## Abstract

Described within this thesis is an approach to educational hypermedia that utilises symbolic AI and connectionist AI to provide generic student modelling. The needs for generic tutoring systems are discussed, in terms of a system that is applicable to a multitude of teaching domains, whilst maintaining diagnostic facilities of the student. The first contribution to knowledge is an approach to educational hypermedia that employs a semantic network allowing the use of automatic reasoning, using symbolic AI, to produce weighted links, the weight being the relevance to the current node according to the semantic network. The novice student is aided by weighted links in that complexity can be reduced by removing the links with the lowest weight and hence relevance. The linking system is further enhanced by the second contribution to knowledge. The second contribution to knowledge concerns a type of student model that is employed to record information about the student so that the weighted links can be tailored towards the student's interests. The student model contains sub-systems to measure and record domain-independent information about the student, which includes their interests, as derived from the semantic network, their ability to complete tutorials and their browsing behaviour. A neural network is used to grade the student into an ability level based upon their interactions with tutorials. A neural network is used because it is able to accomplish the task of grading the student automatically, without manual intervention. A further element of the student model uses a neural network to recognise the movements that students make as they browse the hypermedia. Such movements have been previously identified as offering information to potentially aid the student. This research offers a mechanism to recognise and record these browsing patterns and to map them to the student’s ability and the task that the students were using the hypermedia for.

## Acknowledgements

My thanks and appreciation to:

Dr. David Hobbs, School of Computing, Leeds Metropolitan University

Dr. David Moore, School of Computing, Leeds Metropolitan University

For supervising the project.

Professor J.R. Hartley, Computer-Based Learning Unit, University of Leeds

For providing invaluable advice.

Mum and Dad for a Christmas present in 1982 that was the first step on the road to this thesis.

Finally, to Caroline for her understanding.

## Table of Contents

## Table of Figures and Tables

# Chapter 1

# Introduction

It is widely acknowledged that there is a growing need for computer based learning packages (CBL) to cope with the ever-increasing number of students (Moore et al 1997). In order to combat this need research has been undertaken over a number of years to add artificial intelligence (AI) to CBL packages (electronic textbooks), to directly tailor material to individual students. Such systems are termed "Intelligent Tutoring Systems" (ITS). ITS tend to focus on precise elements of technical domains, for example an ITS for teaching fault diagnosis in electronic circuits. This renders them less suitable for teaching larger domains or a number of different domains. Their nature also requires that the students are measured using a student model that is usually not suitable for other domains (Woods and Warren 1995). By contrast, educational hypermedia, a method for organising information in a non-sequential way, tends to be used for more general domains but does not offer the in-depth diagnosis of the student offered by an ITS (Dillon and Gabbard 1998). Hypermedia is therefore on the opposite end of the computer-based learning spectrum to ITS, in that an ITS constrains the student to learn in a particular fashion and is able to supply direct guidance, whereas educational hypermedia systems tend not to constrain the student in the ways they navigate the material and do not offer guidance (Elsom-Cook 1989). Proponents of hypermedia argue that a student should be able to learn using methods that suit them (Jonassen and Grabinger 1990). There is therefore a gap between ITSs and educational hypermedia. This project is concerned with the investigation of an approach to educational hypermedia that fits into (although not necessarily fills) the gap, and thus potentially gains some of the benefits of both paradigms.

Hypermedia has been criticised for imposing an extra burden upon the student, in terms of navigation, and an extra burden upon the author, in terms of providing every useful link (Theng and Thimbleby 1998, DeBra and Calvi 1998). This project aims to demonstrate that these problems can be alleviated by the addition of several facilities drawn from the fields of AI and ITSs. It will be argued here that the navigation problem can be reduced by simplifying the domain for less advanced students (by offering fewer links, but tailoring them towards the student's particular tasks and interests) and that the authoring problem can be reduced by automatically producing related links. The result is a hybrid system that is, in principle, generally applicable to different domains whilst providing facilities to tailor material individually to a student.

The approach investigated for the current research project uses AI techniques to formally structure a given domain into a set of related concepts. The approach then draws on ideas from ITSs to provide a form of student model (Brusilovsky 1996), which is used to record information about the student that can then be used by inference mechanisms to alter the way information is presented to the student. Information may be tailored according to the student’s level of knowledge of the domain and their interests by using the information stored in the student model.

The approach is potentially applicable to a multitude of domains since evaluations of the student are based upon non-domain-specific metrics, namely their ability to complete on-line tutorials, their movement patterns through the hypermedia and their interests, as derived from their navigation of the formalised domain. The patterns a student makes as they navigate have been identified previously as offering information regarding the task a student is attempting to accomplish (Canter et al 1985, McAleese 1993). The research detailed within this thesis specifies an approach to educational hypermedia that is capable of automatically identifying these "browsing patterns". These browsing patterns are then linked to other information collected about the student, such as ability levels of students (as obtained from tutorials) and the types of task that they are required to undertake (such as "find a specific piece of information").

The student is offered links according to their ability (less for a novice, more for an expert student). However, it is not suitable to randomly exclude links; instead the system makes an evaluation of both the links and the student and offers links that are likely to be the most relevant to the student at a particular time. This requires facilities to evaluate the student and the links. In order to achieve this, ideas from symbolic AI, connectionist AI (neural networks) and fuzzy logic are employed. It is argued within the thesis that neural networks and fuzzy logic are useful in dealing with the problem of extracting information about the student, since recognising student behaviour is essentially recognising non-discrete (fuzzy) patterns in a potentially noisy environment. The research, then, offers the following new contributions to the intelligent tutoring and hypermedia paradigms:

[DJM1]

The first contribution to knowledge of the research is an investigation into the structuring of hypermedia systems to aid both the student and the author. Specifically this involves:

- The examination of a method for organising and structuring a hypermedia domain into a semantic network-like structure.
- A method for using the semantic network-like structure of the domain to provide automatic weighted links.
- A dynamic linking system to tailor links to individual students (via ordering and highlighting).

Research has clearly indicated the need for facilities to aid the users of hypermedia systems to combat the endemic problems associated with navigating a possibly complex structure, in particular the problem of being "lost in hyperspace" (Conklin 1987, Jonassen and Grabinger 1990, Brusilovsky 1996, Theng and Thimbleby 1998). It is argued within this thesis that the addition of the above methods contributes to the solution of these problems. Ensuring that the domain is structured in a logical and hierarchical manner (should the domain allow) aids both the student and the author, in terms of finding links to add (the author) and to navigate (the student). Such an organisation also allows the production of automatic links, by the utilisation of symbolic AI reasoning algorithms. These algorithms determine the relationship between two nodes in the hypermedia by examining the indirect route through the semantic network that connects them (Pereira et al 1991, Tudhope and Taylor 1997). The production of automatic links removes the burden from the author of providing links that are not directly related to the current node, but still may be of use to the students. Links automatically produced are assigned a weight, or relevance factor, thus aiding the student in knowing how relevant a particular link is. This information may also be used to reduce the cognitive loading imposed upon a novice student, by reducing the number of links presented to them, by removing the least relevant links.

Linking may be further enhanced by the addition of dynamic links. These are links that are tailored to an individual student, in terms of their current interests, as determined by the student overlay model.

The use of the semantic network-like structure provides distinct advantages for both the student and the author. The author is potentially aided since the requirement to structure the domain into a logical hierarchy before they begin linking nodes to the hypermedia, helps them to decide how to link nodes to each other. This combats the often-reported problem of "linkitis" (Conklin 1987), whereby the author is unsure how to link a node and hence links it to many nodes with possibly tenuous relationships. The student is aided by being provided with a logically organised, hierarchical structure, which may provide important navigational cues.

[DJM2]

The second contribution to knowledge is concerned with the extraction of information about browsers of hypermedia systems as they use the system, in terms of the browsing patterns that they make whilst navigating the domain (Canter et al 1985). The facilities described above provide an environment whereby this information may be extracted.

Currently, research has identified that users of information structures, such as databases, hypermedia and the World Wide Web (WWW), use characteristic movement patterns (Canter et al 1985, McAleese 1993, Micarelli and Sciarrone 1998). These patterns change in accordance with the type of user and the task that they are attempting to complete. It is reasonable to suggest that the ability to identify this information is of potential use, especially in situations where it may not be possible to identify information in other ways, such as the WWW (Armstrong et al 1995). Once information linking particular browsing patterns to types of user and the task that they are involved in (searching for a particular item of information, browsing generally etc.) is available then it may be incorporated within an "Intelligent Agent" (Armstrong et al 1995). Such an agent may be able to aid a user as they browse the information resource, in terms of helping them to choose a link. Currently, intelligent agents use the frequency of keywords that appear in documents that the user examines (Armstrong et al 1995). This, therefore, provides a content-based evaluation of the user's interests and potential requirements. Browsing patterns might also provide information about the user in terms of a particular task that they are involved in and, potentially, can indicate their degree of expertise. This information may be used by an intelligent agent to more accurately retrieve information. For example, an intelligent agent currently retrieves information based upon keywords. However, it may be that the user only expressed a superficial interest in such keywords; however the agent makes the selection of the keyword more likely by finding items containing the keyword. The agent is therefore not useful to a user who is interested in browsing more generally. If the student’s desire to browse more general information could be obtained beforehand then the agent could be tailored to offer such material.

This research project takes steps towards the goal of obtaining this browsing information, in that it provides the facility to extract it from a set of active browsers. This facility has been verified using simulated browsing data, which demonstrates the practicality of the approach. A further follow-up project would be to use this facility with real browsing data.

The current research project therefore offers a method of evaluating students using non-domain specific metrics, namely by examining their movements through the hypermedia and their interactions with tutorials. Specifically this involves:

- An adaptable neural network for examining browsing patterns.
- An adaptable neural network for grading a student according to their ability to complete tutorials.
- A fuzzy rule-base for linking student ability and tasks to browsing patterns.

The above facilities are adaptable in that they are able to extract browsing information with minimal input from the human researcher and they are robust in that they are able to cope with the potentially non-discrete information that is being collected from the student. This research project has examined the use of neural networks and fuzzy logic as a means of providing this adaptability that would not usually be available.

An approach to educational hypermedia, called "Hypernet" has been developed in order to examine the facilities described above. Hypernet advocates a structured hypermedia system, in that the domain is expressed as a set of nodes and typed links, similar to a semantic network. The educational element is provided by tutorial nodes, which require that the student interact with the domain. A student's performance with the tutorial nodes is then measured by a neural network system termed the "Tutorial Supervisor". A neural network was seen as appropriate since the grading of many students over many domains is a potentially dynamic and complex process, a neural network is able to adapt to dynamic environments and is therefore able to maintain itself without further intervention. Once the student has been graded links can be tailored to their ability level and interests.

A second neural network is used to identify the browsing patterns that the student makes as they navigate the hypermedia. Again, a neural network is employed because browsing patterns are non-discrete in terms of the start and end points and their complexity. Once browsing patterns have been identified then general trends of the types of browsing patterns that are used for particular types of task are extracted and recorded in a fuzzy logic structure called a fuzzy rule-base (FRB). Fuzzy logic is used here instead of a neural network since the information stored within a neural network is not readable, whereas the information inside a fuzzy rule-base is readable. The upshot of this is that information learned by the FRB is interpretable and may be used elsewhere. The FRB is used to map general trends concerning types of students and the tasks that they are using the hypermedia for, to the browsing patterns that the students make whilst using the hypermedia. Small-scale studies conducted as part of the research project have demonstrated that different levels of student browse hypermedia differently and that different tasks require different movement patterns. It may therefore be possible to extract information about a particular student by examining their browsing behaviour, a potentially useful tool for aiding browsers of non-guided systems, such as the World Wide Web.

[DJM3]

In sum, this research project offers new ideas concerning a hypermedia based tutoring paradigm, which includes the extraction of a student’s interests (concerning the domain) and then modifying the presentation of links in order to reflect these interests.

These ideas are centred upon the requirement to aid the student as much as possible whilst still being viable for an author to produce.

## Thesis Overview

The thesis introduces the area of concern and then progresses to discuss the proposed navigational aids and student model. The bulk of the thesis is concerned with the discussion of the sub-components of the student model, after the detailed discussion of these sub-components the thesis closes with a discussion of hypermedia authoring.

Chapter 2 provides the background research into the areas of intelligent tutoring and hypermedia. This chapter argues the need for navigational aids for hypermedia systems if they are to be used as useful tutoring systems and details research in the area of hypermedia navigation.

Chapter 3, The Hypermedia System, presents a high-level view of a prototype tutoring system, Hypernet, used to demonstrate and motivate discussion of the areas of navigational aid. Automatic and dynamic linking is discussed with particular reference to how they relate to the semantic network definition of the domain.

Chapter 4, The Student Model, introduces methods and prototype systems that help to provide the student with an environment that is adaptable to their interests at a particular instant in time. This chapter argues for the use of domain-independent metrics so that there is a greater likelihood of the hypermedia tutoring paradigm being useful for many domains. The student model records information about the types of nodes that the student has visited, in order to present links to similar nodes. This chapter argues for the use of this domain independent metric and introduces the sub-components of the student model, namely the Browsing Pattern Recogniser, the Fuzzy Rule Base and The Tutorial Supervisor. These components are dealt with in depth in subsequent chapters, although the mechanism by which they interact with one another is described in this chapter.

Chapter 5, Hypermedia Browsing Strategies, discusses the patterns made by users of information structures, such as hypermedia. Typical patterns as identified by previous research are discussed and how these might be recognised by an automated system and put to use in aiding the student/user of such systems. This chapter provides background information for the Browsing Pattern Recogniser (Chapter 6), which is used to recognise browsing strategies and the Fuzzy Rule Base (chapter 7), which utilises the browsing patterns and derives information relating to the student's ability level and the tasks that they are undertaking. Small-scale browsing experiments are described and discussed. These experiments go some way to demonstrate that different types of student use different browsing strategies for various tasks.

Chapter 6, The Browsing Pattern Recogniser, is concerned with the identification of the patterns described in the previous chapter. Browsing patterns are identified by the Browsing Pattern Recogniser (BPR), a neural network that is capable of recognising browsing patterns that have not previously been discovered by research. It achieves this by combining the low-level browsing constructs that have been identified (Canter et al 1985) into more complex fuzzy browsing. Results concerning the effectiveness of the neural network are detailed.

Chapter 7, The Fuzzy Rule Base, is concerned with the fuzzy-logic processor designed for the project. The FRB is useful in that it can potentially provide information linking browsing patterns (as identified by the BPR) to types of task (provided by the tutoring system) and the types of student that use them (in terms of their ability and interests). A neural network is a closed system and is not able to offer this facility, whereas the FRB is an open system and can be internally examined. Experimental results concerning the FRB are presented.

Chapter 8, The Tutorial Supervisor, concerns the sub-system for grading the student into an ability level according to their interactions with tutorials. This system provides two services, firstly it is used to give an ability rating of the student, when they periodically undergo a tutorial. Secondly, it is used to update the FRB in order to provide information regarding the student's ability level based upon the student's movements through the hypermedia (provided by the BPR). The Tutorial Supervisor is shown to be adaptable in that it can adapt to domains where student grades for particular ability levels are different. Justification for the employment of a neural network in general and, more specifically, a particular type of neural network is given. The results for various experimental neural network architectures are presented.

Chapter 9, Domain Design, is concerned with authoring hypermedia domains for both hypermedia tutoring system and for hypermedia systems in general. This discussion is intended for authors of any hypermedia system, since it has previously been argued that knowledge representation structures should be a fundamental part of any such system (Pereira et al 1991). The implications of imposing knowledge structures on domains that do not necessarily have a well-defined structure are discussed. A "bottom-up" approach to authoring is described, whereby rather than the author defining the structure of the domain, it is allowed to evolve through interaction with a large number of students.

Chapter 10, Conclusions and Further Work. Further work is detailed, notably the need for further experimental trials with students and authors and the possible utilisation of the intelligent facilities over the World Wide Web.

Page: 3

[DJM1]FIRST CONTRIBUTION

Page: 5

[DJM2]SECOND CONTRIBUTION

Page: 8

[DJM3]END CONTRIBUTIONS

# Chapter 2

# Intelligent Tutoring

## 2.1 Introduction

This chapter is concerned with the difficulties associated with traditional Intelligent Tutoring Systems (ITS) and suggests that a possible alternative to such systems would be to employ a hypermedia system with the addition of AI facilities. The chapter outlines various forms of computer based learning, including ITSs and hypermedia. Deficiencies with both paradigms are suggested. Novel approaches to computer-based tutoring, using neural networks, are discussed. The chapter is intended to argue the case for the addition of AI facilities to hypermedia to render it a suitable teaching medium; the following chapter is concerned with the design of these facilities.

## 2.2 An Introduction to Intelligent Tutoring

ITSs use AI techniques to make decisions about the state of knowledge of a student user. Major (1995) states that an ITS should adapt to a particular student by varying "difficulty of material, presentation style, help offered, the path taken by the student, [and] generality of the material". He further states that the path through the courseware is calculated whilst the student is using the system and is not pre-programmed beforehand. The general paradigm chosen to overcome these problems has been that of Symbolic Artificial Intelligence (SAI) (Woods and Warren 1995).

[DJM1]Typically however an ITS will constrain the student to learn by a predetermined method or strategy (Ridgeway 1989, Kinshuk and Patel 1997). ITSs use a model of the student's knowledge (student model) so that the student is presented with new information only when they require it, to reinforce a point or to progress in the learning and to identify misconceptions and mal-rules (Sleeman and Brown 1982). Such systems have been criticised for constraining the student to solve a problem in a particular way (Ridgeway 1989). In most complex problem domains, there can be many methods to achieve a correct solution - some people may find one particular method suits their way of thinking better than others. It has been argued that students should be able to experiment with their own ideas and find methods that naturally suit them (Ridgeway 1989). However, some of these methods may not be diagnosed by an ITS (Chiu, Norcio and Petrucci 1991, Woods and Warren 1995).

[DJM2]Elsom-Cook (1989) reviews some Computer-Based Learning (CBL) and ITS packages and grades them between two extremes; total constraint and totally unconstrained (most systems were found to be near the total constraint side). He argues that ITSs lie at the constraint end of this spectrum, whereas an electronic textbook type system would be unconstrained. He argues that the perfect tutoring system should be able to "slide" between these two extremes according to the student's needs and state of knowledge. Therefore a system could appear as a traditional ITS to a novice student or a discovery learning, hypermedia system to an advanced student. (Hartley 1993) reinforces this point by stating that when there is a mismatch between strategy (of the learning system) and learning style (of the student) then performance is degraded. He states that support for different styles and viewpoints of users are required. Further, research has shown that learning is improved when a student is allowed to follow pathways of their own choice, at their own pace and able to monitor their progress by instant feedback to questions (Kibby and Mayes 1990). These ideas are similar to the idea of "cognitive scaffolding" (Henze et al 1999, Robotham

1995), whereby a learner is supported in their learning activity when they need support, but the support is reduced as they become more able, and hence are empowered to undergo learning in a style that suits the learner.

A further problem with ITSs is that most tutoring packages to date are applicable only to the domain for which they were specifically produced. A reason for this is that most are of the constrained type, which lend themselves less readily to the implementation of a number of domains. The outcome of this is that every time a tutoring package is required for a new domain, it is usually necessary to rewrite large parts, or all, of the system. Kinshuk and Patel (1997) note that an ITS is invariably an adaptation of an expert system. They then state that “an expert system is time consuming to produce, limited in its expertise and inflexible whereas the hypertext paradigm offers faster development, easy modification and frequent updates”, although to use Elsom-Cook's metaphor, ITS and hypertext/hypermedia exist on opposite ends of the educational media spectrum and therefore may have very different objectives. Kinshuk and Patel (1997) further suggest that there are no ITSs in existence at the time of writing that allow the development of an ITS without “starting from scratch”. It could be argued that this is not necessarily the case, since ITSs are often adapted from one domain to another. However, this is usually between closely related domains, such as various areas of medicine (Rich and Knight 1991). Kinshuk and Patel note that there is therefore a need for an ITS that can be adapted to completely different domains, and can be engineered by experts in the domains themselves and not necessarily by experts in ITSs. Chiu, Norcio and Petrucci (1991) reinforce this point by noting that the development of symbolic AI systems is a time consuming process and other paradigms, such as connectionist technologies (neural networks), can on many occasions provide a quicker alternative.

Bloom (1995) identifies four problems with the Intelligent Tutoring paradigm; firstly ITS authoring is complex, requiring specialist domain authors. Secondly, his research suggests that the “real world communities” (the end users) seem unwilling to accept the ITS paradigm, mainly because they do not understand the technology and nothing is offered to the human teacher. Thirdly, there is very little reuse of ITS architectures across applications. This argument is in line with the criticisms of Kinshuk and Patel (1997) cited above, in that a new ITS is generally developed for each new domain of application. Bloom points out that in order for an ITS to be termed “generic” it must have the ability to reuse its student model, instructional model and knowledge base inferencing mechanisms. Bloom’s final concern is that most ITSs require specialist delivery platforms, i.e. they may not suit the systems that the end-user may already have, although this is less likely to be the case today as computer systems are more powerful and cheaper. Bloom also notes that the development time for an ITS constitutes “an amount of time unacceptable under any circumstances other than a research project”. This is why, generally, ITSs have failed to migrate from the research laboratory.

Given these problems with ITSs hypermedia was therefore examined as an alternative approach to adaptive tutoring.

## 2.3 An Introduction to Hypermedia

Hypermedia involves the presentation of information to a user in a non-sequential way. A user typically has the option to "travel" to many separate destinations from the current information point (node).

Conklin (1987) describes several types of hypermedia architecture. The most common utilises “referential” links, which link together two nodes in a non-hierarchical way, resulting in a largely unstructured domain. Referential links are commonly associated with selectable words within a document or as "hot spots" within a graphic. A second type employs “organisational” links that connect a node to its parent nodes or child nodes, these are traversed via a separate mechanism from the node itself, either via a graphical browser or via a list of available links. Conklin (1987) also describes a third type of link that he terms “keyword” links (dynamic links). Such links use the processing power of the computer to search the documents in the hypermedia for relevant keywords, selected from the current node. Conklin points out that “this type of linking is yet to be fully explored”, a statement which is largely still true (Theng and Thimbleby 1998).

Whether or not it is better to employ non-structured hypermedia (employing largely referential links), hierarchical hypermedia (organisational links) or a mixture of both is a subject of debate. The proponents of unstructured hypermedia argue that it is its unstructured nature that provides the richness and freedom associated with hypermedia. However, navigating in such structures exacerbates the oft-reported problem of cognitive loading (Lowe and Hall 1999). A hierarchical structure is largely dependent on the ability of the defined hierarchy to match the requirements of the user. There are many ways to organise such structures from various points of view and if the point of view of the system is different from that of the user then the system's usefulness is reduced for that user (Conklin 1987, Hartley 1993). Halaz et al (1987) use both types of links in their system NoteCards, although they point out that the most common links are organisational links. However, the domains they employed within their system were invariably computer science domains and Conklin (1987) notes that engineering/science orientated users prefer hierarchical structures whereas arts/humanities orientated users prefer the cross-reference style of the referential links.

Other hypermedia systems such as Microcosm (Lowe and Hall 1999) and Hyper-G (Maurer 1996, Lowe and Hall 1999) also offer structural and referential link types. However, a notable exception that does not offer different link types is the World Wide Web (WWW).

### 2.3.1 The Structure of Hypermedia

Although many hypermedia systems today do use different types of links that attempt to represent structure, the most common hypermedia systems, such as the WWW and help facilities, do not (Lowe and Hall 1999). In these systems the links provided by the author offer no explicit meaning, instead the two nodes each link connects usually imply the link’s meaning. However, Pereira et al (1991) maintain that a hypermedia system should contain three significant components, a text (and other media) database, a semantic network which relates the database components and tools that allow the user to explore the database and semantic network. It is the semantic network that is often omitted from hypermedia systems. Pereira et al (1991) suggest that the authors of hypermedia domains should generate only a few strongly related links instead of linking to anything the author thinks may be related. Thus the user can navigate using this “pseudo hierarchical structure” which they term “cognitive scaffolding”.

Jonassen and Grabinger (1990) suggest that a hypermedia system is a network of ideas. The structure linking the ideas together helps the user comprehend the node content better. Such a structure is seen as suitable for learning, due to the similarity between the associative structure of hypermedia and the associative structures held in the human brain (Jonassen and Wang 1993). “The belief that hypertext can mimic human associative networks implies that an appropriate method for structuring hypertext is to mirror the semantic network of an experienced or knowledgeable performer or expert.” (Jonassen and Wang 1993).

Jonassen and Wang (1993) further argue that such knowledge structures are transferable from the hypermedia to the learner; however, they also found that simply browsing through a hypermedia system was not enough to produce any appreciable transferral of structural knowledge. Instead, it is necessary to provide exercises that explicitly test the learner’s structural knowledge of the domain.

Information can be considered to be stored in the brain in an associative manner (Jonassen and Wang 1993); thus information is linked together to form a network. Such a structure is referred to as a "schema", each one having a number of attributes that enable it to be linked to other knowledge structures. The process of learning is the acquisition of such information schema and/or arranging them into the learner’s knowledge structures. Three processes allegedly govern learning, i) accretion ii) restructuring and iii) tuning (Jonassen and Wang 1993).

"Accretion" is the process of adding additional information to existing schemas, i.e. new content is added that links it to other schemas. "Restructuring" occurs when a schema has been expanded to a point when it is no longer viable. Additional schemas are then created from the existing schema and the new knowledge that allows the learner to access and interpret the existing knowledge in new ways. "Tuning" is the process of making small adjustments to the schema whilst they are being used, i.e. the learner becomes better at using the knowledge with practice.

A useful hypermedia structure would therefore be a representation of the knowledge schema of an expert. The learner attempts to map these structures onto their own by using the hypermedia. It is this transferral of structure from hypermedia to learner which represents the challenge to hypermedia designers, since without an adequate structure the learner will have insufficient cues with which to incorporate the information into their own knowledge structures.

“The less structured the hypermedia is, the less likely users are to integrate what they have learned into their own knowledge structures, because the hypermedia facilitates only the acquisition part of learning” (Jonassen and Grabinger 1990). The above discussion suggests therefore that structural information is paramount for an educational system. Hypermedia has been described as a suitable medium for representing structural information (although not all hypermedia systems do) (Jonassen and Wang 1993). Means of doing so are discussed below.

Currently, there are a growing number of systems being produced to organise information on a computer system in a structured hypermedia fashion (Tunbridge 1999). These systems organise and display information and files stored on a computer as a meaningful hierarchy and not as a potentially meaningless list of files and directories. Two such systems "Hyperbolic" and "Semiomap" (demonstrations are listed in the references under Tunbridge (1999)), use AI techniques to extract information stored within each file so that it may be related to other files in a hierarchical manner. Information is then presented to the user in a structured hypermedia fashion by means of a graphical browser interface. If these, or similar systems, replace existing browsers provided with popular computer operating systems then the structured hypermedia paradigm is likely to become familiar to the computer using population and hence the cognitive overhead associated with using hypermedia is likely to be reduced.

The purpose of the approach to an educational hypermedia system discussed in this research project is to bring these formalised methods for structuring hypermedia together, to aid both the student and the author.

The problem of authoring a domain, for a hypermedia or learning system, is a complex one. Ideally, as Jonassen and Grabinger (1990) point out, "learning should be facilitated by hypermedia that replicates the expert's knowledge structure in the structure of the hypermedia." Some domains are likely to be more suitable to this approach than others. For example, anatomy and astronomy are highly structured; therefore an expert in the field may readily be able to elucidate the structure. An expert of psychology, where the domain may not be as structured and hierarchical, may have more difficulty in elucidating such a structure.

The following section discusses Hypermedia systems for learning, which are set apart from those that are used for more common applications, such as on-line documentation, by the constraint and/or guidance they apply to the learner. The problem is how much guidance/constraint to apply.

### 2.3.2 Hypermedia in Education

Hypermedia has been cited as being of potential benefit for use as an educational system. Tait (1998) suggests that according to constructivist principles a problem should be authentic and meaningful to students and that “hypermedia can provide an optimal platform for the design of problems and their enrichment with authentic pictures, video-materiel, animations or simulations.” However, Jonassen and Grabinger (1990) indicate that the potential for using hypermedia in educational systems is largely unrealised, although they have achieved a high degree of success when applied to on-line help systems and documentation. Jonassen and Grabinger (1990) further state that a hypermedia learning system will shift the responsibility for accessing and sequencing information from the teacher to the student, which will entail a cognitive overhead.

Hypermedia provides information about a given domain in two ways. Firstly, the content of each node provides a chunk of information about a discrete topic. Secondly, the links that give the hypermedia its structure provide additional information about how the current node fits in with the rest of the domain. Each node of information is not stored in isolation, but is linked to other nodes that may provide examples and illustrations, which help to place the current node in context.

Hypermedia has not achieved widespread success in the field of tutoring systems (Brusilovsky 1996). This is because the hypermedia paradigm is plagued by several problems, outlined below. It is the contention of this project that these problems can be eased with the addition of AI techniques, such as the use of a student model to record information about a student's ability and interests.

### 2.3.3 Problems with Hypermedia

Conklin (1987) and Theng and Thimbleby (1998) refer to two types of problems associated with hypermedia. The first are problems of implementation, such as slow response times when a user selects a link, restrictions on naming links and deficiencies with browsers. These are problems that can be solved by better machines and software designs. "Endemic" problems, however, are not likely to be solved by improvements in computer hardware or by software design, Conklin defines the problems, “Getting lost in space” and  “the cognitive task scheduling problem”. Both terms have later been redefined by common consensus as “lost in hyperspace” and “cognitive overhead”.

The problem of Lost in Hyperspace arises from the complexity associated with “having to know where you are in the network and how to get to some other place that you know (or think) exists in the network” (Conklin 1987). Such disorientation problems exist in other media also, but if a reader becomes lost in a linear text document then all that they can do is look further into the text, earlier in the text or jump to the index. Hypermedia however offers far more degrees of freedom and hence the potential for the reader to become lost is far higher (Conklin 1987). Graphical browser systems offer some visual cues, so that readers can orientate themselves by recognising “land marks”. However, Conklin (1987) states that there is no natural topology for hypermedia; hence a reader may have to learn the topology of the hypermedia. Another possible aid to the problem is the use of filters so that the reader is presented with a manageable level of detail, by removing nodes that are not likely to be of interest (Brusilovsky 1996). This method is discussed further in the following chapter.

"The number of learning options available to learners places increased cognitive demands upon the learners that they are often unable to fulfil." (Jonassen and Grabinger 1990). What use then is a system that removes some of a learner's ability to actually learn? Jonassen and Grabinger (1990) wonder if the "information rich" will get richer and the "information poor" will get poorer. Much of the cognitive overhead associated with hypermedia is endemic to other computer systems also (Preece et al 1995). The user of any new software must become familiar with is novelties and intricacies before they can devote their entire attention to completing the task for which the software was designed. This type of cognitive overhead is relieved to a degree by standardised user interfaces and by the general populous becoming more familiar with computer systems as a whole (Preece et al 1995). There is however no standardised hypermedia design for education. Therefore, the user of a hypermedia system is likely to have to learn how to use the hypermedia interface and learn how to use hypermedia generally. However, it may be that these two types of cognitive loading are likely to become less obvious and important as computer interfaces become further standardised and as hypermedia becomes more common in the every day computing environment.

Many researchers have proposed the use of default pathways within the hypermedia in order to support a user who is unable to decide where to go (Trigg 1987, Halaz 1987, Conklin 1987, Beynon-Davies et al 1994). The theory is that a novice user is able to follow a predefined path, but is free to break from it whenever they wish. Such an approach has been criticised, most notably by Jonassen and Grabinger (1990) for providing the unambitious, uninterested or lost learner with a path that they can blindly follow, thus ignoring the wealth of information elsewhere in the hypermedia database.

The original idea for default paths, as indeed the original idea for hypermedia, is often attributed to Bush (1947). However, Bush describes a system where a user builds up their own pathways as they discover relationships between concepts. The guided tours proposed by Trigg (1987), on the other hand, are provided beforehand by the domain author. They therefore may not have any relationship to the student's current learning needs and therefore seem wholly inappropriate for an educational system (Jonassen and Grabinger 1990), although there may be a case for them in non-educational systems.

Cognitive overhead is also induced by the complexity of the domain itself. As a student uses a hypermedia system, they must use cues from the domain in order to orientate themselves and to navigate further into the domain. Hence the cognitive loading problem is exacerbated when the hypermedia is being employed as a tutoring medium, since the student is not just navigating in order to find a particular piece of information but is actively attempting to learn the contents of parts of the domain. However, as noted by Jonassen and Wang (1993), the structure of the hypermedia can be made a key part in the learning activity, as it may provide context and examples. This structural information may therefore provide the learner with a "cognitive scaffolding" (Jonassen and Wang 1993) onto which the learner may "attach" their newly leaned material.

It should be noted however, that Jonassen and Grabinger are explicitly referring to using hypermedia for learning systems and not for a more general browsing system, such as may be found in a museum, when the choice may be between a user not using the system at all, or being allowed to follow a guided tour. For learning systems, the user is not always someone who wants to use the system for purely informative or recreational purposes. It seems reasonable that the student is provided with tutorial tasks to complete whilst using the hypermedia. Hence, for the purposes of this research project, it is considered paramount that the student is encouraged to move through the hypermedia structure in their own way and are not provided with predetermined paths to blindly follow. In order to achieve this the student is provided with tasks to encourage them to browse particular elements of the hypermedia.

## 2.4 Intelligent Hypermedia

The term "Intelligent Hypermedia" refers to the addition of AI to the hypermedia paradigm. This usually involves the utilisation of AI techniques to create dynamic links, i.e. links that may change for each subsequent visit to a node. Dynamic links usually involve the tailoring of individual links to the person who wishes to use it and this inevitably requires the addition of a student/user model (Brusilovsky 1996).

A central argument of this research project is that the addition of such AI facilities to hypermedia will combat the classic problems of hypermedia, most notably the difficulty in navigation and the cognitive overheads associated with navigating hypermedia (Theng and Thimbleby 1998, Conklin 1987).

