group, illustrated by Eq 12.3.

$$L_a = \sum_{i=0}^n L_i$$

Eq. 12.3: Initial weight for manifests

where n is the number of manifests in the LWMC group. The same applies to the group of HWMC.

The absolute value of maximum weight allowable for the entire network, T_{max} , is twice as many as Ha . That means the sum of initial weight for all the manifests is only 50% of the maximum weight possible. This also sets the capping limit of the weight for each manifest. Each manifest can have the maximum as the twice of its initial weight; this prevents the overflow of the entire network.

After each learning session, which is defined by completion of a unit, or if the student logs off before a unit is completed, the weights are redistributed by a mechanism called Retrapolling. Retrapolling operates in the following way:

1. All the manifests are checked for their activation frequency, which is the frequency at which the manifest has been activated during the session. If a manifest's activation frequency is greater than or equal to 1, it is called an active manifest.
2. When a manifest and its opposite manifest are both active manifest in a session (e.g. linear and non-linear navigational pattern), they are both deactivated, and they both are subject to weight deduction in the Weight Redistribution Phase. Their activation frequencies are both set to 1.
3. All the current weights of active manifests are summed, and it is called the Polling Sum.
4. A Prevailing Group is decided according the Polling Sum; when the Polling Sum is positive the HWMC group prevails, and the LWMC group recesses, and vice versa for the Recessive Group.
5. The weight of each of the active manifests is either increased or decreased by the Gradient Constant times the manifest's activation frequency for Prevailing Group and Recessive Group respectively. The Gradient Constant P is expressed as a percentage. When the weight of any manifest is greater than its

capping limit, it is set automatically to the capping limit. This is called the Weight Redistribution Phase.

6. The activation frequencies are re-zeroed for the next learning session.

The Retrapolling is concerned with only the active manifests because there could be scenarios where a manifest has never had a chance to occur. For example, some manifests can only be detected if a long sequence of calculation is present, but if the current topic is only an introduction and has no long sequence of calculation or procedural, then those manifests will just remain inactive and will be ignored during the polling.

In the situation where a manifest and its opposite manifest both are active, it implies that the underlying theory does not very well apply/explain the behaviours of the particular student, and therefore, they should be gradually faded out. For example, it is Huai's (2000 as quoted by Kommers, 2000) theory that suggests: "linear navigational pattern manifests HWMC"; if the student frequently manifests both linear and non-linear navigational pattern in many sessions, it indicates that this particular theory can not portray the actual cognitive processes of this particular student, and therefore should be diminished. It is quite possible in this case that a Huai's (2000 as quoted by Kommers, 2000) theory is 90% accurate and this particular student belongs to the other 10% of the population.

The Polling Sum determines the Prevailing Group. Theoretically, after a period of time, the Polling Sum will be dominantly positive or negative. But at the very beginning, when the ITN has not learned much about the student, there could be still quite uncertain.

Gradient Constant, P , is the constant that determines how fast the weight's value of each active node should change (a node's weight will increase if it is in the prevailing group, otherwise it will decrease), i.e. it determines the gradient of change.

However, the selection of the Gradient Constant P requires great care. If P is too high, the weight would change dramatically in the first few sessions. For example, a student may be misjudged as belonging to the HWMC group just because misleadingly (s)he

just happen to demonstrate more HWMC manifest in the first few sessions. However, if the P is set too low, it may takes years for the system to know the working memory capacity of a particular student. Therefore, a simple computer simulation is prepared to choose the value of P appropriately.

12.3: Simulation for Selecting Appropriate Gradient Constant

In order to select an adequate Gradient Constant P two criteria have to be considered:

1. P has to allow the ITN to judge correctly.
2. P has to be able to reflect the student's temperament in an acceptable period of time.

First, the Gradient Constant P has to be non-biased. When the student's working memory capacity is in the average level, the ITN could not be biased to place the student into either the LWMC or the HWMC group, no matter how long the time has elapsed. And if the student possesses HWMC, the ITN should be able to reflect this fact, and so it should if the student has LWMC. Although at the beginning, when the ITN is still learning to know the student, there could be a learning curve, but after a while, the ITN should be able to tell the student's capacity accurately.

Second, P has to allow the ITN to be able to reflect truly the student's temperament in an acceptable period of time. Although setting P to a very small value will diminish the risk of misjudging student's capacity, it would require a very long period of time for ITN to know the student well enough to be able to provide any benefits to the student.

In order to facilitate easy selection of P , a computer simulation is written. It provides graphic charts to assist the comparison. The simulation allows the modification of two parameters: the Gradient Constant P , and the Working Memory Capacity WMC. The value of WMC represents the working memory capacity of a hypothetical student. It uses a scale between 1 and 10, for 1 is the lowest of the capacity and 10 the highest.

The Gradient Constant P determines how fast the weight increases or decreases. By manipulating the value of P , it is easy to see how an ITN will learn about the working memory capacity of the hypothetical student. P is the focus of this simulation, and an appropriate value of it is sought after in this simulation. Figure 12.1 shows an interface to manipulate the parameters of this simulation.

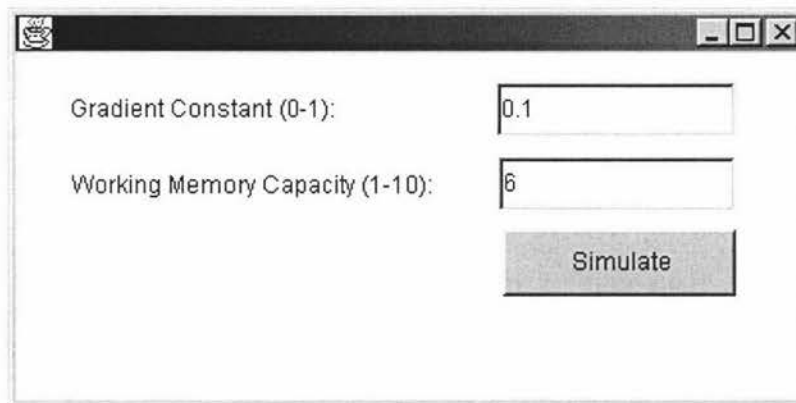


Figure 12.1: Interface for manipulating parameters

Figure 12.2 shows a chart of the simulation. The x-axis shows the number of sessions, it goes from 0 to 400. The y-axis is the Polling Sum which when the Polling Sum is higher, it indicates High Working Memory Capacity (HWMC), and vice versa for Low Working Memory Capacity (LWMC).

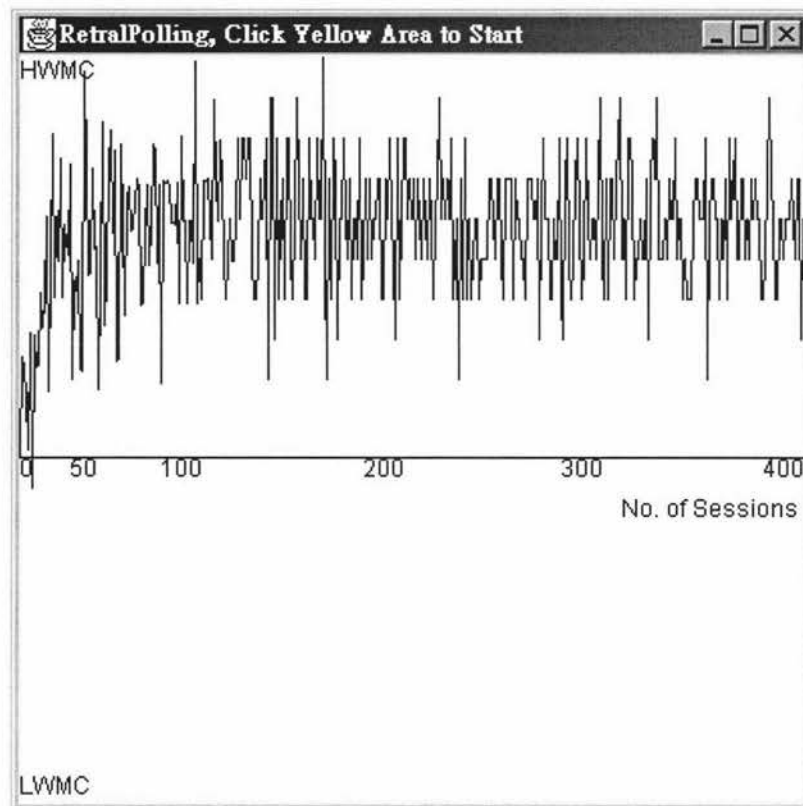


Figure 12.2: Interface for manipulating parameters

Figure 12.3 show the simulation result when $WMC=5$ and $P=0.05$. In this simulation, the hypothetical student has average working memory capacity ($WMC=5$). Therefore, by theory, the line of the chart should centres around the x-axis and that is what happens before about the 35th session. But after the 35th session and before about the 90th session, the weights of HVMC manifests are increased too rapidly, and the weights of the LVMC manifests are decreased too fast. There is no chance for it to come closer back to the x-axis anymore after the 90th session. Therefore, it is obvious that the value of P (0.05) is too high.

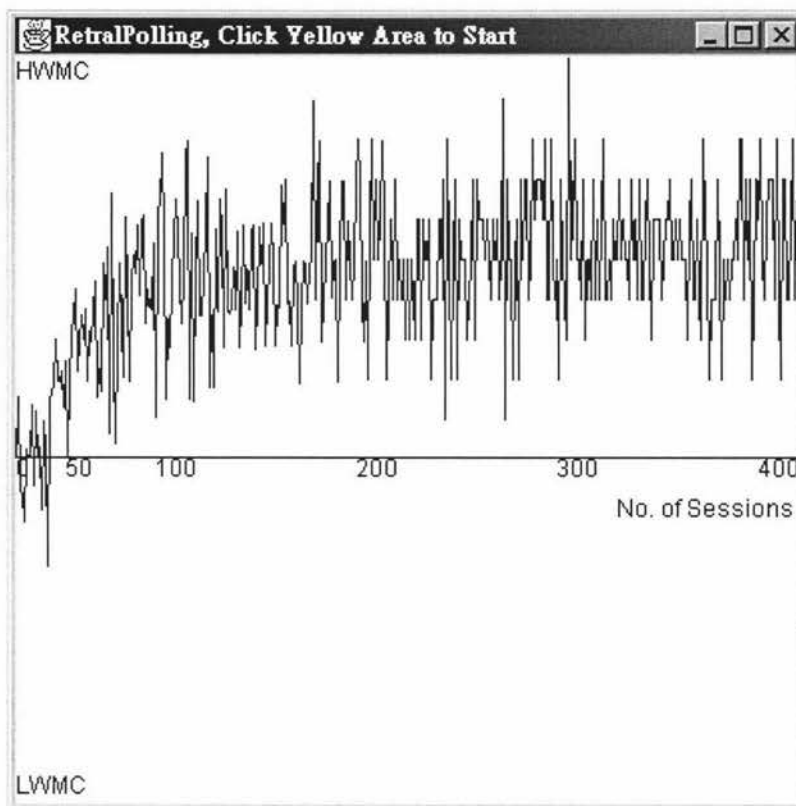


Figure 12.3: WMC=5, $P=0.05$

In figure 12.4, P is set to a smaller value, 0.005, and the rest remains the same. It can be seen that during the first 170 sessions, the graph behaves as expected (centred around the x-axis), but after about 200 sessions, it still goes astray, too. So the value, 0.005, is still too high.

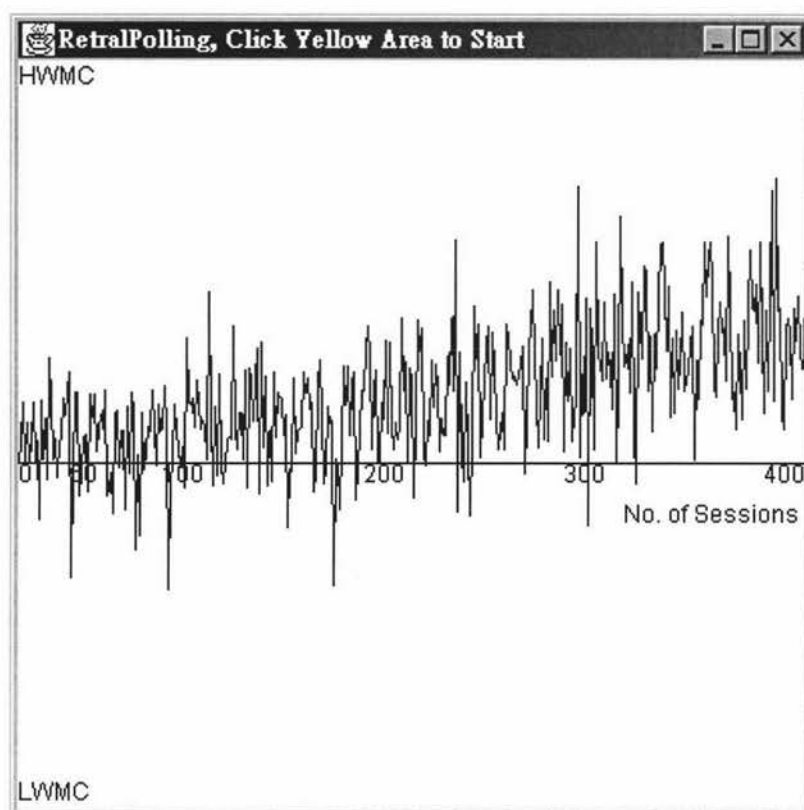


Figure 12.4: WMC=5, $P=0.005$

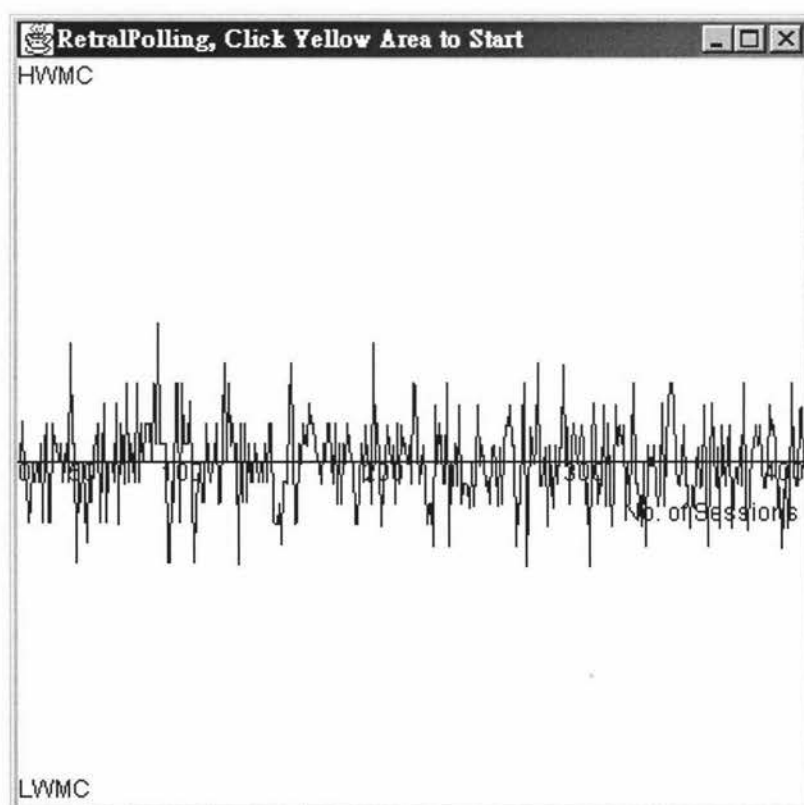


Figure 12.5: WMC=5, $P=0.001$

In figure 12.5, the P is set to 0.001, and the ITN behaves as expected (centred on the x-axis).