An important aspect of the AI facilities discussed in this research project is the use of a semantic network, which enables dynamic linking facilities to be provided. Strath Tutor, developed by Kibby and Mayes (1989,1990), takes a similar approach and employs dynamic links that are generated as the student uses the system. They note however, that the process is extremely time-consuming, invoking delays in response times from the system in the order of tens of seconds for their restricted domain. They also note that the addition of extra nodes to the domain increases the computation time exponentially. It is for these reasons that this research seeks to find an alternative to this system of dynamic linking.

Boyle and Snell (1992) employ an alternative approach, in their SINS system; this model employs a frame model and production rules for each node in the hypermedia. Each rule has a weight associated with it so that the firing of simultaneous rules usually results in one "winner" or several close alternatives, a threshold can be set by the user so that they are presented only with nodes that meet their minimum criteria. A ripple search technique is employed to ensure that the search proceeds to the next most related nodes, according to the semantic network. Unlike Strath Tutor, the SINS system does not continually generate dynamic links as the user moves from node to node. Instead, the user is able to navigate the semantic network without inducing processor load, should they then require additional help they can ask the system to perform its intelligent search, in which case the computationally expensive dynamic links are induced. It could be argued, however, that this system is merely postponing the processor overhead until the student requests it.

Pereira et al (1991) include a semantic network in their definition of hypermedia. They argue that it is paramount that a semantic network or similar is employed for an intelligent hypermedia system. Indeed, a semantic network diagram resembles a diagram of a hypermedia domain, with nodes connected by links. Links in a semantic network are always typed, indicating the relationship between the two nodes that are connected. This is not always the case with hypermedia: most commonly the link has no information associated with it. Similarly, the nodes in a semantic network are used to represent concepts and they contain no information other than their label. Hypermedia nodes by contrast contain the information that the student requires. However, the application of the semantic network paradigm to hypermedia is a natural one in that the “benefit of semantic networks is that they are natural to use, since related concepts tend to cluster together in the network. Similarly, an incompletely or inconsistently defined concept is easy to spot since a meaningful context is provided by those neighbouring concepts to which it is already linked.” (Pereira et al 1991).

Beynon-Davies et al (1994) utilise a form of semantic network in that they use binary relations to define a domain knowledge base, and point out that the knowledge base makes authoring the hypermedia simpler. For example, they define a canal as a waterway which has a start and finish using two binary relations (canal has-a start; canal has-a finish), an author who then wishes to add the Pontypridd canal, for example, to the domain knows that they must define the start and end locations for the canal. The upshot of this is that the author is aided in the authoring process, since the system can explicitly request information from the author.

Tudhope and Taylor (1997) go a step further with their semantic network in that they give specific weighting to link types. The weighting is used in the computation of automatic links in order to give a priority to certain link types. In the case of Tudhope and Taylor’s (1997) system a priority is given to links representing lower-level detail than those representing higher-level detail, since the lower the level of detail the more relevant the information is to the current node, in their system at least. The user can further tailor the links by adjusting thresholds for certain types of links.

In the light of these arguments the educational hypermedia approach developed for this research project utilises a semantic network within the hypermedia. The author is encouraged to formulate a semantic network for the domain, as a web of class nodes. This semantic network forms the basis for the hypermedia domain, in that instance nodes of the classes in the semantic network are added to the domain using the same typed links found in the semantic network. The authoring process is discussed further in chapter 9 when the rest of the hypermedia approach taken for this research project has been discussed.

### 2.4.1 Dynamic Links

Dynamic Links are links that are created at run time according to some stimulus from the user or are modified in some way so as to model the user's current interests. The aim of dynamic linking is to alleviate the "lost in hyperspace" and "cognitive overhead" problems. This is achieved by filtering out some of the links, in order not to present the user with too much information. Also it assists authors, in that it is not necessary for them to think beforehand about every single useful link that should be included in the domain. It is to dynamic links that Conklin (1987) was referring to when he suggested the addition of AI to the hypermedia paradigm.

Similarly, Hammwohner and Kuhlen (1994) argue “the selection of links may be supported by the filtering processes, thus excluding link types from the presentation which - with respect to some user profile - are known to be of no interest to the reader.” Hammwohner and Kuhlen (1994) describe two types of links, extensional and intentional. Extensional links are those links that are defined explicitly by the author.

Since it is virtually impossible to exhaust all the possibilities for creating links and because the process of defining links is a time consuming one, intentional links must be employed to supplement the extensional ones. Intentional links are automated links that are specified as computable rules. Such a process can exhaustively generate all linking possibilities. Further, when links are generated using rules it is much easier to filter out some according to an appropriate model of the user. The system described in Hammwohner and Kuhlen (1994) utilises intentional links if they can be computed in reasonable time, if they cannot then they are computed at the next update of the hypermedia domain and stored permanently. The approach discussed in the following chapter uses a similar system, whereby automatic links are generated off-line. These links may then be further tailored, according to the student model, at run time, in order to provide more useful links for the student. This has the advantage of significantly reducing the processor overhead as the student uses the system. Reducing processor overhead is as paramount today as it has been previously since the increased processor power being offered now and in the future is likely to be consumed by more complex operating systems and media requirements (Denning 1999).[DJM3]

The use of semantic networks also provides a mechanism for producing and organising intentional links in order of relevance. Carbonell (1970) describes relevancy, in terms of a semantic network, as the distance between two nodes. The distance is measured in terms of the number of intervening nodes between the two nodes in question. Hence, a relationship between two nodes that are connected by five intermediate nodes is stronger than that of the relationship between two nodes connected by ten intermediate nodes.

### 2.4.2 Adaptive Hypermedia

Adaptive Hypermedia Systems (AHS) attempt to remedy the problems described above by altering the presentation of material to suit an individual user or student (Brusilovsky 1996). This can be achieved in a number of ways for a number of purposes. There are two elements of a hypermedia system that may be altered, namely the links and the material within the nodes. Brusilovsky (1996) describes these as "link-level adaptation" (termed dynamic links by other researchers) and "content-level adaptation", which refers to altering the contents of the nodes themselves. Link-level adaptation allows different links to be presented to different types of user. This is achieved by guidance, sorting, hiding and annotation. Most AHSs require a model of the domain and/or the user, in order to be able to alter the links towards the student's interests. Individual elements are discussed below.

#### 2.4.2.1 Link Adaptation

"Guidance" is the simplest form of link support and has been employed by Trigg (1987) for supplying paths for the user to follow. "Sorting" is used to order the links so that the most relevant appears at the top of the list. This requires the addition of a system to determine the relevance of individual links, either in terms of the user, in which case a user model is required, in terms of the domain, in which case individual links must specify a relevance, or both. Sorting has been criticised since it has the potential to change the links at a node for subsequent visits, thereby potentially confusing the user. This is exacerbated when the user is not used to the system or domain (a novice) (Debevec, Rajko and Donlagic 1994). However, this criticism can be raised at all adaptation methods if steps are not taken to ensure that the user is made aware that the node is the same and can visit the same nodes that they could before. Research has demonstrated the benefit of ordering in AHS (Armstrong et al 1995, Mathe and Chen 1994). The linking approach discussed in the following chapter uses sorting to order links so that the most relevant links, in terms of the current student's interests, are towards the top of the list. This has the advantage that the novice student can be aided in their link selection, since they are presented in relevance order. This is possible because of the application of the semantic network-like structure of the domain, this is discussed further in the following chapter but is provided here as an aid to the discussion. In order not to fall foul of the criticisms of Debevec, Rajko and Donlagic (1994), the links are presented in several boxes denoting different types of links (authored or extensional links, structural links and related material). The probability of the node appearing radically different is therefore greatly reduced. This is discussed further in the following chapter.

"Hiding" is an approach used to remove certain types or a certain number of links to non-relevant nodes (Brusilovsky 1996, Hook et al 1996). Removing non-relevant links reduces cognitive loading by reducing the navigational possibilities available to the user and since only non-relevant links have been removed this is seen as desirable. However, hiding has been criticised for causing the user to form incorrect mental models of the domain (Brusilovsky 1996). One proposed remedy to this is to use "greying" (Brusilovsky 1996), whereby links are displayed in a lighter colour and are not selectable by the user. Hiding technology also requires facilities to diagnose the ability of the student, in order to determine how much information to hide. The linking approach, discussed in detail in the following chapter, uses hiding since it generates every possible link it is impractical and unsuitable to offer every one to the student. However, since the links have been sorted into relevance order the least relevant can be removed and so the student's cognitive loading is reduced. However, it is important to offer those links that have been offered on a previous visit to a node again, so that the navigational possibilities of a node is not reduced in any way; rather the navigational possibilities branch as a student progresses in levels. Therefore, the approach discussed in the following chapter records the links that have previously been offered to a student so that they can be offered again, regardless of the grading of links. This helps avoid the criticisms of Debevec, Rajko and Donlagic (1994) in that the student does not have links previously offered removed and therefore ensures that the student can return along a path they have previously traversed.

An approach to link adaptation is "Annotation" which augments the link with extra information, which my take various forms, including textual cues, different colours and icons. Link annotation is an effective form of navigation support (Zhao, O'Shea and Fung 1993). The simplest form of annotation is to provide the user with information stating whether they have previously traversed a link or not. Most WWW browsers accomplish this by changing the colour of the link. More advanced link annotation systems offer explanations of the link itself. This may be accomplished by displaying the type of the link, providing the system organises the domain as a semantic network (Pereira et al 1991, Kibby and Mayes 1989, 1990). Systems such as ELM-ART (Schwarz, Brusilovsky and Weber 1996) and ISIS-Tutor (Brusilovsky and Pesin 1994) have a user model to record various levels of detail that the student knows about each node. This information is then annotated to the link so that the students are made aware of their own state of knowledge about a particular node. Brusilovsky (1996) notes that when link greying is used instead of link hiding, this can be then thought of as annotation. Presumably this means that the dimmed links are still selectable by the user but are dimmed in order to indicate to the user that they probably should not be selected. The linking approach discussed in the following chapter uses annotation to show the student the relationship between the current node and the node connected by the link. This relationship is the path through the semantic network. This is useful since it enables the student to make a determination about the relevance of a link, to a degree, without the need to traverse the link and so induce extra cognitive loading.

#### 2.4.2.2 Content Level Adaptation

Content-level adaptation is when the contents of the nodes themselves are altered in order to present the most relevant information to the current user. Systems that employ this approach are predominantly educational hypermedia systems (Brusilovsky 1996). The contents of the nodes may differ if the user is a novice of the domain and therefore requires less detailed information than a more experienced user. Criticisms applying to link hiding (Debevec, Rajko and Donlagic 1994) also apply to this method, since the user may become confused when apparently the same node bears no resemblance to the initial visit. Instead, it may be better to separate the information intended for different levels of users into different nodes, but require that the novice nodes are visited before the expert nodes (Brusilovsky 1996). This introduces a form of dependency, in that one node is dependent upon another and may not be viewed until the former node has been viewed. Such an approach is common in designing educational courseware (Ridgeway 1989), although it should be noted that this is moving away from the traditional concept of hypermedia. However, this approach may lead to added complexity for the author and can potentially confuse the student when they revisit the node and see different information (Debevec, Rajko and Donlagic 1994). Instead, information for different levels of student should be placed in separate nodes, restricting nodes containing more complex information to higher level students by using dependence link types, thus providing a finer grain of hypermedia (McAleese 1990).

### 2.4.2.3 Domain and User Models

In order for a hypermedia system to be adaptive it must have information relating to the structure of the domain and a model of the user (Brusilovsky 1996). The complexity of both these elements can vary enormously. A simple domain model would record which nodes the user has visited and which they have not. A more complex domain would record the detailed structure of the domain. Indeed, Brusilovsky (1996) defines three levels of domain model; level 1 models record independent sets of concepts for each elements of the domain; level 2 models relate these individual concepts together to form a semantic network, thus representing the external structure of the domain; level 3 models provide concepts for the internal structure of the domain, namely the internal structure of nodes themselves; this is analogous to the frame-based model in AI (Rich, Knight 1991). The student model used in this research project and discussed in chapter 4, uses a level 2 model since the domain is represented using a semantic network-like structure. This design decision is based on maintaining a manageable level of complexity for both the author and the student. It would be an arduous task indeed to define every concept within a node as a semantic network to form a level 3 model if the node contains anything more than a few concepts.

The user model is used to represent the user's current state of knowledge. The simplest form of user model is the same as the simplest form of domain model, in that a record of the nodes the user has visited shows what they may know. A more complex form of user model is to record each of the concepts that the user "knows". This is difficult for non-educational systems, since the only practical way to discover whether a user knows a concept is to test them in some way. Therefore, the most complex user models tend to be found in educational AHS, since they belong to an area where it is acceptable (and may indeed be necessary according to Jonassen and Wang (1993)) to directly interrogate the user about their knowledge. Chapter 4 argues for a student-overlay model of this sort to store the types of nodes that the student has visited in terms of their semantic nature. A student-overlay model provides a simple and manageable method for representing information about a user of a hypermedia system (Brusilovsky 1996). It also employs facilities for determining the student's ability and the patterns they make whilst using the hypermedia.

#### 2.4.2.4 Types of Adaptive Hypermedia System

According to Brusilovsky (1996) AHS can be separated into three categories: Page Indexed, Fragment Indexed and Knowledge Based. Page indexed systems connect together nodes (pages) that contain the concepts that are related to the current concept. Page indexing also involves linking nodes together that contain concepts that must be learned in a certain order, thus one node is dependent upon another. ISIS-Tutor (Brusilovsky and Pesin 1994) and ELM-ART (Schwarz, Brusilovsky and Weber 1996) use page indexing. Fragment indexing is similar to page indexing, the difference being that a node is split up into several concepts. This method is often used in systems that present different information to different classes of user. A novice may therefore be shown a different set of concepts from an expert, thus the page itself will appear different. Anatom-Tutor (Beaumont 1994) and KN-AHS (Kobsa, Muller and Nill 1994) use fragment indexing. The Knowledge-based approach organises the whole domain as a set of concepts, each node in the domain representing a concept. This is similar to McAleese's (1990) idea of small granularity in node contents. Thus the hypermedia structure itself represents the domain structure directly. ELM-ART (Schwarz, Brusilovsky and Weber 1996) uses this system (as well as page indexing).

Brusilovsky (1996) argues that none of these approaches are mutually exclusive and may be mixed freely together, as can be seen from his own system ELM-ART. In order to provide as useful a system as possible the approach discussed in this thesis also utilises several of the elements described above. In sum, the educational hypermedia approach described and discussed in detail in subsequent chapters fits into the Adaptive Hypermedia System paradigm, as described in Brusilovsky (1996), in the following way: it adapts link presentation using sorting, hiding and annotation; it models the domain using a level 2 model (a semantic network); the student is modelled using an overlay model. Further, neural network technology is used in the hypermedia approach discussed in this thesis in order to improve the student-overlay model so that it is robust to possible differences in students and domains. The use of neural network in educational systems is now discussed.

## 2.5 Neural Networks in Tutoring Systems

ITSs have been criticised for being too rigid in their approach to an individual learner (Ridgeway 1989, Elsom-Cook 1989, Hartley 1993). For this reason several researchers have examined the possibility of using artificial neural network (connectionist) technology in this area.

Battiti and Serra (1991) use a variation of the back-propagation neural network called the "Bold Driver", in a tutoring system for diagnosing electronic circuits. The student's task is to learn to apply several related equations to solve electronic circuit faults. The neural network’s inputs are a complete list of variables from the complete set of equations, repeated twice. A high input in the first instance indicates the use of that particular variable as a known variable. A high input in the second instance indicates that this is the goal variable, or unknown variable. The output layer of the neural network corresponds to each equation that may be employed in the domain. The result is a neural network that can be trained to examine a given problem, in terms of what is known and what needs to be known, and determine how to proceed at each step, i.e. the network is selecting the relevant equation at each step to solve the problem. The network was trained to operate in two separate ways, the first uses a data-driven approach to solving the problem. This approach is typical of an expert, where an equation is selected that involves as many of the known variables as possible.  The second approach, typical of a novice, involves the repeated selection of an equation with one of the known variables until the final solution is reached. The use of two training patterns for novices and experts provides the neural network with the ability to recognise different behaviours that may produce the same result. The system is therefore able to discern what type of person is using the system, without forcing them down a particular “path of learning”.

It might be argued that symbolic AI could be employed to achieve the same result. The difference between a system comprising a neural network and a system comprising rules arises, however, when a student uses the system. The rule-based system will be sensitive only to those methods previously encoded into it by the author. The connectionist system is given the start and end conditions only when it is trained; it therefore has no methods “hard coded” within it. The neural network may therefore be able to respond to situations where a student has achieved a result by unconventional means.

Chiu, Norcio and Petrucci (1991) have also conducted research concerning the use of neural networks for student modelling. They too note the inadequacy and inefficiency of rule based programs for this task. They criticise the use of expert system technology, stating four fundamental problems. Firstly, they state that the use of expert systems effectively limits the domains for which an expert system is applicable, in a very general sense. This is because a rule base explicitly requires a domain that can be stated as well defined rules which is not always possible. Secondly, the maintenance of an expert system is difficult and costly. Thirdly, information provided by an expert system is often incomplete and subjective and hence fuzzy in nature, coupled with the fact that the student may react in ways not previously envisaged by the system designer, making the system incomplete, ambiguous and uncertain. Finally, an expert system requires extensive processor power, making it impractical for many real time applications. Although this final problem is not as important today as the average personal computer has increased in power many time since 1991, Chiu et al. (1991) argue that these problems give sufficient reason to explore alternative technologies, neural networks being the one they suggest.

Chiu et al (1991) argue that neural networks are a better alternative to a rule-based system because of i) a neural network's pattern recognition ability on imprecise data, ii) a neural network's ability to generalise and learn from specific examples and iii) their speed in execution, which makes them ideal for real time applications. Chiu et al (1991) also argue that categorising students into explicit categorise such as “novice” or “expert” is unsuitable. Instead, a student should be able to belong, to varying degrees, to “stereotype sets”, i.e. a set of predetermined behavioural patterns. A neural network's ability to deal with fuzzy data makes it an ideal choice for such a situation (Kaiser 1990). Bergeron et al (1989) argue on similar grounds for the use of neural networks in their tutoring system.

The tutoring system that Bergeron et al (1989) built operates on a simple but effective pedagogy. A student is offered material at or slightly beyond their current ability, this being seen as the best way to “stretch” the student and thus improve their ability. Such a pedagogy requires two problems to be solved: firstly a student must be graded into an ability range and secondly each question must be graded into an ability range. The neural network is used to grade the student and initially the tutor grades the questions. Thereafter the neural network may modify the question grading in response to how each student ability type deals with it. Questions and students are classified into one of ten grades. A student is offered questions graded at the same level and at the level above. The neural network takes the student’s current ability, the question’s current grading and the student’s success or failure with that question as its input. The output of the neural network is the student’s new ability level. Questions are re-graded off-line, thus correcting any mistakes the tutor may have made with a particular question or a particular group of students.

Bergeron et al's (1989) system is remarkable in several aspects, most notably in its simplicity. The neural network is simple and requires very little processor power, the pedagogy proved to be effective and has the useful side effect of re-grading the questions. Their results showed the system to be extremely powerful. Further, the neural network was totally transparent to the student and the students themselves responded well to the system. Finally, Bergeron et al (1989) remark that their results would not have been achievable by employing a rule based approach, “Compiling thousands of generated rules for each video disk is clearly not feasible, and simply presenting images to students in a rigid, sequential fashion is not acceptable.” They note that their system is a compromise between a full rule based system, with all the difficulties the production of such a system entails and a “simple electronic page flipper”.

Micarelli and Sciarrone (1996, 1998) also employ a neural network to aid a user navigating a hypermedia system. Their system uses Case-Based Reasoning (CBR), which aids a user by attempting to provide them with a path previously identified by the domain author as completing a particular goal. The user’s goals are ascertained by finding the closest match between the user’s partial browsing pattern and the set of browsing pattern cases provided by the domain author. The user may then be offered the nodes missing from their current browsing pattern. Micarelli and Sciarrone use a neural network to perform the mapping of the user’s partial browsing to the stored cases. CBR relies on the domain author identifying browsing paths (cases) beforehand. The neural network then acts as a partial pattern recogniser, in that the user generates a partial pattern or similar pattern that the neural network matches against the pre-defined database of cases. The system can then guide the user over the remainder of the pattern. This system is similar to Trigg's (1987) idea of default paths, except that CBR provides many useful paths and the choice of which path is offered depends upon the user's initial browsing patterns.

Micarelli and Sciarrone's (1996, 1998) work is of particular interest, since this research project also uses browsing patterns to gain information about the student and uses a similar neural network to accomplish this. However, the approach taken as part of the current research does not require that the domain author identify potentially useful patterns; instead this is achieved using a population of students and identifying which patterns produced the best results. The use of neural networks for student modelling is discussed further in Chapter 4, The Student Model.

Student modelling is essentially a classifying task, although traditional approaches go much deeper than classifying a student into a simple category. Such traditional approaches attempt to identify misconceptions and mal-rules (Sleeman and Brown 1982), these approaches are still classifying tasks however, albeit classifying of very specific items. The specificity of such a system, however, is also its downfall in that it makes them expensive to produce and highly specific to the domain in question. Other disadvantages are that it is extremely difficult to provide a system with all the rules defined beforehand and that the more students that use the system the more unexpected behavioural patterns are likely to be encountered.

Neural networks are better classifiers than statistical systems (Gurney 1997), in that they can operate on incomplete and noisy data and can link inputs to outputs in ways that may not have been envisioned by the developer. Neural networks are essentially automatic in the way they are trained, i.e. they are not told what they are doing wrong and how to correct the problem, they are only told that they are wrong and that they should correct the problem. The result of this is that development of a neural network is far quicker than that of a statistical system (Chiu, Norcio and Petrucci 1992), and a neural network provides the system with the ability to continue learning while it is being used.

## 2.6 Summary

This chapter has introduced various computer based learning paradigms, in particular ITS and Hypermedia. It has been argued that the use of AI facilities and neural networks may provide a richer educational hypermedia environment than the standard hypermedia paradigm. The following chapter is concerned with putting this into practice and the details of the argument for and the description of a hybrid symbolic AI and neural network architecture are there presented.

Page: 13

[DJM1]2

Page: 14

[DJM2]3

Page: 30

[DJM3] Note this is very similar to my system. Mine does the bulk of the work off line and then tweaks them in real time.

# Chapter 3

# The Application of Artificial Intelligence to Hypermedia

## 3.1 Introduction

This chapter explains and justifies an approach to educational hypermedia, which has been collectively termed “Hypernet”. The Hypernet approach as a whole was devised to explore issues relating to an educational hypermedia system that goes someway to providing an educational environment that is able to offer some “cognitive scaffolding” (Tait 1998, Robotham 1995) for those learners that require it, whilst allowing those learners that do not the freedom to explore and learn in their own way.

In order to provide such an adaptive environment several additions to the standard hypermedia paradigm are proposed as follows:

- A student model.
- A semantic network-like structure to describe the domain.
- A structured link database.
- A control system for presenting only useful links.

These facilities have been identified by several researchers, most notably Brusilovsky (1996) who suggests several types and complexities of student model for adaptive hypermedia systems (AHA) and Pereira et al (1991) who suggest that a semantic network is paramount to a hypermedia system. The student model utilises non-domain-specific metrics in order to tailor the presentation of the hypermedia domain to reflect the student’s interests and ability. The student model comprises a technique for identifying which particular browsing pattern a student is employing and a device for grading the student according to their ability to answer questions and complete tutorials. A browsing pattern is defined here as a pattern made by a student as they visit and revisit hypermedia nodes; this is a view of the hypermedia at its highest, content-less level. The student's current interests are also ascertained, by examining how the student interacts with the semantic network; this is browsing the hypermedia at a content level; this information is then stored and utilised by the dynamic linking system to tailor links more closely to the student’s interests.

The structured link database is composed of weighted links to nodes, the weighted links being generated automatically. These weighted links are generated during the authoring phase rather than when the student is using the system; each link is weighted according to the relationship between the two nodes it connects. This relationship is defined by the length of the route connecting the two nodes through the semantic network, the weight being useful for determining the relevance of the link. Similar weighting systems have been employed by Pereira et al (1991) and Tudhope and Taylor (1997). Generating the links before the student uses the system and then modifying individual links according to the student model allows the system to rapidly present links. (Since the former process can be highly iterative this is paramount). It is accepted that altering link presentation does not offer the same potential for enriching a student’s learning experience as an ITS might. However, it does provide additional benefits to the standard hypermedia paradigm, without incurring the overhead of the ITS paradigm, previously discussed.

The approach discussed here requires a control system (called the “Hypermedia Manager”), which takes the information, concerning the student’s content-less browsing and content-based browsing, from the student model and the structured link database and on the basis of this determines which links are offered to the student, according to the student's current profile. Content-less browsing involves examining the student’s movements to and from previously visited and previously unvisited nodes, whereas content-based browsing is concerned with the contents of nodes visited by the student.

The Hypernet approach to educational hypermedia has two main objectives:

- To attempt to offer the student navigational aid, thus making it of potentially greater educational value than a traditional hypermedia system.

- To be non-domain-specific in its representation and techniques and hence be viable to employ for different domains.

The Hypernet approach overcomes this problem by using metrics that are not domain-specific, e.g. student interests derived from browsing particular class nodes and student abilities obtained by providing the student with tutorials. The drawback of this method is that these metrics are inevitably less powerful with respect to a specific domain than domain-specific metrics which may have been employed in a traditional ITS. However, as discussed in the previous chapter, a gap has been identified in the tutoring spectrum between ITS and non-guided hypermedia systems (Elsom-Cook 1989, Woods and

Warren 1997). The Hypernet approach is designed to fit between an unguided hypermedia system and an ITS.

## 3.2 Pedagogy

The Hypernet approach offers the domain author a system that requires only information about the structure of the domain and the domain content itself. From this, it is possible to support the student in their navigation, by manipulating how links are presented in order to reflect the student’s current ability and interests, until they have reached a sufficient proficiency level to be able to move more freely in the domain. The guidance takes the form of restricting the choices available to less advanced students, gradually revealing more detail as they become more advanced. The restriction of links is accomplished by removing those links that the systems determines to be the least relevant to the student, in accordance with the student’s current interests and goals, as identified by the student model, the student model being the subject of the following chapter. The Hypernet approach gauges and facilitates the student’s changing ability by a combination of tutorial tasks for the student to complete and a sub-system to measure the student’s ability (the Tutorial Supervisor). Both tutorial nodes and the Tutorial Supervisor are covered in greater depth in chapter 8, after other elements of the student model have been discussed.

The Hypernet approach is based upon Elsom-Cook's (1989) pedagogy, in that an attempt is made to identify the student's ability and interests and information is then presented accordingly (Elsom-Cook's sliding scale). A similar system is used by Woods and Warren (1997) for their generic tutoring system RAPITS. The Hypernet approach advocates a structured hypermedia system, in which the student is offered a number of links (which have either been specified by the domain author or generated automatically according to the semantic network) from the current node, from which they may then select. Subsequently, as their ability with the domain increases, they may be offered further links if they revisit the node. The number of links they are offered is determined by their ability, as determined by the system. Further, the types of links offered, in terms of their content, are determined by the nodes previously visited by the student (which represent the student's interests, or their previous content-based browsing) and/or by preferences entered into the system by a human teacher (which represents what they should be learning). For example, if the student has previously visited the nodes “Jupiter”, which is an instance of the class node "Planet", then it reasonable to assume that they would be interested in similar nodes such as the planet “Earth” or the planet “Mercury” and perhaps be less interested in the nodes "Earth", meaning soil, or "Mercury" the heavy-metal. In reality the diversity of nodes is not likely to be as great as the example given, although such distinctions are still likely to exist in most domains. Link weights may be altered so that nodes that are more relevant to the student's interests are more likely to be presented to them. This is of potential educational benefit to the student because they are aided in selecting links that are relevant to them because the system does not present the least relevant links. This method relies on the system being able to determine the student’s interests, however higher level students are offered more links and hence the system’s constraining effect on them is reduced.

A human teacher may define the number of links offered to the various levels of student. Similarly the number of individual levels may also be set. However, for the purposes of this research, the number of levels has been set to ten levels of student ability. This allows the student to belong to a clearly defined ability level, whilst preventing a student becoming stuck at a level if there are too few and the levels becoming meaningless if there are too many.

The pedagogy employed as part of this research project also incorporates adaptive hypermedia techniques described in Brusilovsky (1996). This allows the author of the domain to assign a hierarchical order to the nodes, in terms of certain nodes containing background information to another node. It may be that certain nodes can only be visited once other background nodes have been visited.

The Hypernet approach represents an improvement over allowing a student to idly wander through the hypermedia encumbered by all the deficiencies of the standard hypermedia paradigm, as outlined in chapter 2, such as getting lost and wasting effort in navigating the hypermedia system. Further, the pedagogy is domain independent and it is therefore potentially applicable to many domains without the need to redesign the whole system. The pedagogy discussed here answers criticisms raised by other researchers, namely over constraining students, forcing them to learn methods that may not suit them and basing evaluations on possibly misconceived concepts (Elsom-Cook 1989, Ridgeway 1989, Hartley 1993, Chiu et al 1991, Linard and Zeiliger 1995). It could be argued on negative grounds that the Hypernet approach does not over-constrain students because it does not measure enough elements of the student to be able to over-constrain them. This is undoubtedly true, but it is intentional that the Hypernet approach should not measure precise elements of the student so that it can, to a larger extent than other educational systems, be domain-independent, whilst avoiding the criticisms of Ridgeway (1989) concerning the over-constraining of students.

## 3.3 Overview of Hypernet

The Hypernet approach uses several sub-systems, which are shown in overview in figure 3.3.A. The Hypernet approach is based on a structured hypermedia system with the addition of a student model. Merging a semantic network with the hypermedia domain forms the structured hypermedia, which in turn enables the system to generate its own meaningful links. The student model (which comprises the Tutorial Supervisor (TS), Fuzzy Rule-Base (FRB), Browsing Pattern Recogniser (BPR) and record of the student’s interests (SER)) examines the student's ability to complete tutorials, their movements through the hypermedia (content-less browsing pattern) and the types of nodes previously visited (content-based browsing pattern), in order to ascertain the ability level and current interests of the student (taking into account what the human teacher thinks should be important). The Hypermedia Manager (HM) then utilises this information to alter the presentation of the hypermedia in order more accurately to reflect the ability and interests of the student in question. Content-based and content-less browsing have been introduced in overview here only and are discussed in greater detail in chapter 5, after the student model as a whole has been discussed and before those elements of the student model that are concerned with student browsing are discussed in detail.

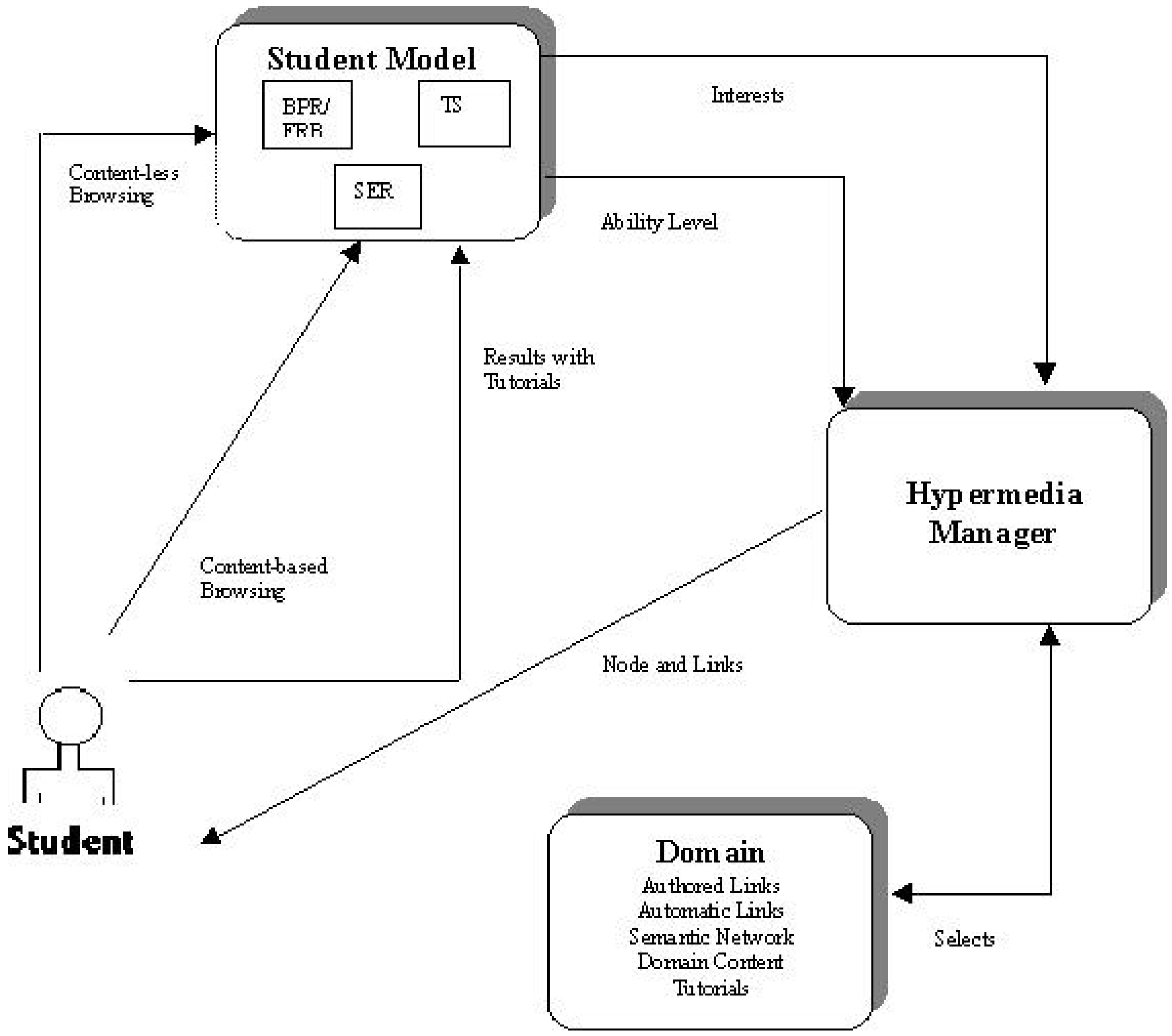


**Figure 3.3.A System Overview**

## 3.4 The Student Model

A student model is employed by Hypernet in order to record information, such as browsing patterns made by the student, student interests and student ability information about the student as the system is in use. This information about the student is then utilised in order to adapt the hypermedia to more accurately suit a particular student. The student model in Hypernet differs from traditional student models in that evaluations are not based on rigid domain-specific metrics (Sleeman and Brown 1982). Using domain-specific metrics has been criticised by Woods and Warren (1995) and Kinshuk and Patel (1997), on the grounds that they are time-consuming to develop and hence are not commonly employed, resulting in tutoring systems for relatively few domains. Instead, Hypernet's student model evaluates the student according to non-domain specific metrics: namely content-less browsing patterns, identified as potentially useful by Canter et al (1985); ability to complete tutorials and answer questions, as employed with success by Bergeron et al (1989); and information about the classes of nodes (according to the semantic network) the student has visited (content-based browsing patterns). It is argued in the following chapter that a combination of these metrics yields enough information about the student to be able to make a useful evaluation of the student. This can then be utilised by the control mechanism (Hypermedia Manager) for the purpose of making the system more suitable for the student.

The student model can also be utilised as a research aid for collecting information about how content-less browsing patterns relate to student abilities and the types of task that students undertake. For example, the student model records information about a student's browsing habits in terms of the actual node classes visited, the content-less browsing pattern employed and the student's ability at tutorial nodes. It then attempts to determine how the browsing patterns relate to particular tasks set by the system at the tutorial nodes (for example tasks that involve finding a particular piece of information, or acquiring structural knowledge etc.). Initially some assumptions may be made about how browsing patterns link to ability. For example, "wandering is bad" and "browsing is good for a more advanced user". However, this might not necessarily always be the case, therefore the system has been designed to offer the potential to adapt to limitations in the original assumptions, this ability being achieved by the use of connectionist/fuzzy logic technology. The FRB sub-system of the student model may alter its interpretation of a particular browsing pattern in relation to ability as it gains more experience with real students and hence has more information on which to base its assessments. The FRB is a research aid in that the information linking browsing patterns to types of student undertaking different tasks is interpretable by the researcher once it has been collected by students using a Hypernet-based hypermedia system. These facilities are discussed in depth in subsequent chapters.

The student model comprises several parts:

- Browsing Pattern Recogniser (BPR)
- Fuzzy Rule Base (FRB)
- Tutorial Supervisor (TS)
- Student Experience Record (SER)

The BPR and TS employ neural networks. The TS is used to grade the student into an ability level according to their performance with tasks set by tutorials stored within the domain. The BPR is used to recognise the content-less browsing patterns (sometimes referred to as browsing strategies (Canter et al (1985)) that the student makes whilst using the hypermedia. The FRB then links the content-less browsing pattern to other information obtained (from the TS and tutorials), thus potentially providing presently unknown information linking content-less browsing patterns to the types of student that have used them. The Student Experience Record is a general record of the node classes that the student has previously visited (their content-based browsing pattern, the class of the node being derived from the semantic network, for example "Earth" is of the class "Planet"), tutorial results (obtained from the tutorial nodes) and current ability level, (as previously determined by the Tutorial Supervisor). The information stored in the SER is then used to tailor the hypermedia links to reflect the student's interests and to reduce the number of links offered to more novice students.

## 3.5 The Hypermedia Manager

The Hypermedia Manager (HM) takes the ability level and interests of the student from the student model and translates these, using symbolic rules, into the number of links that the student should be offered, as previously determined by the domain author. It also utilises the Student Experience Record (SER) from the student model and any predefined human teacher preferences to dynamically tailor links to more closely match the student's interests. The SER provides information about which classes of node the student has visited and therefore shown an interest in, the human tutor may also manually set interests should it be required that the student examines a particular area of the domain. The HM then alters the weights of the links (previously generated and based upon the length of the path through the semantic network that connects two nodes) according to the SER and takes the top number of links for that particular class of student. The actual number of links offered to each level of student may be changed in line with the best pedagogic practice.

The Hypermedia Manager (HM) contains all information regarding the levels of students and the number of corresponding links each particular student is offered. These parameters can be changed for further research or may be set to different values should a particular domain require.

The following parameters in the HM have been set to "common sense" values. In the absence of experimental trails to determine the optimum value for each parameter, which was beyond the scope of this research, it was decided to adopt the values described below. “Number of Levels” has been set to ten. This means that a student can be graded as one of ten levels. Level zero is the least advanced and level nine the most advanced. The system is able to process many more levels should different domains and classes of students require. “The number of links” offered to each level of student, in each link box on the Hypernet interface, has been set as four for level zero, then four plus the level number for levels one to eight and finally all available links for level nine. It is conceivable that a domain author may opt for fewer levels with a larger increase of links per level. This would make the increase more apparent to the student, which may produce benefits in that the increase could be perceived as a reward for good progress, although the opposite could also be true if a student were to drop a level. It is probably better to have a smoother links-to-levels curve to avoid a student becoming trapped in a particular level. The interface of Hypernet offers links in one of three categories (discussed further below), “direct links”, “structural links” and “related material”. This helps avoid certain types of fundamental links being excluded from the selectable links; it also means that the student may be offered more than the set number of links for each level. The “number of links” field is therefore used as a limiting value for each category. This helps avoid criticisms raised by Debevec et al (1994) concerning the consistency of the user interface, since the class structure of the domain is always accessible to the student through the structured links. Note that inconsistency can result only in two situations: firstly if the student has dropped a level and secondly if their interests have changed resulting in them being offered some different links. Hypernet shows links that were previously available to the student in a different colour, or greyed out (Brusilovsky 1996), this may alleviate the problem of interface inconsistency.

Since the hypermedia domain is structured according to a semantic network, the human teacher may tag certain classes in the network as having greater priority for a particular student than others. The Hypermedia Manager contains these tags which are then used (by the Dynamic Link Manager) to modify link weights so as to present those links which the human teacher thinks may be relevant to the student. The importance of the tags relative to the student’s interests decreases as the student becomes more advanced, since expert students should be allowed to browse freely (Jonassen and Grabinger 1990). The SER uses the same system to alter the links to reflect the student's interests. Instead of class nodes being tagged manually by a human teacher, the system tags class nodes that the student has visited and hence shown an interest in. This helps more relevant links to rise to the top of the list of links.

## 3.6 The Semantic Network

A semantic network as defined by Rich and Knight (1991) is a representation of knowledge using nodes and typed links. Concepts link together to form a knowledge base. A semantic network is used in Hypernet to represent the knowledge structure of the domain. It is not the case that the semantic network represents all the knowledge stored within the domain (such as concepts contained within the text of individual nodes), since this would be an arduous task for even the simplest of domains and would result in a domain that was difficult for the student to read; this would qualify as Brusilovsky's (1996) level 3 representation of the domain. Instead the semantic network represents enough of the structural knowledge of the domain to aid the author in generating the domain, since it forces them to think carefully about the domain and where each particular node belongs. Additionally, the use of a semantic network enables the provision of automatic links, in that a link can be produced directly between two nodes that are already connected indirectly through the semantic network. The student is aided because the links and nodes are laid out in a logical and hierarchical fashion and by the automatic linking system that is based on the semantic network. The semantic network forms a skeleton upon which the rest of the domain may be attached (attaching instance nodes to class nodes). Note that each class node in the semantic network represents structural knowledge about the domain. It may also be used by the hypermedia as a node in the traditional hypermedia sense, in that it may hold text, graphics, video etc.

The semantic network forms the basis of the automatic linking system developed as part of the current research project. Since each node in the domain forms part of the semantic network, as either a class or instance of a class, a logical relationship between nodes can be calculated and hence a relevance factor from one node to another can be found.

It is further anticipated that the addition of a semantic network will help solve the problem of "Linkitis" (Conklin 1987). Linkitis is the problem of the author not knowing how to link a node to the rest of the hypermedia network with the result that the node may be linked to nodes that may have little relevance. Further, most hypermedia systems in common use have no facility for qualifying one particular link as more or less relevant than another (Lowe and Hall 1999). Thus the student’s navigation problem is increased and hence the student is given a complex learning task that is not necessarily associated with their learning needs, this being the "cognitive loading" problem. The requirement of a semantic network, however, forces the domain author to consider the structure of the domain and to impose it upon the hypermedia. The structure helps the author to decide whether a particular link should be created (Pereira et al 1991). For example, a class node such as “Satellite” may contain the specific requirement that an instance node must be attached to an instance node of another class node such as "Planet". In this case “The Moon” would be linked to a host “Planet”, "Earth". Further, automatic links reduce the burden on the author to produce links and since these links have a relevance factor (the length of the route through the semantic network) the student is provided with information regarding which link may be of most use to them.

### 3.6.1 Combining a Semantic Network with Hypermedia

A traditional semantic network represents knowledge using a structure of nodes (classes and instances) and links (relationships). Hypermedia more usually employs nodes of various media types and bi-directional, un-typed links (Brusilovsky 1996, Lowe and Hall 1999). It is a natural expansion of hypermedia to employ the typed links and nodes of a semantic network (Pereira et al 1991). It has long been the contention of hypermedia researchers that semantic networks or semantic network-like structures should be employed for hypermedia systems (Jonassen and Wang 1993, Kibby and Mayes 1991, Tudhope and Taylor 1997, DeBra and Calvi 1998). Further, using a semantic network to represent the domain is in line with Brusilovsky's (1996) knowledge-based approach to structuring the domain. Indeed, this has been achieved by other hypermedia researchers (Kibby and Mayes 1989, Beynon-Davies et al 1994, Tudhope and Taylor 1997). This method allows a natural clustering of nodes of the same type. In the case of the current research project, a semantic network is created for a domain to represent the essential structure. Hypermedia content, in terms of what might be thought of as traditional hypermedia nodes, may then be attached to, or incorporated within, the semantic network. The semantic network thus represents the hierarchical structure of the domain (and is provided by the domain author) and it is this structure that the automatic linking system uses to generate additional links which render the manual linking of nodes at the same level of hierarchy unnecessary. For example, figure 3.6.1.A shows that Mars and the Earth are linked indirectly via the class node “Planet”. It is therefore unnecessary for the domain author to provide this link. The automatic linking system may provide links at several levels of the hierarchy, links that are closely related are stronger than those that are linked by longer paths through the semantic network. The Hypernet approach helps remove the Linkitis problem (Conklin 1987), since the system provides the majority of the links, once the author has described the structure.

Additionally these indirect links provide information about the relationship between any two nodes in the hypermedia in terms of the strength of the relationship, namely the number of indirect semantic nodes connecting them together. It also provides information about how the nodes are related in terms of the link and class types connecting them together. Both of these elements are exploited in the Hypernet approach. The former is exploited in order to provide real time dynamic linking and the latter to provide contextual information to the student.

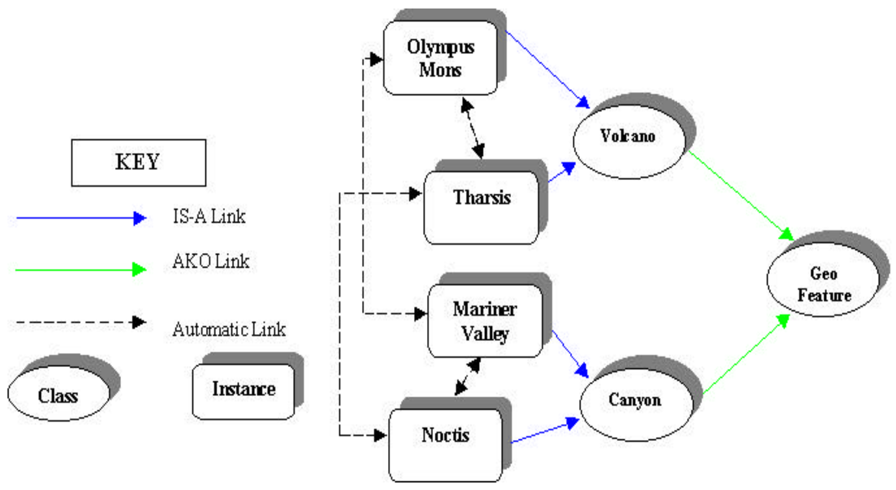


**Figure 3.6.1.A Solar System domain example**

Figure 3.6.1.A shows a simple semantic network representing a subset of a solar-system domain.  The nodes "Geo Feature”, “Volcano” and “Canyon” represent class nodes. The relationship between these nodes is demonstrated by the “AKO” (a-kind-of) link (e.g. "Volcano" [is] a-kind-of "Geo Feature", in that a "Canyon" inherits properties of "Geo Feature", such as the fact that they features on the surface of a planet). The instance nodes (“Olympus Mons”, “Tharsis”, “Mariner Valley” and “Noctis”) are attached by “Is-a” links to their semantic class node. Chapter 9 takes the discussion of link types further, the remainder of this discussion pertains to Hypernet's architecture.

## 3.7 The Automatic Linking System

Using a Solar System domain as an example, the nodes "Olympus Mons" and "Tharsis" are linked by an "is-a" link to the class node "Volcano". This gives a simple knowledge base, namely the knowledge that both "Olympus Mons" and "Tharsis" are "Volcanoes". Thus the system can automatically produce a link between "Olympus Mons" and "Tharsis", since they are both "Volcanoes". Further links can be produced automatically between "Olympus Mons" and "Mariner Valley" and "Olympus Mons" and "Noctis", via the weaker relationship through the "Geo Feature" class node (further automatic links can also be produced between "Tharsis" and "Mariner Valley" and "Tharsis" and "Noctis" via the "Geo-Feature" class node). These automatically generated links are potentially useful since they offer related material, further, the weight allows the system to reduce the number of links it offers to novice students without randomly pruning them, in that links that have a weight below a certain threshold can be excluded from presentation to the student.

It may be argued that many of the links generated by the system may not be useful to the student, since it is normally the case that the longer the path through the hierarchy the less obvious the relationship between the two nodes it connects. This is a primary argument for the need for facilities to aid the student using a hypermedia system, since a large corpus of links that are not obvious to the student increases their cognitive loading and thus reduces the system's effectiveness as a tutoring system. However, the links that are provided through the semantic structure are of use to students of higher levels, since it removes the need to navigate through a domain structure that they in all likelihood have already learned (as indicated by their high level) and possibly expanded upon (Jonassen and Wang 1993). Freed from the need to navigate the entire domain structure they are able to embark more fully upon their own discovery learning (Elsom-Cook 1989). Traditionally, it has been the responsibility of the author to provide all links. This has been criticised for producing "Linkitis" (Conklin 1987). Links used in traditional hypermedia systems do not provide the student/user with information regarding the relevance of the particular link (Brusilovsky 1996b), and therefore the student/user is given a profusion of links that may look identical. By contrast the automatic links may be presented in order of relevance, they may be presented with a subset only (the most relevant), or the relevance factor may be altered dynamically to further tailor the links to the student (discussed in the dynamic links section later in this chapter).

The nodes that represent classes provide information for both the students and authors. For the student they provide general information and a starting point from where they may seek more detailed information should they require it. For authors they may provide information about how to add new nodes to the domain, should the initial domain author have provided it. For example, the "Geo feature" class node, if properly defined by the author, should explicitly state that an instance node of "Geo Feature" “has-a” host "Planet", for example, so that it is clear that "Olympus Mons" should be connected to a host "Planet".

### 3.7.1 Practical Domain Issues

Once the domain structure has been defined then the automatic linking algorithm is executed before students use the system. The process is highly iterative and consumes an exponential amount of processing power for each node in the domain. It is not the concern of this project that the algorithm is excessively slow (in terms of a few minutes for a domain of tens of nodes), since the speed of the processor on which the program is run and the programming language used are of little consequence, this process being activated before a student uses the system. However, the processing power required is orders of magnitudes greater than any other part of the system, such as executing the neural networks and displaying large bitmaps. It is therefore sensible to execute this stage when no one is requiring a response from the system. This process is required only when a new domain is authored, or additional nodes are added to the domain. Once the process has completed then the links that are generated are saved to a file that is then loaded each time the system is used.

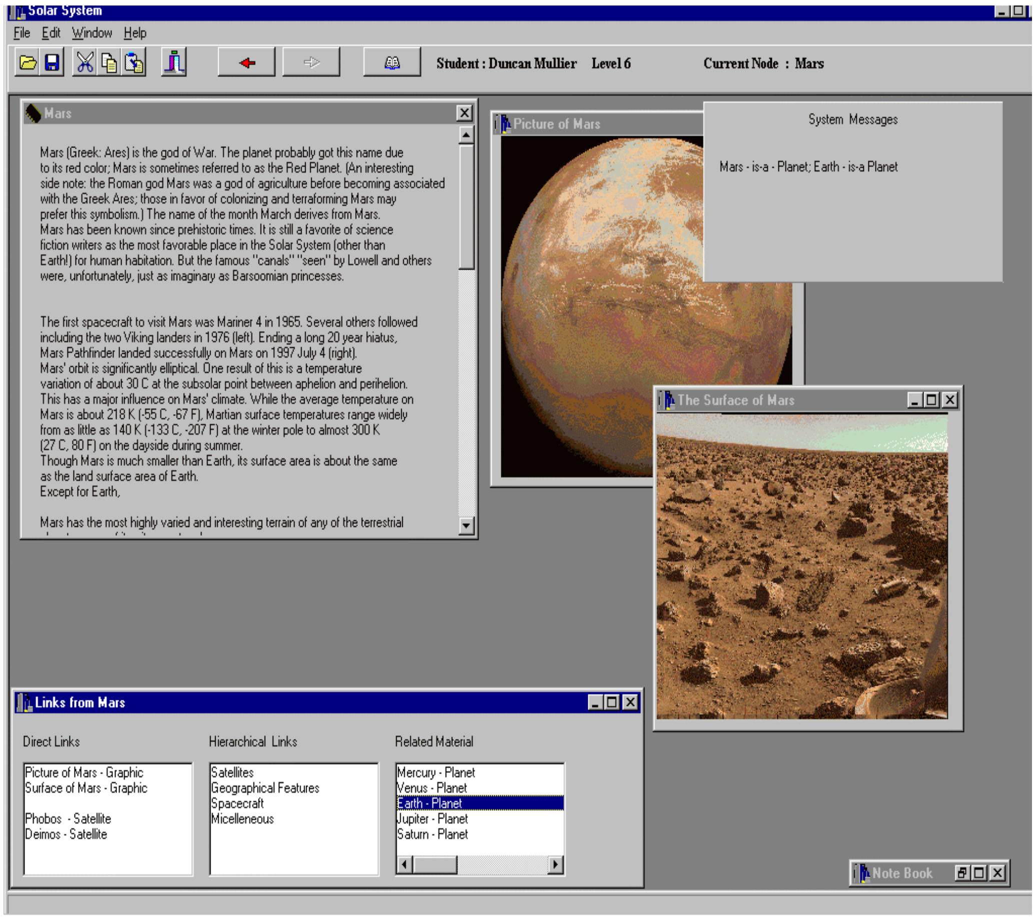
This method differs from other intelligent hypermedia systems in that the "intelligent" linking is effected when the node is added to the system, rather than at run time where it would require a massive processor overhead (Kibby and Mayes 1989). The automatic linking system is contained in Appendix C.

Defining a domain does not involve directly inputting a graphical semantic network, although there is no reason why software could not be written to make this possible. It is suggested however that the author first defines the domain as a diagrammatic semantic network, which can then be converted into binary relations, as described by Beynon-Davies et al (1994).

## 3.8 The User Interface

The interface has been designed to present multimedia nodes to the student. The main point of interest is the system for presenting links. Links are presented to the student separately from the nodes themselves. Many hypermedia systems use “hot words” within the text of the node itself to represent links to other nodes. This means that a word representing a link must be present within the text of the node, which may not necessarily always be the case. A further criticism of this method is that it induces more cognitive load on the student, since the student is not presented with links in a logical manner; instead they are presented with links in the arbitrary order that they occur in the node text (Jonassen and Wang 1993). Most of the adaptable hypermedia facilities described in chapter 2 and by Brusilovsky (1996) require that the links are separate from the node itself, so that they can be sorted and clearly annotated. For these reasons the Hypernet approach presents links outside of the node text itself, in three link boxes. The “Direct Links” box contains links that are directly connected to the current node (such as “Phobos” to “Mars”, or “Picture of Mars” to “Mars”) that have been defined by the domain author (although, as described above, some may have been requested by the system), the “Hierarchical Links” box contains the links that connect the current node to the semantic network (for example, “Moon” is-a “Satellite”). The "Hierarchical Links" box is also used to provide a method for presenting many direct links together, for example, the nodes "Olympus Mons" and "Tharsis", which are connected directly to "Mars", can be excluded for a novice student, but can still be visited through the hierarchical link "Geographical Feature" (shown in Figure 3.8.B).

The “Related Material” box contains the links generated by the automatic linking algorithm. Thus the three boxes of links present three different kinds of links. The order of the links in the boxes is determined by the strength of the relationship with the current node (the most relevant are at the top). The number of links offered is determined by the student model (more for more advanced students).



Mars (Greek: Ares) is the god of War. The planet probably got this name due to its red color; Mars is sometimes referred to as the Red Planet. (An interesting side note: the Roman god Mars was a god of agriculture before becoming associated with the Greek Ares; those in favor of colonizing and terraforming Mars may prefer this symbolism.) The name of the month March derives from Mars.
Mars has been known since prehistoric times. It is still a favorite of science fiction writers as the most favorable place in the Solar System (other than Earth!) for human habitation. But the famous "canals" "seen" by Lowell and others were, unfortunately, just as imaginary as Barsoomian princesses.

The first spacecraft to visit Mars was Mariner 4 in 1965. Several others followed including the two Viking landers in 1976 (left). Ending a long 20 year hiatus, Mars Pathfinder landed successfully on Mars on 1997 July 4 (right).
Mars' orbit is significantly elliptical. One result of this is a temperature variation of about 30 C at the subsolar point between aphelion and perihelion. This has a major influence on Mars' climate. While the average temperature on Mars is about 218 K (-55 C, -67 F), Martian surface temperatures range widely from as little as 140 K (-133 C, -207 F) at the winter pole to almost 300 K (27 C, 80 F) on the dayside during summer.
Though Mars is much smaller than Earth, its surface area is about the same as the land surface area of Earth.
Except for Earth,

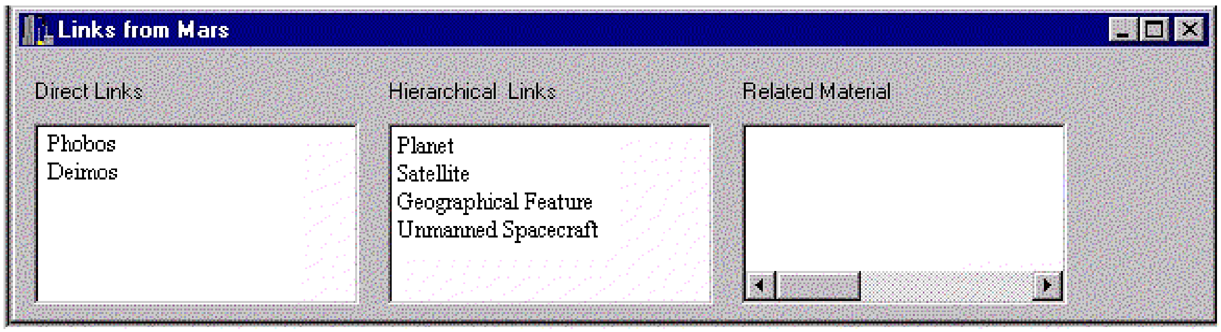
Mars has the most highly varied and interesting terrain of any of the terrestrial

System Messages

Mars - is-a - Planet; Earth - is-a Planet

| Direct Links | Hierarchical Links | Related Material |
|---|---|---|
| Picture of Mars - Graphic | Satellites | Mercury - Planet |
| Surface of Mars - Graphic | Geographical Features | Venus - Planet |
|  | Spacecraft | Earth - Planet |
| Phobos - Satellite | Micelleneous | Jupiter - Planet |
| Deimos - Satellite |  | Saturn - Planet |

**Figure 3.8.A The Interface**



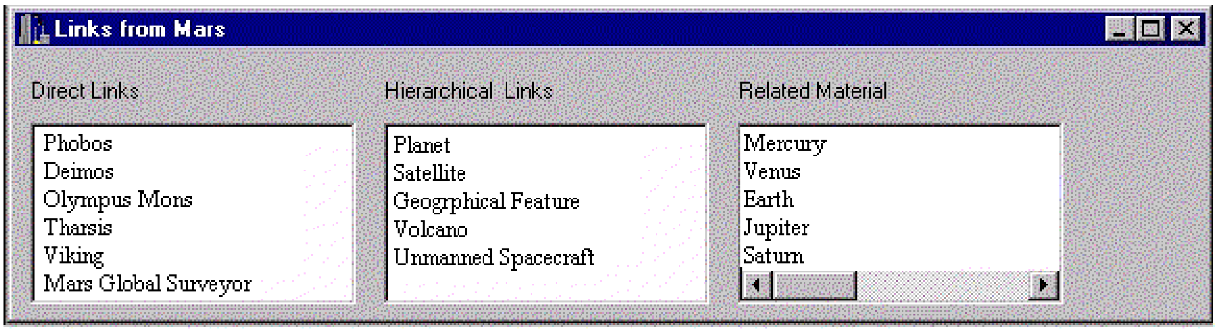
Links from Mars

Direct Links: Phobos, Deimos

Hierarchical Links: Planet, Satellite, Geographical Feature, Unmanned Spacecraft

Related Material:

**Links offered to a "Novice" student**



Links from Mars

Direct Links: Phobos, Deimos, Olympus Mons, Tharsis, Viking, Mars Global Surveyor

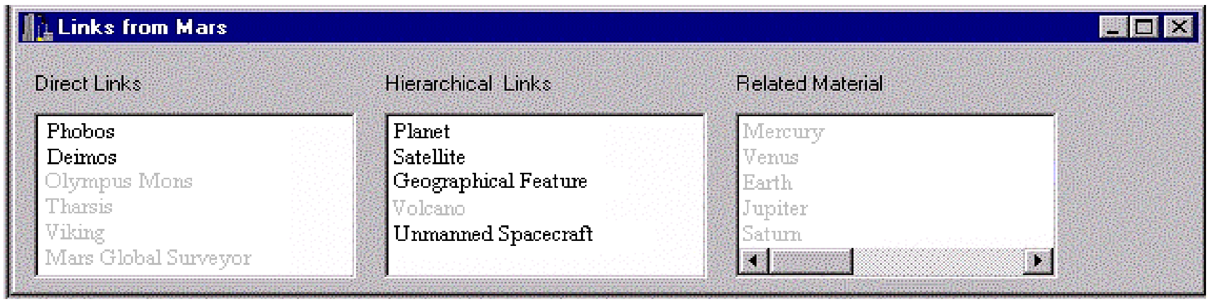
Hierarchical Links: Planet, Satellite, Geogrphical Feature, Volcano, Unmanned Spacecraft

Related Material: Mercury, Venus, Earth, Jupiter, Saturn

**Links offered to an "Expert" student**

**Figure 3.8.B Links Offered to Different Students**

Figure 3.8.B Shows links from the same node presented to two different levels of student. The higher level student is offered more links in the "Direct Links", "Hierarchical Links" and "Related Material" link boxes. In order to combat the problems of an inconsistent interface (Debevec et al 1994), a student who has been offered links before (as either a higher level student or because their previous interests were different) is offered the links they were offered previously. However, if they are below the current threshold they are greyed out (but still selectable) as in Figure 3.8.C. A similar system is advocated by Brusilovsky (1996).



Phobos
Deimos
Olympus Mons
Tharsis
Viking
Mars Global Surveyor

Planet
Satellite
Geographical Feature
Volcano
Unmanned Spacecraft

Mercury
Venus
Earth
Jupiter
Saturn

**Figure 3.8.C Greying of previously offered links**

## 3.9 Dynamic Links

### 3.9.1 User Specific Dynamic Links

The linking described above is generated according to the domain and not the user of the domain. The user may be interested in a very specific area of the domain whereas the hypermedia itself is modelled on a higher level. The link that the user is interested in may be buried at the bottom of a pile of links that the system has graded as more relevant in accordance with the semantic structure. If links more relevant to a particular student are to be brought to the top of a link box a method of user modelling is required to keep track of the student's interests and state of knowledge.

The student model provides this information in terms of previous node types visited (their content-based browsing). The student model stores the class of the node in the semantic hypermedia that the student has been most interested in. For example, the student may have expressed an interest in “Planets” because they have visited the nodes “Earth” and “Mars”, and it can therefore be inferred that they may be interested in other “Planet” nodes. The system can then use this information, held in the student model, to modify the already present weights generated previously by the automatic linking algorithm and stored with each link. Thus the links are dynamically modified for a particular student at a particular instant in time. Since the student is presented a subset of the available links only (according to experience level) this facility has the result of bringing the link, which may have been buried under semantically stronger links, nearer the top of the list and thus beyond the threshold at which point they are offered to the student. As described above, a possible pitfall of this approach is that links that a student has seen at a node on a previous occasion may not be present at subsequent visits, because the system has determined that the student's interests have changed. The student may have returned to the node in order to navigate to another node that they saw at their first visit. A student is likely to become confused if the node that they are looking for is no longer present. Therefore, a record of nodes previously presented to a student is stored within the student model so that they can be represented, albeit greyed out if they are below the current threshold.

### 3.9.2 Teacher Specific Dynamic Links

A third method of dynamic linking allows a human teacher to modify the link weights so as to favour certain areas of the semantic hypermedia. For example, the teacher may want the student specifically to look at a certain area of the domain. This would be accomplished by assigning a priority to certain class nodes in the semantic network. The system then increases the weight for each instance of this class, thus making it more likely to be selected. It may be appropriate to reduce the effect of the teacher’s preferences as a student becomes more advanced. This crossover point is defined as a preset in the Hypermedia Manager for each individual student level.

In implementation both the User and Teacher Dynamic links use the same mechanism. The algorithm employed processes the set of links and alters the strength of those automatic links that are more closely related to the classes of node that the student has previously visited. For larger domains the depth of this search may be limited by another preset in the Hypermedia Manager, since the further down the list of links the less likely the chance of a weight modification being sufficient to make it presentable. The algorithm adds on an amount (which may be varied as an experimental parameter) to the link weight for those links that connect to a node of the correct semantic type. This has the effect of increasing the chance of the link being presented to the student. The algorithm also examines links that are removed further from the desired node class and adds on a degree of the weight increase. This portion steadily decreases as the node types become further removed. In essence this is a similar algorithm to the automatic link generator. However it is limited in its depth to enable its response time to be unnoticeable to the student (by avoiding processing links which will never be displayed to the student, due to the number of links they are offered). For example, if the student or teacher has shown a preference for "Planet" nodes, then all nodes of class "Planet" have their weights increased and all nodes of class "Satellite" also have their weight increased, but to a lesser degree. An example of the usefulness of Hypernet is for a domain that represents a diverse content, such as an encyclopaedia domain. If the student is interested in the solar system, then it might be helpful to present them with nodes representing the Roman gods, after whom the planets are named, but it would be unhelpful to offer links to other areas of Classical study (such as Roman poetry), which would inevitably be linked to nodes concerned with Roman Gods. The semantic network allows the distinction to be made and thus enables the system to give a higher priority to those nodes associated with the solar system. It might be argued that it is impossible to tell which nodes might help a student put their current learning into context. However, once the students have achieved a sufficient amount of learning, as determined by the system, then they will be able to visit such nodes as more and more nodes become available to them.

### 3.9.3 Link Presentation

Links are altered in several ways so as to present the most relevant links to the student, without inducing excessive cognitive loading. The Hypernet approach uses three of the four methods of link adaptation described in Brusilovsky (1996), namely, annotation, sorting and hiding. Brusilovsky's fourth method, direct guidance is not employed, in that the student is not encouraged to follow any predefined complete paths, largely because of taking account of criticisms raised by Jonassen and Grabinger (1990) that this inevitably leads to the student blindly following the path. Students may tend to follow the most relevant links as defined by the semantic structure, and therefore indirect guidance is employed. Annotation is used in that the student is able to click on a related material link and see the path that connects it to the current node through the hypermedia structure. Sorting is used by having the most relevant links appear towards the top of the link boxes. Hiding is used to remove certain types of links, depending upon the link box, for particular levels of student. The simplest form of hiding is used for the related material link box. The related material box displays a certain number of links for a particular level of student; the remainder (which potentially includes the rest of the nodes in the domain) are not displayed. The hierarchical link box shows the structure of the domain to a set depth for each level of student, therefore a more advanced student sees more of the structure. The direct link box operates differently in that certain types of links may be excluded for certain levels of student - this is in line with Brusilovsky's (1996) knowledge-based approach, in that different complexities of nodes may be presented to different classes of student. Certain link types may be deemed unsuitable for a novice student, for example a link type denoting a counter-example might be considered confusing for someone who is not proficient with the domain. It is a matter for the domain author to describe the domain using whatever link type they consider necessary and to exclude some from certain classes of students. The automatic link generator algorithm itself is robust to new link types, since its purpose is to connect nodes via the semantic network. It bases its structural links upon those link types defined above (isa, hasa etc.), the algorithm that generates the related material links does not take into account the type of the links (except to exclude those already in the structure links box).