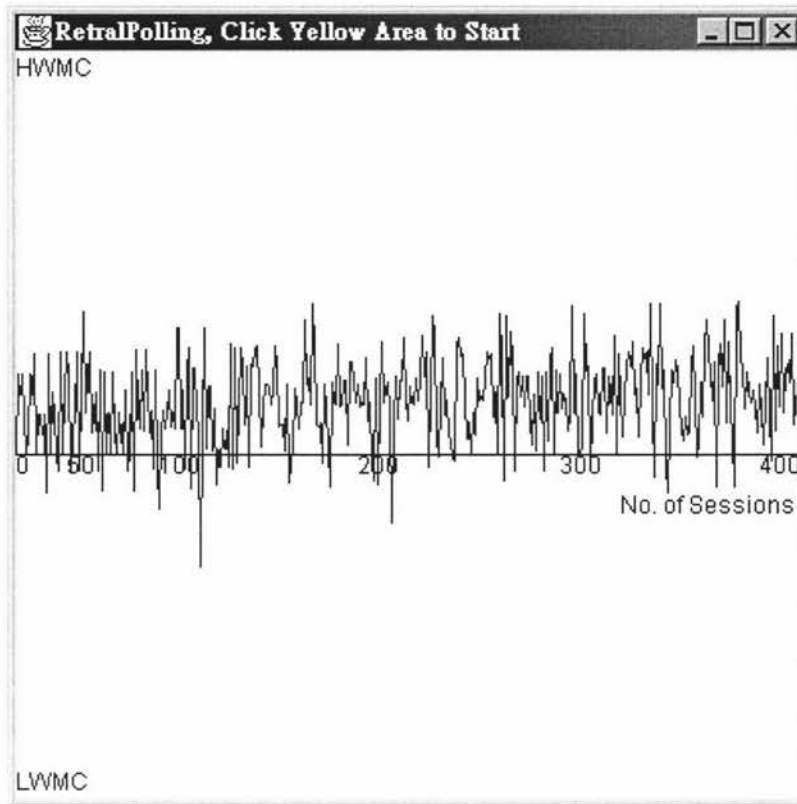


Figure 12.6: WMC=6, $P=0.001$

In figure 12.6, WMC is set to 6, and therefore, the student is expected to have above average working memory capacity. But from figure 12.6, it is clear that the student is having HWMC from about 120th session onwards.

In figure 12.7, P is increased to 0.003 and WMC still remains at 6. The graph shows that after about the 60th session, there is very little noise. So 0.003 could be a good value for P . This value has to be further tested by setting WMC back to 5 to see whether the line of the graph could centre on the x-axis. Figure 12.8 gives the answer “yes”.

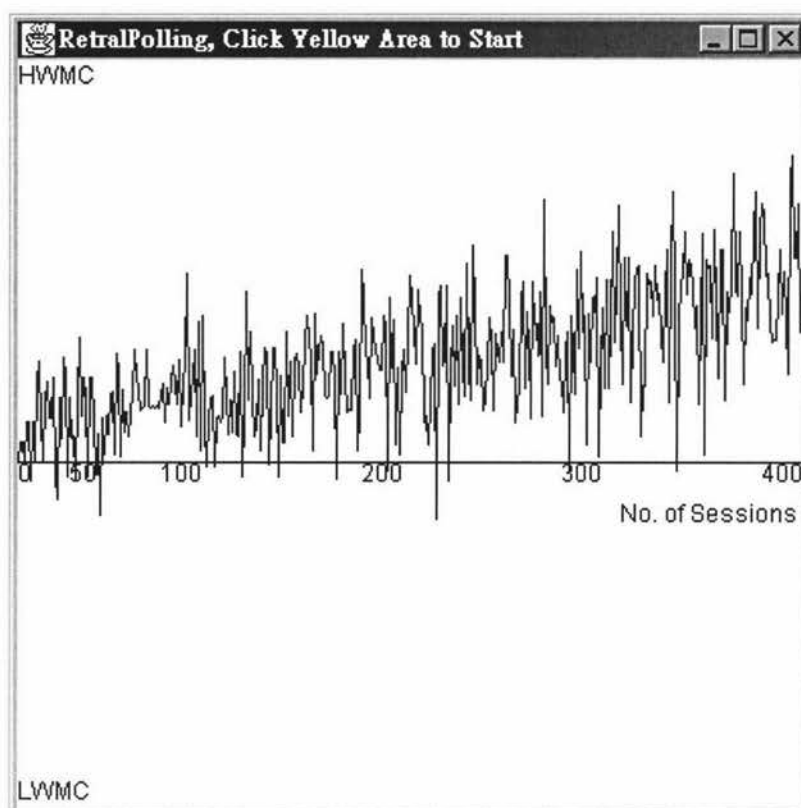


Figure 12.7: WMC=6, P=0.003

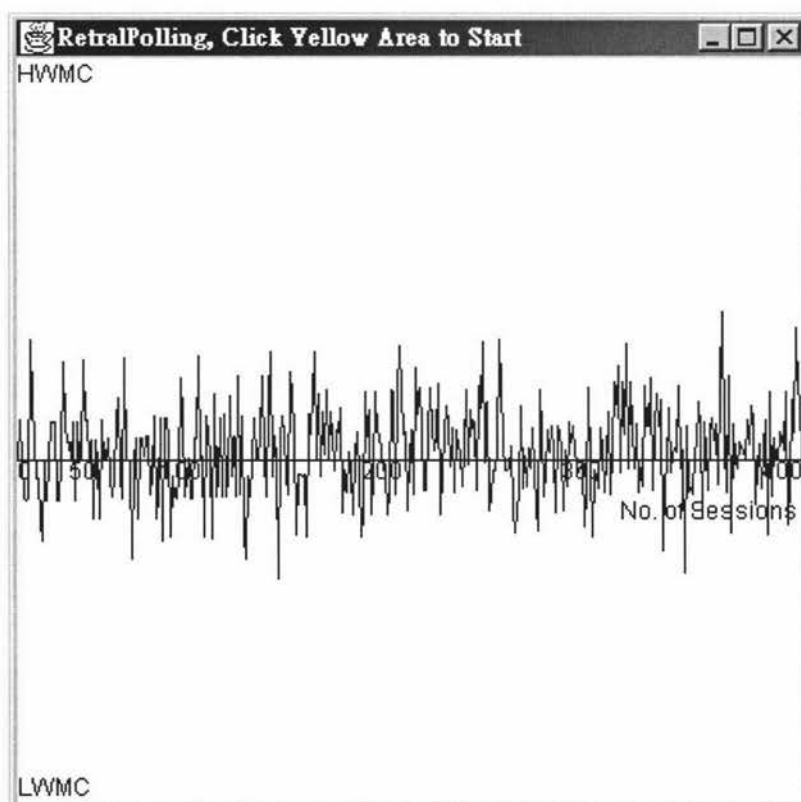


Figure 12.8: WMC=5, P=0.003

Furthermore, in figure 12.7, the ITN starts to provide accurate predictions very early of the sessions. In fact, after about the 60th session, the network is able to achieve 97.05% accurate predictions, and after 200th session, 99.55% (Appendix 1 shows the statistics in tables). However, the number of the sessions does not go proportionally with time because CTM is designed to be applicable to multiple domains. For example, a university student takes four papers each semester with an average of 20 sessions a paper; the ITN can easily accumulates experience of 80 sessions for this particular student in one semester.

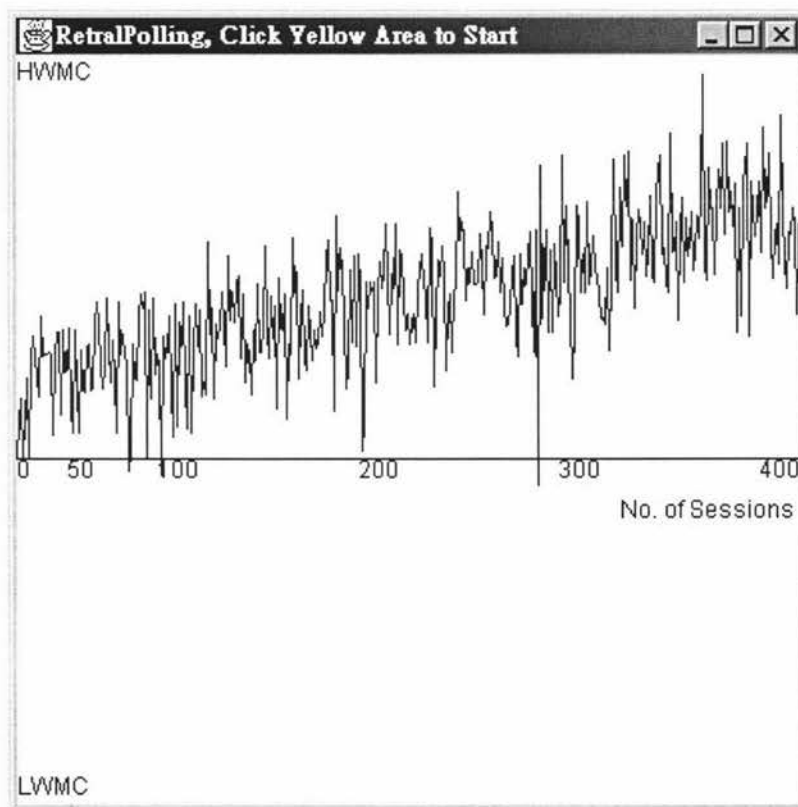


Figure 12.9: HPOS=7, LPOS=3, P=0.003

And please note, the statistics is taken when WMC=6, which is suppose to be the most difficult case for the ITN to learn. If the value of WMC is not 6 or 4, the correct prediction rate increases (figure 12.9 and figure 12.10).

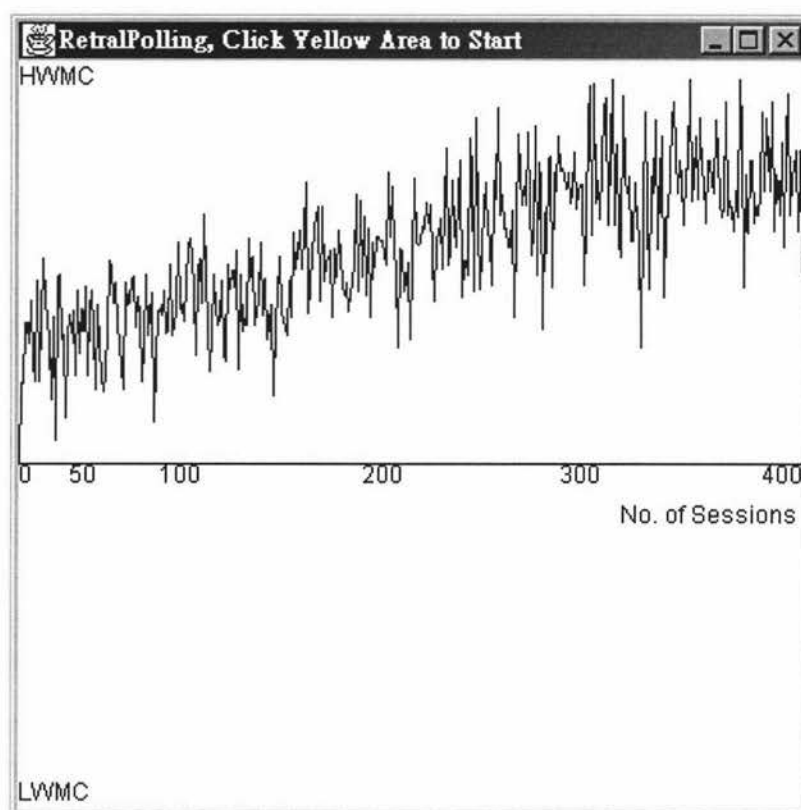


Figure 12.10: WMC=8, $P=0.003$

12.4: Summary

Individualised Temperament Network (ITN) inherits the networked structure of an artificial neural network, and the main idea of ITN is to let the student's behaviours determine which theory or method works for him/her the best.

Retrapolling is carried out at the end of every session to update the weights in ITN. The weights of active manifests' are redistributed during the retrapolling process. The rate of change for the weights is called the gradient constant. Due to the difficult nature for real student experiment, a computer-based simulation is created to determine the appropriate value for the gradient constant, which found 0.003 to be a suitable value.

Chapter 13: Prototype Web-Based Tutorial with Cognitive Trait Model

13.1: Introduction

A prototype web-based tutorial of recursion was created to demonstrate the implementation of Cognitive Trait Model (CTM) and its components. The purpose of this prototype is not to create a complete system but to show the potential of CTM. The full implementation of CTM requires further research and therefore is subjected to the future improvement.

This chapter of the thesis first discusses the structure of the domain. Then description of the tutorial is presented, and finally an exemplary analysis is carried out at the end of this chapter for a hypothetical learning session.

13.2: Structure of the Domain

In this prototype, a simple tutorial on the use of recursion in computer programming language is created. The examples and listing in the tutorial is in C language but any other programming language that supports recursion can use this approach. The main focus of this tutorial is on the idea of recursion, principles of writing recursive functions, and good programming practice.

The tutorial is completely web-compatible. It consists of twenty-two nodes (Figure 13.1) each of which can roughly approximate to single web-page except the recursion nodes which bring the learners to other websites of related topic. This tutorial has sixty-one relations which are mainly of six types: *IsBasisFor*, *IsBasedOn*, *HasPart*, *IsPartOf*, *ExcursionTo*, and *IsExcursionOf*.

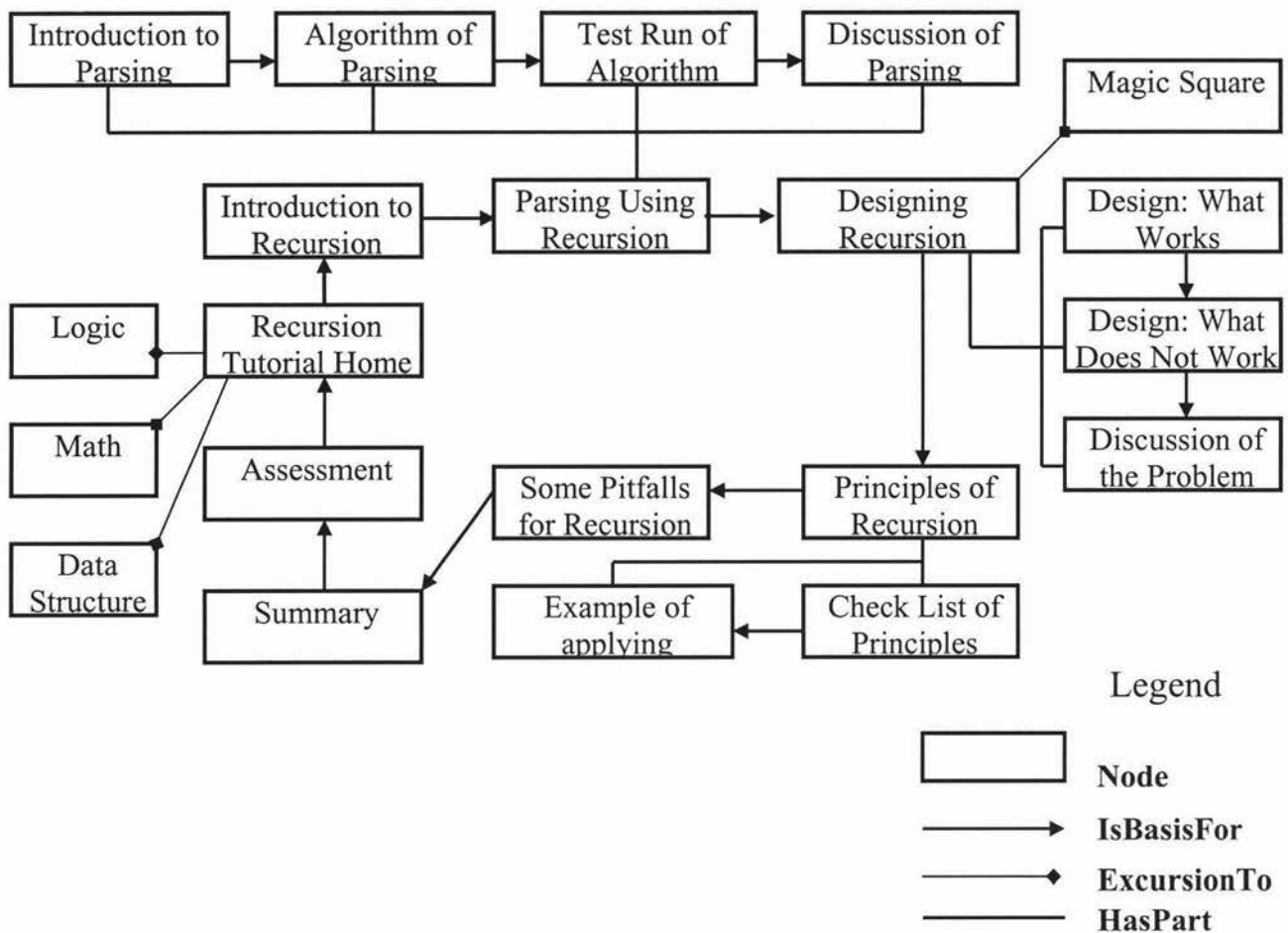


Figure 13.1: Structure of the Domain: Tutorial of Recursion

The logical start of the tutorial is Recursion Tutorial Home node, but it is a strict limit. The learners can go through the tutorial in a linear fashion by following the IsBasisFor and HasPart relations, or they can explore more examples and different perspectives about recursive programming by the ExcursionTo relations. After the assessment page, the learner is automatically brought back to the Tutorial Home. This provides a means in this prototype to experiment with sessions of different length.

13.3: Description of the Tutorial

On the top of every page, there is a navigation bar in which the “next” links to the pre-determined page and in the bottom of every page there is a view report link. There are sometimes excursions available for a particular node, which bring the learners to

other websites of related concepts. For example, excursions available in the **Introduction to Recursion** (figure 13.2) are recursions to **Data Structure and Algorithm, Logic, and Mathematics**.

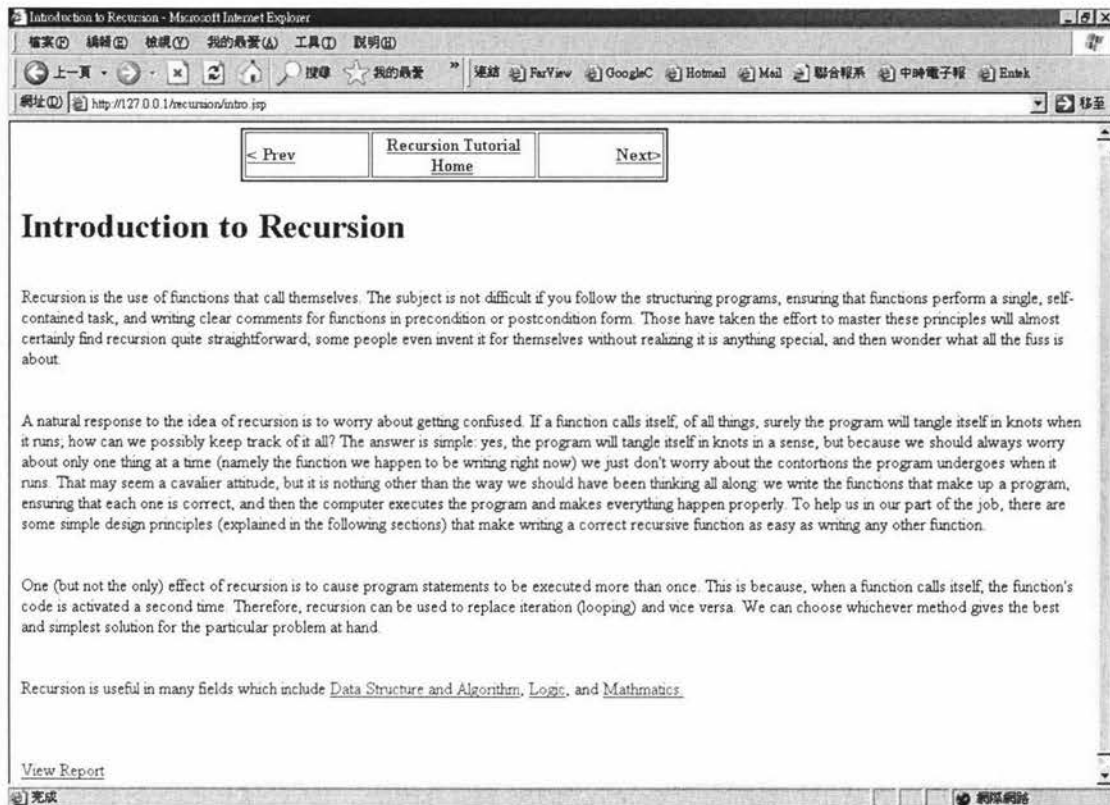


Figure 13.2: Introduction to Recursion

Next, figure 13.3 shows the excursion to the **Mathematics**. Recursions take the learner away from the host, called main host, where the tutorial is located, and thus learners activities in the excursion web site are un-trackable. However, the activations the relations *ExcursionTo*, and *IsExcursionOf* are still in the control of main host and therefore these activations can be recorded in the Student Behaviour History (figure 10.1).

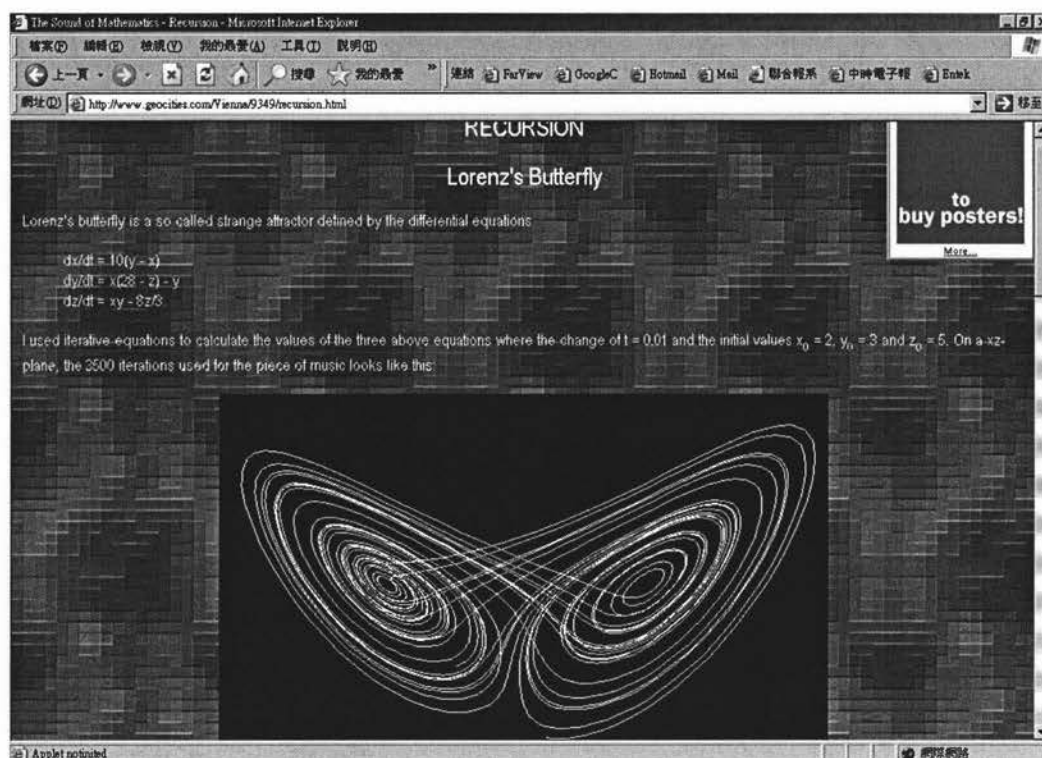


Figure 13.3: Excursion to the Mathematics node

When the learner clicks the Next button in the **Introduction to Recursion**, the tutorial brings the student to the **Introduction to Parsing** (figure 13.4).

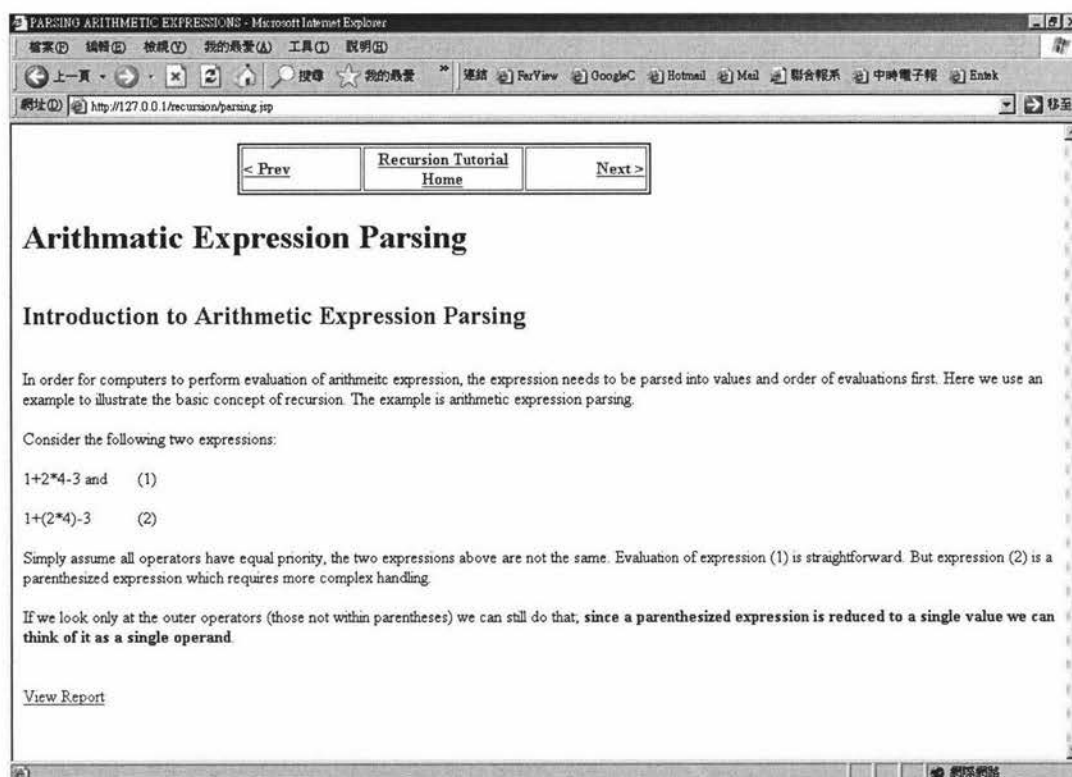


Figure 13.4: Page for the introduction to the arithmetic expression parsing

Please note that in figure 13.1, the **Introduction to Recursion** has an **IsBasisFor** relation to the **Introduction to Parsing**, which is actually a logical learning object only. Logical learning objects are used to relate learning objects semantically, but not necessarily physically (by hyperlinks). A single action of transferring from Introduction to Recursion to Parsing Using Recursion actually activates two relations: an **IsBasisFor** relation (from **Introduction to Recursion** to **Parsing Using Recursion**) and a **HasPart** relation (from **Parsing Using Recursion** to **Introduction to Recursion**).

13.4: Technical Details of the Prototype

The creation of the web-based exemplary tutorial includes a simple Cognitive Trait Model (CTM). This tutorial implements the interface module, the student behaviour history, and trait analyser, and a mark-up student performance model.

13.4.1: The Interface Module

The tutorial is designed to be web-based and the hence its interface to the learner is via the web-browser. The nodes in the tutorial are implemented by Java Server Page™ technology in order to be able to create content dynamically. Tomcat, which is a free, open-source implementation of Java Servlet™ and JavaServe Pages™ technologies developed under the Jakarta project at the Apache Software Foundation, is used as a standalone web server that the tutorial is installed on.

Recording of learner navigations is implemented in the form of query string. In the case when an excursion link is clicked, the learner intends to move to a page that is not in the server (host) of this tutorial, and therefore a buffer page is created for every excursion which will automatically re-direct the web browser to the excursion site. The buffer page serves the function to record the activation of **ExcursionTo** and **IsExcursionOf** relations. But the subsequent pages after the first excursion page are beyond the control of the host, therefore they cannot be not recorded.

13.4.2: The Student Behaviour History

The student behaviour history is implemented by database records of the learner's navigational history. A navigation consists of the source, the destination, the time, and the relation type of this navigation. The source and destination of a navigation can be used to know which of the learning objects have been visited and when. The relation of a navigation is used by the trait analyser in which the relations are of primary importance for many analysis tasks. The exemplary trait analysis presented later in this chapter will show how the relations are used.

13.4.3: The Trait Analyser

The Trait Analyser is implemented as Java Beans, which are component-oriented Java objects. The activation of the Trait Analyser should be scheduled at the end of every learning session. However, for the ease of demonstrating how the Trait Analyser works, the trait analyser is available at any point of the tutorial view the "View Report" link.

The Trait Analyser has three parts that form a chain of operation (a pipeline), they are:

1. **Pattern Detector,**
2. **Individualised Temperament Network, and**
3. **CTM Updater.**

Upon the activation of the Trait Analyser, the Pattern Detector begins to collect data of current learning session from the Student Behaviour History, and performs pattern-matching operations. The pattern-matching operations require the data of relation types and the relations between learning objects. These data are stored in an internal database, and the data is created during the creation of the tutorial. Implementation patterns are sought in the learner's navigational path, and all the implementation patterns (such as the linearity, reverse navigation, and so on) detected are outputted to the Individualised Temperament Network (ITN) with their frequency of occurrences.

Once the ITN get a set of data that is the output from the Pattern Detector, it starts the retrapolling process. Retrapolling involves the current status (weights) of the ITN and the new set of data. The new set of data is processed according to their corresponding weight values in the ITN, and a polling sum is created. The ITN has to redistribute its weight according to the polling sum, and signals the CTM Updater to update the CTM accordingly.

13.4.4: Student Performance Model

This is a mark-up performance model created in order to demonstrate some functions of the trait analysis. The performance model records the result of the assessment page at the end of the tutorial. In a complete system, there should be a module that is responsible to communicate to the independent performance model.