## 3.10 Discussion

Hypernet, as it has been introduced in this chapter, raises several important issues. Firstly, does it aid the student more than a totally passive educational system (an electronic textbook) and secondly, does it achieve any such aiding of the student without incurring a prohibitive amount of effort from the domain author? Both of these issues apply to other educational systems and it has been argued by Ridgeway (1989) and Woods and Warren (1997) that most ITSs fail on the second issue, in that they do require an excessive effort from the author (or system designer). Further, Ridgeway (1989) raises the issue that it is not desirable to force a student to learn a method that may not suit the student’s learning style. However, it is not a simple matter to answer these issues, since it is difficult to definitively evaluate educational systems (Jonassen and Wang 1993, Dillon and Gabbard 1998). Hypernet is an attempt to take a step in the right direction, towards the goal of producing a system that provides benefits to the student, in that it aids them in their navigation without preventing them from exploring their own goals (at expert student levels). There is a requirement on the part of the author to define a domain in a structured fashion and to define tutorial tasks. However, the preparation of course material is a requirement for any educational system, passive or not. The approach introduced in this chapter does not require that software components are reproduced. There is therefore a high degree of reuse of system, as argued for by Lowe and Hall (1999). Therefore, it is open to educational debate as to whether the approach is useful to the student, but it is likely that it requires less effort on the part of the author than say an ITS.

At a lower level the approach described here raises several more issues. Firstly, does the adaptation of links offer educational benefits to the student and secondly, does Hypernet achieve this adaptation? Link level-adaptation has been identified as the most fundamental method available to the adaptive-hypermedia designer to tailor the presentation of material towards a student (Brusilovsky 1996). The ideal situation would be to automatically present the student with the next node of information that will enable them to learn the next concept. This is not possible because it would require a deep understanding on the part of the educational system of both the student and the domain being taught. Therefore it is necessary to offer the student several choices of which only one can be selected at any given moment. Therefore, the links that are not selected are a hindrance to the student since they must decide which one to select.

## 3.11 Conclusion

This chapter has described the Hypernet approach to educational hypermedia. The facilities were described in detail, notably the semantic network used to represent the structural knowledge of the system and the dynamic linking system. It has been argued that a semantic structure should be a requirement for every hypermedia system, in order to provide the student with structural information that may aid them in incorporating the knowledge that they have learned. It also provides an essential navigational aid, since it forces the author to structure the domain hierarchically and logically. The use of the semantic structure enables the use of an automatic linking system that is able to produce links with a relevance factor (weight) based upon the semantic structure. The addition of a student model allows the production of a dynamic linking system, since the aforementioned weights can be changed in the light of the student model. Clearly then, the student model is a crucial part of the system. It is discussed further in the following chapter.

# Chapter 4

# The Student Model

## 4.1 Introduction

This chapter is concerned with a student model designed for the Hypernet approach to educational hypermedia. The student model is designed to be domain independent, basing its evaluations upon the student's ability to complete tutorials and the patterns that students make as they move through the hypermedia domain, e.g. visiting some nodes and not others. This chapter is concerned with these metrics and the interactions between the sub-systems that are designed to utilise them. Subsequent chapters then deal with the internal composition of these sub-systems.

## 4.2 Student Model Overview

The student model comprises;

- The Tutorial Supervisor (TS)
- The Browsing Pattern Recogniser (BPR)
- The Student Experience Record (SER)
- The Fuzzy Rule Base (FRB) to link the outputs of the other components

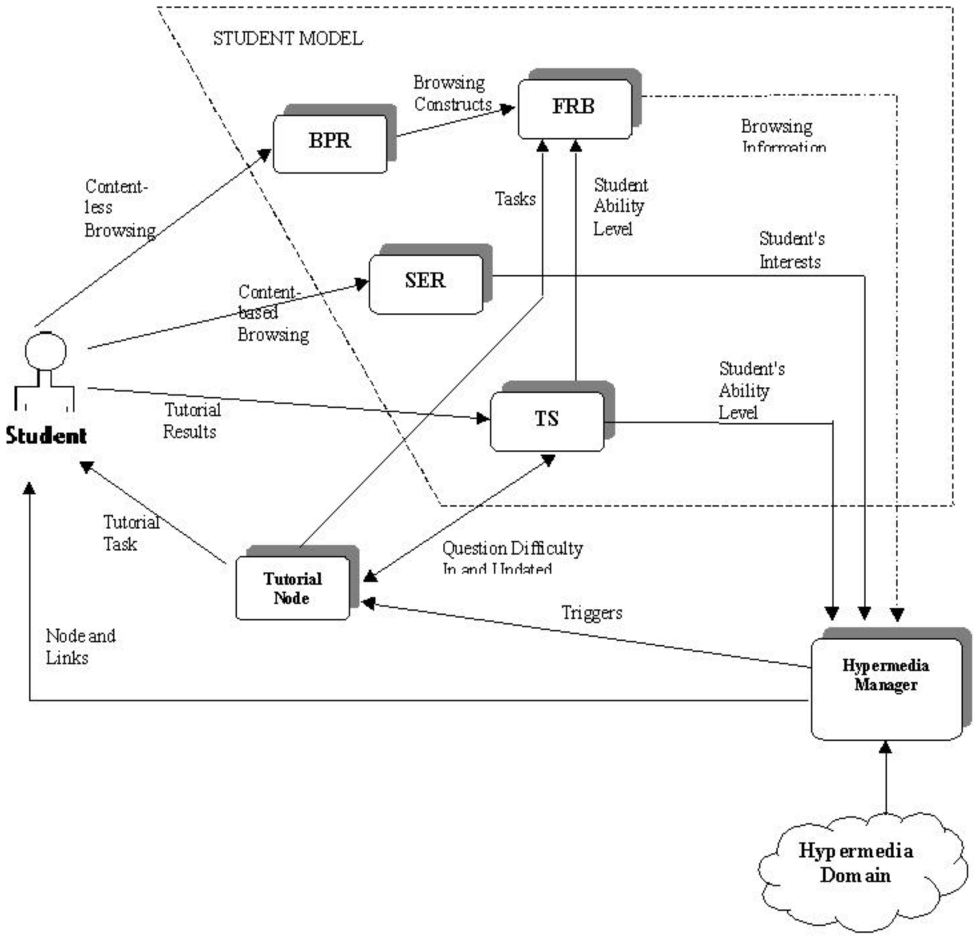


**Figure 4.2.A The Student Model**

Figure 4.2.A demonstrates the interactions between the individual parts of the student model and the student. The SER records the student's content-based browsing, in terms of the links that have previously been offered to the student and from this the SER infers the student's interests in terms of the node classes that they have visited. For example if they have visited "The Earth" and "Mars", which are connected to the "Planet" class node, then they are determined by the student model to have shown an interest in "Planets", this determination is used by web agents such as Web-watcher (Armstrong et al 1995). The BPR monitors the student's content-less browsing, in terms of the student's movements through the hypermedia, which is represented by nodes that they have previously visited and nodes that they have not visited. Browsing patterns are covered in depth in the following chapter, before the discussion of the sub-systems that utilise browsing patterns begins. The TS examines the student’s results with tutorial questions and tasks and grades the student into an ability level. The student ability level is then passed to the FRB, along with the results of the BPR, which are the low-level browsing constructs, described in the following chapter. The FRB then utilises this information to form a link between the ability of the student, the types of tasks that the student undertakes whilst using the hypermedia (gained from the tutorial nodes) and the content-less browsing patterns that the student makes within the hypermedia. This element of the student model requires exposure with many students before it is fully trained and fully utilised by the rest of the student model (as indicated by the dashed line output of the FRB on figure 4.2.A). It is the intention of the FRB information to be used as further research in that it has the potential to provide information concerning how different types of students use hypermedia for various tasks. This information may be of use in agent-based navigation aids (Armstrong et al 1995). The student level information is also utilised to ensure that the grading of the tutorial questions is accurate (e.g. a question may be set incorrectly by the domain author). Finally, the Hypermedia Manager utilises the student level information and the student's interests in order to determine how many links to offer the student (less for novice students etc.) and provides dynamic links (in terms of adjusting the link weights to more closely resemble the student's interests). These interactions and facilities are discussed in detail below.

## 4.3 The Metrics

Intelligent Tutoring Systems exhibit “intelligence” towards the student partly by altering their behaviour according to the perceived cognitive state of the student at a particular instant in time. This is achieved by recording information about the student as they use the system. Rules relating this information to tutoring strategies are then applied, with the aim of presenting the student with the tutoring strategy that best suits them at a particular moment. The information recorded about the student is in the form of a number of metrics, a metric being what is measured by the system (Woods and Warren 1995). The Hypernet approach is not intended as a direct replacement for an ITS. Instead, it is intended to fit into the gap between a system that offers no guidance facilities (such as an unguided hypermedia system) to the student and one that rigidly guides them (such as an ITS). Such a system that offers some assistance to the student is a step in the right direction, even though it cannot measure every element of the student's understanding (Elsom-Cook 1989, Woods and Warren 1995). However, in common with ITSs, it is necessary to measure some attributes (metrics) of the student.

The metrics employed for this research project are domain independent and measure only two attributes of the student, namely their ability and their interests. These metrics would not usually be employed alone by an ITS, since they do not directly measure the student's understanding of concepts pertaining to the domain in question. However, other more accurate metrics would be domain dependent and are not usually available to hypermedia designers, since this prevents the system from being used for many domains (Brusilovsky 1996, DeBra and Calvi 1998). The metrics employed in this research project provide some measure of the student’s ability at the given domain, since the student is asked to complete tutorial tasks and their success with these tasks can be measured. This ability level can be utilised to tailor the system to a particular ability of student. Furthermore, the student's interests can be extracted from their movements through the semantic network and the links can then be tailored to suit these interests. This is discussed in more detail below.

### 4.3.1 Domain Dependent Metrics

Domain dependent metrics, such as those employed by domain dependent ITSs, measure complex and intricate elements of a student’s understanding of a particular domain (Sleeman and Brown 1982, Woods and Warren 1995). They may be identifying something as specific as a misconception about a particular mathematical method. Such a metric could not be employed for any other domain. They do enable a system employing them to accurately identify a particular predefined misconception that the student is employing and may therefore enable the system to act to remedy it. However, it has been argued that such a method may be restrictive to the student, since they are being forced to learn by a particular method, a method that may not be suited to them individually (Ridgeway 1989). Since many domains and all students are complex, it has been argued that domain dependent metrics themselves may be inadequate, since they are likely to be based on incomplete knowledge of the domain and of individual students (Chiu, Norcio and Petrucci 1991). Therefore, it may be that using an ITS that employs such methods may encumber the student. Further, domain dependent metrics are expensive to produce in terms of the effort in identifying and implementing them. They do not necessarily encompass a large section of the domain, resulting in a multitude of such metrics. The result is a complex set of rules about the domain and possible students who will encounter it.

The ability of a student at a particular domain is by definition domain dependent. The student's ability may change from domain to domain (they may be good at one subject but poor at another). However, their ability for a particular domain can still be measured in the same way for many domains, i.e. using the same tools. This can be achieved by providing the student with tasks to complete (at tutorial nodes) and then measuring their success with these tasks by using the Tutorial Supervisor. Tutorials could take the form of multiple choice questions or more complex tasks that require that the student visits several nodes in the domain. Differences in scale (student abilities may be more tightly clustered in one domain than another) are catered for by the TS since it uses an adaptable neural network architecture (covered in detail in Chapter 8). The student's interests, in terms of the classes of nodes that they have previously visited, can also be seen as domain independent since they are inferred from the semantic network that forms part of the domain; similarly their interests may change from domain to domain, but they are extracted in the same way. Therefore, these metrics can be used across a variety of domains, dramatically reducing the cost in terms of time for development.

### 4.3.2 Domain Independent Metrics

The difficulties with domain dependent metrics are avoided by domain independent metrics. They are cheap to produce for a single domain, because they are not measuring intricate elements of the domain, they are cheaper still because they can be applied across domains. They measure less precise elements of the student than domain dependent metrics so that a rule base devised of several is not likely to restrict the student and force them to learn a particular method. Since they are not measuring exact patterns of behaviour, they are not forcing the student to exhibit those elements of exact behaviour. The student is thus freed to a large extent to employ their own methods that are more suited to them and not necessarily those more suited to the author of the system, although the tutoring system can aid in their navigation (by providing them with set of relevant and manageable links from a rich set of total links) and reduce the student's "cognitive loading". The argument for the employment of domain independent metrics for systems that would otherwise have no measurement is compelling, despite the fact that they are limited in the information they provide.

### 4.3.3 The Metrics Employed by the Hypernet Student Model

The first metric utilised by the student model is the student’s ability within a particular domain, at a particular instant in time, expressed as an integer from a predefined set of integers. This ability level directly affects the number of links that the student sees, in that the navigational possibility at each node increases as the student progresses up the ability levels (or decreases if they drop in ability levels), this goes someway towards Elsom-Cook’s (1989) idea of providing a sliding scale of support from novice to expert. The actual number of levels can be varied according to the needs of the domain or population of students in question. This metric is measured using the Tutorial Supervisor, which is the subject of chapter 8.

The current research project also investigates the potential to extract information about the student based upon their content-less browsing patterns only. This has been identified as potentially useful by Canter et al (1985), in that the patterns a student makes as they browse the hypermedia may indicate "psychologically significant information". Such information may indicate a student's browsing task and the type of student that they are, in terms of ability (i.e. expert student use particular patterns that novices do not, etc.). This "psychologically important" information is not currently available, since it has proved to be difficult to extract (Canter et al 1985), but may be obtained as a population of students use the system. Once obtained, the information may be of use in other hypermedia systems, for example as an addition to a browsing aid, since such information may be able to help determine the granularity of information that the student requires. Several researchers have attempted to explain which types of patterns and strategies relate to particular activities and how these activities relate to a particular ability of a student (Dillon 1990, Canter et al 1985, Macarelli and Sciarrone 1996, 1998). However, it is by no means clear how these browsing patterns relate to different domains and different students. Therefore, a novel approach has been adopted for this problem, namely the use of connectionist technology and fuzzy logic. These technologies are utilised in order that the browsing pattern metric, as measured by the BPR, is dynamically altered according to new information from the TS and results from the tutorial nodes. The result of this is recorded in the FRB, the benefit of this being that the information in the FRB can then be interpreted by the researcher and may be applicable to other hypermedia systems. Browsing information could be useful for gaining an insight into the student’s information seeking goal (are they looking for specifics or do they want more general information?), it could also be useful in situations where it is difficult to extract information about the student or user, such as the World Wide Web. These issues are discussed in greater detail in the following chapter, which is primarily concerned with browsing in hypermedia. This part of the discussing has centred on collecting content-less browsing information, the next part is concerned with content-based browsing information, and this is where the actual content of nodes is considered.

## 4.4 The Student Experience Record

The SER is based upon an overlay model (Brusilovsky 1996) and is used to extract information regarding the student's content-based browsing. Several adaptive hypermedia systems utilise overlay models to record information about the student/user (Brusilovsky 1996). These can be as simple as recording whether a node has been visited or not, or may store information about which concepts, contained within the node, have been encountered or learned by the student. A key element of the overlay model is that it "can measure independently [of the domain] the user's knowledge of different topics [domains]" (Brusilovsky 1996). This is a key feature of the domain-independent paradigm employed for this research project.

The SER is a record of the student's interaction with the semantic network. This provides information about the types of nodes that the student has visited and is therefore taken to be interested in. For example, the student may be examining the "Olympus Mons" node in figure 4.4.A. This may indicate that the student is interested in "Volcanoes", since "Olympus Mons" is-a "Volcano". Further reasoning infers that the student may have a high likelihood of being interested in “Geo Features” and hence may be interested in “Canyons” and “Mountain Ranges”. However, examining the semantic network demonstrates that the same size route through the semantic network links “Volcano” with “Planet”. The student may therefore be offered links to other Planets at the same relationship strength as Canyons. This may not be desirable if the student is really only interested in Geographical Features (at this particular point in time). This interest can be ascertained by virtue of the student visiting related nodes. If the student has visited several nodes that are connected to the “Geo Feature” class node, then the system can gauge their interest in this area and increase weights to these links. Thus nodes representing “Geo Features” are presented above other links of the same weight, according to the semantic network. The student's changing interests are modelled by giving precedence to the most recent and most frequently visited class nodes that have been visited (or indirectly visited by virtue of a visit to an instance of a class). Therefore, if a student visits a particular class (or instance of a class) then it becomes an immediate interest of the student. If the student revisits the class (or other instances) then the interest value of that class is reinforced.

The process of defining the student’s interests is accomplished by the SER. As a student visits a node its class (e.g. the class associated with “Olympus Mons” is “Volcano”) is stored in the SER, along with all sub-classes to a shallow depth (the depth is dependant upon the number of class nodes in the domain). The first class is given the highest score, subsequent classes are given diminishing scores. Every time the student visits a node, the process is repeated; however, all values are added to values previously calculated. Reducing the value for each class that is not relevant to the current node represents a student’s changing interests. Values are decreased in such a way so as to reduce the values back to zero, if the node is not re-involved. This method allows a student to change their interest over time, without the SER changing what it perceives as the student's interests after the student visits a single example of a previously unvisited class of node. The upshot of this is that the student’s interests are built up as they visit similar class nodes and then may diminish as they no longer visit these nodes (because they have lost interest). This method has the advantage that the system does not continually change what it determines as the student’s current interests in response to a student visiting a new class of node (possibly accidentally). However, it has the possible drawback of not having the ability to react quickly should a student suddenly change their interest, in that a student may think “I’ve seen enough Planets now”. However, the SER is an attempt to select a number of what are determined to be relevant links from a set of all links, it is not necessarily the case that a student’s new interest will not be offered. Further, it is possible to offer class nodes in a separate link box, which is always available to the student, the result being that the student always has access to the complete domain at a general level.

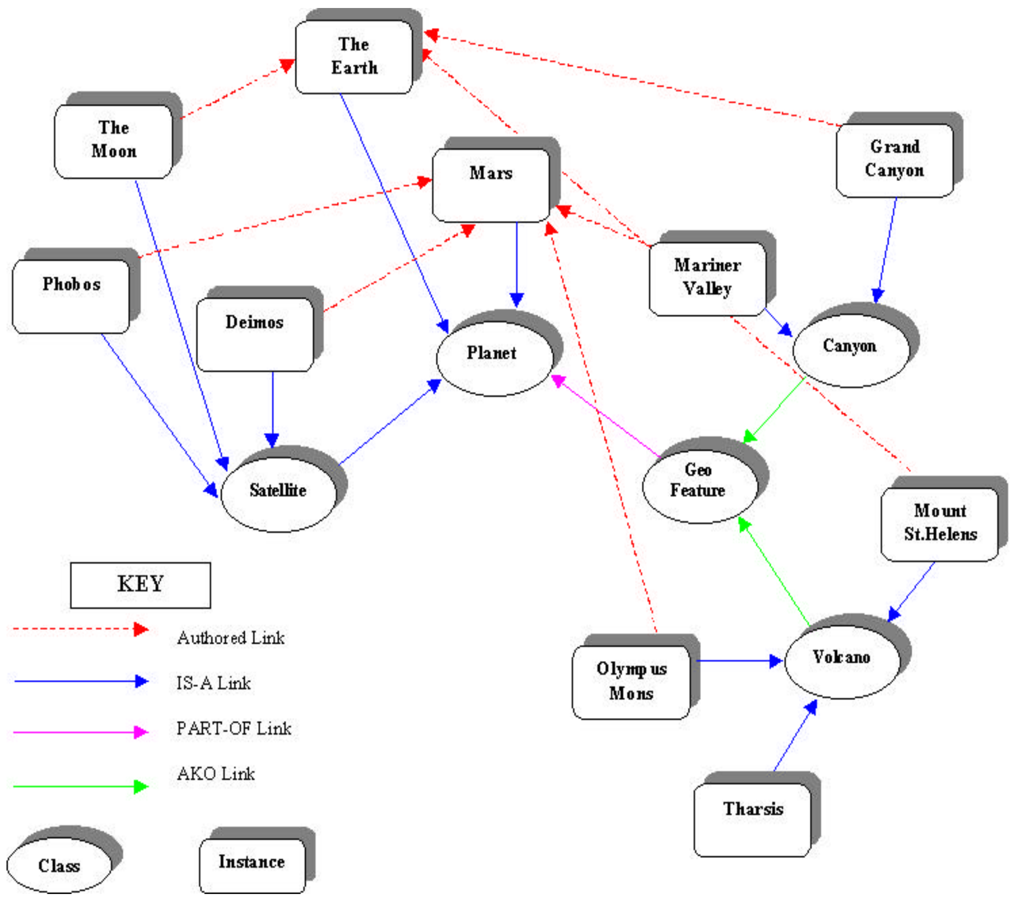


**Figure 4.4.A A Solar System Sub-Domain**

### 4.4.1 The Operation of the Student Experience Record

Information recorded within the SER is used to alter the presentation of links that have been generated beforehand by the automatic linking algorithm. For example, using figure 4.4.A, if the current node was "Olympus Mons", then the "Direct Links" box would contain a link to "Mars", the "Hierarchical Links" box would contain the link "Volcano" and possibly "Geo-Feature" and "Planet", depending upon the student level. The "Related Material" box can contain potentially every other link in the domain, although in practice this is reduced in accordance to the student level (figure 4.4.1.A). The links in this box are ordered according to their relevance, or weight, as generated by the automatic linking algorithm. However, it is often the case that several links have the same relevance even though they are indirectly connected by very different routes through the semantic domain. It is sensible, therefore, to use these different routes through the semantic domain to make a further distinction about which links are the most relevant and hence alter their presentation to the student in order to reflect this, since if this is not done then the student is likely to be offered material at the same level of relevance that is related in totally different ways to the current node. For example, the "Related Material" links are formed from the following:

Current Node : Olympus Mons

Tharsis, same parent "Mars" and "Volcano"

Mariner Valley, same parent "Mars" and "Volcano - Geo-Feature - Canyon"

Phobos, same parent "Mars" and "Volcano - Geo-Feature - Planet - Satellite"

Deimos, same parent "Mars" and "Volcano - Geo-Feature - Planet - Satellite"

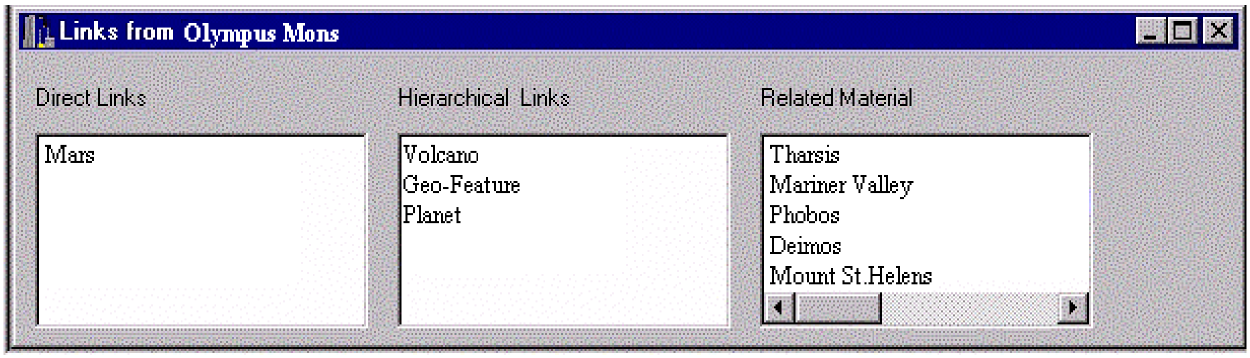
Mount St. Helens, "Volcano"

Grand Canyon, "Volcano - Geo-Feature - Canyon"

The Earth ," Volcano - Geo-Feature - Planet"

The Moon, "Volcano - Geo-Feature - Planet - Satellite"

Note that in this case there are two possible relationships between the current node and the other nodes, namely via the intermediate class nodes and via a direct authored link (in this case the parent node "Mars"). Further, the links from "Olympus Mons" to "The Earth" and "Olympus Mons" and "Grand-Canyon" have the same relevance, since they have the same size path through the semantic network. It may be the case, however, that the student is interested in geographical features of the solar-system, in which case it would be better to ignore the relationships through the authored links and therefore present the link to "Mount St. Helens" ahead of the links to "Phobos and Deimos", since Mount St. Helens and Olympus Mons are both volcanoes. It would also be better to make a distinction between "Grand Canyon" and "The Earth", and present "Grand Canyon" in preference, since it shares a strong relationship with Olympus Mons in that they are both geographical features (figure 4.4.1.B). The result of this process is that the student is encouraged to seek more detailed information. This is a general criticism of agents like the SER, in that they tend to have a “snow-balling” effect on the student, since once they have shown an interest in something they are then offered more material on it and are therefore more likely to, or even forced, to select this more detailed information (Armstrong et al 1995). However, this is countered in the Hypernet approach by the use of the student’s content-less browsing pattern, which may indicate that they are more interested in general information.

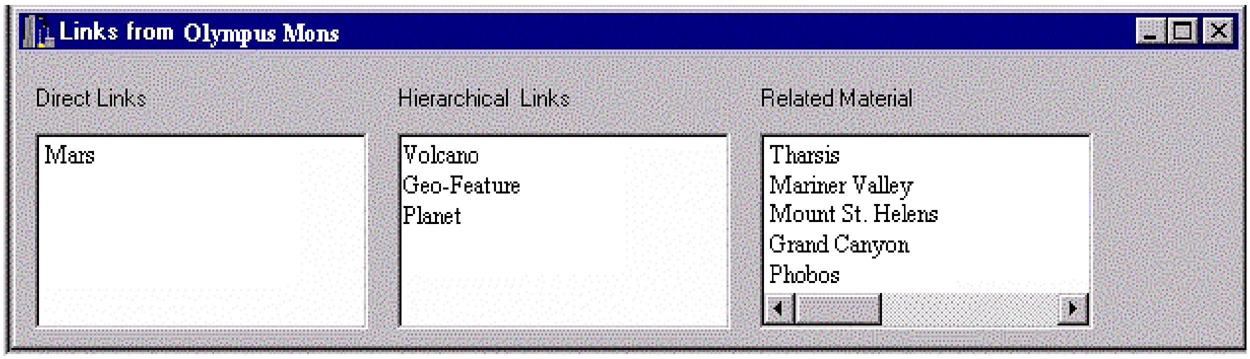

Links from Olympus Mons

Direct Links: Mars

Hierarchical Links: Volcano, Geo-Feature, Planet

Related Material: Tharsis, Mariner Valley, Phobos, Deimos, Mount St.Helens

**Figure 4.4.1.A Automatic Links (not tailored for an individual student)**


Links from Olympus Mons

Direct Links: Mars

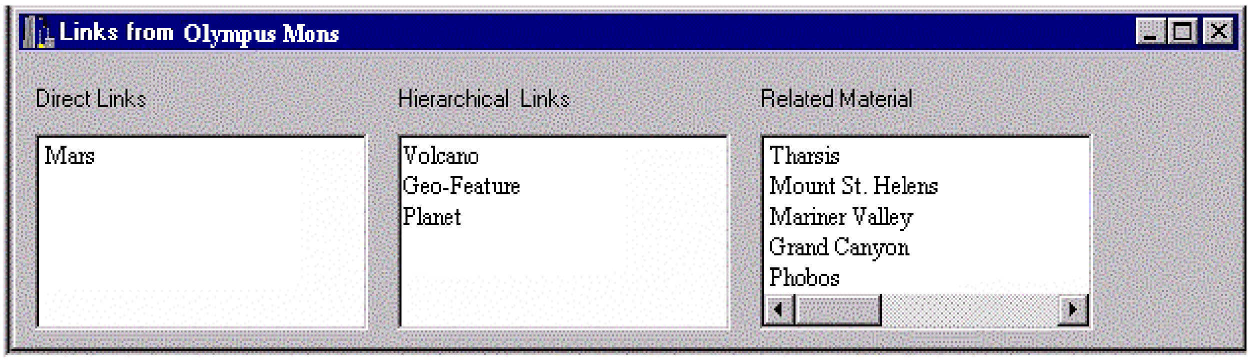
Hierarchical Links: Volcano, Geo-Feature, Planet

Related Material: Tharsis, Mariner Valley, Mount St. Helens, Grand Canyon, Phobos

**Figure 4.4.1.B Dynamic Links towards Geo-Features**

The SER serves the purpose of providing this interest information. It may be that the student's content-based browsing involves an even selection of class nodes, thereby demonstrating a general interest. On the other hand, it may be that the student tends to browse a particular area of the domain; for example, by repeatedly visiting nodes connected to "Geo-Feature" demonstrates a more specific interest. This content-based browsing information can then be used to tailor the links more closely to the student's interests. The difficulty is determining whether the student is interested in specific information or is more generally browsing the domain. This could be achieved by a combination of content-less browsing and content-based browsing, in that content-less browsing patterns have been identified as offering the potential to discern this information (Canter et al 1985). Content-based browsing could also identify this by examining how a student browses the semantic network. In the absence of information regarding content-less browsing it would be necessary to use content-based browsing alone. The content-based approach to browsing has been utilised successfully by Micarelli and Sciarrone (1998).

The dynamic linking algorithm starts by assuming a general undeclared interest for the student. It therefore gives preference to links that have a strong relationship, both via the semantic network and via direct links. If, however, the student is interested in specific information, then this will force them to browse the class nodes in the semantic network to find the information that they are interested in. The SER then records this information as described previously. If a set of related class nodes has more visits than other class nodes then it is determined that the student is interested in these class nodes and the "Related Material" is altered to reflect this. Secondly, once a specific interest has been recognised then the link-types can be used to present specific information rather than general information. For example, the links from "Olympus Mons" to "The Earth" and from "Olympus Mons" and "Grand Canyon" have the same relevance factor (as calculated by the automatic linking algorithm); however, the link-types connecting them are different. "Grand Canyon" is connected by two "a-kind-of" links, whereas "The Earth" is connected by one "a-kind-of" link and one "part-of" link. The "a-kind-of" link is used to show the relationship between similar nodes and hence further detail or further examples. Further content-based browsing could demonstrate that the student has become very interested in volcanoes and so the "Mount St. Helens" link, could eventually rise above the "Mariner Valley" link (Figure 4.4.1.C).


Links from Olympus Mons

| Direct Links | Hierarchical Links | Related Material |
|---|---|---|
| Mars | Volcano | Tharsis |
|  | Geo-Feature | Mount St. Helens |
|  | Planet | Mariner Valley |
|  |  | Grand Canyon |
|  |  | Phobos |

**Figure 4.4.1.C Dynamic Links (Volcano - Geo-feature)**

## 4.5 The Tutorial Supervisor

The TS is a neural network and is used to grade the student into an ability level in relation to the student's ability to complete tutorials; this provides several distinct advantages. Research by Chiu, Norcio and Petrucci (1991) details the difference between a neural system and a symbolic AI system for generating university timetables. Their results demonstrate that neural systems are orders of magnitude quicker to produce and, for the time-tabling problem at least, produce better results than the symbolic AI counterpart, since this problem involves the modelling of fuzzy data, namely student behaviour. It is generally accepted that neural systems are easy to build, if not necessarily easy to train (Masters 1993). In terms of this project it was desirable to investigate neural networks since they offer potential benefits over their symbolic counterparts, most notably their ability to adapt without human intervention, which would be useful in a tutoring system that is applied to a variety of domains and thus there is a potential for the modelling of students not to be consistent. A second reason for employing connectionist technology is because the system is attempting to map fuzzy (incomplete or changeable) information, namely the student's current ability onto a question's difficulty. This is discussed further in chapter 8, when the TS is discussed in detail.

## 4.6 The Browsing Pattern Recogniser and Fuzzy Rule Base

The BPR is a neural network device for recognising the content-less browsing patterns a student makes whilst navigating the hypermedia. The FRB is a fuzzy logic processor and is used to link the student's ability level (as defined by the TS) and tasks (as provided by the tutorial nodes) to the browsing patterns provided by the BPR. The BPR and FRB together form a research system that offers the potential to link browsing patterns to classes of student and the types of task that they use the hypermedia for.

The BPR is designed to identify movement patterns employed by the student as they move through the hypermedia domain. Once these patterns are identified, they are linked to student ability and tasks. If the ability of the student differs from the output of the TS then the FRB is altered in such a way so as to make the next output more likely to produce the same output as the TS, therefore the FRB is trained by the TS (like a neural network) to link browsing patterns to types of student. This is similar to connectionist methodology in that the system is trained by presenting data to it although, unlike connectionism, it may also be directly programmed. This is a complex issue and is dealt with in detail when the FRB is discussed in chapter 7.

## 4.8 Discussion

The major issue that must be considered when developing an educational system is how much support a student needs and how much support the system can provide to the student. Elsom-Cook (1989) argues that it is necessary to ask and answer the former question continually as the student uses an educational system, so that the system can provide them with a changing degree of aid that matches the student's needs and experience. However, the above needs must be balanced with how much aid can be practically offered by an educational system. Research indicates that it is not practical to accurately gauge how much aid a student needs and to then provide them with the required aid (Woods and Warren 1995, Kinshuk and Patel 1997). The only options are therefore to offer no help at all or to offer as much help as can be practically given. The approach described in this thesis has been designed to explore issues pertaining to offering as much help as possible whilst balancing the need for a practical system, in terms of it being applicable to a multitude of domains without asking for prohibitive effort from the author. In order to achieve this the student model discussed above attempts to measure the student's ability with tutorials and determine their interests by examining their content-based browsing with the semantic hypermedia.

The Hypernet approach attempts to measure the student's ability and interests in order to gain an indication of their overall information seeking strategy and to do this in a way that can be accomplished regardless of the domain being studied. This raises the questions of whether the Hypernet approach achieves this and if so is it useful to do so? Hypernet has one major advantage over a standard hypermedia paradigm, in that it is able to engage the student in a dialogue, since it is an approach to educational hypermedia. This enables the student's ability to be gauged since the students can be provided with tutorial questions and tasks and therefore their performance with these tasks and questions can be measured. However, determining whether collecting content-based and content-less browsing information is useful for an educational hypermedia system is useful remains the subject of further research, although previous research has indicated that this is a promising area (Canter et al 1985).

The Hypernet approach cannot be said to be a discovery learning approach in the same sense as an ideal hypermedia approach. This is because tasks are imposed upon the student and if they are to complete these tasks then they cannot be said to be carrying out their own discovery learning. However, this is countered to some extent in that tutorials are designed by the author to be targeted at various ability levels and there is no reason why higher abilities of students should have tutorials. Consequently Hypernet can act as a constrained hypermedia-based tutoring system to a novice student and as a more traditional hypermedia system to an expert student. This sliding-scale approach is in line with Elsom-Cook's (1989) pedagogy.