13.5: Example of Trait Analysis

Learners' actions of navigation are recorded in Student Behaviour History. An exemplary trait analysis will be demonstrated.

For example, a learner performed the sequence of navigation shown in the figure 13.5.

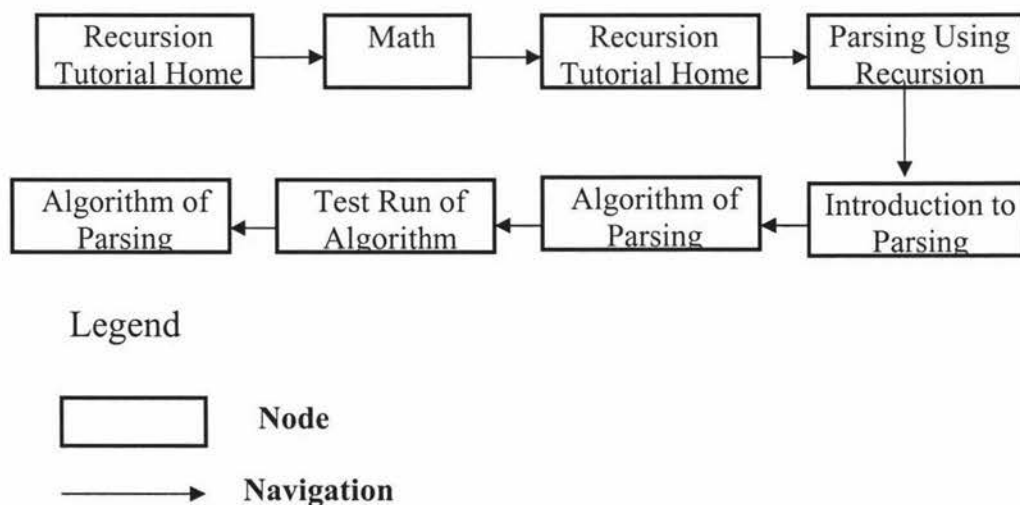


Figure 13.5: An exemplary session

The relations recorded in those navigation are sequentially: ExcursionTo, IsExcursionOf, IsBasisFor, HasPart, IsBasisFor, and IsBasedOn. Those relations can only construct three types of implementations, i.e. navigational linearity, reverse navigation, and excursions. Those implementation patterns will be discussed next. And for the sake of discussion, we assume that this learner failed the assessment after those navigations.

13.5.1: Navigational Linearity

The implementation pattern for navigational linearity is by the measure of LM:

$$LM = (No(B) / ISBASISFOR(V)) / (No(E) / EXCURSIONTO(V))$$

Where

LM is the measure of linearity

W is the set of all visited learning objects.

V is a learning object in the set of **W**.

No() returns the number of items in a set

ISBASISFOR (V) returns the set of learning objects that V has IsBasisFor relations with.

B is the subset of **ISBASISFOR(V)** where for every item **bi** in **B**, (**bi** $\not\subset$ **W**)

$\not\subset$ means not in the set of.

EXCURSIONTO(V) returns the set of learning object that **V** has ExcursionTo relations with.

E is the subset of **EXCURSIONTO(V)** where for every item **ei** in **E**, (**ei** \subset **W**)

\subset means in the set of.

If LM is greater than 1, then the navigational sequence is said to be more linear, otherwise it is more non-linear.

For the number of relations, please refer to figure 13.1. In the above navigational sequence, the total number of excursions, ExcursionTo(V), are three, and the number

In the above navigation, the learners take excursion to the **Mathematics** website, and in the end failed the assessment. The result of the trait analysis of this implementation pattern is “LWMC”.

Due to the types of relations available in this tutorial, other implementation patterns do not exist. All three available implementation patterns indicate that the learner has low working memory capacity.

Figure 13.6 shows the page summarising the result of the analysis. The “View Report” link is made available on every page of the tutorial in order to facilitate the ease of examining the status of the Individualised Temperament Network.

All three implementation pattern indicates LWMC. The values of the implementation pattern belonging to the group LWMC are all intensified by -0.003, whereas the HWMC groups have all their values decreased by 0.003.

of excursions visited, $No(E)$ is one. The number of $IsBasisFor$, I relations of all visited node is four whereas the $No(B)$ is equal to one.

Therefore, the value of $LM = (1/4) / (1/3) = 0.75 \rightarrow$ Non-Linear
Non-linear navigational pattern manifests LWMC.

13.5.2: Reverse Navigation

The implementation pattern for reverse navigation are as follows:

In a session of learner's interaction, there exist "a certain number of R " an ordered pair and represent a navigational action and at least one of the pair is visited. $R = (fromLearningObject, toLearningObject)$ $fromLearningObject$ represent the source of the navigational action and $toLearningObject$ the destination.

In the above navigation, it is obvious that the learner indeed perform navigation from Test Run of Algorithm to Algorithm of Parsing. And in this particular implementation pattern, the result of the analysis is "LWMC" means the increase of the score of low working memory capacity, and decrease of high working memory capacity.

13.5.3: Excursions

The implementation pattern LWMC for Excursions is:

*For every vi in V , if vi has any **ExcursionTo** relation, and if any of its **ExcursionTo** relation is activated and the result of that unit is equal to fail.*

Where

V is the set of all visited learning object in a session.

vi is a learning object in V .

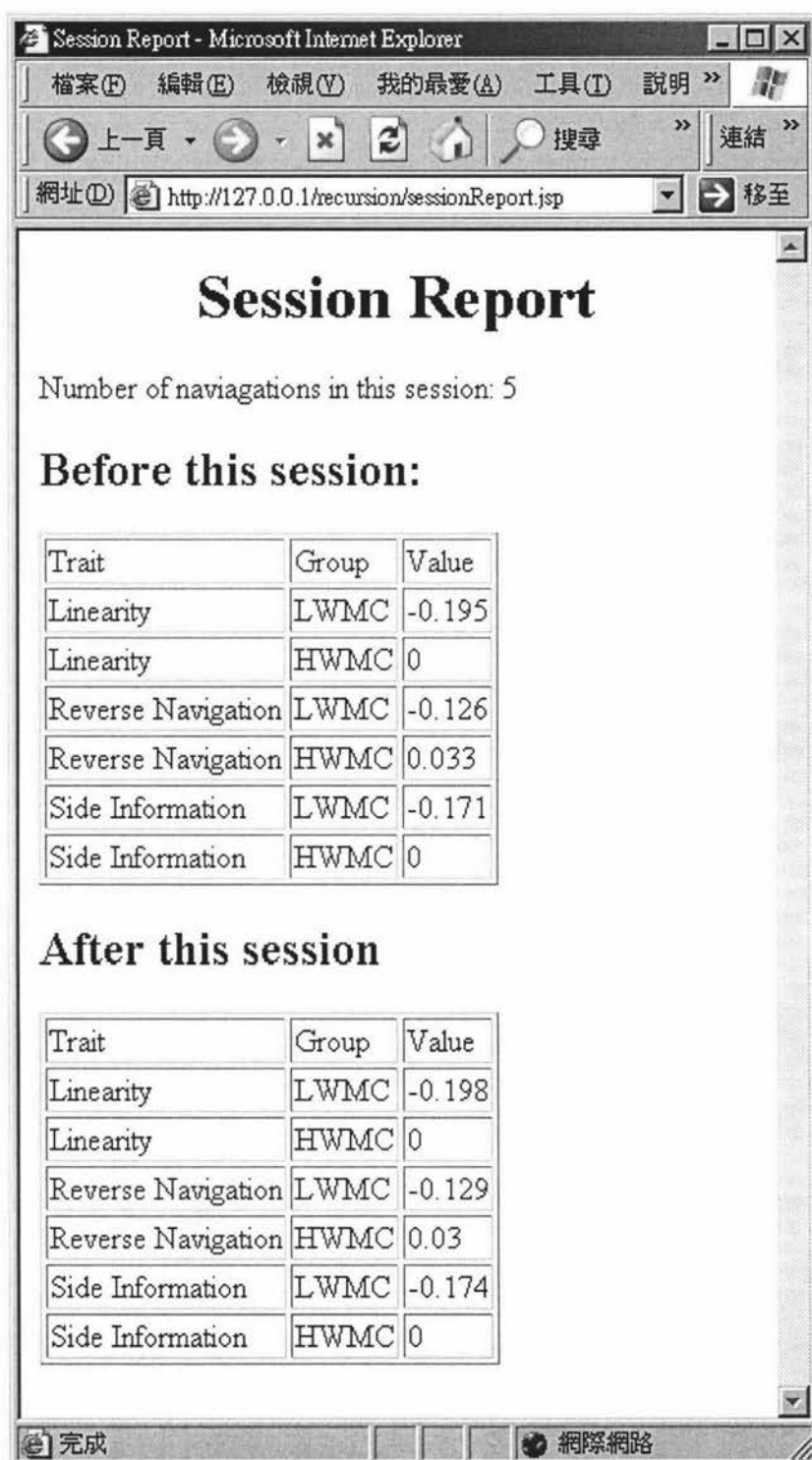


Figure 13.6: Page for the result of the analysis

Please note that in figure 13.6, the Side Information and Linearity in the HWMC group remains zero because for the HWMC group, zero is the minimum value and one is the maximum value whereas for the LWMC group negative one is the minimum literal value and zero is the maximum literal value.

13.6: Formative Evaluation of the Prototype

A formative evaluation was carried out for the prototype. The evaluation was presented in the form of questionnaires. The aim for these questions is to evaluate the educational value of the Cognitive Trait Model approach, and the performance and effectiveness of the prototype developed.

13.6.1: Participants of the Evaluation

There were total six participants including:

- 2 Lecturers, one in New Zealand and one in Iran,
- 1 Assistant Lecturer,
- 1 Junior Research Officer, and
- 2 Master Degree students (1 of them is a Postgraduate Assistant)

The evaluation questionnaires along with the URL of the prototype and relevant documentations were electronically delivered to the participants (see Appendix 2). The participants were asked to go through the documentations and the prototype in order to answer the questionnaire. The content of the questionnaire along with the overall rating and the summarized feedback for each question are presented next.

13.6.2: Evaluation Questionnaire and Summary

There are totally 7 questions asked.

1. Do you think that the Cognitive Trait Model presented in this prototype can assist students during their learning?

Strongly disagree

Strongly agree

1

2

3

4

5

Please explain, why.

The average rating for question 1 is 3.58. All the participants agreed that the approach of Cognitive Trait Model could assist students during their learning. Participants' reasons for the marks they gave to this question include:

- save student's learning time;
- improve the system's efficiency to deliver knowledge;
- provide individualised instruction;
- reduce frustration of student's learning; and
- improve learning effect.

2. Did the prototype demonstrate enough information to show that the Cognitive Trait Model can be used in learning systems?

Strongly disagree

Strongly agree

1

2

3

4

5

Please explain, why.

The average rating for question 2 is 3.25. The participants' reasons include:

- the prototype's structure, that is constructed and organised using learning objects, could prevent overloading the students;
- the start page of the tutorial for it does not have enough information (e.g. table of content) for the students;
- proper assessment unit could better show the effect of the prototype; and
- there should be some kind of feedback components following the assessment unit.

3. If you are a courseware developer, would you consider using Cognitive Trait Model in your systems?

Strongly disagree

Strongly agree

1

2

3

4

5

Please give three reasons, why.

a.

b.

c.

The average rating for question 3 is 3.67. The participants' reasons include:

- can help to improve student's learning results;
- can keep track the student' cognitive pattern with their learning results;
- could prevent overloading students with a lot of information;
- can track which part is most browsed by students;
- possible to assist different student group with different navigation background;
- can try to provide more local information (internally) when the instructor see some extrusion activity from students;
- the concept of learning objects is useful in developing teaching materials;
- *"A system without CTM is one-way communication system, it can't get any feedback online. Compared to a non-CTM system, a system using CTM has some two-way communication characters; it can track the activities of the student using the system. This is the basis for design a more flexible e-learning system."*;
- the learning environment can also learn from the students to improve itself;
- CTM may increase the course develop effort a lot; and
- assessment methodology of the tutorial needs to be strengthened.

4. If you are a student, would you prefer the courses that use Cognitive Trait Model over those that do not?

Strongly disagree

Strongly agree

1

2

3

4

5

Please give three reasons, why.

a.

b.

c.

The average rating for question 4 is 4.00. The reasons include:

- CTM could prevent discouraging students from studying;
- adaptivity provided by CTM could increase the performance and willingness of the study;

- allowing good learning effect by not overloading;
- the navigational history could allow the students to reflect upon their learning process if needed;
- CTM could be used to provide adequate feedbacks to the students; and
- the adaptivity may limit the freedom of navigation

5. How experienced are you in the field of e-Learning?

Not experienced

Highly experienced

1

2

3

4

5

The average rating for question 5 is 3.25. It shows that on average the participants have some experiences with e-learning either as a learner, a instruction provider, or both.

6. Based on your experience with this prototype, if you were to pick one aspect of the Cognitive Trait Model that is the most valuable for educational purposes, what would that be?

The feedback is summarised as follows:

- structuring the curriculum by learning objects;
- providing individualised instruction;
- CTM could be used for both the presentational adaptivity and the navigational adaptivity;
- its focus on the cognitive abilities of students;
- allowing student to have a choice to study related topics, obtain more detailed information, do more exercises etc. Hence, having optional learning blocks; and
- providing possibility to track student learning steps and its ability to record the learning history of the student for the course administrator.

7. Based on your experience with this prototype, if you were to pick one aspect of Cognitive Trait Model that needs most improvement, what would that be?

The feedback is summarised as follows:

- there is not enough information to guide the student in the first page of the tutorial;
- the assessment unit better to be fully implemented; and
- the system should be careful when making decision for the students who may not like those decisions because of their limitations.

13.6.3: Discussion of the Evaluation

Even though this is only formative evaluation, the overall rating of the Cognitive Trait Model approach is positive. All participants agreed that this approach could assist students in their learning. The benefits include time saving for the students, reduced frustration in learning, and increase learning effect.

The prototype uses the learning objects (discussed in Chapter 9) as the building block of curriculum. It was commented by one of the participant to be a very good way to structure and organise learning material because this structure provides manageable learning unit to the students and therefore reduces the possibility to overload the students. However, one participant felt and suggested that if the assessment unit was fully implemented, it could provide a more realistic feeling of the prototype to the users.

The evaluation also shows that the educational value of the Cognitive Trait Model can benefit both the courseware developers, and the students. The courseware developers can see if their teaching material (courseware) is suitable for the students for their different attributes (such as different ages). The instructor can easily keep track of the students' learning results, and possibly provide assistance when students need it. The students can be benefited by increased learning results, and improved willingness to engage learning more actively due to the helpful feedbacks they get.

One point mentioned by multiple participants in many parts of the evaluation is that the CTM could prevent overloading the cognitive load of the students in their study.

The emphasis and frequency of appearance indicates that nowadays information overloading is an issue that need urgent attention.

13.7: Summary

This chapter demonstrated a web-based tutorial that incorporates a Cognitive Trait Model, and the evaluation of it. The prototype shows how the CTM can be used in technology supported learning environment.

The creation of the content and the metadata (including relations) of the learning objects were done manually for this tutorial. It was found to be quite a time-consuming and tedious task. An authoring tool for specifying the metadata and integrating learning objects into a curriculum would greatly enhances the usability of the CTM, and therefore is listed for future improvement.

Chapter 14: Discussion and Future work

14.1: Discussion

Quinton (2001) pointed out, in the discussion of meaningful learning, that learning should be “*truly holistic, flexible, dynamic, multidirectional, and adaptable*”, and the role of a learning theory should accommodate those properties in order to provide a guide to comprehensive learning by customisable education in a technologically focused future.

The strength of adaptivity in education does not just lie in its novelty, experimental studies have substantially proven its ability to enhance learning speed, memorisation and comprehension; its customisable property makes it a preferable choice for life-long learning systems.

This chapter provides a very broad outline on the issue of user adaptation in virtual learning environments (VLEs). The discussion of value of multimedia in learning systems and the increasing popularity of adaptive hypermedia systems shows the significance of the evolutionary idea of custom-made education and adaptativity in learning environments. The discussion of adaptive navigational support and adaptive content presentation demonstrates the state-of-the-art adaptive techniques used in most of the current adaptive VLEs. The need of integrated theory on adaptation is obvious, as most the adaptive techniques do not identify with each other due to the lack of a firm theoretical background; therefore Exploration Space Control (Kashihara et al., 2000) and Multiple Representation approach (Kinshuk et al., 1999) were cited to address this issue. As the benefits of those integrating theory are known, Exploration Space Control Formalisation serves as a tactical plan on how the strategies suggested by the theories can be carried out.

As most of the current student models are only competence-oriented, the adaptation can only be provided in a performance-based manner which means that the VLEs adapt itself in those areas that the student's performance are identified (or predicted)

to be sub-optimal. The fact is that human uses similar mechanisms to solve problems in a variety of tasks (Reisberg, 1997), and therefore, there exists an opportunity that the improvement of those mechanisms can lead to improved learning. This fact is left un-recognised, or deliberately un-tackled due to its inherent complexity and the fact that *“trained cognitive psychologists with experience in information technology are few and far between in industry and are certainly not available to most product designers ‘on tap’ to contribute to the design of specific products under development”* (May et al., 1995).

Therefore, the research on Exploration Space Control Formalisation (Lin, 2002) and Cognitive Trait Model are profoundly based on cognitive science as the keystone for all formalisation tasks, and the reason for that is due to the fact that not very instructional method can be directly compared with other alternatives (it could be very expensive to do so) (Koedinger, 2001). This necessitates a way to guide the generation of instructional designs which should help to trim parts of the design that are improbable to enhance students’ learning (Koedinger, 2001), and cognitive science serves this purpose well.

Cognitive Trait Model (CTM) tries to provide a supplementary module to any existing VLEs that wish to support adaptation also in the cognitive level. CTM can be integrated in existing framework of AHSs by addition of interaction modelling, and trait analysing facilities. The existing competence model can be used for performance-based adaptivity while CTM can supply the adaptivity that address the differences of each individual’s cognitive abilities. However, due to the complex and interdisciplinary nature of the CTM approach, more evaluation is required in terms of its accuracy and implementability. Several opportunities for improvement are identified and will be discussed next.

14.2: Future Improvement

As this is just a preliminary probe into a new ways of looking at student modelling, it indeed requires a lot more research, experiments, and developments. Some of the

possible future improvements on both the theoretical and practical aspects of the points raised in earlier chapters are discussed in this section of the thesis.

More work has to be done on the research of the student's navigational pattern and their implications. Nelson et al. (1993), and Beasley & Vila (1992) had substantive research on the user interaction and the linearity of the interaction, they are both cited by Andris & Stueber (1994).

Nelson et al. (1993) had researched quantitative and qualitative techniques for collecting and analysing user interaction data in hypermedia systems. Beasley & Vila (1992) developed a system called LinkWay folder, and measured how the linearity and non-linearity of students navigational patterns related to their gender and their ability. Their results show that low ability students tended to navigate in a more linear manner. *"However, the literature does not reveal any standard measures of navigational patterns, and most especially such measures whose reliability and validity have been established"* (Andris & Stueber, 1994). Other aspects of the navigational pattern including reverse navigation, excursion, and others also requires more efforts in order to acquire more scientific backing.

So far, the CTM approach provides only qualitative technique. Further research work on experimental cognitive psychology (Richards-Ward, 1996; Reisberg, 1997), and psychometrics (Revelle, 1995), which is the branch of psychology concerned with the design and analysis of research and the measurement of human characteristics, is required in order to provide qualitative and explicit techniques for the trait analysis in CTM.

More advanced techniques to collect student data or analysis could increase the accuracy of the CTM. Calvo (2001) used on-line eye-movement monitoring facility to examine the elaborative inference, which is an ability that is heavily dependent on the resource available in working memory. The evaluation of this technique showed improved accuracy of the measurement (Calvo, 2001). Similar techniques can be used when this kind of technology is mature and wide-spread enough.

Authoring tool for learning object metadata insertion, and integrating learning objects is also an area that could improve the usability of the CTM model and creating curriculum using the concepts of learning objects. As the learning object metadata has been standardised (LOM, 2002), there will be more researches and learning materials appear around this concept. It is certainly a good opportunity for researching and developing an automated authoring tool for this purpose.

At the discussion of the implementation pattern of simultaneous tasks, tasks are defined by their structures, i.e. activation of a learning object A is thought to be within the same task to the another learning object B, if A and B has the same structural parents by the IsPartOf and HasPart relations. Researches are required for more detailed definition of tasks, or other method to define tasks other than by their structures.

Another area that requires more effort is the study about the temporal information of student's action. Recording student's actions by a time stamp could allow the system to find out the duration of a task which could be learning a concept in a learning object, or performing a procedure that is represented by a series of learning objects. The question need to be asks about whether the temporal information has any value to the system or they are just an overhead, and how it can be best adopted in CTM.

References

- Anderson, J. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.
- Anderson, J. R., Reiser, B. J. (1985). The Lisp Tutor. *Byte*, 10, 159-75.
- Andris, J. & Stueber, A. (1994). Studying Student Navigational Patterns in a Hypermedia Geology Lab Simulation and their Relationship to Learning Style. *Proc the Seventeenth Annual Conference of the Eastern Educational Research Association* (pp. 16), Sarasota, FL.
- Arthur, W. B. (1994). Inductive Reasoning and Bounded Rationality. [WWW page] URL http://www.santafe.edu/arthur/Papers/El_Farol.html
- Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. In K. W. Spence, and J. T. Spence, *The Psychology of Learning and Motivation: Advances in Research and Theory*, vol. 2. New York: Academic Press.
- Ausubel, D. (1963). *The Psychology of Meaningful Verbal Learning*. New York: Grune & Stratton.
- Baddeley, A. D. (1986). *Working Memory*. Oxford: Oxford University Press.
- Baddeley, A. D. (1992). Working memory. *Science*, 255, 556-559.
- Barnard, P. J. (1993) Modelling Users, Systems and Design Spaces (Esprit Basic Research Action 3066). *Proceedings of HCI 93* (pp.331-342). Available at URL http://www.mrc-cbu.cam.ac.uk/amodeus/abstracts/id/id_wp17.html

- Barnard, P. J., & May, J. (1993). Cognitive Modelling for User Requirements. In Byerley, P.F., Barnard, P.J., & May, J., *Computers, Communication and Usability: Design issues, research and methods for integrated services*. Amsterdam: Elsevier.
- Beasley, R. & Vila. J. (1992). The identification of navigational patterns in a multimedia environment: a case study. *Journal of Educational Multimedia and Hypermedia*, 1, 209-222.
- Beaumont, I. (1994). User modelling and hypertext adaption in the tutoring system: Anatom-tutor. [WWW page]. URL <http://www.edfac.usyd.edu.au/projects/ah/Beaumont.html>
- Benford, S. D., Taylor, I., Brailsford. D., Koleva, B., Craven, M., Fraser, M., Reynard, Ga. T., & Greenhalgh, C. M. (2000). Three-dimensional Visualisation of the World Wide Web. *ACM Computing Surveys Symposium on Hypertext*.
- Burton, R.R., & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. *Intelligent Tutoring Systems*, In Sleeman D. and Brown J., Orlando, FL: Academic Press.
- Bouillion, L. M., and Gomez, L. M. (2001). The case for considering culture entailments and genres of attachment in the design of educational technology. In K. D. Forbus, and P. J. Feltovich, *Smart machines in education* (pp.331 - 348). Menlo Park, California: AAAI Press/ MIT Press.
- Bower, G. H. & Hilgard, E. R. (1981). *Theories of Learning*, Englewood Cliffs: Prentice Hall.
- Boyle, C., & Encarnacion A. O. (1994). Metadoc: An Adaptive Hypertext Reading System. *User Modeling and User-Adapted Interaction*, vol. 4 (pp. 1-19).
- Brooks, J. G., & Brooks, M. G. (1993). *In search of understanding: the case for constructivist classrooms*, Alexandria, VA: American Society for Curriculum Development.

Brown, P. & Hirst, S.B. (1983). Writing Reading Courses: The Interrelationship Of Theory and Practice. In Brumfit, C.J. (ed) (1983). *Language Teaching Projects For The Third World. ELT Documents 116*. England: British Council English Teaching Information Centre.

Bruner, J. (1973). *Going Beyond the Information Given*. New York: Norton.

Brusilovsky, P. (1994). Adaptive Hypermedia: an attempt to analyse and generalise. [WWW page]. URL <http://www.edfac.usyd.edu.au/projects/ah/Brusilovsky.html>

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, vol. 6 (pp. 87-129).

Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware Computer Networks and ISDN Systems. *Proceedings of Seventh International World Wide Web Conference*, vol. 30 (1-7, pp. 291-300).

Brusilovsky, P., Eklund, J., and Schwarz, E. (1997). Adaptive navigation support in educational hypermedia on the World Wide Web. *Proceedings of INTERACT97: the 6th IFIP World Conference on Human-Computer Interaction* (pp. 278-285). New York: Chapman and Hall.

Brusilovsky, P. & Pesin, L. (1998). Adaptive Navigation Support in Educational Hypermedia: An Evaluation of the ISIS-tutor. *Journal of Computing and Information Technology*, vol. 6 (1), 27-38.

Brusilovsky, P. and Schwarz, E. (1997) User as student: Towards an adaptive interface for advanced Web-based applications. In: A. Jameson, C. Paris and C. Tasso (eds.) (Proceedings of 6th International Conference on User Modeling, Chia Laguna, Sardinia, Italy, June 2-5, 1997) Wien: Springer-Verlag, pp. 177-188.

Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman, & J. S. Brown, *Intelligent tutoring systems* (pp 157-182) London: Academic Press.

Byrne, M. (1996). A Computational Theory of Working Memory. [WWW page].
URL http://www.acm.org/sigchi/chi96/proceedings/doctoral/Byrne/mdb_txt.htm

Calvo, M. G. (2001). Working memory and inferences: Evidence from eye fixations during reading. In S. E. Gathercole, *Short-term and working memory*. East Sussex, UK: Psychology Press Ltd.

Campbell, J. I., & Charness, N. (1990). Age-related declines in working memory skills: Evidence from a complex calculation task. Special Section: Cognitive training in later adulthood. *Developmental Psychology*, 26, 879-888.

Carroll, J., Mack, R., Lewis, C., Grischkowsky, N., & Robertson, S. (1985). Exploring exploring a word processor. *Journal of Human-Computer Interaction*, vol. 1. 283-307.

Case, R. (1995). Capacity-based explanations of working memory growth: A brief history and reevaluation. In F. E. Weinert & W. Schneider, *Memory performance and competencies: Issues in growth and development* (pp. 23-24). Mahwah, NJ: Erlbaum.

Clancey, W. J. (1983). GUIDON. *Journal of Computer-Based Instruction*, 10 (1), 8-14.

Cofer, C. N., Chmielewski, D. L., & Brockway, J. F. (1975). Constructive processes and structure of human memory. In C. N. Cofer, *The structure of human memory*. San Francisco: W. H. Freeman and Company.

Coorough, C. (1998). Chapter1, Getting started with multimedia. *Introduction to multimedia* (pp. 1-24). Orlando, FL: Dryden.

- Costa, P. D, Oliveira, A & Vaz, J. (1991). Hypermedia and ITS. In. Oliveria A, *Hypermedia courseware: Structures of communication and intelligent help*. Berlin: Springer-Verlag.
- Craik, F. & Lockhart, R. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning & Verbal Behavior*, vol. 11, 671-684.
- Cronbach, L., & Snow, R. (1989). Aptitude-Treatment Interaction as a framework for research on individual differences in learning. In P. Ackerman, R.J. Sternberg, & R. Glaser, *Learning and Individual Differences*. New York: W.H. Freeman.
- Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behaviour*, 19, 450-466.
- De Bra, P., Brusilovsky, P., & Houben, G. J. (1999). Adaptive Hypermedia: From Systems to Framework: *ACM Computing Surveys*, 31 (4), [WWW page]. URL http://www.cs.brown.edu/memex/ACM_HypertextTestbed/papers/25.html
- De Bra, P., & Calvi, L. (1997). Creating Adaptive Hyperdocuments for and on the Web. *Proceedings of the AACE WebNet Conference* (pp. 149-155). Toronto.
- De Bra, P., & Calvi, L. (1998). Towards a Generic Adaptive Hypermedia System. *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia* (pp. 5-11). Pittsburgh, PA.
- Duffy, T. M., & Jonassen, D. H. (1991). New implications for instructional technology? *Educational Technology*, 31 (3), 7-12.
- Eliot, C., & Woolf, B. P. (1995). An adaptive student centered curriculum for an intelligent training systems. *Journal Of User Modeling and User-Adapted Interaction*, vol. 5 (1), 67-85.

Eklund, J. & Zeiliger, R. (1996). Navigating the Web: Possibilities and Practicalities for Adaptive Navigational Support. [WWW page]. URL

<http://ausweb.scu.edu.au/aw96/tech/eklund1/>

Eliot, C., Neiman, D., & Lamar, M. (1997). Medtec: A Web-based intelligent tutor for basic anatomy. WebNet'97, World Conference of the WWW, Internet and Intranet, AACE. 161-165.

El-Sheikh, E. (1997). A Functional Student Model for an ITS [WWW page]. URL

<http://www.cse.msu.edu/~elsheikh/cse845/paper.html>

El-Sheikh, E., and Sticklen, J. (1998). Using a Functional Approach to Model Learning Environments. *Model Based Reasoning for Intelligent Education Environments Workshop, ECAI'98*. European Conference on Artificial Intelligence, Brighton, UK.

Felder, R. M. (1988). Learning and teaching styles in engineering education.

Engineering Education, 78 (7), 674 - 681.

Fischer, G. (2001). User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction 11*, 65-86.

Forbus, K. D., and Feltovich, P. J. (2001). From this revolution to the next. In K. D. Forbus, and P. J. Feltovich, *Smart machines in education* (pp. 421 - 425). Menlo Park, California: AAAI Press/ MIT Press.

French, B. M. (n.d.). Working Memory. [WWW page]. URL

http://www.psych.ualberta.ca/~mike/Pearl_Street/OldDictionary/W/working_memory.html

Glibert, J.E., Han, C.Y. (1999). Arthur: An Adaptive Instruction System Based on Learning Styles. In *Proceedings of ED-MEDIA 1999-World Conference on*

Educational Multimedia, Hypermedia & Telecommunications, Seattle, Washington, USA.