## 4.9 Conclusion

The novel student model conceived and designed for this research project is based on domain independent metrics. The advantage is that this student model has the potential to be employed for a variety of domains with little or no modification, whereas a traditional student model would usually require reengineering for a different domain. The disadvantage is that the metrics employed are inevitably less suited to a given domain. It is argued that the student model has the potential to aid the student considerably, in comparison to no student model at all. The student model contains several sub-systems, namely the Tutorial Supervisor (TS), the Browsing Pattern Recogniser (BPR), the Fuzzy Rule Base (FRB) and the Student Experience Record (SER). The TS grades the student into an ability level according to their ability to answer questions provided by tutorial nodes. The BPR is used to identify the student's content-less browsing patterns through the hypermedia. The FRB then links this information so that it may become possible to extract information about a student by examining their movement strategies. The SER records information regarding the types of nodes that the student has visited so that the hypermedia can be tailored towards them.

The following chapters are concerned with the individual components of the student model, namely the BPR, FRB and TS. The next chapter serves as an introduction and explanation of hypermedia content-less browsing patterns and the possible uses to which they may be put.

# Chapter 5

# Hypermedia Browsing Strategies

## 5.1 Introduction

This chapter is concerned with the patterns made by the users of hypermedia systems as they navigate from one node to another. It is hypothesised within this chapter that these patterns may be used to aid a student model in the extraction of information about a particular student, which may then be used to aid that student. These patterns are termed “content-less browsing patterns” here to distinguish them from browsing the hierarchy or semantic structure of the domain (content-based browsing). For example, a content-based browsing pattern involves a system extracting information about the student moving from one node to another whilst taking into account what it is that each node represents, e.g. moving from “Planet” nodes to “Satellite” nodes. By contrast, content-less navigation is concerned with patterns that a user of an information structure makes as they move from one node to another regardless of what it is that the node represents. Note that the distinction between content-less and content-based browsing patterns exists only when a system is attempting to recognise them. A student will always navigate the hypermedia by using a content-based approach, since nodes represent information and the student navigates by using this information. However, recognising the content-based pattern has the drawback that the system must have some knowledge about what the information is inside the nodes, this renders the approach domain-dependent, the drawbacks of which were discussed in chapter 4.

Content-less browsing patterns, on the other hand, are not domain-dependent and when viewed globally have been identified as being of potential use to extract information about the person who has formed them. Research has identified (Canter et al 1985, Machionini 1989, Micarelli and Sciarrone 1996, 1998) that certain tasks tend to induce the user to make distinct patterns within the hypermedia (or other information structure). These content-less browsing patterns (hereafter referred to simply as browsing patterns) may vary according to the type of the student that is making them, since one student may have different needs or learning styles from another (Mumford 1995, Robotham 1995) and is therefore likely to use the hypermedia differently. Indeed Canter et al (1985) state "it should not be expected that novice and expert users exhibit the same behaviour or make the same demands on any particular system". These patterns are therefore of potential use in determining what type of student is using the hypermedia and what they are using it for. A method for extracting this information is a key element of this research project. Canter et al (1985) indicate that these patterns may be used to "capture some of the psychologically significant aspects of the movements of users within interactive data-bases". Such information may for example be that the user is worried about becoming lost, or is wandering aimlessly away from their goal (Canter et al 1985), or is interested in general information only.

This chapter discusses the kinds of patterns that have been identified by previous research and how they are made up from lower level browsing constructs or primitives. Note the distinction between browsing *patterns* and browsing *strategies*. A browsing pattern is a content-less pattern etched onto the hypermedia as a user moves through it. Chapter 6 discusses a system for recognising these browsing patterns A browsing strategy is a method employed by the student in order to fulfil their task whilst using a hypermedia system. A browsing strategy is therefore a browsing pattern with some extra information indicating what the pattern is being used for. Chapter 7 discusses a system designed to convert the browsing pattern into a browsing strategy, by combining the browsing pattern with information collected elsewhere in the system (from tutorial types and their results).

Having defined in more detail what content-less browsing patterns are, this chapter moves on to discuss some experiments concerning how browsing patterns are used by various learners. The purpose of the browsing experiments is to provide some empirical evidence as to whether different learners use hypermedia in different ways and whether relationships exist between similar types of learners.

## 5.2 Browsing Patterns and Browsing Strategies

The following is a discussion of browsing patterns and browsing strategies employed by students and users of hypermedia systems in general. The aim is to derive information about the student's intentions and state of knowledge, without recourse to the complex and domain specific structures used for traditional Intelligent Tutoring Systems. This may involve, for example, identifying that the student is "lost in hyperspace" and requires assistance or that they are searching for a specific item of information and so do not require links that offer a broader scope of information. The need for deriving information about the student is driven by the complexities of hypermedia navigation, discussed in chapter 2. It was argued in chapter 2 that information concerning what the student requires with a hypermedia domain could be used to simplify the domain, for that particular student. Such a process would render the hypermedia domain more useful for a given individual. The tailoring of the hypermedia domain could range from offering links that are consistent with the student's current interests, to potentially encouraging particular browsing strategies that are known to be beneficial. Knowing whether a particular browsing strategy is useful or not for a given individual is currently unknown, however, such information may be collected automatically, a process that is discussed further in chapters 6 and 7.

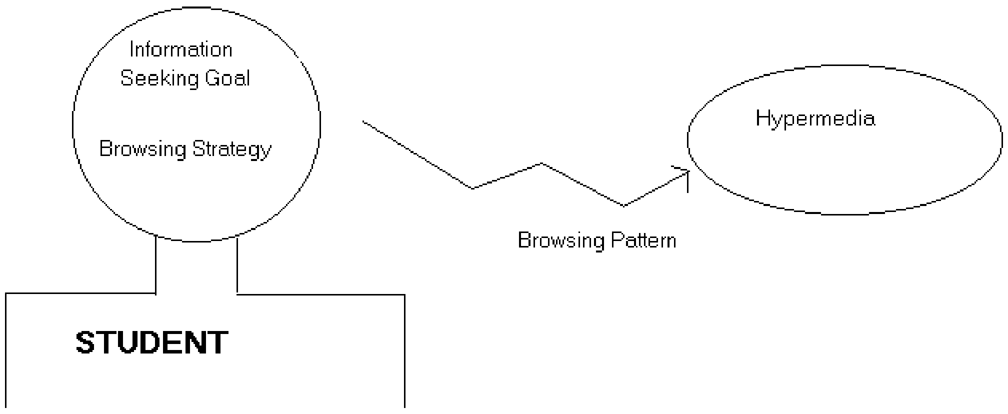


**Figure 5.2.A Browsing Strategies and Browsing Patterns**

Hypermedia browsing strategies, as outlined by Canter et al (1985), are characteristic methods for moving through a network of information nodes. Canter et al are referring to browsing strategies since they are referring to a user undertaking a particular information seeking strategy, as illustrated in figure 5.2.A as an operation internal to the student and being derived from a student's information seeking goal or needs. For example, a student searching for a specific piece of information (the information seeking strategy) may induce a "seeking" browsing strategy. Therefore the student may "quickly skip from one node to the next, seldom returning to a previously visited node, seldom spending more than a few seconds at each node and ignoring the associative trails that lead off in varying directions" (Canter et al (1985)'s description of scanning). It is this movement through the hypermedia that can be determined by the hypermedia system and is termed a "browsing pattern". The browsing pattern may then be used as a starting point to identifying the student's browsing strategy.

Canter et al (1985) have formally identified five browsing strategies, shown in figure 5.2.B.

| | |
|---|---|
| 1) Scanning | - covering a large area without depth. |
| 2) Browsing | - following a path until a goal is achieved. |
| 3) Searching | - striving to find an explicit goal. |
| 4) Exploring | - finding out the extent of the information given. |
| 5) Wandering | - purposeless and unstructured "globe trotting". |

Further, Canter et al (1985), suggest several lower level browsing structures as follows in figure 5.2.B.

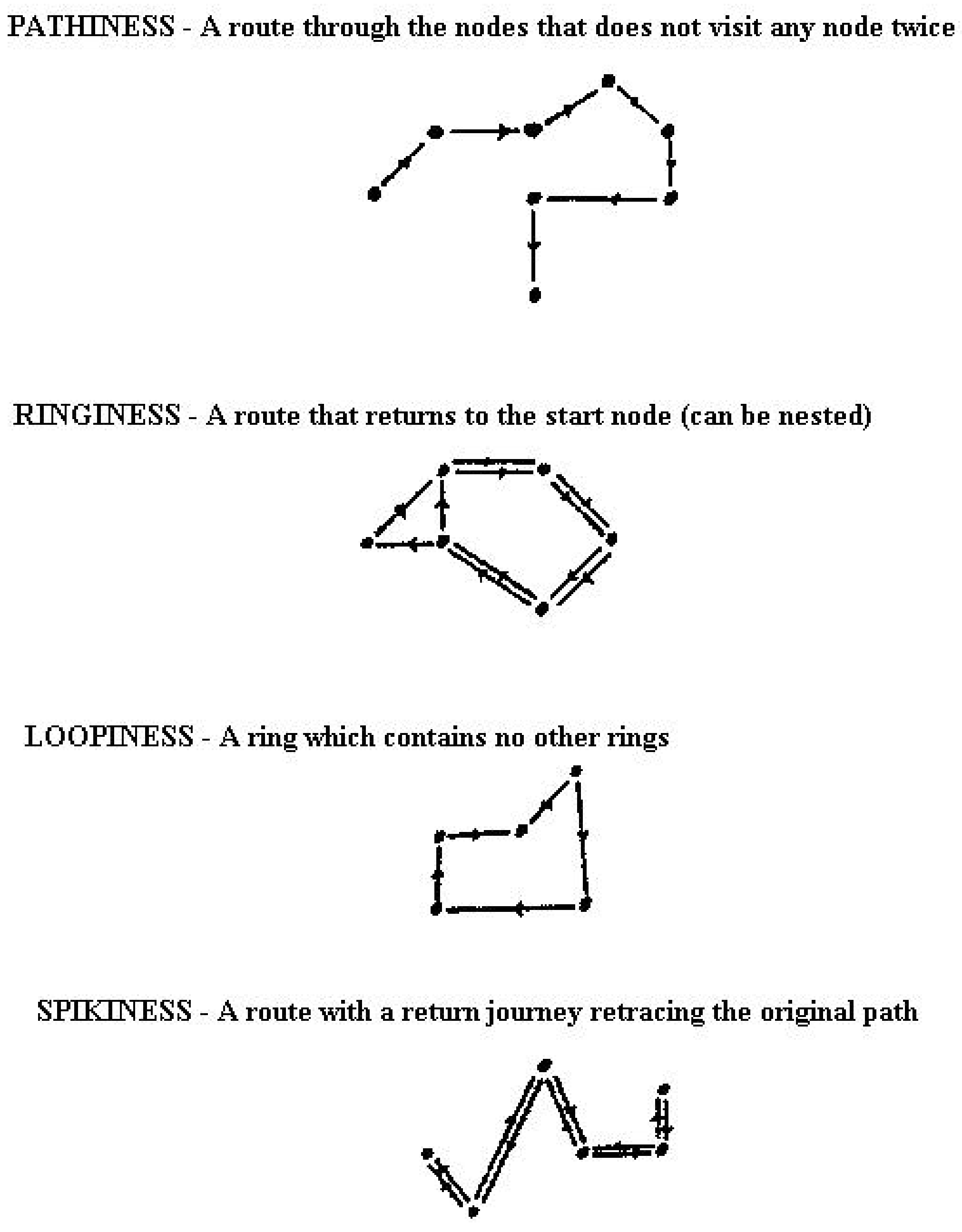
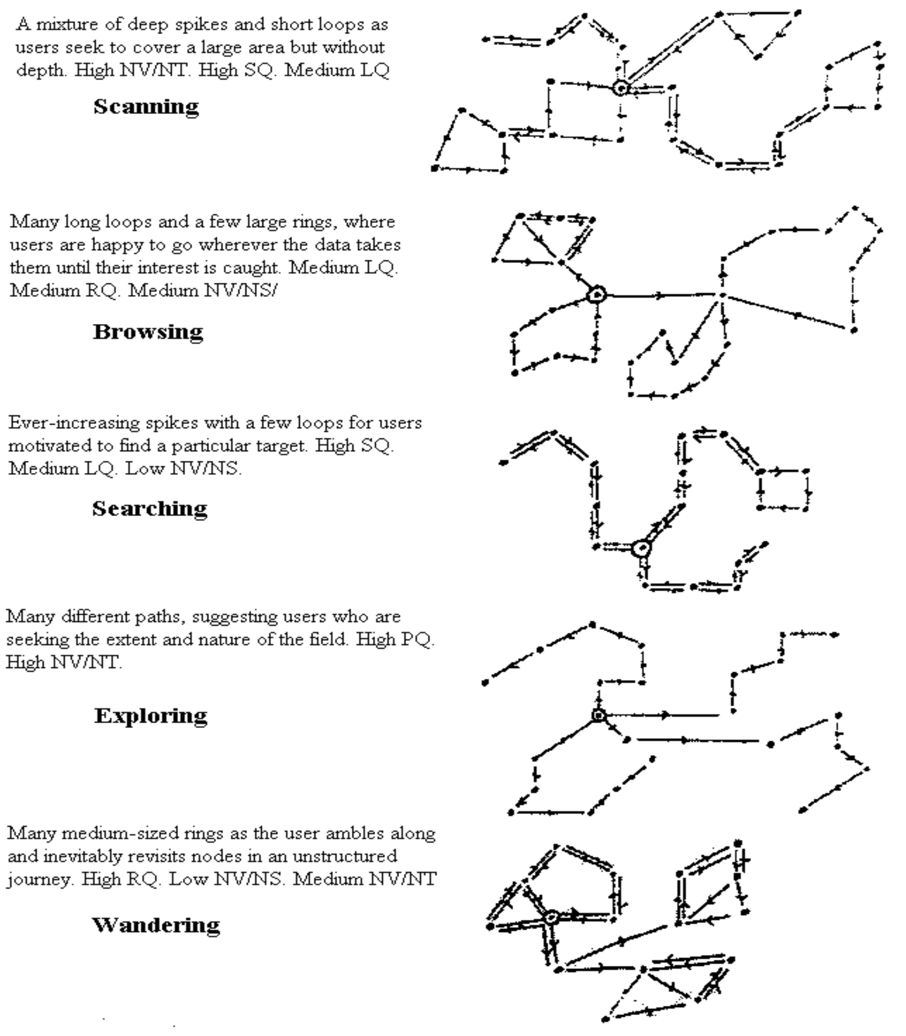

PATHINESS - A route through the nodes that does not visit any node twice
RINGINESS - A route that returns to the start node (can be nested)
LOOPINESS - A ring which contains no other rings
SPIKINESS - A route with a return journey retracing the original path

**Figure 5.2.B Browsing Constructs, from Canter et al (1985)**

These low-level browsing constructs (spike, ring, path, loop) connect together to form higher level browsing patterns (searching, scanning etc.). These browsing patterns may then form part of a student's overall browsing strategy. This presents a problem for automatic recognition, since it is not readily apparent where one construct ends and another construct begins. It could also be argued that constructs merge into each other. For example, a large loop can also be thought of as a path, until the point where it crosses a previously visited node. Further, the diagrammatic representations above and in Canter et al (1985) are not accurate representations of the actual constructs and browsing patterns. This is because the diagrams are drawn in two-dimensional space, whereas the constructs themselves do not exist in such a space. For example, the ring diagram shows a trail arcing over upon itself until it crosses a previously crossed node. In reality, it is only because one node has been visited twice that this construct becomes a loop. Although Canter et al (1985) do not note this deficiency in representation explicitly they do provide a mechanism for overcoming the problem. They do not attempt to identify constructs individually, instead they describe a user's browsing pattern as being comprised of several indices, namely the path, spike, ring and loop indices, identified on figure 5.2.C as Path PQ, Ring RQ, Loop LQ and Spike SQ.  However, they admit that their indices present "calculation problems" since indices can overlap and become fuzzy in nature and there is therefore a requirement for "more complex and subtle algorithms". It can be seen then that the low-level browsing constructs merge into one another to form the complex browsing patterns. The task then, is to identify these constructs and relate them to the overall browsing pattern. The browsing pattern may then be used to infer information about the student's overall browsing strategy.

A further important characteristic when identifying browsing patterns is the time spent at each node. It may be that the student is using the node as a navigational cue only, i.e. they are using it as a stepping stone to another node. Conversely, the student may be interested in the contents of the node itself and may examine its contents thoroughly. It may be that the student is using the node as a navigational cue but is unable to navigate to the next node by examining the links presented. It may be the case that the student must examine the contents of the node in order to determine their next destination. There may therefore be several types of spike, path, loop and ring, in that one version of the construct may involve using the node as a navigational cue only and another version may involve using the node as an information resource, or more likely a combination of the two. A possible solution to this complex recognition task is discussed in chapter 6.


A mixture of deep spikes and short loops as users seek to cover a large area but without depth. High NV/NT. High SQ. Medium LQ

**Scanning**

Many long loops and a few large rings, where users are happy to go wherever the data takes them until their interest is caught. Medium LQ. Medium RQ. Medium NV/NS/

**Browsing**

Ever-increasing spikes with a few loops for users motivated to find a particular target. High SQ. Medium LQ. Low NV/NS.

**Searching**

Many different paths, suggesting users who are seeking the extent and nature of the field. High PQ. High NV/NT.

**Exploring**

Many medium-sized rings as the user ambles along and inevitably revisits nodes in an unstructured journey. High RQ. Low NV/NS. Medium NV/NT

**Wandering**

**Figure 5.2.C Browsing Strategies, from Canter et al (1985)**

Figure 5.2.C shows how the browsing constructs identified in figure 5.2.B combine to form more complex patterns, which occur as the student employs a browsing strategy. These patterns were found in experiments performed by Canter et al (1985). They note that several factors affect the formation of these patterns, namely the ability of the user, the task that they are performing and the interface style of the system. A key element of the current research project is to formulate the "complex and subtle algorithms" that Canter et al call for, so that these patterns can be distinguished as accurately as possible, given their fuzzy nature, and used to identify the "psychologically significant aspects" of the user of the system.

It has been suggested by Robotham (1995) and Honey and Mumford (1992) that different learners may approach the same task in different ways and that forcing a learner to approach a task in one way can result in problems such as the learner losing motivation or failing to benefit from the learning experience. However, Robotham (1995) also warns against allowing the learner to use one style only and suggests that "the learner will be forced to increase [their] learning versatility through the required use of unfamiliar learning approaches." There is much debate, therefore, about styles of learning and it is possible that a student's learning style is related to their browsing strategy whilst using hypermedia. This is further to previous work by Canter et al (1985) and Micarelli and Sciarrone (1996, 1998) that strongly suggests that browsing patterns relate to the types of task that the hypermedia (or other information structure) is being used for. Browsing strategies therefore offer the potential to investigate elements of a learner's learning style, which may then be used to investigate the issues discussed by Robotham (1995).

## 5.3 Automatic Recognition

The above discussion indicates that different learners tend to use different browsing strategies when using hypermedia (and hence form different browsing patterns) depending upon the task that they wish to complete. It is therefore of potential benefit to automatically recognise these browsing patterns so that some information can be gained about the user without the need to resort to direct dialogues. Again this point is reinforced by Canter et al (1985) who argue that recognising such information may provide important psychological information about the student.

With many browsing systems it is not always possible to involve the student/user in a direct dialogue, (for example the World Wide Web (WWW)). Research into aids for users of the WWW usually centres upon extracting the user's interests by examining the frequency of keywords in documents that they have selected (Armstrong et al 1995, Cartledge and Pitkow 1995). However, as Cartledge and Pitkow (1995) point out, such a method cannot take into account the possible dynamic behaviour of the user, in terms of the potential for the user to change their goals and strategy at any instant. Similarly, Armstrong et al (1995) note that their browsing support system "Web Watcher" is restricted to being an adviser only because it "might not perfectly understand the user's information seeking goal". These drawbacks are endemic to the approach of identifying a user's interests based upon keywords alone. Browsing patterns, on the other hand, offer the potential to provide extra information about the user in terms of their overall information seeking goal and browsing strategy. For example, it may be possible to determine that a user is only generally interested in the information presented and is therefore not necessarily searching for information described by a few keywords. There is therefore a case for producing a system for identifying browsing patterns and ascribing meaning to them and thus identifying the student’s overall browsing strategy.

Automatic recognition of browsing patterns, however, presents several problems and their interpretation several more. Firstly, recognising the high level patterns is made complex because a user may adopt more than one pattern for a particular session and may appear to be utilising different patterns depending upon how many movements are examined (Bates 1989). For example, a student may be involved in an overall “searching” strategy (and thus invoke a searching pattern), i.e. they are looking for a specific piece of information. However, they may become interested in a piece of information that they have discovered whilst engaged upon their “searching”. The student may then make a short deviation from their current “searching” strategy and "browse" around the area that has caught their interest, before returning to their original strategy. How then is this to be interpreted? Are there three distinct strategies? Are there two? Or is this to be interpreted as one complex strategy? If a symbolic artificial intelligence (SAI) approach to pattern recognition were to be employed then the answer to such questions must be decided upon before a student uses the system. This is because the SAI designer must decide beforehand the construction of every browsing pattern (that is invoked by a particular strategy) that the system must recognise and encode appropriate production rules.

A further problem is that a student may use a particular strategy at one instant and then swap to another in response to some stimulus from the hypermedia domain itself. The rate of sampling, i.e. how many movements the user makes before the system determines the browsing pattern, can therefore result in the system providing differing results. For example, a user may be involved in an overall browsing strategy that is built up of several smaller browsing strategies, in that the student's overall goal may be served by the employment of several distinct strategies. It is difficult to say whether the student has engaged one overall browsing strategy or several strategies. It is therefore difficult to determine exactly what the student is doing at one particular instant in time.

The result of these problems is that different results are obtained from a number of recognition methods. Figure 5.3.A shows two browsing samples, the first shows the student using a spike; the second shows the continuation of the student's pattern from the spike into a ring. Therefore, using the simplified example in figure 5.3.A, the system can make three judgements about the student's browsing behaviour; they are performing a spike, they are performing a ring, or they are producing a pattern composed of a spike and a ring. Which pattern is chosen depends upon when and how often the system evaluates the student's browsing behaviour. How then should the analysis be accomplished? The best method, in terms of accuracy, would be to examine a student's complete browsing pattern from the beginning to the end of a session and determine a general browsing pattern. This approach, however, renders the information obtained useless in terms of helping the student to use the system more efficiently, while they are using it. However, it may be useful in terms of coaching the student after they have used the system, or in terms of collecting general information about a population of students that may be useful for designing more efficient systems. To give real-time help, however, smaller sampling rates would be necessary, so that individual browsing patterns could be recognised (e.g. the student is currently searching). It is not however, a simple matter to decide an appropriate sampling rate, since as described above, different rates are likely to give different results.

As a result of this it is likely that a symbolic AI approach to recognising browsing patterns will be difficult, incomplete and inflexible. For example, it may be that the student is employing characteristics of several constructs, a “spikey ring” for example. This situation is further complicated because each construct may be being employed by the student to varying degrees. This situation renders the application of a rule-based system impractical. It may be possible to design a fuzzy rule-based system; however such a system will still be restricted by the design of the initial rules. The following chapter is concerned with a potential solution to these problems.

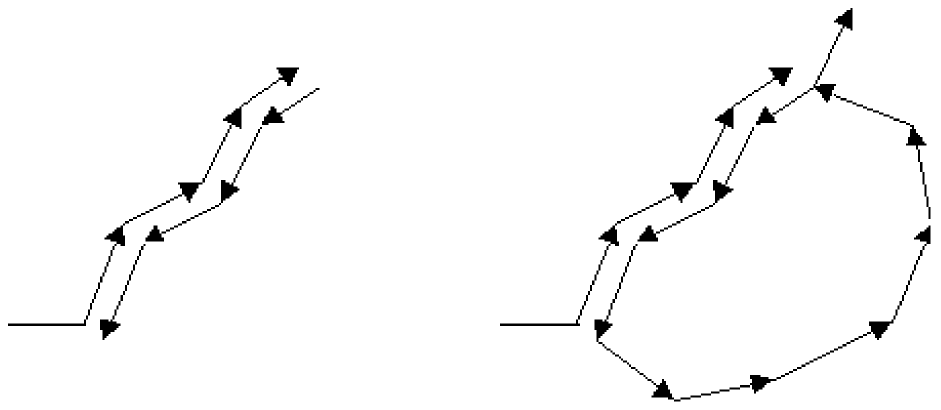


**Figure 5.3.A Different Sampling rates for a browsing pattern**

## 5.4 The Meaning of Browsing Patterns

Once a particular browsing pattern has been identified and some inference has been made to link it to a student’s browsing strategy then, it may be possible to tailor the presentation of information to a student in such a way so as to model their interests more closely. For example, if the student is generally interested in the domain then it may be useful to present them with a selection of different topics from the domain, whereas it would be better to offer more detailed information if they are looking for a specific item of information. Methods of link adaptation that could be used to aid the user of a hypermedia system were discussed in chapter 2 (c.f. section 2.4.2). Other possibilities for utilising browsing information may arise as part of further research into interface design.

## 5.5 Low-level Constructs to Browsing Pattern

Low-level constructs, e.g. spikes, paths, rings and loops, are combined together by the student user of the hypermedia system into browsing patterns, such as those shown in figure 5.2.C and potentially other browsing patterns that have not been previously identified. The BPR, described in chapter 6, constantly examines a student's movements through the hypermedia. The BPR is able to identify the points in the movements that uniquely identify particular constructs. The higher level browsing patterns, however, do not have discrete points with which to identify them, indeed it is extremely difficult to say where a browsing patterns starts and ends.

It is necessary however, to identify high-level browsing patterns since the low-level constructs do not provide enough information about the student, since they are only formed in a relatively short time period and encompass a small amount of the domain (in comparison to an overall browsing pattern formed by the student to solve a particular information seeking problem). In order to recognise a browsing pattern, however, it is necessary to identify where the pattern begins and where it ends. This is possible in Hypernet because it can identify the start of a browsing pattern. The start point can be defined as the point at which time the student leaves a tutorial node, i.e. they have been given a task to complete by the tutoring system. They then engage upon a browsing strategy, which induces a browsing pattern, in order to complete the requirements of the tutorial node. Once they have satisfied the requirements, the browsing pattern has concluded. Once the high-level browsing pattern has been identified, in terms of its start and end points, then the system may process it. It is stressed that Hypernet is a research system and is used here to extract these browsing patterns (and record them) so that they can be used afterwards in other systems (such as the WWW). It is therefore not necessary to always identify the start and end points of a browsing pattern, once Hypernet has recorded a representative sample the start and end points are patterns themselves within the sample. This situation is analogous to recognising the handwriting of an unknown script, in that enough examples should make the script clear. This is similar to the Case-Based Reasoning approach used by Micarelli and Sciarrone (1998) to identify content-based browsing patterns, in that recognition of the current case is based upon similar cases encountered previously. A possible criticism of the Hypernet approach is that browsing patterns recorded as part of Hypernet may be specific to Hypernet and may not exist in other systems. As McAleese (1993) points out certain browsing strategies are facilitated by certain interface styles (e.g. graphical browsers and text-based browsers), it therefore follows that the interface style (in this case Hypernet's interface) has a direct influence upon the browsing pattern produced by the student's browsing strategy. However, it is likely to be the case that there is a core of patterns that are common to all styles and the recognition system, described in the following two chapters, is not dependent upon the interface style. The recognition system may therefore record general information as part of Hypernet and may then adapt to individual browsing systems that it may subsequently be employed within.

## 5.6 Browsing Experiment

### 5.6.1 Introduction

The browsing experiment described and discussed here was designed to test the hypothesis that for each type of learner it is possible to identify distinctive browsing strategies and browsing patterns. If this hypothesis was proven to be the correct then there is a strong case for developing a method to automatically recognise these browsing patterns and hence determine the student's browsing strategy. Once the student's browsing strategy has been identified the student may then be aided in their particular strategy by manipulation of the links presented to them. Previous research by Canter et al (1985) has provided some evidence that users of databases produce these characteristic movement patterns, described above. Further, research by Micarelli and Sciarrone (1996, 1998) demonstrates that link manipulation based upon a user's content-based browsing patterns can aid a user of hypermedia. The experiment outlined below was designed to expand upon Canter et al's (1985) experiments by bringing them into the hypermedia domain and to incorporate some temporal information. Temporal information has been cited above (c.f. section 5.2) as being of potential benefit, since it may allow a determination to be made as to whether a node is being used for its contents or is being used solely as a navigational cue.

### 5.6.2 Method

The Browsing Experiment was conducted as a distance experiment. A prototype Hypernet browser (discussed in Chapter 3) with an Astronomy domain was dispatched on a CD-ROM to 11 volunteers contacted by a BITNET email group (ASTRO-L).

#### 5.6.2.1 Design

The independent variable (IV) is the element that is being manipulated in the experiment. For the Browsing Experiment the IV is the subject, since each subject may have a different self-perceived ability with the domain.

The subjects were asked to grade themselves into one of the following groups, with regard to the Astronomy domain:

No knowledge

Very little knowledge

Basic knowledge

Advanced knowledge

Expert knowledge

The dependant variable (DV) is the element that is to be measured. In this case it is the browsing patterns used by the subject as they navigate the Astronomy domain. The browsing pattern is recorded by the Hypernet interface and stored in a file that the subject was then required to email back at the end of the experiment. Once browsing patterns were received from all the subjects then the browsing patterns were analysed. The analysis process is discussed in the results section below.

Other variables under the control of the experiment relate to the Hypernet interface, such as how the links are presented and how many links are presented. For this experiment the Hypernet interface was set to behave in an identical fashion, regardless of the IV (the subject’s perceived ability with the domain). This is discussed further in the apparatus section.

### 5.6.2.2 Apparatus

The apparatus of the browsing experiment was the prototype Hypernet interface, the listing of which is presented in Appendix C, with an astronomy domain and functioning facilities for dynamic link adaptation, described in Chapters 3 and 4. The Astronomy domain was a small hypermedia domain, in the order of 150 nodes, hierarchically structured, with many links generated automatically and potentially available to the student.

In order not to introduce other variables into the experiment, and therefore potentially confound the results, the number of links that were offered to each subject was kept constant, regardless of the subject’s self-perceived ability grading. For the purpose of the experiment, the maximum number of links offered to a subject, in either the hierarchical or related material link boxes (described in Chapter 3), was set to 8. The number of links was chosen to offer enough links so that the AI facilities for link adaptation had some effect, but not enough to confuse the subjects and/or possibly the results of the experiment. The direct link box was set to display all links that were created by the author.

The Hypernet interface was able to record a subject’s complete browsing session. It achieved this by recording, when a node was visited, when it was last visited and how many times it had been previously visited. Hypernet also recorded special events, such as whether the subject clicked the “Back” or “Forward” button on the browser. This information enabled a full replay of a subject’s browsing session, although it was not possible to determine how much or little the subject read at each node and whether they understood individual concepts presented at each node.

Browsing patterns adopted by the subject were analysed using the Browsing Pattern Recogniser trained using the Virtual Student Program (discussed in detail in the chapter 6) and by manually examining the history file returned by each subject. The data from the BPR is available in Appendix D. The history file contains the following fields:

| ITEM | NODE | C | BPR | V | CLICK | TIME | Comment |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

"Item" is used to identify individual lines and is used to refer to the table within the body of Appendix D. "Node" is the name of the node that the subject has navigated to. "C" is the count value that shows how many times a node has been visited minus 1 (since the initial visit is recorded by the visited flag). "V" is the visited flag, which is 0 for a node that has not previously been visited or 1 otherwise. The BPR value is used as the real value input to the BPR and is not easy for the experimenter to interpret, it is for this reason that "C" and "V" values are recorded. However, it can be judgmental and difficult to interpret these values, so the output of the BPR was used as a more authoritative guide to the subject's overall content-less browsing pattern, since the BPR has been trained to perform this potentially difficult recognition task (discussed further in Chapter 6). The BPR value decreases by 0.01 each time some other node is visited. For experimentation the amount this value decreases can be changed after the browsing pattern has been recorded. The "CLICK" field indicates whether the subject navigated using a link from a link box or used the "FORWARD" or "BACK" button on the browser. "Time" records how long the subject spent at each node, thus giving an indication as to whether they were reading the node contents or using the node as a navigational cue. The “Comment” field has some general comments that were annotated by the author to the browsing data after they had been sent back by the subjects. These comments are not intended to be a rigorous and definitive description of the subject’s browsing; instead they are judgmental comments annotated to the data during interpretation.

The Hypernet browser generated a summary file for each subject (shown along with the browsing data in Appendix D). Contained within the file is a figure that represents the total number of nodes visited divided by the number of new nodes that were visited. A high value for total/new demonstrates that the subject made many navigations but visited a relatively small proportion of the domain. The number of visits to an instance of a class was also recorded in the summary file. For example, if a subject visited “Earth” and “Mars”, which are connected as instances of “Planet”, then the student’s “Planet” count would be increased by 2. This class data enables a simple determination as to which areas of the domain the subject did or did not visit. Such a measure may be useful for determining whether areas of the domain are difficult to navigate to and for determining the student’s interest in the domain. For example:

```
Path 0.5
Ring 0.7
Spike 0.4
Loop 0.1
Total nodes visited : 119
New nodes visited : 65
total/new = 1.830769
Classes visited : 3
Planet visited 28 times
Satellite visited 24 times
Volcano visited 12 times
```

The path, ring, spike and loop values are the outputs of the BPR after it was presented with the subject’s browsing pattern, which was done when the subject’s browsing file had been received.

#### 5.6.2.3 Procedure

The experimental procedure involved contacting subjects by placing a message on the Astro-L BITNET email group and then dispatching the Hypernet interface and Astronomy domain on CD-ROM along with instructions as to how to install Hypernet and what to do for the experiment.