Goldman-Rakic, P.S. (1987). Circuitry of primate prefrontal cortex and regulation of behavior by representational memory. In F. Plum, Ed.), *Handbook of physiology: The nervous system: Section 1, Vol. 5: Higher functions of the brain, Part 1* (pp. 373-417). Bethesda: American Physiological Society.

Goldstein, P. (1982). The genethic graph: a representation for the evolution of procedural knowledge. In D. Sleeman, J. S. Brown, *Intelligent tutoring systems* (pp 51-78). London: Academic Press.

Green, A. J. K. (1994) An Exploration of the Relationships between Task Representations, Knowledge Structures and Performance. Interacting with Computers, 6, 61-85. [WWW page] URL http://www.mrc-cbu.cam.ac.uk/amodeus/abstracts/um/um_wp12.html

Guthrie, E.R. (1938). *The Psychology of Human Conflict*. New York: Harper.

Halff, H. M. (1998). Curriculum and instruction in automated tutors. In M.C. Polson & J.J. Richardson, *Foundation of intelligent tutoring systems* (pp. 79-108). NJ: Lawrence Erlbaum.

Han, S. G., Yoon, J. S., & Jo, G. S. (2001). Case-based student model using knowledge markup language for intelligent e-learning systems [WWW page]. URL http://www.ifcomputer.com/inap/inap2001/program/inap_han.pdf

Heit, E. (2000). Properties of inductive reasoning. *Psychonomic Bulletin & Review*. [WWW page]. URL <http://www-psych.stanford.edu/~jbt/205/heit.pdf>

Henry, L. (2001). How does the severity of a learning disability affect working memory performance? In. S. E. Gathercole, *Short-term and working memory*. East Sussex, UK: Psychology Press Ltd.

Hohl, H., Boecker, H., & Gunzenhaeuser, R. (1996). HYPADAPTER: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modeling and User-Adapted Interaction*, vol 6 (2-3), 131-155.

Hothi, J., & Hall, W. (1998). An Evaluation of Adapted Hypermedia Techniques using Static User Modeling. *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia* (pp. 45-50). Pittsburgh, PA.

Hume, G. D. (1995). Using student modeling to determine when and how to hint in an intelligent tutoring system [WWW page]. URL <http://www.csam.iit.edu/~circsim/documents/ghdiss.pdf>

Jacobs, G. (1992). Hypermedia and discovery-based learning: A historical perspective. *British Journal of Educational technology*. Vol23 (2, pp. 113-121).

Jameson, A., Schlfer, R., Weis, T., & Weyrath, B. T. (1999). Making Systems Sensitive to the User's Time and Working Memory Constraints [WWW page]. URL <http://www.iuiconf.org/99pdf/1999-001-0014.pdf>

Jonassen, D., & Wang, S. (1990). Adding intelligence to hypertext with expert systems and adding usability to expert systems with hypertext. In Oliveria A, *Hypermedia courseware: Structures of communication and intelligent help*. Berlin: Springer Verlag.

Kashihara, A., Kinshuk, Oppermannn, R., Rashev, R., & Simm, H. (2000). A Cognitive Load Reduction Approach to Exploratory Learning and Its Application to an Interactive Simulation-Based Learning System. *Journal of Educational Multimedia and Hypermedia*, 253-276.

Kausler, D. H. (1991). *Experimental psychology, cognition, and human aging (2nd ed.)*. New York: Springer-Verlag.

Kay, J., & Kummerfeld, R. J. (1994). An individualised course for the C programming language. [WWW page]. URL

<http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.htm>
1

Kearsley, G. (1998). Theory into practice (TIP) database. [WWW page]. URL
<http://www.gwu.edu/~tip/index.html>

Kearsley, G. (2001). Explorations in Learning & Instruction: The Theory Into Practice Database [WWW page]. URL <http://tip.psychology.org/scandura.html>

Keates, S., & Robinson, P. (1997). User performance modeling and cognitive load [WWW page]. URL
<http://rehab-www.eng.cam.ac.uk/papers/lsk12/resna97/resnah.htm>

Kinshuk, Oppermann, R., Patel, A., Kashihara, A. (1999). Multiple representation approach in multimedia based intelligent educational systems. In S. P. Lajpie & M. Vivet. *Artificial Intelligence in Education: ISO Press* (pp. 259-266). Amsterdam.

Kobsa, A., Miller, D. & Nill, A. (1994). KN-AHS:AN adaptive hypertext client of the user modeling system BGP-MS. *Fourth International Conference on User Modeling*. Massachusetts, USA: Mitre Corporation.

Koedinger, K. R. (2001). Cognitive tutors as modelling tools and instructional model. In K. D. Forbus, and P. J. Feltovich, *Smart machines in education* (pp.145 - 168). Menlo Park, California: AAAI Press/ MIT Press.

Kommers, P. (2000). Imagination through Virtuality for in-depth Learning. Keynote Speeches. IWALT 2000. [available as WWW page]. URL
<http://lttf.ieee.org/iwalt2000/keynotes.html>

Laurillard, D. (1993). *Rethinking University Teaching, a Framework for the effective use of Educational Technology*. London: Routledge.

Lin, T. (2002). Exploration Space Control Formalisation. *Research Report*. Palmerston North, New Zealand: Massey University.

Lovett, M. C., Daily, L. Z., & Reder, L. M. (2000). A Source Activation Theory of Working Memory: Cross-task Prediction of Performance in ACT-R* [WWW page]. URL <http://www.psy.cmu.edu/~LAPS/pubs/Lovett00CogSysResearch2.pdf>

LTSC IEEE (2002). Draft Standard for Learning Object Metadata. Retrieved 10 September, 2002 from http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf

May, J, Barnard, P. J., & Blandford, A. (1995). Using Structural Descriptions of Interfaces to Automate the Modelling of User Cognition. *International Journal of Human Computer*. Available at URL http://www.mrc-cbu.cam.ac.uk/amodeus/abstracts/um/um_wp01.html

Mayo, M. J. (2001). Bayesian Student Modelling and Decision-Theoretic Selection of Tutorial Actions in Intelligent Tutoring Systems. *PhD Thesis*. Available at URL http://www.cosc.canterbury.ac.nz/research/reports/PhdTheses/2001/phd_0102.pdf

Martin, B. (1999). Constraint-Based Modeling: Representing Student Knowledge. *New Zealand Journal of Computing*, 7_(2), pp 30-38. Available at URL http://www.cosc.canterbury.ac.nz/~bim20/paper_nzjc.pdf

Merrill, M. D. (2000). Knowledge object and mental models. In Kinshuk, J. Chris, T. Okamoto, Proc. Of International Workshop on Advanced Learning Technologies: Advanced Learning Technology: Design and Development Issues: IWALT 2000 (pp. 244-246). USA: IEEE Computer Society.

Meyer, B., Sit, R. A., Spaulding, V. A., Mead, S. E., & Walker, N. (1997). Age Group Differences in World Wide Web Navigation [WWW page]. URL <http://www.acm.org/sigchi/chi97/proceedings/short-talk/bm.htm>

Miller, G. (1956). The magic number seven, plus or minus two: Some limit of our capacity for processing information. *Psychology Review*, vol. 63 (2), 81 –96.

Mitchell T. (1997). *Machine Learning*. The McGraw-Hill Companies, Inc.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal on Artificial Intelligence in Education*, 10 (3-4), 238-256.

Muan, H. N., Hall, W., Maier, P. J., & Armstrong, R. (2001). History-based Link Annotation for Self-Exploratory Learning in Web-based Hypermedia. [WWW page]. URL <http://www.wis.win.tue.nl/ah2001/papers/hong.pdf>

Mukherjea, S. (2000). Information Visualization for Hypermedia Systems. *ACM Computing Surveys, Symposium on Hypertext and Hypermedia*.

Mullier, D. (1999). The Application of Neural Network and Fuzzy Logic Techniques to Educational Hypermedia. *PhD Thesis*. [Available as WWW page] URL <http://www.mullier.co.uk/thesis/thesis.html>

Munro, A., & Pizzini, Q. A. (1996). RIDES Reference Manual [WWW page]. URL <http://btl.usc.edu/rides/documentn/refMan/index.html>

Najjar, L. J. (1996). Multimedia information and learning. *Journal of Educational Multimedia and Hypermedia*, vol. 5, 129-150.

Nakabayashi, K., Maruyama, M., Kato, Y., Touhei, H., & Fukuhara, Y. (1997). Architecture of an intelligent tutoring system on the WWW. In B.D. Boulay and R. Mizoguchi, *Proceedings of AI-ED'97, World Conference on Artificial Intelligence in Education*. Amsterdam: IOS Press.

Nelson, W., Harmon, S. Orey, A., & Palumbo, D. (1993). Techniques for analysis and evaluation of user interactions with hypermedia systems. *Proceedings of ED-MEDIA 93N World Conference on Educational Multimedia and Hypermedia* (pp 585-588), Orlando, Florida.

Paiva, A, Self, J. & Hartley, R. (1995). Externalising LearnerModels. In J Greer, *Proceedings of AIED95*. AACE publication.

Paivio, A. (1986). *Mental Representations*. New York: Oxford University Press.

Pascoe, R., & Sallis, A. (1998). A pedagogical basis for adaptive WWW textbooks. *Proc of North American Web Developers Conference*. [WWW page]. URL http://webfuse.cqu.edu.au/Information/Resources/Readings/html_papers/pascoe/index.html

Pitkow, C. (n.d.). Characterising Browsing Strategies on the Web. [WWW page]. URL http://www.gatech.edu/lcc/idt/Students/Catledge/browsing/user_1.html

Quinton, S. R. (2001). Meaningful learning in a complex information environment. In C. H. Lee, *Enhancement of quality learning through information and communication technology (ICT): Proceedings of the ICCE International Conference On Computers in Education /SchoolNet2001*, 3 (pp. 1326-1333). Seoulm, Korea: Incheon National University of Education.

Raiford, C., & Dudley, J. (2000). Beyond 3,000 WPM. [WWW page]. URL <http://www.integratedcommerce.com/speedreading/>

Revelle, W. (1995). Personality processes. *Annual Review of Psychology*, 46.

Recker, M., Ram, A., Shikano, T., Li, G., & Stasko, J. (1995). Cognitive media types for multimedia information access. *Journal of Educational Multimedia and Hypermedia*. vol 4 (2/3), 183-210.

Reigeluth, C. (1992). Elaborating the elaboration theory. *Educational Technology Research & Development*, vol. 40(3), 80-86.

Reisberg, D. (1997). *Cognition: Exploring the science of mind*. New York: W. W. Norton & Company.

Reddy, E. R. (1980). Machine models of speech perception. In R. A. Cole, *Perception and Production of Human Speech* (pp. 215-242). Hillsdale, NJ: Erlbaum.

Richards-Ward, L. A. (1996). *Investigating the relationship between two approaches to verbal information processing in working memory: An examination of the construct of working memory coupled with an investigation of meta-working memory*.

Palmerston North New Zealand: Massey University.

Rogers, E., Kennedy, Y., Walton, T., Nelms, P., & Sherry, I. (1995). Intelligent multimedia tutoring for manufacturing education. *Frontiers in Education conference*, Atlanta, Georgia.

Salomon, G., (1977). *Interaction of Media, Cognition and Learning*. San Francisco: Jossey-Bass.

Salthouse, T. A., Babcock, R. (1991). Decomposing adult age differences in working memory. *Developmental Psychology*, vol. 27, 763-776.

Salthouse, T. A., Mitcheel, D. R. D., Skovronek, E., & Babcock, R. L. (1989). Effects of adult age and working memory on reasoning abilities. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15, 507-516.

Salvucci, D. D. (2002). Modeling Driver Distraction from Cognitive Tasks [WWW page]. URL http://hfac.gmu.edu/~cogsci/final_ind_files/Salvucci.pdf

Scandura, J.M. (1973). *Structural Learning I: Theory and Research*. London: Gordon & Breach.

Shortliffe, E. H. (1976). *Computer-based Medical Consultations: MYCIN*. New York: American Elsevier Publishers.

Smith, S. (1998). Tutorial on [WWW page]. URL <http://www.cs.mdx.ac.uk/staffpages/serengul/table.of.contents.htm>

Staff, C., 2001. HyperContext: A Framework for Adaptive and Adaptable Hypertext. PhD Thesis, University of Sussex.

Specht, M., Weber, G., Heitmeyer, S., & Schöch, V. (1997). AST: adaptive WWW-courseware for statistics, In P. Brusilovsky, J. Fink and J. Kay, *Proceedings of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling: UM97*, 91–95.

Stankov, S. (1996). Student model developing for intelligent tutoring systems. *International Journal for ENGINEERING MODELLING*, 9 (1-4), 35-41. Available at URL http://www.pmfst.hr/~stankov/zn_radovi/1996/coins/COINS_rad.PDF

Stauffer, K. (1996). Applications of Student Modeling [WWW page]. URL http://ccism.pc.athabascau.ca/html/students/stupage/Project/sm_app.htm

Sun, C.T. and Ching, Y.T. (1995). Hypermedia browsing pattern analysis. *International Journal of Educational Telecommunication* 1(2/3), pp.283-308.

Tam, M. (2000). Constructivism, Instructional Design, and Technology: Implications for Transforming Distance Learning. *Educational Technology & Society* 3_(2).

Tang, T. Y., Chan, K., Winoto, P., & Wu, A. (2001). Forming student clusters based on their browsing behaviors. In C. H. Lee. *Enhancement of quality learning through information and communication technology (ICT): Proceedings of the ICCE International Conference On Computers in Education /SchoolNet2001*, 3 (pp.1229-1235). Seoul, Korea: Incheon National University of Education.

Tarpy, E. M., & Mayer. R. E. (1978). *Foundations of learning and memory*. London: Scott, Foresman and Company.

Tebo, A. (1994). Artificial neural networks: a developing science. [WWW page] URL http://www.spie.org/web/oer/september/neural_net.html

Tweedie, L. and Barnard, P. J. (1992). The interactive talk: a new tool for presenting complex theory. *Psychology Software News*, 3, 43-45. Available at URL http://www.mrc-cbu.cam.ac.uk/amodeus/abstracts/um/um_wp04.html

Urban-Lurain, M. (1996) Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology [WWW page]. URL <http://www.cse.msu.edu/~urban/ITS.htm>

Vassileva, J. (1997) Dynamic Course Generation on the WWW. Proceedings of AIED'97.

Virvou, M., Maras, D., & Tsiriga, V. (2000). Student Modelling in an Intelligent Tutoring System for the Passive Voice of English Language. *Educational Technology & Society* 3 (4). Available at URL http://ifets.ieee.org/periodical/vol_4_2000/virvou.html

Webb, G. I., Pazzani, M. J., & Billsus, D (2001). Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction 11*: 19-29. Available at URL <http://www.ics.uci.edu/~pazzani/publications/webb.pdf>

Weber, G., & Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In Anthony Jameson, Cecile Paris, and Carlo Tasso (Eds.) *User Modeling: Proceedings of the Sixth International Conference, UM97* (pp. 289-300). Vienna, New York: Springer Wien New York.

Wickelgren, W. A. (1977). *Learning and memory*. N.J: Prentice-Hall Inc.

Wiley, D. (2000). learning object Design And Sequencing Theory. *PhD Thesis*. Brigham Young University.

William, M. K. (2001). Deductive and Inductive Thinking. [WWW page] URL <http://trochim.human.cornell.edu/kb/dedind.htm>

Zao, Z, O'Shea, T., & Fung, P. (1993). Visualisation of semantic relations in hypertext systems. In T. Ottman & I. Tomek, *Educational multimedia and hypermedia: Proceedings of Ed-Media 93* (pp. 556-564). AACE.

Zhou, Y., Evens, M. W. (1999). A Practical Student Model in an Intelligent Tutoring System. Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence. Chicago, IL, 1999, pp. 13-18. Available at URL <http://www.csam.iit.edu/~cirsim/documents/yzitai99.pdf>

Appendix 1: Statistical Result of the Simulation

While the parameters are set: HPOS=6, LPOS=4, P=0.003,
The results were obtained when the simulation runs 400 sessions:

No of Correct Prediction	No of Incorrect Prediction	Incorrect Rate
379	21	0.0525
386	14	0.0350
381	19	0.0475
383	17	0.0425
380	20	0.0500
382	18	0.0450
377	23	0.0575
374	26	0.0650
384	16	0.0400
379	21	0.0525

Average Incorrect Rate: 0.04875

Number of Incorrect Predictions after 50 th Session	Number of Incorrect Predictions after 200 th Session
9	1
16	2
13	2
8	2
13	0
16	0
15	3
6	2
12	3
10	3
Average Incorrect Rate: 0.0295	Average Incorrect Rate: 0.0045

Appendix 2: Evaluation of Cognitive Trait Model

This tutorial is available at: <http://www.entek.net.nz/recursion>

Other information relating to the Cognitive Trait Model can be requested from me by email me at tt3838@hotmail.com.

Brief Introduction to Concepts Used in This Evaluation

Student modelling is the process of creating a representation of student in learning systems (usually technology-based systems), and this representation is often named as the student model. Student models contain the information about individual students.

This work intends to describe a new approach for student modelling, in particular an approach for modelling students' cognitive traits.

Cognitive traits are those characteristics of human beings that relatively persistent in time, and consistent across domains; for example working memory capacity.

Working memory is also referred as short-term memory. It denotes the memory capable of transient preservation of information and is functionally different from the memory that stores historical information (the long-term memory).

The aim of this work is to develop a student model that can assists the learning systems to provide adequate learning materials to students based on the students' cognitive ability. For example, when a student's working memory capacity is low, the system should provide more hints. Another aim of the Cognitive Trait Model is to prevent overloading the students' cognitive capacity (such as too much information) that may discourage them to engage future learning.

Introduction to the Prototype

A prototype web-based tutorial of recursion was created to demonstrate the implementation of Cognitive Trait Model (CTM) and its components. The purpose of this prototype is not to create a complete system but to show the potential of CTM.

The domain of the prototype (i.e. the Recursion Programming) is constructed by learning objects. A learning object in this prototype corresponds to a web page. Learning objects has relations to other learning objects. For example, learning object A may has an **IsPartOf** relation to learning object B, and has a **IsBasisFor** relation to learning object C. Relation-based navigational analysis is used in this work to analyses students' navigational behaviour in the learning session. More information of learning objects and relation-based navigational analysis is available in the Chapter 9 of the thesis.

Structure of the Domain (Optional for Evaluators to Fully Read this Section)

In this prototype, a simple tutorial on the use of recursion in computer programming language is created. The examples and listing in the tutorial is in C language but any other programming language that supports recursion can use this approach. The main focus of this tutorial is on the idea of recursion, principles of writing recursive functions, and good programming practice.

The tutorial is completely web-compatible. It consists of twenty-two nodes (Figure 13.1) each of which can roughly approximate to single web-page except the recursion nodes which bring the learners to other websites of related topic. This tutorial has sixty-one relations which are mainly of six types: **IsBasisFor**, **IsBasedOn**, **HasPart**, **IsPartOf**, **ExcursionTo**, and **IsExcursionOf**.

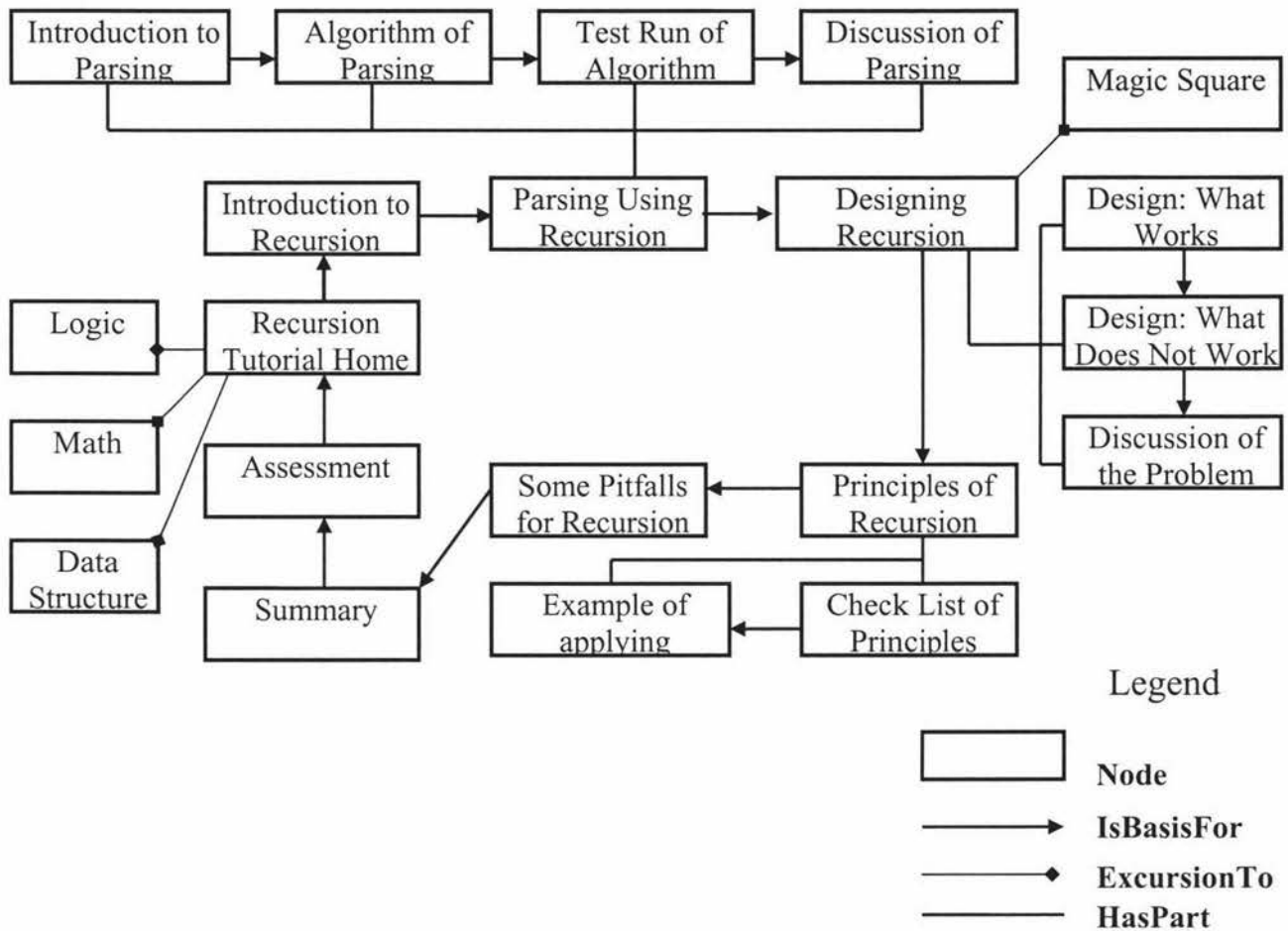


Figure 13.1: Structure of the Domain: Tutorial of Recursion

The logical start of the tutorial is Recursion Tutorial Home node, but it is a strict limit. The learners can go through the tutorial in a linear fashion by following the IsBasisFor and HasPart relations, or they can explore more examples and different perspectives about recursive programming by the ExcursionTo relations. After the assessment page, the learner is automatically brought back to the Tutorial Home. This provides a means in this prototype to experiment with sessions of different length.

Brief Description of the Tutorial

On the top of every page, there is a navigation bar in which the “next” links to the pre-determined page and in the bottom of every page there is a view report link. There are sometimes excursions available for a particular node, which bring the learners to other websites of related concepts. For example, excursions available in the

Introduction to Recursion (figure 13.2) are recursions to **Data Structure and Algorithm, Logic, and Mathematics**.

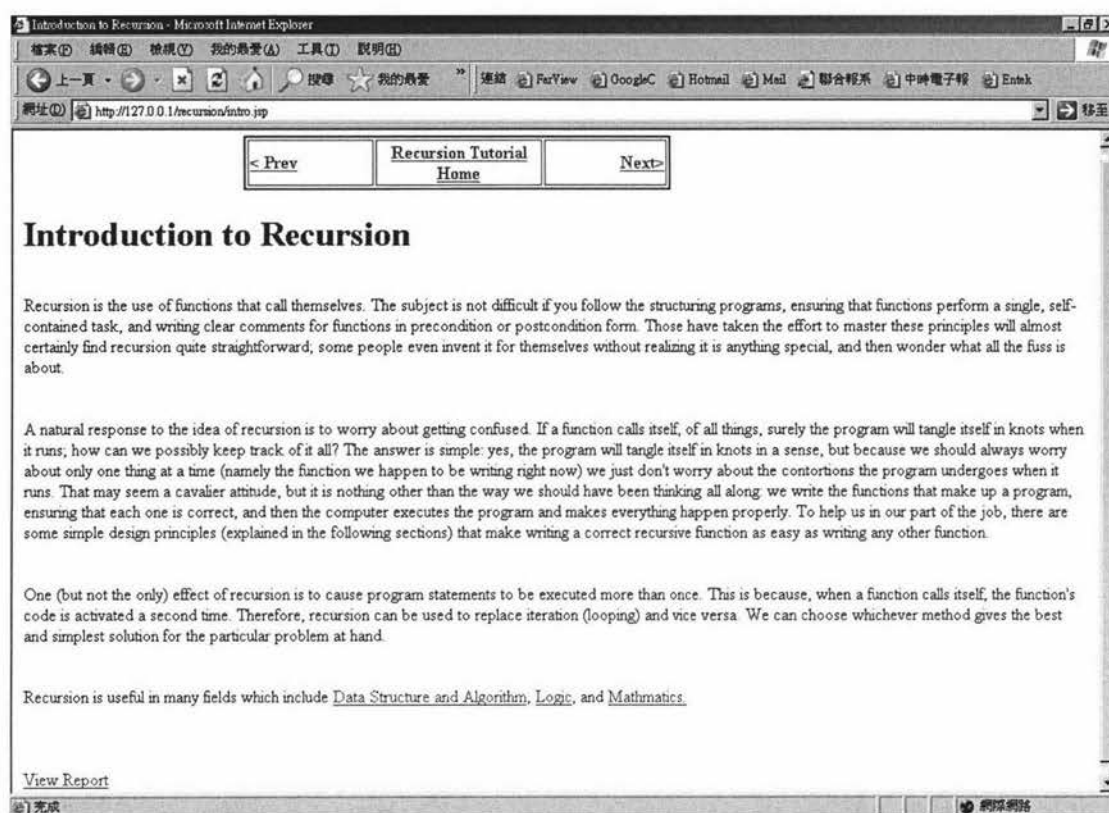


Figure 13.2: Introduction to Recursion

Next, figure 13.3 shows the excursion to the **Mathematics**. Recursions take the learner away from the host, called main host, where the tutorial is located, and thus learners activities in the excursion web site are un-trackable. However, the activations the relations *ExcursionTo*, and *IsExcursionOf* are still in the control of main host and therefore these activations can be recorded in the Student Behaviour History (figure 10.1).

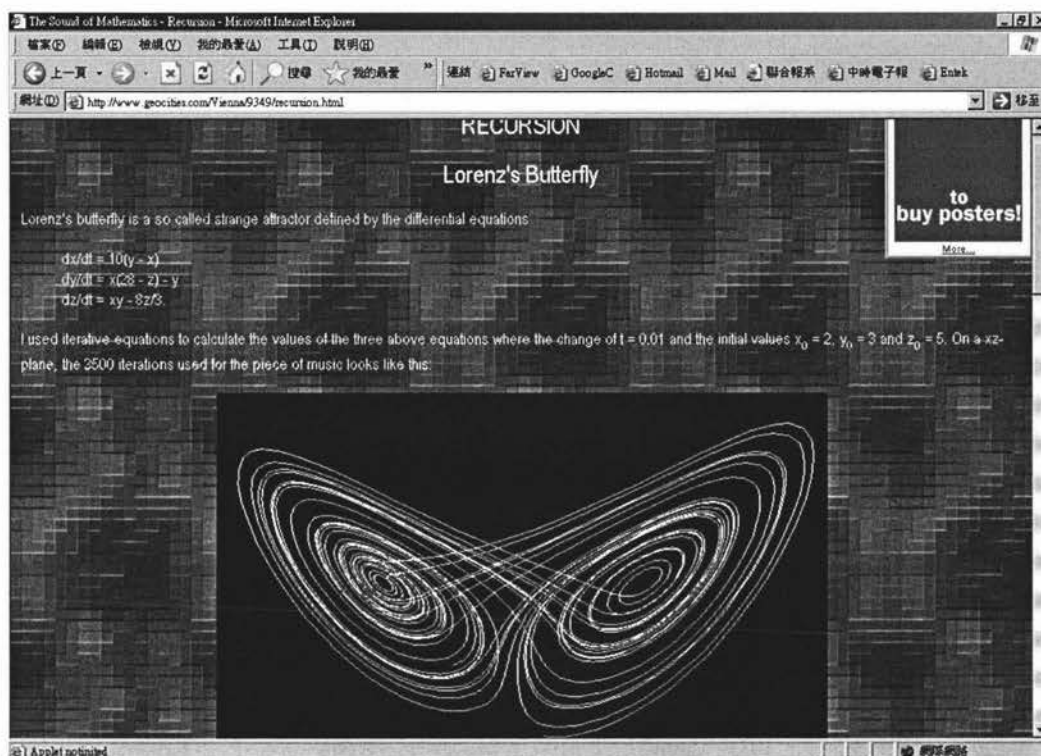


Figure 13.3: Excursion to the Mathematics node

When the learner clicks the Next button in the **Introduction to Recursion**, the tutorial brings the student to the **Introduction to Parsing** (figure 13.4).