Once Hypernet was installed on the subject’s computer they were then asked to grade themselves into one of the abilities (with astronomy and the solar system) described above. They were then asked to complete a set task and finally to browse the domain generally for as long as they wished. The set task required the subject to identify which is the biggest satellite of the solar system. The subjects were asked to browse the domain until they could answer this question to their satisfaction (by providing Ganymede, Titan or both), at which point they were requested to indicate their answer in an email along with the browsing pattern file recorded by Hypernet. A second element of the experiment asked the subject to browse the domain for as long as possible so that larger browsing patterns could be collected.

### 5.6.3 Results

The results described below are from a sample of 11 volunteers. These were separated into three categories: 6 novices, 2 intermediate and 3 expert, since there were not enough subjects to warrant the use of all the groupings shown in the method above.

#### 5.6.3.1 Subject 1

Subject 1 was self-assessed as an expert with the domain. Examination of their browsing patterns demonstrates that they solved the set task very quickly. Subject 1 probably knew the answer beforehand since they were able to locate it immediately via the structure of the domain, which they presumably also understood. Once the task was complete subject 1 examined most of the nodes within the domain. This was achieved by navigation of the hierarchical links as can be seen by the relatively high counts at the class nodes (shown in Appendix D).

The browsing pattern shows a high degree of rings with a mid value for paths and spikes and a low value for loops. It can be seen that the subject used class nodes or a major node (such as an instance of Planet with lots of authored links connected to it) as a means of navigating to instances of that class. This resulted in relatively short paths, as they returned to a class or major node before navigating to the next instance node of interest. Many rings were also produced as the subject circled their area of current interest until the nodes have been exhausted, at which point they returned to the class or major node and exited the current area in search of a new, unvisited area.

Subject 1's browsing appears to be quite structured and resulted in a coverage of most of the domain whilst not revisiting many instance nodes twice. In the case of Subject 1 the ratio of total navigations to total different nodes visited is relatively small. This is indicative of someone who is comfortable with the structure of the domain. For example (from the results table):

**Table 5.1 Extract from Subject 1's data from Appendix D**

| Item | Node | C | BPR | V | CLICK | Time |
|---|---|---|---|---|---|---|
| 35 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:03 |
| 36 | Canyon | 0 | 0 | 0 | LINK | 0:00:09 |
| 37 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:18 |
| 38 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:31 |
| 39 | Noctis | 0 | 0 | 0 | LINK | 0:00:31 |
| 40 | Canyon | 0 | 0.97 | 1 | LINK | 0:00:02 |
| 41 | Geographical Feature | 0 | 0.95 | 1 | LINK | 0:00:04 |
| 42 | Planet | 2 | 0.92 | 1 | LINK | 0:00:09 |
| 43 | Vulcan | 0 | 0 | 0 | LINK | 0:00:18 |

This shows a ring around the Geographical Feature class node. Subject 1 visited the sub-node Canyon and then every instance of Canyon, before returning to Canyon and Geographical feature. This pattern was fairly typical of Subject 1, indicating that their goal did not change whilst they examined the hypermedia.

### 5.6.3.2 Subject 2

Subject 2 was a self-assessed expert with the domain. Subject 2 progressed to answer question one straight away, indicating that they probably knew the answer beforehand. Subject 2 then progressed to answer question 2 via the hierarchical links. Interestingly, the approach of using hierarchical links was not used very much afterwards. Instead Subject 2 began to use the related material links (this information has not been specifically recorded but can be surmised as navigation takes place between two nodes of different classes). Generally, this strategy resulted in some large rings and longer paths, with few spikes. However, Subject 2 appeared to change their strategy around the Volcano nodes. It could be surmised that Subject 2 became interested in, or became less sure of the area of, Volcanoes and changed their strategy accordingly. The new strategy produced a high degree of spikes and less rings. The new strategy can be seen in the excerpt from the table below at lines148-176. The first part of the strategy, lines 148-161, corresponds to Subject 2 exhausting the Related Material links. Afterwards Subject 2 began to use hierarchical nodes to further their exploration and began to revisit some nodes. Subject 2 was sporadic with the time spent at each node, generally not spending very long at each node. However, Subject 2 began to spend longer at certain nodes, most notably some Volcano nodes and most Asteroid type nodes.

**Table 5.2 Extract from Subject 2's data from Appendix D**

| Item | Node | C | BPR | V | CLICK | Time |
|---|---|---|---|---|---|---|
| 149 | Planet | 1 | 0.83 | 1 | LINK | 0:00:05 |
| 150 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:03 |
| 151 | Volcano | 0 | 0 | 0 | LINK | 0:00:05 |
| 152 | Asteria | 0 | 0 | 0 | LINK | 0:00:23 |
| 153 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:13 |
| 154 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:03 |
| 155 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:08 |
| 156 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:02 |
| 157 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:05 |
| 158 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:13 |
| 159 | Versuvius | 0 | 0 | 0 | LINK | 0:00:02 |
| 160 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:12 |
| 161 | Volcano | 0 | 0.91 | 1 | LINK | 0:00:03 |
| 162 | Isis | 0 | 0 | 0 | LINK | 0:00:20 |
| 163 | Mount Etna | 0 | 0.93 | 1 | LINK | 0:00:04 |
| 164 | Asteria | 0 | 0.89 | 1 | LINK | 0:00:06 |
| 165 | Volcano | 1 | 0.96 | 1 | LINK | 0:00:06 |
| 166 | Montes Haemus | 0 | 0.89 | 1 | LINK | 0:00:56 |
| 167 | Geographical Feature | 0 | 0.84 | 1 | LINK | 0:00:02 |
| 168 | Volcano | 2 | 0.97 | 1 | LINK | 0:00:06 |
| 169 | Eruption | 0 | 0 | 0 | LINK | 0:00:12 |
| 170 | Volcano | 3 | 0.98 | 1 | LINK | 0:00:01 |
| 171 | Geographical Feature | 1 | 0.96 | 1 | LINK | 0:00:02 |
| 172 | Canyon | 0 | 0 | 0 | LINK | 0:00:04 |
| 173 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:14 |
| 174 | Noctis | 0 | 0 | 0 | LINK | 0:00:12 |
| 175 | Geographical Feature | 2 | 0.96 | 1 | LINK | 0:00:04 |
| 176 | Planet | 2 | 0.73 | 1 | LINK | 0:00:02 |

#### 5.6.3.3 Subject 3

Subject 3 was a self-assessed expert with the domain. Subject 3 began by pressing the BACK button several times and then the FORWARD button several times. This was probably caused by a fault in the browser software. Afterwards Subject 3 restarted the system and went directly to both answers to the browsing task. This suggests that the subject was familiar with the answers. After Subject 3 had completed the set task then they appear to have spent some time reading other Satellite nodes (as indicated by the long time spent at them). Afterwards Subject 3 entered a new area (Volcanoes) and spent less time at them, perhaps indicating a reduction in interest. After leaving the Volcano area Subject 3 then revisited the Satellite area, but they did not spend more than a second or so at each node. It is also apparent that Subject 3 did not navigate by using hierarchical nodes and must have therefore been using Related Material links. The section of browsing data presented below indicates a revisit to the Satellite nodes and could be indicative of Subject 3 becoming frustrated with the number of links being offered, since it could be that they exhausted the Related Material links and became stuck in an area. This may be useful information for determining whether more links should be offered or that the current interface style does not suit a particular person.

Subject 3 exhibited several strategies. Firstly they appeared to be interested in Satellites and appeared to be familiar with the layout of the domain, resulting in paths and rings. Later on Subject 3 changed their strategy, possibly because they became lost, disinterested or frustrated, this is indicated by Subject 3 spending only a second or so at each node. Finally, Subject 3 appeared to change their strategy once more as they entered a new area (Asteroids) and began to spend more time at each node. This can be seen in the table below.

**Table 5.3 Extract from Subject 3's data from Appendix D**

| Item | Node | C | BPR | V | CLICK | Time |
|---|---|---|---|---|---|---|
| 297 | Jupiter | 0 | 0.37 | 1 | LINK | 0:00:00 |
| 298 | Ganymede | 1 | 0.72 | 1 | LINK | 0:00:10 |
| 299 | Satellite | 0 | 0.37 | 1 | LINK | 0:00:03 |
| 300 | Titan | 0 | 0.37 | 1 | LINK | 0:00:01 |
| 301 | Ariel | 0 | 0.37 | 1 | LINK | 0:00:01 |
| 302 | Callisto | 0 | 0.37 | 1 | LINK | 0:00:01 |
| 303 | Charon | 0 | 0.37 | 1 | LINK | 0:00:01 |
| 304 | Deimos | 1 | 0.49 | 1 | LINK | 0:00:02 |
| 305 | Encelydus | 0 | 0.37 | 1 | LINK | 0:00:01 |
| 306 | Europa | 2 | 0.5 | 1 | LINK | 0:00:22 |
| 307 | Iapatus | 0 | 0.37 | 1 | LINK | 0:00:12 |
| 308 | Io | 0 | 0.37 | 1 | LINK | 0:00:32 |
| 309 | Janus | 1 | 0.46 | 1 | LINK | 0:00:11 |
| 310 | Saturn | 0 | 0.37 | 1 | LINK | 0:00:13 |
| 311 | Montes Haemus | 0 | 0.39 | 1 | LINK | 0:00:01 |
| 312 | Mount St. Helens | 0 | 0.39 | 1 | LINK | 0:00:01 |
| 313 | Mount Rainier | 0 | 0.39 | 1 | LINK | 0:00:01 |
| 314 | Earth | 0 | 0.39 | 1 | LINK | 0:00:02 |
| 315 | Deimos | 2 | 0.89 | 1 | LINK | 0:00:02 |
| 316 | Europa | 3 | 0.9 | 1 | LINK | 0:00:03 |
| 317 | Janus | 2 | 0.92 | 1 | LINK | 0:00:01 |
| 318 | Europa | 4 | 0.98 | 1 | LINK | 0:00:01 |
| 319 | Europa Orbiter | 0 | 0.39 | 1 | LINK | 0:00:01 |
| 320 | Galileo | 0 | 0.39 | 1 | LINK | 0:00:01 |
| 321 | Spacecraft | 0 | 0.39 | 1 | LINK | 0:00:01 |

#### 5.6.3.4 Subject 4

Subject 4 was a self-assessed intermediate with the domain. Subject 4 located the first answer to the set task immediately, indicating that they knew the answer beforehand, but then took a little longer to find the second answer, indicating that they probably did not know the second answer. Afterwards Subject 4's browsing was very similar to Subject 3's (lines 379 – 473 in Appendix D). However, Subject 4 made more use of the hierarchical structure of the domain and as a result did not appear to become lost or frustrated.

#### 5.6.3.5 Subject 5

Subject 5 was a self-assessed intermediate with the domain. Subject 5 found both answers to the set task immediately (similar to expert Subjects 1,2 and 3). During the free browse Subject 5 was quite methodical and used the BACK button more than previous subjects (resulting in a much higher spike value). Subject 5 tended to use the hierarchical nodes as landmarks, but did not spend very long at most nodes and does not appear to have read many nodes in the domain.

#### 5.6.3.6 Subject 6

Subject 6 was a self-assessed novice with the domain. Subject 6 was interesting because it took them some time to find the answers to the set browsing tasks, indicating that they did not know the answers beforehand. Subject 6 appeared to employ a very methodical browsing pattern consisting of spikes and/or tight rings. It would appear that Subject 6 used Moon and Earth as base nodes for further exploration. They then appeared to search for the largest satellite by visiting each planet node in turn. This resulted in a kind of ripple search that eventually found the solution. However, such a method could potentially take a long time if the solution happens to be a long way off. After Subject 6 found both answers to the set task, they then continued with their browsing strategy until all the satellite nodes had been visited. Afterwards Subject 6 tended to use this same strategy to explore other areas of the domain. Towards the end of the session Subject 6’s browsing strategy changed and they began to use class nodes more and spend less time at each node (around entry 675 in the table in Appendix D). This may indicate that the subject had become bored, lost or frustrated. This particular strategy was continued until the exit was found.

#### 5.6.3.7 Subject 7

Subject 7 was a self-assessed novice with the domain and appears to have used a similar strategy to that of Subject 6 although it seems they became side-tracked from the task in hand and began to examine the Volcano nodes. One difference between Subject 7 and Subject 6 is that Subject 7 did not explore the domain based upon the actual structure of the domain. For example, whereas Subject 6 used the hierarchical structure and therefore visits the planet node before the host planet and then onto satellites, Subject 7 tended not to use the hierarchy and instead used the related links. This strategy resulted in a less clear browsing pattern since they did not browse the planets in their logical order. This is an issue that relates directly to the interface and demonstrates the clear link between what the user of the interface can do and what the interface allows them to do. Eventually Subject 7 found the first answer, but afterwards seemed to change their strategy and failed to find the second answer. Afterwards, Subject 7 spent less time at each node and continually revisited several nodes, spending very little time at each one. At one point Subject 7 spent nearly 4 hours at a node (presumably they had left the system for this duration) and upon further navigation spent very little time at each node, indicating their disinterest.

Similarly to Subject 6, Subject 7 tended to use spikes and rings, rather than paths and loops.

#### 5.6.3.8 Subject 8

Subject 8 was a self-assessed novice with the domain. Subject 8, in a similar fashion to Subjects 6 and 7 began by finding a familiar node and then navigating around it, producing a high spike value. Subject 8 also used the BACK button on the browser, which again produced a high spike value. Subject 8 appeared to be quite methodical but took a different approach from that of Subjects 6 and 7, in that Subject 8 used the Satellite class node as a basis for their navigation, whereas Subject 6 used the Planet node and Subject 7 used Related Material links. As with Subject 6's browsing strategy, Subject 8's browsing strategy is guaranteed to locate both solutions (if it is taken to its conclusion). Subject 8 used the Earth node as a base node throughout their browsing. After Subject 8 had located both answers to the set task, then they appeared to lose some interest in the domain and spent relatively little time at each node.

#### 5.6.3.9 Subject 9

Subject 9 was a self-assessed novice with the domain. Subject 9 behaved very differently from the other novice subjects (6,7 and 8). Subject 9 did not seem to employ a methodical strategy; rather, they clicked links without regard to how far away from the initial node they were travelling. This strategy resulted in a low value for spike and a high value for path. This strategy continued through different classes of nodes, before the set task had been completed. Eventually, Subject 9 began to revisit nodes, and thus the spike and ring values increase. Subject 9 then revisited the Satellite node, which may have reminded them of their task, and they began to spend more time at the nodes. Subject 9 then found the second answer to the set task. Finally, they appeared to be satisfied and made a direct exit from the system.

Subject 9's strategy is interesting, but is difficult to interpret. It could be a useful strategy to rapidly explore the domain to gain an overview before homing in on the solution. However, it may also be possible that Subject 9 randomly encountered the answer.

#### 5.6.3.10 Subject 10

Subject 10 was a self-assessed novice with the domain. In common with Subject 9, Subject 10 rapidly ventured into the domain and spent very little time at each node. The only patterns that can be discerned are that the Earth node appears to be being used as a landmark node. Subject 10 visited both solution nodes for the set task several times, but only reported a solution to one. The second solution node appears not to have been visited for long enough to have been read. Subject 10 continued in this vein and eventually found the exit. It is quite possible that subject 10 was using the Related Links to navigate the domain and became stuck or lost when they had exhausted them at each node.

#### 5.6.3.11 Subject 11

Subject 11 was a self-assessed novice with the domain. Subject 11 was similar in their browsing habits to Subjects 6 and 7, in that like Subjects 6 and 7 they were quite methodical and reluctant to move too far away from familiar nodes. This resulted in a high spike value and a low path value. A slight difference in Subject 11's browsing strategy was that they appeared to use the Related Material link box since the Moon node would appear in this link box for each Satellite node, this could have provided a convenient way to return to a familiar point. This browsing strategy found both solutions to the set task quite quickly. Subject 11 continued to use this browsing strategy to explore much of the rest of the domain before they found the exit.

### 5.6.4 Discussion

#### 5.6.4.1 Discussion of Experimental set up

The set task was designed to encourage the subject to seek out some information stored within the domain. Their ability to do this would depend upon their previous knowledge of the structure of the domain. In the case of the Astronomy domain the structure is closely related to the physical composition of the solar system. It is therefore highly likely that the subject's previous knowledge of the subject would allow them to navigate the hierarchy of the domain. The set task was to identify which is the biggest satellite of the solar system. This task was chosen deliberately since there are two possible answers. One answer is that Ganymede is the biggest in terms of physical size; however, Titan is bigger if the atmosphere is taken into account (this information is held within the contents of the astronomical domain's nodes and was taken to be accurate at the time of authoring).

The second element of the experiment asked the subject to browse the domain for as long as possible. The purpose behind this element of the experiment was to attempt to gain an insight into how a subject's browsing habits may change as they encounter new information as they browse the domain. Such browsing information could ultimately be used to support the student's change in interests, perhaps by presenting more of the links that relate to those interests.

The choice of the ASTRO-L email group was made on the basis of the domain being used for the experiment and because it is a moderated email group. A moderated email group has fewer postings than an unmoderated group, but the postings are generally highly related to the subject. Further, it was anticipated that a request for volunteers would be less likely to be interpreted as an advertisement in an unmoderated email group. The choice of a distance experiment was made on the basis of practicality, since it was anticipated that a greater number and variety of volunteers could be obtained in a non-local environment. Further, the use of a distance experiment allows each subject to spend a potentially unlimited amount of time with the prototype system, which would not have been possible in a laboratory environment. However, the use of a distance experiment imposes some burdens upon the experiment. There is no real way to determine how knowledgeable each subject was about the domain. It may have been possible to use a pre-test but in this case it would not have been possible to verify the results as genuine, since the subject could not be interrogated directly, and so nothing would have been gained. This would not have been a problem for a local experiment, since a pre-test could have been conducted in controlled conditions. It is for this reason that subjects were self-assessed the consequence is that they could have had different ideas about what each ability category meant.

A further complication of using non-local volunteers was the lack of control of other variables, such as whether the volunteers were disturbed or otherwise prevented from full engagement during their browsing with the system. This can be countered by Hypernet recording complete timings of the browsing session so that it is possible to determine whether the volunteer has "gone to sleep" by virtue of there being a long interval at a node. This can be seen in the results when one subject spent nearly 4 hours at one node. However, it was anticipated that the disadvantages of using a distance experiment would be outweighed by seeking volunteers, who as volunteers would be more likely to take an active interest in the experiment and by there being a much reduced "experimenter effect", since it was not possible to influence a subject's behaviour beyond the written instructions.

#### 5.6.4.2 Discussion of Results

Results for expert level subjects (subjects 1 – 3) demonstrate that they tended to use rings and loops with a high degree of pathiness. This suggests that they were sure of the layout of the domain (since it is organised in a structured manner) and that they are only crossing previously visited nodes as a means of getting to their destination. Further evidence of this is that they only spent a second or so at each node on their return journey. The free browsing exercise produced more spikes, although these tended to be more distinct than for the novice subjects, in that they were likely to be returning along the same path to get to their next point of interest. Both expert and intermediate subjects (subjects 1 – 5) appeared to use one strategy throughout. This is probably because they are comfortable with the domain and interface and are able to locate information within it. By contrast novice students (subjects 6 – 11) tended to adopt more than one strategy, for example Subject 7 appeared to use a ripple searching technique (circle the current area whilst slowly moving outwards) similar to that used by Subject 6 for locating the answers to the set browsing task. However, at one point during Subject 7's navigation they encountered the "Volcano" nodes, which appeared to take their interest, whereupon they then temporarily abandoned their "searching" strategy and began to browse the area around Volcanoes, before finally returning to their original strategy. It could be argued that the expert subjects are less likely to have their interests caught in this way, if they are already familiar with the domain and this may account for the manifestation of one strategy from the expert subjects.

Novice subjects (subjects 6 – 11) tended to be more complex in their behaviour than expert subjects although several distinct patterns did emerge. Novice subjects tended to use spikes and rings much more than expert subjects, and they tended to be short and often centred on a small cluster of nodes. This is indicative of the subject being reluctant to leave an area with which they are familiar. Novice subjects tended not to use large loops or large rings, suggesting that it is important to have an understanding of the structure of the domain before the subject can return to a previously visited node via a different path.

Novice subjects tended to fall into two categories; subjects in the first category seemed reluctant to leave the area with which they are familiar, resulting in a low path count and high spike and ring counts. The second category of subjects tended to be more adventurous and move away quickly from the area, resulting in a high path value, but low values for spike, ring and loop. It is debatable which strategy is generally the more productive. However, the different browsing patterns used by novice subjects does suggest that, for this experiment at least, there were two approaches taken by novice subjects (larger experiments may uncover more). It may be that these two strategies undertaken by the different novices may be facilitated by different presentation of the hypermedia material. The reticent navigators (Subjects 6,7,8 and 11) were obviously intent upon the domain and did not want to become lost, thus affirming their interest and commitment. These subjects used very similar content-less browsing patterns, in that they tended to use small loops and short spikes. However, they tended to use different nodes as local landmarks and thus their content-based browsing patterns look different. It is possible that the use of content-less browsing patterns may enable the identification of similar browsing patterns by filtering out the content-based information that could otherwise confuse the issue. This can be seen by subjects 6,7,8 and 11 appearing to use different strategies when content-based information is taken into account (they use different nodes as navigational cues, for example), but they appear to be using similar strategies when the explicit information about individual nodes is removed. Further, it is possible that the more adventurous navigators (Subjects 9 and 10), who tended not to use short spikes and small loops, were just aimlessly wandering (a strategy identified as being undesirable by Canter et al (1995)). If it could be determined that a student was aimlessly wandering then the hypermedia system could alter its presentation of the domain in an attempt to reengage the student, perhaps by offering links more closely related to the student’s interests. It was not possible to determine whether a student was aimlessly wandering with the free browsing task, since if the student is not tested then it is difficult to determine whether they are involved in their own novel browsing strategy. The set task did, however, provide some information with which to make a determination on a particular browsing pattern, since a student’s success with the task is an indication of the usefulness of their strategy. Results demonstrated that Subject 9 probably was wandering, since they were unable to fulfil the task. However, Subject 10, who also used a more adventurous strategy, did fulfil the task.

The browsing experiment conducted as part of the current research project used a relatively small domain; it could be that larger or more complex domains more effectively enable students to exhibit a wider range of browsing strategies that better suit their learning styles. This may be especially true of the expert subjects, since the domain was in all probability not taxing for them and they were therefore able to locate information using simple browsing strategies. However, this limited experiment has demonstrated a clear distinction between subjects who were familiar with the subject matter beforehand and those subjects who were not. It also demonstrated differences between subjects with different navigational (or learning) styles, since expert subjects browsed the hypermedia in a different ways to novice subjects and not all novice subjects used the same browsing strategy, for example. What the experiment did not show is that every subject browsed the domain in a totally different way from every other subject. There is therefore a strong case for stating that there is a finite number of browsing patterns that relate to browsing strategies and learning styles and therefore people with similar backgrounds (in terms of previous knowledge and learning styles) will generally appear to behave in similar ways with a hypermedia domain. This relationship could therefore be exploited to extract information about a user of hypermedia with a view to aiding them in their navigation.

### 5.6.5 Further Experiments

The major criticism of the browsing experiments is the low number of subjects. It was originally anticipated that an email list would have provided scope for many more volunteers. However, this did not prove to be the case, and was probably due to the effort required for a volunteer in terms of replying and setting up the prototype Hypernet system. In retrospect, a better approach might be to use a non-local version of the Hypernet system. This would involve Hypernet being redesigned to function on an Internet server, so that it could be accessed remotely by volunteers. This approach would make the experiment more viable for both the researcher and the volunteers. The researcher would be assured that the system was functioning identically on each machine, since it must comply with the Internet protocols, and also the volunteers would be freed from the inconvenience of there being a delay between volunteering and receiving the system and the complications of installing the system. Such an experiment could also be more open-ended, since data could be actively collected from volunteers over a prolonged period.

A further weakness with the experiment was that it involved only a single, small domain. It could therefore be argued on negative grounds that the browsing patterns produced were idiosyncratic to the domain used in the experiment. It is likely that differently structured domains will require that the learner use different browsing strategies and hence produce different browsing patterns. However, this does not necessarily mean that the contention that similar learners use similar browsing patterns does not still hold for differently structured domains. Rather, it means that these other potential browsing patterns were less likely to have been employed in the browsing experiment described above. Further experiments could shed light on the issue by employing several differently structured domains.

The small-scale experiments presented above provided some evidence, along with Canter et al (1985) and Micarelli and Sciarrone (1996, 1998) that browsing patterns warrant further investigation as a means of extracting information about a user of hypermedia. For this reason a Fuzzy Rule Base (discussed in Chapter 7) was built, so that information could be collected, over time, and inferences could be automatically made about how browsing patterns relate to types of learner. Such a system would be a useful tool for further larger scale experiments, since it could reduce much of the burden upon the researcher to manually examine browsing patterns.

## 5.7 Conclusion

This chapter has discussed the patterns made by users of hypermedia systems and the potential use this information may be for aiding that user. The browsing experiments discussed in this chapter suggest that different types of student do browse the hypermedia in a distinctive manner. However, a further result from the experiment was that there could be a wealth of different browsing strategies and that it is not a simple case of similar abilities of students behaving in similar ways. It follows that different learners could be aided by different mechanisms, such as presentation of different types of links, and it would therefore be useful to identify these browsing strategies for further analysis.

The following two chapters are concerned with the mechanism used to identify browsing patterns and the mechanism designed to map browsing patterns onto different styles of learners.

# Chapter 6

# The Browsing Pattern Recogniser

## 6.1 BPR Introduction

The previous chapter described some of the content-less browsing patterns and strategies employed by the users of hypermedia systems and the potential uses to which they may be put. Previous research by Canter et al (1985) indicates that browsing pattern data offers the potential to extract some psychological information about the student, which may in turn be used to tailor the hypermedia towards the student. Canter et al (1985) state that a suitable algorithm needs to be devised to accomplish the task of extracting browsing information from the student. It is the role of the Browsing Pattern Recogniser (BPR) to extract this information.

This chapter is concerned with the practical aspects of neural network design, training and implementation, with specific reference to the neural network systems designed for the BPR, a device for recognising content-less browsing patterns, so that some indication of the student's overall browsing strategy can be made. The BPR recognises browsing patterns as the student moves from one node to another, a process that is common to all hypermedia systems. The BPR therefore offers the potential to be employed within many hypermedia systems. The benefit is that an indication of the student's overall browsing pattern could be used to supplement other information seeking aids, such as those that use keyword extraction. Browsing pattern information could supplement such aids in that it may enable the aid to distinguish a user who requires more detailed information from a user who requires more general information and it may be able to diagnose that a user requires help in finding information. Developers of information seeking aids and agents have noted the lack of this kind of information (Armstrong et al 1995).

The BPR has previously been described as a black box system inside the student model. Within this chapter, the BPR is described in terms of its architecture, training and results. Note that the BPR identifies browsing patterns and not necessarily the browsing strategies described by Canter et al (1985) and McAleese (1993). The distinction is that the pattern itself does not necessarily imply the student is performing a particular task (as the aforementioned strategies do). This extra information must be extracted by other means. This is the role of the Fuzzy Rule Base, described in the following chapter.

## 6.2 Rationale for Using A Neural Network

A compelling reason for employing a neural network is the fuzzy nature of the browsing pattern data. For example, spikes, rings, loops and paths have been identified by research. However they may not appear in a browsing pattern as discrete items; instead they may merge into hybrid constructs. A neural network is able to cater for this fuzzy nature (Gurney 1997), whereas a rule-base system would need every possible combination explicitly encoded within it.

Bailey and Thompson (1990) suggest that a neural network approach is desirable if the application data is intensive and depends on multiple interacting criteria, the problem area is rich in historical data or examples and the data set is incomplete and/or noisy and/or contains errors. In Hypernet's case the application data is intensive; this is described further below. The data is potentially noisy due to the multitude of possible patterns the student can exhibit, again this is described more fully below.

General arguments for the application of neural networks in educational systems can be found in Chapter 2, Masters (1993) and Hagen et al (1996) and can be summarised as follows. A neural network system is fast to execute and is the only practical choice for situations where the input data may be imprecise or not fully understood. Several low-level browsing constructs have been previously identified by research (Chapter 5). However it may be the case that more, or different, low-level browsing constructs exist for newer hypermedia designs, such as the semantic hypermedia designed for this project. A neural network system could be quickly updated with the extra parameters representing these constructs and retrained, whereas a rule-based system would likely require to be completely rewritten.

A final reason for employing a neural network is the relative speed of development, compared to a rule-based system. Research has indicated that the development of a neural network system is several times faster than a rule-based system (Chiu et al 1991, Hagan et al 1996). This is because although the neural network acts as a rule-base, its rules are not explicitly encoded within it, but instead learned from a set of examples from the chosen domain. It is argued that as long as sufficient examples exist a neural network can learn its rules much more quickly than a human designer can formulate them (Chiu et al 1991).

Canter et al (1985) remark that the browsing constructs described in the previous chapter "present some calculation problems", specifically the tendency for them to merge together and hence become fuzzy in nature. They note the need "to develop more complex and subtle algorithms". The remainder of this chapter demonstrates that such a complex and subtle algorithm can be interpreted as a neural network. Further, the neural network described below is capable of identifying browsing patterns more complex than those described by Canter et al, in that the neural network described here has been trained to recognise temporal information, in that it is able to distinguish between nodes being visited at different points in the past.

## 6.3 Representation of Hyperspace (Interfacing the Hypermedia to the Neural Network)

The neural network must be provided with information on which to base its evaluations. As noted above, this involves imposing some spatial structure upon the hypermedia itself, in order to recognise a student moving through this abstract space. The term “hyperspace” is perhaps a misleading one, since hypermedia and other information storage systems do not literally occupy anything that can usefully be thought of as space. The physical representations of such paradigms (how they are stored on the computer) are often equally unhelpful and therefore computer scientists impose other, logical, structures. Further, the concept of a student moving from one node to another is difficult to reconcile with spatial concepts.

Rather, the “space” utilised by the neural networks in this research project is conceived of as temporal space. Such space is common to all hypermedia systems, in that all hypermedia systems involve a student or user moving from one node to another at some point in time. This is therefore an approach that would render the BPR facilities useful for a multitude of other systems. Temporal space is defined in a hypermedia system by a student moving from one node to another. Therefore, in terms of a student’s navigation, a change in time is defined by this transition. Initially each node in the domain can be defined as being either “visited” or “unvisited” and this allows the neural network to interface to the hypermedia via a simple integer connection. As a student visits a node a flag is set within the node that indicates that the node has been “visited”. The neural network is therefore able to determine if a node has not been visited or not. Note that this information is sufficient to model all of Canter's low-level constructs, since such constructs can be defined as being streams of unvisited nodes, streams of previously visited nodes, a change from visited to unvisited, or a change from unvisited to visited. This is outlined below.

For example, given that “1” represents a node previously visited and “0” represents a node previously unvisited, (once an unvisited node is visited its values is set to 1), then 0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1 represents a spike, since it can be seen that the student has followed a trail of new nodes before returning along the same path, as indicated by the stream of visited nodes.

Similarly, the following patterns represent other browsing constructs:

0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0 represents several loops.

Which can be seen as streams of unvisited nodes, followed by a visited node as the student loops across the path they have previously taken.

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 represents a ring.
A ring is similar to a spike except that a ring indicates the returning to a start of a journey via a different path, whereas as spike is where a student returns to a node via the same path. This can be seen as the final node has been visited, whereas the others have not.

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 or 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 represents a path.
A path is a stream of previously unvisited nodes or visited nodes, although usually a path of previously visited nodes would be seen as part of a spike.

A difficulty with the above representation of hyperspace is that the domain has the potential of becoming a profusion of “visited” nodes and it is therefore impossible to tell the difference between a node visited recently and a node visited a long time ago. One possible solution to this problem that was examined was to change the values of “visited” nodes to “visited a long time ago”, after a certain time has elapsed (after the student has moved a set number of times). A major drawback of this approach is that it is difficult to determine how many student movements should be used before a node is set to “Visited a long time ago”. Further, the temporal information that can be represented using this system is still limited, as is the representation system described by Canter et al (1985).

Therefore, a further expansion was made to the hyperspace representation. This is described below.

Representing more temporal information was accomplished by using real values rather than integers for the nodes visited flag. As a student visits a node it is set to “visited” (1). As each subsequent time interval elapses (as a student continually moves from one node to another), the value of each node’s visited flag in the hypermedia whose value is not “unvisited” (0), has its value reduced by a small amount. Thus information is encoded within a node about how long ago that node was visited, until the node becomes marked as zero, or “visited a long time ago”. There is therefore always a distinction between a “visited” and an “unvisited” node, as well as a distinction between nodes visited at various points in the past.  This enables the neural network to model browsing constructs more complex than Canter et al's, since temporal information is taken into account.

The utilisation of decreasing real values for representing when the node was visited (or not) helps the neural network to distinguish between nodes that have been visited recently and nodes that may have been visited a longer period ago. Therefore, using real data, a spike can be distinguished, from say random jumping, since the stream of previously visited nodes will appear as a stream of steadily decreasing "visited" values, whereas, randomly jumping to previously visited nodes is likely to produce random "visited" values. Note also, that moving from one node to another can be via a link, in this case selected from one of the link boxes, or via a jump. Since each node in the hypermedia is connected to every other node, indirectly via the semantic network, a jump (via a search query, for example) is still travelling to another node via a link that has previously been calculated and weighted by the automatic linking algorithm. Therefore, no extra information needs to be supplied to the neural network to indicate a such jump.

## 6.4 Differences between low-level Browsing Constructs

It is important to consider what part(s) of the browsing pattern the neural network defines as a particular browsing construct. It may be apparent to the human observer what the distinction is between constructs; however, the neural network is not "sensing" its environment (input data) in the same way as the human observer. The neural network "sees" the browsing constructs as a series of real numbers, flowing through its inputs. The question is, then, how the neural network learns to distinguish the individual browsing constructs from such a sparse input.