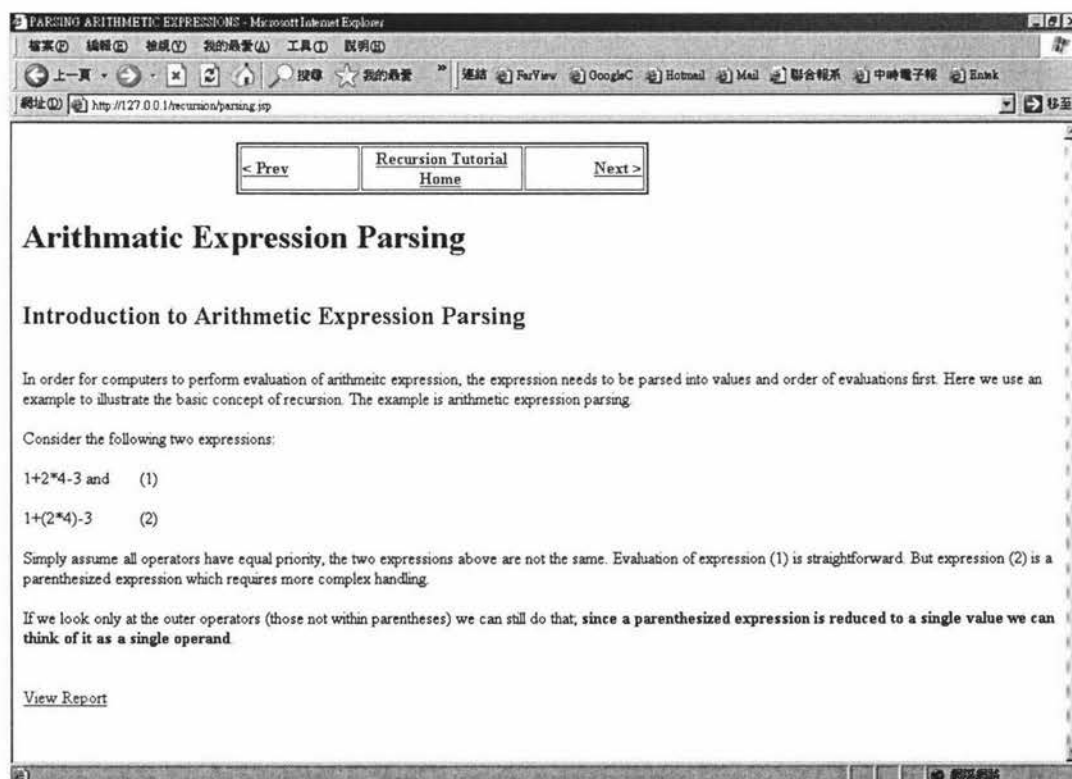


Figure 13.4: Page for the introduction to the arithmetic expression parsing

Please note that in figure 13.1, the **Introduction to Recursion** has an **IsBasisFor** relation to the **Introduction to Parsing**, which is actually a logical learning object only. Logical learning objects are used to relate learning objects semantically, but not necessarily physically (by hyperlinks). A single action of transferring from Introduction to Recursion to Parsing Using Recursion actually activates two relations: an **IsBasisFor** relation (from **Introduction to Recursion** to **Parsing Using Recursion**) and a **HasPart** relation (from **Parsing Using Recursion** to **Introduction to Recursion**).

Technical Details of the Prototype

The creation of the web-based exemplary tutorial includes a simple Cognitive Trait Model (CTM). This tutorial implements the interface module, the student behaviour history, and trait analyser, and a mark-up student performance model.

The Interface Module

The tutorial is designed to be web-based and the hence its interface to the learner is via the web-browser. The nodes in the tutorial are implemented by Java Server Page™ technology in order to be able to create content dynamically. Tomcat, which is a free, open-source implementation of Java Servlet™ and JavaServe Pages™ technologies developed under the Jakarta project at the Apache Software Foundation, is used as a standalone web server that the tutorial is installed on.

Recording of learner navigations is implemented in the form of query string. In the case when an excursion link is clicked, the learner intends to move to a page that is not in the server (host) of this tutorial, and therefore a buffer page is created for every excursion which will automatically re-direct the web browser to the excursion site. The buffer page serves the function to record the activation of **ExcursionTo** and **IsExcursionOf** relations. But the subsequent pages after the first excursion page are beyond the control of the host, therefore they cannot be not recorded.

The Student Behaviour History

The student behaviour history is implemented by database records of the learner's navigational history. A navigation consists of the source, the destination, the time, and the relation type of this navigation. The source and destination of a navigation can be used to know which of the learning objects have been visited and when. The relation of a navigation is used by the trait analyser in which the relations are of primary importance for many analysis tasks. The exemplary trait analysis presented later in this chapter will show how the relations are used.

The Trait Analyser

The Trait Analyser is implemented as Java Beans, which are component-oriented Java objects. The activation of the Trait Analyser should be scheduled at the end of every learning session. However, for the ease of demonstrating how the Trait Analyser works, the trait analyser is available at any point of the tutorial view the "View Report" link.

The Trait Analyser has three parts that forms a chain of operation (a pipeline), they are:

4. **Pattern Detector,**
5. **Individualised Temperament Network,** and
6. **CTM Updater.**

Upon the activation of the Trait Analyser, the Pattern Detector begins to collect data of current learning session from the Student Behaviour History, and performs pattern-matching operations. The pattern-matching operations require the data of relation types and the relations between learning objects. These data are stored in an internal database, and the data is created during the creation of the tutorial. Implementation patterns are sought in the learner's navigational path, and all the implementation

patterns (such as the linearity, reverse navigation, and so on) detected are outputted to the Individualised Temperament Network (ITN) with their frequency of occurrences.

Once the ITN get a set of data that is the output from the Pattern Detector, it starts the retrapolling process. Retrapolling involves the current status (weights) of the ITN and the new set of data. The new set of data is processed according to their corresponding weight values in the ITN, and a polling sum is created. The ITN has to redistribute its weight according to the polling sum, and signals the CTM Updater to update the CTM accordingly.

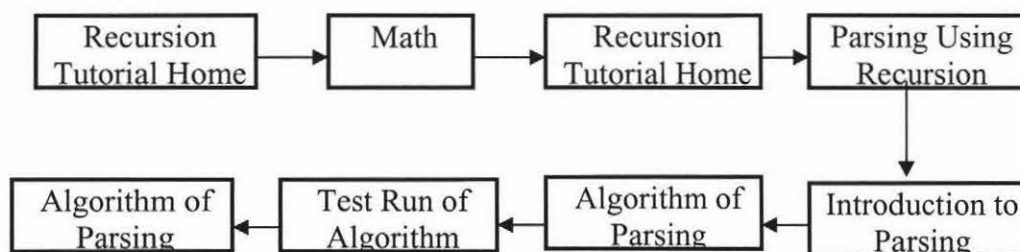
Student Performance Model

This is a mark-up performance model created in order to demonstrate some functions of the trait analysis. The performance model records the result of the assessment page at the end of the tutorial. In a complete system, there should be a module that is responsible to communicate to the independent performance model.

Example of Trait Analysis (Optional for Evaluator)

Learners' actions of navigation are recorded in Student Behaviour History. An exemplary trait analysis will be demonstrated.

For example, a learner performed the sequence of navigation shown in the figure 13.5.



Legend

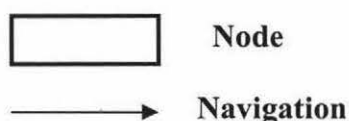


Figure 13.5: An exemplary session

The relations recorded in those navigation are sequentially: ExcursionTo, IsExcursionOf, IsBasisFor, HasPart, IsBasisFor, and IsBasedOn. Those relations can only construct three types of implementations, i.e. navigational linearity, reverse navigation, and excursions. Those implementation patterns will be discussed next. And for the sake of discussion, we assume that this learner failed the assessment after those navigations.

Navigational Linearity (Optional for Evaluators)

The implementation pattern for navigational linearity is by the measure of LM:

$$LM = (No(B) / ISBASISFOR(V)) / (No(E) / EXCURSIONTO(V))$$

Where

LM is the measure of linearity

W is the set of all visited learning objects.

V is a learning object in the set of **W**.

No() returns the number of items in a set

ISBASISFOR (V) returns the set of learning objects that V has IsBasisFor relations with.

B is the subset of **ISBASISFOR(V)** where for every item **bi** in **B**, (**bi** $\not\subset$ **W**)

$\not\subset$ means not in the set of.

EXCURSIONTO(V) returns the set of learning object that V has ExcursionTo relations with.

E is the subset of **EXCURSIONTO(V)** where for every item **ei** in **E**, (**ei** \subset **W**)

\subset means in the set of.

If LM is greater than 1, then the navigational sequence is said to be more linear, otherwise it is more non-linear.

For the number of relations, please refer to figure 13.1. In the above navigational sequence, the total number of excursions, $\text{ExcursionTo}(V)$, are three, and the number of excursions visited, $\text{No}(E)$ is one. The number of IsBasisFor , $\text{IsBasisFor}(V)$, relations of all visited node is four whereas the $\text{No}(B)$ is equal to one.

Therefore, the value of $\text{LM} = (1/4) / (1/3) = 0.75 \rightarrow \text{Non-Linear}$
Non-linear navigational pattern manifests LWMC.

Reverse Navigation (Optional for Evaluators)

The implementation pattern for reverse navigation are as follows:

*In a session of learner's interaction, there exist "a certain number of R " where R is an ordered pair and represent a navigational action and at least one of the items in the pair is visited. $R = (\text{fromLearningObject}, \text{toLearningObject})$, where *fromLearningObject* represent the source of the navigational action and *toLearningObject* the destination.*

In the above navigation, it is obvious that the learner indeed performed reverse navigation from Test Run of Algorithm to Algorithm of Parsing. And therefore, for this particular implementation pattern, the result of the analysis is "LWMC" which means the increase of the score of low working memory capacity, and decrease of the score of high working memory capacity.

Excursions (Optional for Evaluators)

The implementation pattern LWMC for Excursions is:

*For every vi in V , if vi has any **ExcursionTo** relation, and if any of its **ExcursionTo** relation is activated and the result of that unit is equal to fail.*

Where

V is the set of all visited learning object in a session.

v_i is a learning object in V .

In the above navigation, the learners take excursion to the **Mathematics** website, and in the end failed the assessment. The result of the trait analysis of this implementation pattern is “LWMC”.

Due to the types of relations available in this tutorial, other implementation patterns do not exist. All three available implementation patterns indicate that the learner has low working memory capacity.

Figure 13.6 shows the page summarising the result of the analysis. The “View Report” link is made available on every page of the tutorial in order to facilitate the ease of examining the status of the Individualised Temperament Network.

All three implementation pattern indicates LWMC. The values of the implementation pattern belonging to the group LWMC are all intensified by -0.003, whereas the HWMC groups have all their values decreased by 0.003.

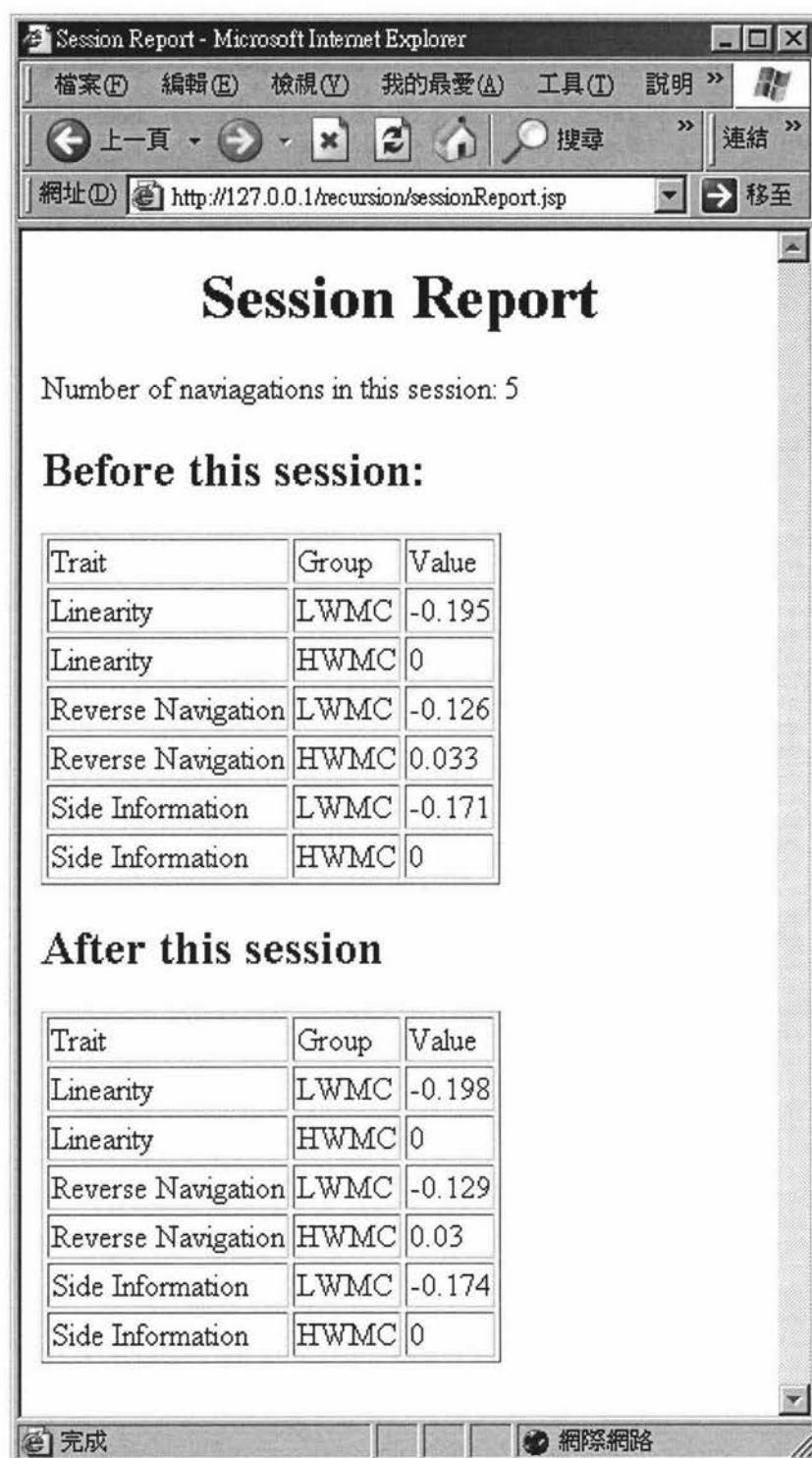


Figure 13.6: Page for the result of the analysis

Please note that in figure 13.6, the Side Information and Linearity in the HWMC group remains zero because for the HWMC group, zero is the minimum value and one is the maximum value whereas for the LWMC group negative one is the minimum literal value and zero is the maximum literal value.

Evaluation Questions - Cognitive Trait Model

1. Do you think that the Cognitive Trait Model presented in this prototype can assist students during their learning?

Strongly disagree Strongly agree

1 2 3 4 5

Please explain, why.

2. Did the prototype demonstrate enough information to show that the Cognitive Trait Model can be used in learning systems?

Strongly disagree Strongly agree

1 2 3 4 5

Please explain, why.

3. If you are a courseware developer, would you consider using Cognitive Trait Model in your systems?

Strongly disagree Strongly agree

1 2 3 4 5

Please give three reasons, why.

a.

b.

c.

4. If you are a student, would you prefer the courses that use Cognitive Trait Model over those that do not?

Strongly disagree Strongly agree

1 2 3 4 5

Please give three reasons, why.

a.

b.

c.

5. How experienced are you in the field of e-Learning?

Not experienced					Highly experienced
1	2	3	4	5	

6. Based on your experience with this prototype, if you were to pick one aspect of the Cognitive Trait Model that is the most valuable for educational purposes, what would that be?

7. Based on your experience with this prototype, if you were to pick one aspect of Cognitive Trait Model that needs most improvement, what would that be?

**** END OF EVALUATION ****

Thanks for participating this evaluation.