It can be seen that a spike is a series of new nodes followed by a series of old nodes, or more accurately, recently visited nodes. This distinguishing feature makes the spike unique. A path is recognised by the neural network as an unchanging sequence of new nodes. Note that a path as defined by Canter et al (1985) is represented by a route away from an initial node that does not return to any previously visited nodes. The path construct is therefore a length construct, i.e. it may appear "inside" another construct. For example, a large loop would exhibit some "pathiness" whereas a small loop would exhibit no "pathiness". The neural network therefore identifies a path as a high degree of new nodes. The ring construct is similar to a spike in that it is a mixture of new and old nodes, however the series of new nodes and old nodes do not cluster together, they are interspersed. Rings are essentially a cluster of small loops, indicating a student crossing previously visited nodes several times in a short space of browsing-time. Conversely, loops are larger structures, with a high degree of pathiness. They indicate a student who has travelled away from their starting node, but has then returned.

## 6.5 BPR Architecture

There are many neural network architectures available (Gurney 1997, Masters 1993, Maren 1991), most of which are the subjects of ongoing research. The practical neural network literature indicates that the Multi-Layer Feed Forward neural network (MLFF), sometimes referred to as the Back-Propagation neural network, is the most popular (Masters 1993, Hagen et al 1996, Gurney 1997). Mathematically the MLFF neural network is capable of simulating any function (mapping a set of inputs onto a set of outputs) (Maren et al 1991). In practice, however, it is sometimes very difficult to train a neural network to map a given function. This problem is endemic to training neural networks. Other neural network paradigms are more usually designed to aid this training process (Masters 1993). The choice of neural network is therefore simplified. If a MLFF neural network can be trained within a practical period of time then it is suited to the task (since it can mathematically perform it). If not, then an alternative neural network architecture should be sought (Masters 1993). The MLFF was therefore examined as a possible architecture for the BPR. The MLFF neural network requires supervised training, in that the neural network must be trained by examples provided by the neural network designer before it is put into active use. This involves the neural network designer identifying all the elements that must be identified by the fully trained neural network. Supervised learning therefore requires that the items to be identified are known beforehand, so that the network can be trained with them. In this case Canter et al's browsing constructs provide the training data, although they have been modified to incorporate temporal information.

Further support for the use of the MLFF neural network can be found in Skapura (1996) (the MLFF being referred to as the Back-propagation network (BPN) by Skapura, who is referring to the learning algorithm and not the architecture). Skapura (1996) gives complete examples of various neural network solutions to various problems. The closest problem described by Skapura (1996) to that of recognising browsing patterns is the problem of character (or handwriting) recognition. The problems are similar because they both involve the classification of a finite number of patterns from a potentially infinite number of possible inputs. For example, there are twenty-six letters in the alphabet but each one can be represented in many ways. Similarly, there are four browsing constructs to be identified, but they can appear in many different ways. The problem of identifying browsing constructs differs from that of character recognition in that the size of each input browsing construct is not known (i.e. they can be potentially any length), whereas characters can be represented (or at least scaled to fit) by a finite grid (8 by 8 say). Furthermore, the number of outputs differs greatly in that for browsing constructs there are only four, whereas for character recognition there would usually be 26 uppercase, 26 lowercase, 10 numbers and possibly some extra characters. Therefore, the browsing construct problem has extra complexity due to the unknown length of input, but the character recognition problem has extra complexity due to the number of outputs required. Skapura (1996) suggests that the outputs of the character recognition neural network should be designed so that each character to be recognised has a separate output and is not based upon a single output producing a character code (such as an ASCII code). The reason for this is that certain characters such as “O” and “Q” are very similar to each other, but their ASCII codes do not take this relationship into account. This can result in a network producing unpredictable results. For the browsing construct problem the drawback of using an output for every pattern is small since there are only four and as Skapura (1996) suggests, it will improve the network’s performance.

The use of a supervised learning paradigm is appropriate if the neural network is trained to recognise the previously identified low-level browsing constructs (Canter et al 1985). However, these low level structures combine to form more complex browsing patterns, which may be numerous and complex. The BPR neural network has been designed to recognise the low level browsing constructs only, since the supervised learning paradigm is not suitable for application to the problem of recognising high-level browsing patterns. This is because recognising high-level browsing patterns would require the identification of every possible browsing pattern, before the system could be trained. However, since the higher-level browsing patterns are composed of the low-level patterns it is still possible to identify them as being a combination of the low-level constructs. This method is also necessary since the high-level patterns described by Canter et al (1985) are really examples from a possible group of browsing patterns. For example, the "scanning" pattern shown in Figure 5.2.B in the previous chapter, represents one example of a "scanning" pattern. In reality there are many similar browsing patterns that can collectively be thought of as "scanning". It is therefore necessary to identify the low-level constructs that are present within a pattern so that they can be combined into the more general groups. Recognising higher level browsing patterns is achieved by further processing of the BPR’s output by a fuzzy logic based system, termed the Fuzzy Rule Base (FRB), which is the subject of the following chapter. This is analogous to handwriting recognition, in that letters are built up of similar constructs (lines, curves and dots) and combine in similar ways to form the handwriting of different people. It is the role of the BPR to recognise these low-level constructs of the browsing patterns, which are analogous to lines and curves in hand-writing and it is the role of the FRB to combine the low-level browsing constructs into high level browsing patterns, which are analogous to characters in hand-writing recognition.

Initial neural networks were designed with integer inputs that are limited to 0 for a previously unvisited node and 1 for a previously visited node. Nodes are marked as visited or unvisited, it is not taken into account whether a node was visited minutes or hours earlier. These networks were then expanded by adding a time relation to the input data, utilising real values as inputs rather than integer values achieves this. The same input structure, described below, was used; however the input data could be a range from zero to one instead of either zero or one. The closer to one the input, the “older” the node. This requires a simple modification to the hypermedia software, when a user visits a node it has its “newness” flag set to zero (as previously). However, at each time interval the newness of the node decreases by a small amount until it settles back to zero. The time interval itself is a parameter that may be adjusted according to the level of user and the number of nodes in the domain, e.g. a novice user may have a shorter time interval, so that the nodes revert to new relatively quickly, representing the novice's narrow knowledge of the domain. The same neural network architectures were used for this expanded browsing data problem as the first, integer data problem. However, the simulated student browsing data was changed to incorporate real values.

## 6.6 BPR Neural Network Design

### 6.6.1 The Virtual Student Program

A major drawback of novel hypermedia systems, as noted by Jonassen and Wang (1993) and Dillon and Gabbard (1998), is the difficulty in obtaining empirical data concerning their usefulness or otherwise. A true hypermedia system is designed to organise a large corpus of knowledge (Conklin 1987). If such a system is then to be used as part of a tutoring system it requires extensive trials with students to accurately evaluate it. This is true of Hypernet, in that many of the facilities require extensive interaction with students in order for the emergent abilities of the various parts of the student model to become apparent. Testing with real students was, however, beyond the scope of the current research project and is suggested as further work, detailed in Chapter 10. However, if it can be demonstrated that the BPR can recognise a set of browsing patterns that "look like" real student browsing patterns, then it is highly likely that it can recognise the real patterns. To this end the neural networks were trained with “Virtual Students”; a program that can mimic the behaviour of students using hypermedia systems. The basis of the movements simulated by the Virtual Student Program (VSP) are those low-level browsing constructs identified by Canter et al (1985), with the addition of temporal information, of which the high-level patterns are composed. The output of the VSP is a browsing pattern that is composed of many browsing constructs. The VSP allows the neural network to be trained to recognise the well-understood low-level browsing constructs identified by Canter et al (1985). Linking these low-level constructs to higher level browsing patterns is accomplished by the Fuzzy Rule Base described in the following chapter. Further details concerning the operation of the VSP can be found in Appendix B.

### 6.6.2 BPR Training

Appendix A gives an overview of neural network technology and provides background information concerning architectures and training that is relevant to this chapter. The neural network in the BPR is provided with information representing a student's temporal browsing pattern, as described above. The inputs of the neural network provide a series of time shifted inputs. For example, if there are *n* inputs then the first input is the status of the current node and the *n*th input is the status of the node visited *n* nodes ago. Each status therefore, enters the neural network at input number one and progressively moves to the next input as the student moves from one node to another. This can be seen in figure 6.6.2.A. The difficulty with this approach is determining how many inputs are required for the neural network. The more inputs, the more complex the neural network and hence the more difficult it is to train. The number of inputs was determined by experimentation and is described later in this chapter.

The neural network was trained by using the VSP to generate a stream of separate browsing constructs. The network is then presented with each construct and allowed to form an output. The output is a real value (between 0 and 1) and can be thought of as a confidence factor, e.g. if the output for a particular construct is 0.7 then network is 70% certain that its input contains a spike. The network may output more than one value at once, for example it may determine its input is a spike (0.7) and a path (0.7) (the actual value indicating a certainty factor for each individual construct), indicating a long spike. The percentage values in the tables shown in section 6.7 represent the average confidence output of the neural network.

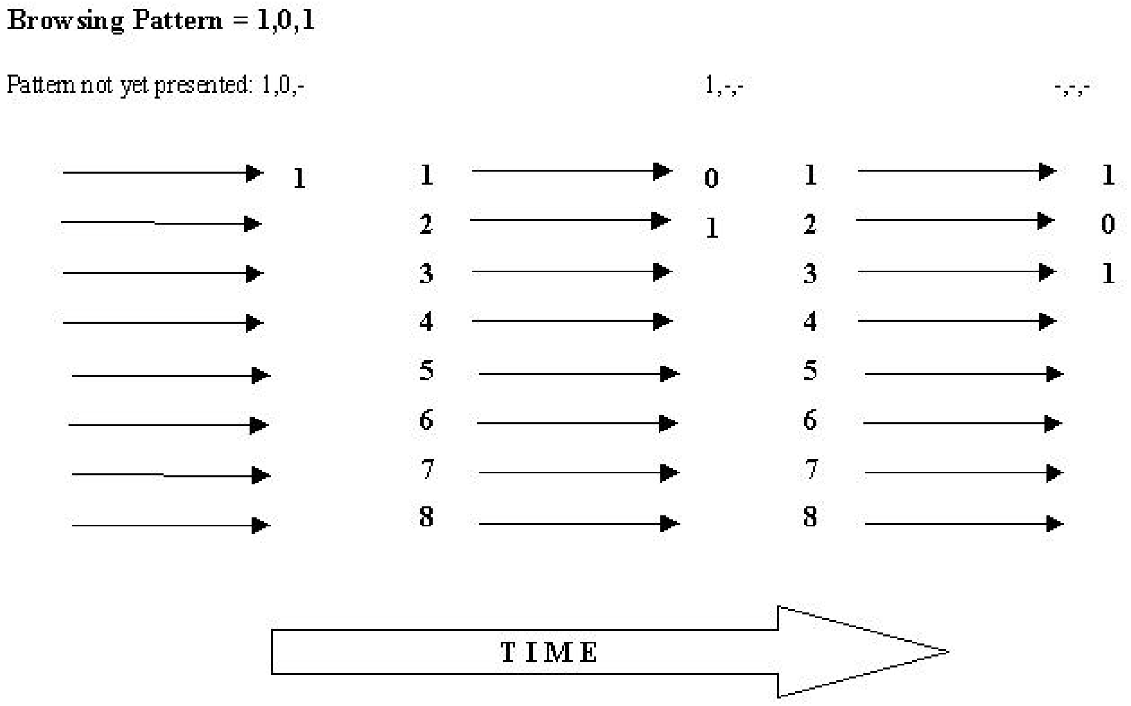


**Figure 6.6.2.A Time-Shifted Neural Network Inputs**

The neural networks were trained in line with practical guidelines suggested by Masters (1993); in that the number of hidden neurones was steadily increased. The training data was presented to the neural network in a rotational (sequential) fashion and in a random fashion. This is explained further in the sections below.

## 6.7 BPR Experiments

### 6.7.1 Experimental Method

Below is a description of the various parameters that form the neural network configuration. Each parameter is listed on the results tables below. The parameters were changed in accordance with the following method:

*The number of inputs and outputs*. The number of inputs is the number of inputs used to represent the time-shifted browsing pattern data. A sensible range to begin experimentation was between 8 and 16; fewer and not enough history information is presented and too many and too much weight is given to browsing data that occurred in the distant past. Determining the optimal value is achieved by examining the neural networks results and making a general determination as to whether it has produced a sensible grading. Those inputs designated "Integer", represent inputs where 1 represents a new node and 0 represents a previously visited node. Those inputs designated "Real", represent inputs which may lie between 1 and zero, indicating how long ago the node was visited. The number of outputs is the directly driven by the problem, in that there are 4 outputs, one for the four constructs to be identified (Path, Spike, Ring, Loop).

*The number of hidden neurones* (the neurones between the inputs and outputs of the neural network that provide the processing power of the network) directly affects the neural network's ability to learn. The more neurones the more the network can learn and hence the more it can memorise (Gurney 1997, Masters 1993). Too many neurones, however, can result in the "over-fitting" problem, where the neural network memorises its training data and does not identify only the salient points of the training data that allow it to generalise (Gurney 1997). There is no theoretical way to determine the required number of hidden neurones, although general guidelines have been suggested, such as a number midway between the number of inputs and the number of outputs (Masters 1993). However, it is generally accepted that the only way to determine the number of hidden neurones is by experimentation (Masters 1993, Hagen et al 1996, Gurney 1997). Therefore, various configurations of network were applied and the number of hidden neurones was varied by 2 neurones from the initial value as determined by Master's (1993) approximation (midway between inputs and outputs). Thus the first configuration was 4 outputs with 8 inputs, giving a starting value for hidden neurones as 6. Note that whilst the problem dictates that the number of outputs remains the same (4), the number of inputs is changed as an experimental parameter, in that the number of inputs represents the size of browsing data sample presented to the network. When the number of inputs was changed, initially the number of hidden neurones was not, to determine if the network could still perform as well.

*"Training set size"* is the number of examples presented to the neural network in order to train it. In order to train a MLFF neural network adequately  it must be presented with a

rich enough set of examples to categorise the general population. If the training set is not rich enough then the network is likely to encounter data in application that it has not been trained to recognise (Masters 1993). Two training set sizes are documented below. 50 was chosen as the standard size, since it allowed the networks to train relatively quickly. After the initial experimental neural networks were tested, 500 examples were used to determine if the first training set was sufficient to model the distinctive elements of each construct.

*"Results Training Set"*, this column displays the results the network achieved with the data it was trained with. This is not a good measure of the network's final abilities, since the network may have "over-fitted" (memorised) the examples and be poor with examples that have not previously been presented to it. It is however, a good indication of whether the network has undergone as much learning as it can with the training set data. The percentage value itself represents the degree of error expressed by the network. For example the neural network may be presented with pattern composed of 10% Spike, 50% Path, 25% Loop, 15% Ring (this is a known composition since it has been generated by the VSP). The neural network responds to this example and output the following values at its outputs; 8% Spike, 51% Path, 28% Loop, 14% Ring, giving the errors 2, 1, 3, 1 (i.e. the difference between the correct result and the actual result obtained). The total error for the presentation is determined by adding the individual errors together, e.g. 7% error or 93% success. This method of determining the success criteria of a neural network system, which is usually designed to model non-discrete or fuzzy data (and hence not likely to produce discrete or "crisp" results) is common in neural network applications (Kosko 1997). For example, recognising handwriting is usually accomplished by the neural

network outputting a certainty factor for each character it has been trained to recognise, a threshold level is then used to determine the "crisp" output, this is sometimes referred to as "defuzzification" (Masters 1993, Kosko 1996).

The *"Results Test Set*" column is the most relevant for determining the success of the neural network, since the test set is data that the neural network has never seen before. It is this value that is used to determine the overall success of the neural network architecture and configuration. It is not a simple matter, however, to determine what represents a good value. The data that the network is presented with is fuzzy and indistinct; therefore, it is not likely to produce 100% results, although it is desirable to obtain results as close to 100% as possible. The highest scoring neural network may then be selected and tested further with real browsing data (described later in this chapter).

### 6.7.2 Results

**Table1 Integer Inputs, 50 Examples**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 8 Integer | 4 | 6 | 50 | 89% | 79% |
| 2 | 10 Integer | 4 | 6 | 50 | 95% | 82% |
| 3 | 12 Integer | 4 | 6 | 50 | 95% | 87% |
| 4 | 14 Integer | 4 | 6 | 50 | 95% | 91% |
| 5 | 16 Integer | 4 | 6 | 50 | 97% | 91% |
| 6 | 18 Integer | 4 | 8 | 50 | 97% | 94% |
| 7 | 16 Integer | 4 | 10 | 50 | 97% | 94% |
| 8 | 16 Integer | 4 | 12 | 50 | 97% | 94% |

**Table 2 Integer Inputs, 500 Examples**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 8 Integer | 4 | 6 | 500 | 88% | 81% |
| 2 | 10 Integer | 4 | 6 | 500 | 95% | 82% |
| 3 | 12 Integer | 4 | 6 | 500 | 96% | 86% |
| 4 | 14 Integer | 4 | 6 | 500 | 95% | 88% |
| 5 | 16 Integer | 4 | 6 | 500 | 95% | 86% |
| 6 | 18 Integer | 4 | 8 | 500 | 97% | 91% |
| 7 | 16 Integer | 4 | 10 | 500 | 98% | 92% |
| 8 | 16 Integer | 4 | 12 | 500 | 97% | 93% |

**Table 3 Real Inputs, 50 Examples**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 8 Real | 4 | 6 | 50 | 69% | 52% |
| 2 | 10 Real | 4 | 6 | 50 | 72% | 57% |
| 3 | 12 Real | 4 | 6 | 50 | 77% | 63% |
| 4 | 14 Real | 4 | 6 | 50 | 79% | 64% |
| 5 | 16 Real | 4 | 6 | 50 | 85% | 68% |
| 6 | 16 Real | 4 | 8 | 50 | 86% | 70% |
| 7 | 16 Real | 4 | 10 | 50 | 90% | 75% |
| 8 | 16 Real | 4 | 12 | 50 | 91% | 75% |

**Table 4 Real Inputs, 500 Examples**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 8 Real | 4 | 6 | 500 | 61% | 48% |
| 2 | 10 Real | 4 | 6 | 500 | 62% | 48% |
| 3 | 12 Real | 4 | 6 | 500 | 68% | 58% |
| 4 | 14 Real | 4 | 6 | 500 | 68% | 63% |
| 5 | 16 Real | 4 | 6 | 500 | 70% | 66% |
| 6 | 16 Real | 4 | 8 | 500 | 78% | 72% |
| 7 | 16 Real | 4 | 10 | 500 | 84% | 79% |
| 8 | 16 Real | 4 | 12 | 500 | 88% | 83% |

**Table 5 Increased Hidden Neurones with 500 Examples**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 16 Real | 4 | 14 | 500 | 87% | 80% |
| 2 | 16 Real | 4 | 16 | 500 | 81% | 77% |
| 3 | 16 Real | 4 | 18 | 500 | 81% | 79% |
| 4 | 16 Real | 4 | 20 | 500 | 79% | 75% |
| 5 | 16 Real | 4 | 22 | 500 | 80% | 75% |
| 6 | 16 Real | 4 | 24 | 500 | 78% | 76% |
| 7 | 16 Real | 4 | 26 | 500 | 79% | 74% |
| 8 | 16 Real | 4 | 28 | 500 | 78% | 71% |

**Table 6 Increasing the number of inputs further**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 1 | 18 Real | 4 | 14 | 500 | 85% | 81% |
| 2 | 20 Real | 4 | 14 | 500 | 82% | 81% |
| 3 | 22 Real | 4 | 14 | 500 | 82% | 79% |
| 4 | 24 Real | 4 | 14 | 500 | 79% | 76% |
| 5 | 18 Real | 4 | 18 | 500 | 82% | 78% |
| 6 | 20 Real | 4 | 18 | 500 | 83% | 78% |
| 7 | 22 Real | 4 | 18 | 500 | 84% | 79% |
| 8 | 24 Real | 4 | 18 | 500 | 82% | 77% |

### 6.7.3 Results Conclusion

Table 1 demonstrates that the neural networks trained to identify browsing patterns composed of only 0 (new node) and 1 (old node), integer inputs, produced extremely high results. The best results were produced with a network of 8 hidden neurones. Table 2 shows the same neural network configurations except that they were trained with a larger training set. There is no significant difference in the results, suggesting that 50 epochs for these configurations of networks is suitable.

The introduction of real values in the data (introducing temporal information) (Tables 3,4 and 5), produced networks that do not produce as high test set results. Increasing the number of hidden neurones does increase the performance of the network. This is in-line with expectations, since the real input data is more complex than the integer data, it was therefore reasonable to expect that a more complex network would be required to process the real data.

Table 3 has a higher difference between training set and test set than does table 4. This indicates that there are not enough training examples, in that the network performs well with data it has seen, but performs less well with data, taken from the same population, that it has not seen. Both tables 3 and 4 demonstrate that an increasing number of hidden neurones improves the network's performance. Table 5 demonstrates that increasing the hidden neurones still further results in a slight reduction in performance, indicating that the optimal configuration has been passed. This is roughly in line with the heuristic proposed by Masters (1993), that the number of hidden neurones should usually be between the number of inputs and the number of outputs.

Table 6 demonstrates that increasing the number of inputs beyond the values in the previous tables produces a reduction in performance, even if extra neurones are added. This may be due to the patterns presented becoming too complex.

### 6.7.4 Results Discussion

The percentage results described in the tables above can be ascertained by virtue of the fact that the VSP generated the test data and so the number and type of constructs in each browsing pattern is known. The network is designed to examine the browsing pattern as it is shifted through its inputs and output a certainty value for each of the constructs it has been trained to recognise. If for example, a browsing pattern begins with a spike, then the spike is steadily “pushed through” the neural network, at first it will appear as a path (a succession of new nodes) and the network’s path output will be high and all the other outputs will be low. When the student (or virtual student) turns round and begins to revisit nodes, then the path output will diminish and the spike output will become high. The upshot of this is that, as described in the previous chapter, a browsing pattern can look different at different rates of sampling, in this case a spike looks like (or is) a path until the student begins to retrace their steps.

Results for the real input neural networks were not as high as for the integer input neural networks. The poorer results can be explained in terms of the browsing patterns being blurred by the real data. The distinction between a new and old node is less distinct. For example the following pattern represents a very simple spike (see above):

0,0,0,0,0,0,0,0,0,0,0,0,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0,0 (using real data)

0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0 (using integer data)

The “0”s represent a user traversing along a path of new nodes, once they have reached some point they decide to go backwards along the return path (perhaps via the back button). The return path is clearly distinguishable from the initial path when examining the integer data since the input activations between a new node and an old node are as far apart as they possibly can be. The distinction is less clear for the real data since the values can be very closely clustered. The neural network may not see that the pattern is simply a sequence of steadily increasing numbers. With further training it might make this distinction, but it would be likely to lose its generalising ability with other sequences, for example a succession of loops. This is an example of the over-fitting problem of neural networks, where a network begins to memorise its training data but is unable to generalise to data presented to it that it has not been specifically trained with (Masters 1993).

The result of this blurring of the data is that the neural network takes longer to “make up its mind” about what particular browsing pattern it is seeing. For example if a spike is presented to the neural network followed by a ring then the network has no discernible output for sometime after the spike has been presented, in fact it does not signal a spike until the next pattern has been fully presented to it. Thus the networks are always one step behind. This does not represent a problem for this application, since there is no need to interrogate the BPR every time the student moves in the hypermedia, this is because a single move does not represent a browsing pattern, similarly a single construct does not represent a complete browsing pattern. Instead several constructs combine together to form a browsing pattern that may be indicative of a particular task. The BPR may therefore be presented with a series of browsing data at relevant points, i.e. when the student has completed a tutorial their entire browsing from beginning to the end of the tutorial is presented to the BPR. However, it is interesting to see how the neural network works. It appears to be using the next browsing pattern in its recognition process. It is impossible to accurately determine on what basis the network makes its decisions. It could be that the network has distinguished a pattern which is present between browsing patterns and is in fact not looking at the complete browsing pattern itself.[DJM1] However, further research was undertaken in order to correct the problem, since there may be other applications, using a similar neural network, where the information is required instantaneously.

In order to solve this problem it was necessary to try and determine what it was that the neural network was identifying as each construct. This was achieved by examining the inputs when the neural network changed from outputting one construct to another.

It was reasoned that the constructs exhibited successions of new nodes, successions of old nodes and rapid changes from new to old and old to new. For example, a path can be described as a succession of new nodes or a succession of old nodes, similarly a loop is a path with the addition of an old node on the end. The neural network may therefore identify a path as a succession of new nodes and a loop as a path with a sudden change from new nodes to an old node. Similarly a spike is a succession of new nodes followed by a succession of old nodes and a loop is a mixture of new and old nodes. It can therefore be seen that there are two distinct elements that the neural network may be using to identify a particular browsing construct, namely successions of the nodes and a change from new to old and vice versa. This is summarised below.

Path succession of new or old nodes (note a succession of old nodes would usually be part of a spike).

| | |
|---|---|
| Spike | succession of new nodes, followed by succession of old nodes. |
| Loop | succession of new nodes, followed by a rapid change to old node. |
| Ring | many rapid changes from new node to old node, with a higher degree of new nodes. |

### 6.7.5 The Feature Detector Neural Network

A new neural network, called the "feature detector", was therefore designed to take the above information directly into account (described in the results table below). The purpose of the feature detector neural network was not to increase the performance of the BPR; instead it was designed to provide further insight into how the neural networks were identifying browsing constructs. It is not necessarily a problem that the first BPR neural network configuration identifies browsing constructs one step behind. The situation is analogous to recognising a hand-written sentence; it does not matter that the micro process of recognising lines and curves is a step behind, so long as the macro process of recognising the sentence can be carried out. Further, the feature detector neural network is tightly coupled to the hyperspace representation and current browsing constructs. If further research indicated that other constructs exist and/or new hyperspace representations were to be employed (such as adding extra parameters) then the feature detector may no longer be valid.

The “feature detector” neural network uses two types of inputs, in contrast to the standard BPR neural network, which uses one (a set of time-shifted inputs). The first type of input to the feature detector uses several time-shifted inputs in the same fashion as the original neural network, although a reduced amount were used in comparison to the original neural network. This input type is the feature detector element, since it is attempting to identify the particular feature of new nodes changing to old nodes. The second type of input, the history data, is a single input representing the mean value for input data, previously presented to the neural network. For example, the last 16 movements can be presented to the neural network using 5 inputs, the first four being the feature detector and the remaining 12 are converted into a mean value and supplied as one input (the history data). The new neural network therefore was reduced in complexity by virtue of it having less inputs then the original neural network and is shown in figure 6.7.5.A.

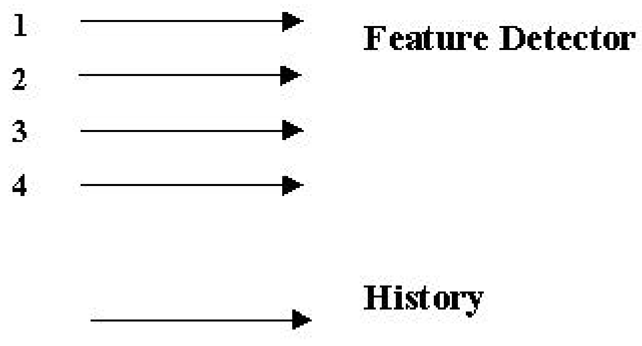


**Figure 6.7.5.A The Feature Detector Neural Network**

The feature detector is designed to detect the rapid changes from new to old nodes and vice versa and the history data determines the average node value and is thus able to detect path elements.

**Table 7 "Feature Detector" neural network**

| Network | Inputs | Outputs | Hidden | Training Set Size | Results Training Set | Results Test Set |
|---|---|---|---|---|---|---|
| 2 | 4 Real | 4 | 10 | 500 | 88% | 85% |
| 3 | 4 Real | 4 | 12 | 500 | 86% | 85% |
| 4 | 4 Real | 4 | 14 | 500 | 85% | 83% |
| 5 | 6 Real | 4 | 18 | 500 | 80% | 75% |
| 6 | 6 Real | 4 | 18 | 500 | 78% | 76% |
| 7 | 6 Real | 4 | 18 | 500 | 79% | 74% |
| 8 | 6 Real | 4 | 18 | 500 | 78% | 71% |

Results with the modified neural network produced similarly high results to the previous neural network. Upon examination of the output data itself, it was discovered that the neural network was indeed able to identify constructs instantaneously.

The “feature detector” configuration is able to recognise constructs as soon as they have been completed, whereas the standard network recognises a construct one step behind. However, this is not necessarily an advantage in the case of identifying browsing patterns, since a browsing patterns is composed of many such browsing constructs. The above experiments demonstrate that the neural network is able to identify browsing patterns based upon Canter et al’s (1985) browsing constructs. Chapter 10 describes some future work that may provide more information regarding browsing patterns.

### 6.7.6 Recommended Configuration

On the basis of this experimentation the recommended neural network configuration is 16 real inputs, 4 outputs, 12 hidden neurones with 500 training examples (table 2, row 8). This configuration of neural network produced trained networks that were good at extracting browsing patterns, whilst being simple enough to train and execute quickly. The feature detector neural network is recommended for producing networks that can recognise individual constructs immediately. The configuration for this neural network is 4 real inputs for the “feature detector”, plus one history input, 4 outputs, 10 hidden neurones with 500 training examples (table 7, row 2). Both these configurations have been selected as the best in the above tables, again because they produce good results with acceptably simple configurations. Note only one of these neural networks needs to be incorporated in a finalised Hypernet design.

### 6.7.7 Using the BPR with real student movement data

The BPR has been tested with real movement student data in an attempt to establish whether it is able to generalise and recognise real movement data. The evidence, shown in the results tables, is that it is able to identify browsing constructs as the hypermedia is navigated. As the hypermedia is navigated the relevant construct values can be seen to change (this was accomplished by recording browsing information as Hypernet was used and then using it with the BPR). Initially all output values are low or zero. As movements are made the path construct increases, indicating a longer and longer path. If navigation is turned round then the spike construct rises and the path construct slowly decreases. Long spikes give higher values of path than short spikes, since they exhibit a higher degree of “pathiness”. Rings too exhibit some pathiness, if there are a lot of rings close together, as indicated by intermittent old nodes, then the path value reduces to almost zero and the ring value rises to a high value. Loops are always associated with mid to high path values, since they invariably have large sections of new nodes. The experiments described above demonstrate that a system has been developed that can recognise the browsing constructs identified by Canter et al and predicted as being "more complex and subtle" than systems previously available. Further, the system is able to identify more complex, temporal, patterns not previously proposed. This therefore provides the potential to identify "the psychologically significant aspects of the movements of users of data structures" (Canter et al 1985, Chapter 5).

## 6.8 Discussion

Representing browsing patterns is not a simple matter and there are a number of alternatives. The chosen method of using a value to indicate when the node was last visited was used because it fits well with Canter et al’s (1985) description of browsing patterns. However, although Canter’s browsing patterns can represent the movements made by a user of an information structure, they do not represent the entire wealth of browsing information that is potentially available in a hypermedia system. Further information could, for example, be obtained by examining time spent at each node by the student, link types used and granularity of the node. However, this extra information might cloud the issue of diagnosing the student’s overall information seeking goal. For example, time spent at each node could be influenced by factors external to the hypermedia, such as the student being interrupted. Other factors such as node and link types were not used for the BPR because their use would more tightly couple the BPR to the hypermedia architecture and therefore reduce the value of the browsing patterns obtained in respect to other systems, most notably the WWW. The use of link types as part of the browsing pattern passed to the BPR would also limit the author to using only those link types that were used in the training of the BPR. Currently the author is free to invent their own link types since the BPR is unaware of any specific link types. The key point is that link types are content-based in that they are designed to convey a particular element of structure or idiosyncrasy of the domain that they are used to represent. As such, link types have a direct influence upon how a student navigates the domain that is unlikely to translate to a domain that does not use that particular link type.

The addition of these extra elements of information leads away from content-less browsing patterns and towards content-based browsing patterns, in that the information collected is more likely to be idiosyncratic to a particular domain representation. Therefore a balance must be struck between the amount of information that can be collected from the student and the portability of the overall information collected from a population of students. If the correct balance is maintained then it is possible that the information can be used to extract information about the student in other domains or even other hypermedia systems. If the weighting of information extracted is balanced towards a particular domain representation then it is likely that the information extracted will not be applicable to other domains, although it may appear that the information extracted is more useful for that particular domain. This raises the issue as to whether the information that is extracted in order to maintain a domain-independent balance is useful. Canter et al (1985) suggest that it may be, although Micarelli and Sciarrone (1998) take a different approach to browsing patterns in that they do attempt to extract as much information as possible and therefore take a totally domain-dependent approach. The BPR and FRB, discussed in the following chapter, may therefore be used as a tool for further research to investigate this issue.

Another major issue, discussed within this chapter, was the suitability of the neural network approach to the problem of recognising browsing constructs. Canter et al (1985) noted that although their diagrams of browsing patterns look simple, in reality the patterns become intermixed and complex and hence “present computation problems”. The problem itself can be thought of as a pattern recognition problem in a potentially noisy environment, so that it is necessary to distinguish patterns that are mixed together and therefore obscuring one another to a degree. The neural network literature states that that this sort of problem is exactly suited to a neural network solution (Masters 1993, Skapura 1996).

The neural network architecture used produced interesting results concerning the speed of recognition of individual browsing constructs. The initial design for the neural network could only recognise browsing constructs one step behind. However, it has been argued above that this is not necessarily a problem since an overall browsing pattern is formed of many browsing constructs and the final construct is eventually recognised. Nevertheless, a second architecture was designed to correct this problem. The major issue arising from this is that there is a choice between the two architectures, since the second architecture can recognise browsing constructs immediately but has the disadvantage of being more closely related to the hypermedia design and so may not be so successful with other hypermedia systems. In the light of the above issue concerning the portability of browsing pattern data it is strongly suggested that the former architecture is the more appropriate, for the following reasons. The argument is similar to that between the approach employed by Micarelli and Sciarrone (1998) and the system employed by the current research project (BPR and FRB). Micarelli and Sciarrone’s approach is domain-dependant because it uses information from within nodes (indirectly), thus allowing the gathering of more information, resulting in a system that becomes useable much more quickly. However the information so gathered cannot be reapplied to other domains. Similarly, the initial neural network design takes longer to recognise browsing patterns, but achieves this in a way that is not tightly coupled to the domain itself and so the results offer much more potential for portability. Therefore, this chapter has demonstrated a design for a system that is capable of recognising the browsing patterns identified by Canter et al (1985) in a time appropriate for aiding the student while they use the system.

## 6.9 Conclusion

This chapter had detailed the design of the neural network for the Browsing Pattern Recogniser. Experiments with numerous configurations of neural network have demonstrated that a Multi-Layer Feed-Forward neural network with 16 time-shifted inputs, ten student level outputs, 18 hidden neurones is able to correctly classify browsing constructs generated by the Virtual Student Program. Although this paradigm could recognise browsing constructs successfully, it was determined that it was unable to accomplish this instantaneously, instead it was a step behind. Although this was not a serious impediment for this problem, a second neural network design was implemented. This neural network used two types of inputs, a feature detector and a history input and was able to identify hypermedia browsing constructs instantaneously.

The following chapter is concerned with combining the browsing construct information generated by the BPR neural network described in this chapter into a browsing pattern and utilising it to determine information about the student, using the Fuzzy Rule Base.

Page: 157

[DJM1]NOT SURE ABOUT THIS PARAGRAPH

# Chapter 7

# The Fuzzy Rule Base

### 7.1 Introduction

It would be educationally useful if a system agent could infer, from a student's use of a hypermedia domain, information regarding the student's ability with the domain and the tasks for which they are using the domain. This would be educationally useful in that the complexity of the domain and user interface could be reduced or increased, in-line with the pedagogy proposed by Elsom-Cook (1989) and Woods and Warren (1995). Further, if the task that the student were using the domain for was known then it may be possible to tailor links to more accurately facilitate this task. This chapter is concerned with investigating one possible approach to this problem, namely the use of fuzzy logic techniques to record and process hypermedia browsing information.

The Fuzzy Rule Base (FRB) is part of the student model and has been designed to map the browsing constructs, as identified by the Browsing Pattern Recogniser (BPR), to a student ability level, as identified by the Tutorial Supervisor (TS), and tasks as provided by tutorial nodes. This provides a potentially useful method for obtaining information from the student without the need for direct interactions. The FRB could therefore be used in other hypermedia based systems that are not necessarily tutoring systems and hence do not involve the user in direct tutorial sessions, which would render the use of a Tutorial Supervisor impossible. The potential benefits of using browsing information are described in chapter 5 and Canter et al (1985) and can be summarised as the ability to extract psychological information about the student or user of a hypermedia system, such as their information seeking goal and their ability. This psychological information may indicate, for example, that the student is lost, or that the student is interested in the domain generally. Such psychological information may provide additional information about that student that cannot be obtained from traditional methods, such as extracting the frequency of keywords from nodes that the student has previously visited. Once obtained this information may be used to tailor the domain to more closely resemble the student's information seeking requirements. For example, if the student is seeking general information then it is not desirable to use keyword extraction to further tailor the domain to a more specific level. This drawback is acknowledged by advocates of keyword based agents (Armstrong et al 1995), and the use of content-less browsing patterns has been suggested as a possible solution (Canter et al 1985).

The FRB serves two purposes; firstly it is an aid for the student in that it extracts the aforementioned psychological information about the student which enables the system to alter the way it presents links to more suit the student. The information stored within the FRB could also be used to augment a browsing agent that would ordinarily base its evaluations on keyword extraction alone. The keyword approach effectively limits the agent to only being able to offer more detailed information, based upon what the student has previously been interested in (Armstrong et al 1995). In reality the student may not be interested in what it is the agent is offering. The FRB offers the potential to diagnose this information. Secondly, the FRB acts as an aid for the hypermedia researcher, in that it offers the potential to identify and store the aforementioned information of psychological importance for later study. This may be useful in that it may allow future hypermedia systems to be designed with beneficial browsing in mind. McAleese (1990) states that different interfaces can induce different browsing strategies from hypermedia users (text based and graphical browsers), it therefore seems logical that altering the interface dynamically as the system is in use may also alter they way students browse. This is a potentially difficult area, however, since the range of possibilities by which the interface can be altered is likely to be constrained by the need to maintain some generality, so as not to confuse the student (Preece et al 1995).

The FRB uses ideas from both neural networks and fuzzy logic; an approach termed "Neurofuzzy" (Jang 1997). The purpose of the FRB is to provide an adaptable structure, similar in operation to a neural network, whilst providing interpretable information, similar to a rule base. The advantage offered by neural networks is that they are to a large extent autonomous and robust to noisy or indistinct data, such as complex browsing patterns (Gurney 1997). However, the advantage of being able to read the data is that the hypermedia researcher could use the information to better design hypermedia systems. A neural network could be employed to perform the conversion of browsing patterns to student ability, and the tasks that are being undertaken by a student. However, it would not be possible to determine how this process is achieved, because the information inside a neural network is not directly readable (Kosko 1997). The FRB, on the other hand, is interpretable because it employs elements of neural networks and elements of fuzzy logic, which allows the FRB to act as an adaptable database, recording how browsing patterns relate to different types of student and task. The ability to directly examine the rules that the FRB has extracted from a population of students is useful, since the information may be used to design hypermedia systems better suited to browsing, or to better understand a student’s information seeking requirements. This is because the FRB is a record of a complete student population’s interactions with the hypermedia domain. This can then be used in different ways. For example, if it is suspected that students were forced by inadequacies of the interface or domain design to browse in an inefficient way, this could be determined by examining the information that the FRB has recorded. Inadequacies in the interface design are likely to be identifiable as inconsistent browsing patterns, an example of which would be the “wandering” pattern identified by Canter et al (1985).

The overall argument for the employment of the FRB is that if it is used to record information from a sufficiently generic hypermedia system, then this information could be used as a benchmark for other hypermedia systems. For example, a new hypermedia design might generally encourage more beneficial browsing than the generic design; it is therefore likely to be a better design than the generic design. The second major advantage of using a FRB is that once it has recorded enough browsing information then the information may be used to gain an insight as to what it is that the student is trying to achieve with the hypermedia, presentation of links may then be altered so as to aid the student in this task. Both Canter et al (1985) and Micarelli and Sciarrone (1998) have indicated the validity of this approach.

A further potential capability of the FRB is the ability to link browsing patterns to particular tasks that a student may undertake whilst using a hypermedia system. As with the link to ability, this is an emergent capability of the FRB in that such information is obtained over time as students use the system. The FRB can link browsing patterns to tasks as set by a tutorial node and defined as a particular task by the original author ("acquire structural knowledge" or "find a particular piece of information", for example). A detailed taxonomy for hypermedia browsing tasks was beyond the scope of the current research project. Once the FRB has been trained on such data it is potentially capable of determining that a student is undertaking a particular task by examining their movements alone; again this has been cited as a possibility by Canter et al (1985). This is useful for other hypermedia-based systems that may not involve a student or user in a direct dialogue, such as the World Wide Web, as argued previously in Chapter 5, and summarised as the potential to extract psychological information about the student based upon previous student's interactions with a hypermedia domain. This information may then be used with other, more traditional methods, such as keyword extraction (Armstrong 1995) to obtain information about a WWW user, with a view to aiding the user in selecting links and finding relevant information.

The FRB is therefore potentially a useful addition to hypermedia systems. This chapter therefore contains details of the FRB, in terms of its design, training and experimental results with a Virtual Student Program (VSP).

## 7.2 Fuzzy Browsing Patterns

The FRB is a set of fuzzy rules linking fuzzy browsing patterns to student ability and the tasks that they were undertaking when they used the hypermedia. Fuzzy browsing patterns are designated "fuzzy" because they are not discrete, in that several similar patterns may be grouped together as being similar. They are described by values of the lower level browsing constructs. This is because the BPR outputs a browsing pattern as a set of levels for the low-level browsing constructs (spike, ring, path and loop) that it has been trained to recognise. For example, a browsing pattern may consist of:

10% Spike 50% Path 30% Ring 10% Loop

These represent running totals, as identified by the BPR. There are a large number of possible browsing patterns (any combination of the percentage values). However, many browsing patterns are closely related to each other, in that they blur into one another (e.g. 11% Spike 49% Path 30% Ring 10% Loop, is highly related to the above browsing pattern, since it varies from the original pattern by only 1% in two of the four constructs). Fuzzy logic allows the distinction between browsing patterns to be blurred, thus removing the potentially counter productive burden from the system designer of defining cut off points between browsing patterns. The use of low-level browsing constructs allows the system to recognise any possible browsing pattern based upon the previously identified low-level constructs, given the argument of Canter et al (1985) that all browsing patterns can be expressed as a combination of the constructs. One possible drawback of this method is that some hypermedia designs may enable the use of other, possibly more complex, constructs, which would allow the identification of more complex browsing patterns. However, it has been previously argued by Canter et al (1985) that the aforementioned constructs represent the basic components of all browsing patterns and that they therefore can be used to represent the majority of browsing information. Further constructs may be used to provide more finely tuned browsing information; this is discussed further in this chapter and in chapter 10 as part of further research.

It may become apparent that certain types of browsing pattern are commonly used (the FRB demonstrates this by showing higher activations for the relevant pattern). Indeed labels, such as "scanning" and "wandering" could be assigned to such common browsing patterns. This would be useful since the majority of browsing patterns that are actively used by students may be clustered together into groups (browsing patterns that are similar being those that differ in their construct compositions by a small amount, as previously described). Each browsing pattern group may then have some information associated with it, for example, “generally used by low level students who are seeking a specific item of information”. It is natural to provide such groups of patterns with an identifying name so that they can be more readily discussed.

Note that the above constructs (spike, path, ring, and loop) do not combine in isolation to form a browsing pattern. For example, as argued in the previous chapter, the path construct represents length; therefore a long spike would be denoted by high values for both spike and path. The sequence of the constructs is not taken into account by the FRB. This is a constraint imposed on the current FRB to ensure that the number of identifiable browsing patterns is kept small, since incorporating the order of constructs effectively increases the number of distinct browsing patterns. This was determined as acceptable since the FRB is currently a test case and there is no reason (other than training time, described later in this chapter) that it could not be expanded to incorporate this information in the future. The FRB therefore currently views a browsing pattern as a list of ingredients (its composition is known) but not as a recipe (the order of the ingredients is not known). This is in line with Canter et al's (1985) indices system, in that potentially useful information is extracted, whilst maintaining manageability of the information collected. Canter et al do this by breaking a browsing pattern down into a number of indices representing the browsing constructs. A browser could therefore be described as having a high or low value for each of their indices. They further describe each of their example browsing patterns, shown in figure 5.2.B in chapter 5, as having high or low indices values. Canter et al note that extracting each index was not possible at the time. However, as described in the previous chapter, the BPR is able to extract these indices as a student browses the hypermedia.

## 7.3 The Application of Fuzzy Logic

Fuzzy logic was chosen for this problem because there is a requirement for modelling the above fuzzy browsing patterns (Canter et al 1985, Micarelli and Sciarrone 1998) and it would be desirable to be able to view this information. Once the information has been obtained the hypermedia designer can use it to structure their systems to encourage more beneficial browsing and the information can be used to aid students currently using the system, by altering links to suit the student’s information seeking needs. As described above, a pattern such as "searching" (Canter et al 1985) is a set of similar browsing patterns, of different sizes and compositions, but with similar unique characteristics, that makes the pattern "searching". The “searching” set cannot be a “crisp” set because there is no logical method for separating one browsing pattern into one set and not another. For example, given that one browsing pattern is highly related to another browsing pattern (they may have close construct values, as described above), then it is a simple progression in that pattern *a* is related to pattern *b,* which is in turn related to pattern *c*. However, the relationship between pattern *a* and pattern *c* may not be so strong as the relationship between pattern *b* and pattern *c*. Therefore, it is not desirable to set crisp start and end points for a browsing set, in that it would be necessary to exclude the strong relationship between pattern *b* and pattern *c*, if it is determined that pattern *c* is not sufficiently closely related to pattern *a* to belong to the same set. Fuzzy logic allows degrees of membership; therefore pattern *c* might be a member of pattern *a*'s fuzzy set but to a lesser degree than pattern *a* and *b* (it may also belong to other sets) (Kosko 1997).

Determining to which fuzzy sets a particular browsing pattern belongs is accomplished by using a membership function (described later in this chapter) (Kosko 1997). Members of a fuzzy set should share key elements, this is analogous to handwriting, in that letters may be written differently by different people, but each letter must contain elements that make it recognisable. Neural networks have been used with a high degree of success for handwriting recognition (Gurney 1997, Hagen et al 1996, Masters 1993). The handwriting recognition problem, however, does not require that the neural network provide to the human designer the elements of the letters that make it uniquely a particular letter. It is enough that the neural network is able to recognise the letter. The problem of identifying browsing patterns is different in that it is highly desirable to extract information regarding what browsing patterns look like, which types of students use them and for what purposes they are used (Canter et al 1985). As argued above, such information may help the hypermedia designer to produce hypermedia systems that better suit beneficial browsing, or allow a system to infer information about a student by how they are using the domain.

Since the problem of identifying browsing patterns and linking them to types of student and the tasks they are undertaking requires the identification of non-discrete patterns, this would usually be accomplished with a neural network. However, given that neural networks do not lend themselves to interpretation and in order to allow the hypermedia researcher to examine how the system determines browsing patterns link to student ability, an alternative approach was sought. Symbolic AI is not suitable for the problem since it does not have suitable learning mechanisms that would allow the extraction of patterns without manual intervention from a human (Kosko 1997, Masters 1993). A possible solution was to employ a paradigm that falls between Symbolic AI and connectionist AI. The field of fuzzy logic is such a paradigm (Kosko 1997). A neural network can be trained to act as a rule-base, although it is a black-box system and can therefore be viewed from the outside only, in that rules can only be examined by presenting the network with an input and then examining the output. By contrast, fuzzy logic rule-based systems can be “opened” for examination and modification (Kosko 1997).

Given that fuzzy-logic is appropriate methodology to investigate for the problem of extracting browsing patterns, the major concern was to find a method for enabling the fuzzy system to learn and adapt as a neural network is able, so that the FRB can extract browsing patterns without the need for manual intervention, since browsing information is not currently available. In order to do this some elements of connectionism were used to build a hybrid neural network-like construct; this process is termed "Neuro-fuzzy" in the fuzzy logic literature (Jang, Sun and Mizutani 1997). As described above, the advantage of this approach is that a researcher can interpret the rules. The disadvantage is that the number of rules in a fuzzy rule-base is many times greater than the number of neurones required for a neural network. A greater number of rules reduces the speed at which the system adapts (learns). For Hypernet the number of rules turns out to be manageable, as the experimental trials described below show.

## 7.4 Fuzzy Rules

The BPR neural network has been trained to recognise and output four attributes. These are the browsing constructs identified by Canter et al (1985), namely paths, loops, rings and spikes. The path attribute can be thought of as a length attribute and is used to indicate the size of the individual constructs, e.g. big spikes, big rings etc. Figure 7.4.A shows a browsing pattern as output by the BPR neural network. The BPR outputs an activation level for each of the four browsing constructs.

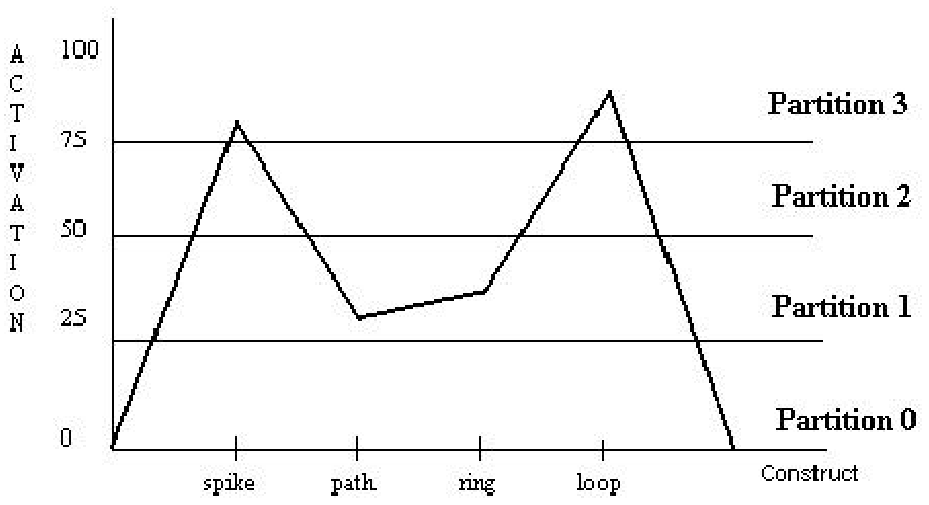


**Figure 7.4.A A Fuzzy Browsing Pattern, using Partitions**

The fuzzy rules are based upon a standard fuzzy rule-base approach (Kruse et al 1994). A browsing pattern (as determined by the BPR) is examined to determine its membership of a particular fuzzy set (by the FRB); it is then mapped to the degree of success that the student achieved whilst using the pattern (as dictated by the TS) and the tasks for which it was employed (as defined by a tutorial node). The fuzzy rule represents all of this information, in that it incorporates the fuzzy set (the set of all possible patterns with which the rule is associated) and information regarding how useful the set of patterns are and the types of tasks for which they are used (based upon previous interactions).

The number of fuzzy rules, used to represent the complete set of possible browsing patterns, is dependent upon the partitioning fuzzy logic method (Jang 1997). The number of fuzzy rules represents a sampling frequency or resolution and not the ability to recognise a discrete number of browsing patterns. Each fuzzy rule has the potential to represent many browsing patterns. However, all of the browsing patterns encompassed by a single rule should be similar (in that many browsing patterns could usefully be thought of as "scanning"). Therefore, the number of rules in the FRB affects how many browsing patterns each rule must represent, in that if the overall number of rules is small then individual rules must represent more browsing patterns, thus increasing the possibility that a single rule must represent so many browsing patterns that not all of them can be thought of as similar. Increasing the number of rules reduces the likelihood of the aforementioned situation happening. However, increasing the number of rules necessitates more training examples (resulting in the FRB taking longer to train). An increased number of rules also increases the likelihood of a set of related browsing patterns ("scanning" for example) being represented by several rules, however, this is not necessarily a problem in fuzzy logic systems as several rules can be "defuzzified" into one crisp value (Kosko 1997). In order to separate an input space into a finite number of rule fuzzy logic partitioning is used.

Fuzzy rule partitioning is a method of representing an input space with a finite number of rules. Several methods are available, namely grid, tree and scatter. Grid partitioning allows a simple separation of the domain based upon its inputs. The disadvantage is that the number of rules increases exponentially as the number of inputs increases. However, grid partitioning offers the quickest method to select a rule, since each rule has a unique index number. Tree and scatter partitioning use mathematical methods to separate the input space. Tree partitioning is complex and computationally expensive, since it requires the use of many membership functions and scatter partitioning is difficult to interpret (Jang 1997). Therefore, the use of grid-partitioning was useful in the case of the FRB since the number of inputs is small, producing a manageable number of rules, whilst maintaining the advantage of quick and efficient identification of rules.

Grid partitioning gives a number of rules equal to the number of inputs raised to the power of the number of partitions shown on the diagram. In this case 4 inputs are used to represent the 4 browsing constructs (path, spike, ring, loop) identified by Canter et al (1985). Determining the number of partitions to use directly affects the resolution of the FRB, since this is a changeable parameter whereas the number of inputs is not. The resolution of the FRB is the number of rules used to represent the total number of possible browsing patterns. In practice the resolution is finite so that a manageable number of rules (in terms of training the rules) can be used. Determining this number for the current research project was not possible since real student browsing data was not employed. Instead, a value was chosen to demonstrate the FRB with the VSP. Four partitions are used in the test case (shown as partitions on Figure 7.4.A), giving a total of 256 distinct rules to represent browsing patterns. This number is then multiplied by the number of student levels the system is to identify, in the test case 10, giving a total of 2,560 rules to represent all possible browsing patterns that can be distinguished by the FRB. It can be seen that if the input space is partitioned twice, then there are only ($4^2$) 16 rules to represent the wealth of possible browsing patterns (multiplied by 10 student levels giving 160 rules in total). Having five partitions gives a total of 5,120 rules. This was determined to be too large for the test case, since each rule must be presented with sufficient training data for it to represent the relationship between browsing and students (cf. 7.5). Note that rules in a fuzzy-logic system do not behave like crisp rules in a rule-base, since relationships can be formed between rules/sets (Kosko 1997). The rules are therefore not separate from each other, which allows the clustering of related rules, should the clustering exist in the browsing patterns presented to the FRB. The upshot of this is that the FRB has a resolution of 2,560 rules (in the test case) to represent potentially fewer major browsing patterns (since a "scanning strategy" may in reality be composed of many related rules).

In the test case, grid partitioning allows the output of each browsing construct to be in one of 4 possible categories. The output from the BPR of each construct may fall in the range of 0-99. Partition 0 is where the value falls in the range 0-24, partition 1, 25-49, partition 2, 50-74 and partition 3, 75-99. Therefore, there is a rule in the FRB to represent every possible combination of partitions and constructs ($4^4$). Each fuzzy rule therefore represents a range of possible browsing patterns. The rules themselves may be distinguished by their binary patterns. Each construct has two bits associated with it (giving four possible combinations); a fuzzy rule may then be represented by its eight-bit pattern (4 constructs with 2 bits each). Figure 7.4.A shows the activation level being split into 4 partitions (partitions 0-3). The activation level for each browsing construct (from the BPR) will fall into one of the four partitions. In Figure 7.4.A the spike construct falls into partition 3, the path into partition 1, the ring into partition 1 and the loop into partition 3. The browsing pattern therefore belongs to a fuzzy set described by the partitions 3,1,1,3. Other patterns may also fall into these partitions and are therefore similar but not necessarily identical since the actual construct values may not be identical. For example, 78 Spike, 27 Path, 40 Ring, 80 Loop belongs to the 3,1,1,3 fuzzy set/rule and so does 85 Spike, 45 Path, 50 Ring, 77 Loop. They are similar but not identical. Grid partitioning therefore groups these similar patterns together and represents them with a single fuzzy set/rule (Jang 1997). Representing every possible browsing pattern separately would produce $100^4$ or one hundred million rules. This is far beyond the realms of practicality in terms of training and execution. This is a clear indication that fuzzy logic techniques, such as grid partitioning, are required to reduce the problem to a manageable level, however, it does impose a burden upon the system designer to specify the number of partitions, which directly affects the resolution or granularity of the FRB.

The fuzzy-rule itself is used to record and hold information about the types of students that have used the browsing pattern, in terms of their ability as determined by the TS and the types of task that the browsing pattern was used to fulfil. Therefore, independence is maintained between the student ability level and the browsing pattern, in that one pattern may be beneficial for an expert but not for a novice. The fuzzy-rule also records how successful a pattern is, generally, with that particular ability level of student and the tasks that the matching browsing patterns have been used for (as provided by the tutorial nodes). A fuzzy rule takes the form:

**If** (*browsing pattern, student level*) **then** *browsing* is x *beneficial and tasks likely to be undertaken* are $t_1,t_2,t_3,t_4..t_n$.

The "browsing pattern" is a fuzzy set. The student level is involved, to ensure that differences between a browsing pattern for different types of student can be catered for. The fuzzy browsing patterns and student level are linked in the rule to how beneficial the pattern is determined to be (by the TS, since it is able to determine how well the student performed with the browsing pattern that was used in response to a tutorial node task) and the task(s) that the pattern was used to complete (from the tutorial nodes). For example, if a student employed a particular pattern and was determined to have achieved some success, by the TS, then it may be said that the pattern was useful, for this particular student. It may be that with repeated exposure to similar browsing patterns a general trend emerges, in that a pattern turns out to be beneficial for certain types of students generally (Casti 1998).

Using grid partitioning, the FRB takes the form:

| **IF** | $P_1P_2$ | $S_1S_2$ | $R_1R_2$ | $L_1L_2$ | Level | **THEN** Beneficial Value task 1, task 2,..taskn |
|---|---|---|---|---|---|---|
| | 0 0 | 0 0 | 0 0 | 0 0 | 1 | |
| | 0 0 | 0 0 | 0 0 | 0 1 | 1 | |
| | . . | . . | . . | . . | | |
| | 1 1 | 1 1 | 1 1 | 1 1 | 1 | |
| | 0 0 | 0 0 | 0 0 | 0 0 | 2 | |
| | . . | . . | . . | . . | | |
| | 1 1 | 1 1 | 1 1 | 1 1 | 2 | |
| | . . | . . | . . | . . | | |
| | 1 1 | 1 1 | 1 1 | 1 1 | 10 | |

$P_1P_2$ =path, $S_1S_2$=spike, $R_1R_2$=ring and $L_1L_2$=loop and

Where each construct has two bits $X_1$ and $X_2$ and

00 = 0-24%, 01=25-49%, 10=50-74%, 11=75-99% active output from the BPR

Tasks are represented as a running total for each time the browsing pattern has been used for a particular task.

Since grid partitioning is used the columns representing the browsing constructs (P,S,R and L) are organised in a binary fashion (Jang 1997), in that the table starts at 00000000, following entries are then the consecutive binary values (00000001,00000010,00000011 etc.) for each level of student that is to be identified.

Figure 7.4.A shows a sample browsing pattern as identified by the BPR. The matching fuzzy rule would take the form (11 01 01 11) since this is the binary value for the pattern 3,1,1,3 described earlier. There is a matching rule for every student level identifiable by the system. For example if the pattern in figure 7.4.A were employed by a level 6 student then the fuzzy rule (11 01 01 11, level 6) would represent the current browsing pattern. The "then" side of the rule represents the information that has been extracted from repeated interactions with students and shows how successful the pattern has generally been and the types of task to which it has been put. For example, the beneficial value may indicate that the pattern is 80% successful and has been used for task a x times, task b y times etc. The beneficial value is dependent upon how well individual students performed with their task using the browsing pattern. This is discussed in detail below.

### 7.4.1 Generating a Browsing Pattern

In order to determine which browsing pattern a student employed for a tutorial, the complete browsing pattern is processed by the BPR. This involves moving the browsing pattern through the BPR as the student moves from one node to another. The output of the BPR is then continually monitored and recorded. These recorded values for the browsing constructs are then passed to the FRB, once a student has completed a tutorial, and are matched against the fuzzy-rules and hence used to identify the browsing pattern. This is achieved by employing the standard fuzzy logic process of grid partitioning. This is accomplished by dividing each activation value by the partition size (Jang 1997). Since the output of the BPR is in the range of 0-99 and there are 4 partitions, then the partition size is 25, giving partitions 0,1,2 and 3. Once all constructs have been converted into partitions, then the resulting numbers and current student level define the matching fuzzy rule. For example, if the activations for path, spike, ring, loop are 76,55,27,34, then the partition values are 3,2,1,1 (expressed in binary as 11 10 01 01). If the current student is of level 6 then the matching fuzzy rule is represented by the values 3,2,1,1,6 and can be found in numerical order in the FRB. The above method is used commonly in fuzzy logic for classification tasks (Kosko 1997, Jang, Sun and Mizutani 1997) and has been applied here to the specific problem of identifying browsing patterns, since it is highly efficient in terms of processing speed.

Once a fuzzy rule has been found to match the current browsing pattern then the information stored within it, i.e. its right hand ("then") side, is used to provide an insight into the task that the student was attempting to achieve and the likely success associated with the particular browsing pattern. This information linking student abilities and tasks to browsing patterns will ultimately be based upon previous interactions with many students. In effect the FRB must learn this information, in a similar fashion to a neural network, from many examples presented to it, because this information does not currently exist and must therefore be collected. This is the role of fuzzy-rule learning described below.

## 7.5 Fuzzy Rule Learning

In order not to fall foul of the same criticisms of traditional student models, namely overly constraining students to learn in ways that may not suit them, based on possibly incorrect assumptions about the student and the domain (Woods and Warren 1995, Kinshuk and Patel 1997, Hartley 1993, Ridgeway 1989, Bergeron et al 1989, Elsom-Cook 1989), the fuzzy rule base is capable of adapting to misconceptions in the original rules. In practice the rules are learned from a population of students. A mismatch may occur if the population of students with which the FRB was initially trained is not representative of the current population that the FRB is being used with.

In this case the FRB may be allowed to continually train and therefore adapt its rules accordingly. A drawback of this is that it is not possible to allow the FRB to adapt to non-tutoring hypermedia system, such as the WWW, since there is no TS or tutorial node information with which to make a determination about the applicability of the rules. It is therefore important to select the initial training population carefully.

If there is a mismatch between the student’s ability level as output by the Tutorial Supervisor and the beneficial value from the FRB (for example the TS says that the browsing pattern is good but previous interactions say that it is bad) then the rule corresponding to the browsing pattern must be altered in such a way so as to make this mismatch less likely. The TS is used to provide the student’s ability since it has collected the information as part of the tutorial exercises and the information is put to a secondary use by the FRB. Thus, the "beneficial" value of each rule must be altered towards the current value (from the TS). This process is exactly how a supervised learning neural network operates, in that a “total solution” can never be presented to a neural network; instead examples of correct solutions can be presented and the neural network must learn to extract the elements that are common to all correct solutions. For example, there is not one solution for handwriting recognition that can be presented to a neural network so that it can recognise all handwriting; instead a representative sample is presented. Due to this similarity and the need for the FRB to learn to map browsing patterns to student abilities and the types of task that the student is using the browsing pattern for, without the need for human intervention, connectionist search techniques were examined as a possible solution for the problem, this approach is termed NeuroFuzzy (Jang, Sun and Mizutani 1997). Fuzzy systems using connectionist learning techniques are becoming increasingly prevalent and have achieved practical success (Von Altrock 1996, Bossley et al 1994).

### 7.5.1 Multiple Rule Learning

In order to increase the speed of learning in fuzzy rule bases some researchers apply learning to several rules at once, according to a fuzzy membership function (Jang, Sun and Mizutani 1997). The suitability of this approach is dependent upon the application. There must be a strong relationship between rules that undergo learning at the same time. In the Hypernet context one would intuitively expect that there is a strong relationship between browsing patterns at similar ability levels. For example, in the FRB there is a separate rule for a particular browsing pattern at each student ability level. The relationship between the same browsing pattern at, say, levels 6 and 7 is likely to be extremely strong, since the student ability levels are set points along a spectrum of ability. The relationship is likely to reduce as the level gets further away; for example the relationship is stronger for the same browsing patterns at levels 6 and 7, than for levels 1 and 7. This enables the FRB to deal with novice students potentially using different browsing patterns from expert students for the same tasks, or for a browsing pattern being good for an expert but bad for a novice. A second relationship exists between similar browsing patterns; for example, one fuzzy set may represent a browsing pattern that differs in one element only from the original pattern. Other patterns may differ in more than one element and so the relationship becomes weaker. Fuzzy logic allows membership to be defined in degrees (a pattern may be a member of a fuzzy set by 75% say). A membership function can therefore be employed to decide how much learning is applied to related rules (Jang, Sun and Mizutani 1997).

The FRB employs relationships between rules with the same browsing pattern but different student ability level. The strength of each relationship can be increased and reduced for experimentation. Generally, however, the stronger the relationship the quicker the rule base trains, since more than one rule is learning at once. The drawback is that the strength of the relationship should match reality, in that it is not advisable to provide a strong relationship between similar browsing patterns if such a strong relationship does not exist in reality. Therefore the strength of the relationship should be defined by experimentation with real student browsing data. The FRB does allow the same student browsing data to be reused, in that browsing data can be recorded as students use Hypernet. This information can then be applied to any configuration of FRB off-line, thus substantially reducing the time taken to train the FRB. Experimentation with different configurations of FRB does not therefore require the collection of different student browsing data although it is possible for an optimally configured FRB to learn, or continue to learn, on-line, as students use Hypernet. Therefore, the actual strength of the relationship between related rules can be determined by collecting student browsing data and then applying it to several configurations of FRB, one say with a weak relationship and one with a strong relationship. It could then be determined which FRB in practice produced the best diagnosis of a student’s browsing pattern.

### 7.5.2 The Learning Paradigm

Each individual fuzzy rule learns the relationship between a browsing pattern and types of students and the tasks they are using the hypermedia for in response to an output from the Tutorial Supervisor and tutorial nodes. The simplest learning mechanism would be to set a direct relationship between the fuzzy rule’s “beneficial” value and the output of the tutorial supervisor. This would result in the fuzzy rule mimicking the last output of the TS, without taking into account any information concerning other interactions with the TS. However, linking student ability to browsing patterns is not likely to produce such a direct mapping, since it is unlikely that a browsing pattern will be used by every student in exactly the same way. It is likely that occasionally a result is obtained from the TS that is completely different from the current trend of results; therefore the above method would alter the learning to reflect only this information. The result would be a system that can on occasion fluctuate wildly from the correct solution. Instead, a method is required that tracks the best possible solution. Such methods are common in both symbolic artificial intelligence and connectionist artificial intelligence (Rich and Knight 1991).

One possibility is Perceptron learning, which is attractive since the Perceptron is a simple neural network with a simple learning function. Its learning algorithm is:

$W^n = W^{(n-1)} + eP$

Where W = the weight vector

eP = the error expressed as (t – a) target value – actual value.

P = the training example or classification vector presented to the neural network.

(Hagan et al 1996)

**Equation 1 The Perceptron Learning Algorithm**

This learning function operates by moving the Perceptron's weight vector towards a particular solution, as described by several presentations. The solution is represented by the line (vector) separating them. The algorithm operates by moving and rotating this vector. The algorithm is therefore unable to solve problems that are not linearly separable Gurney (1997).

The Perceptron learning algorithm moves the vector towards the solution, by taking into account previous presentations (i.e. the new weight vector is related to the previous weight vector). The vector is therefore a direct product of the neurone's weights. The Perceptron can be expanded to a Multi-Layered Perceptron, which is able to solve non-linearly separable problems. This is achieved by adding more neurones and hence more vectors. These too must be trained into the correct position.

The solution space for the FRB problem is one-dimensional, since the rule must move the "beneficial" value either up or down towards the true solution. The issue is therefore whether the current solution is above or below the true solution. This simplifies the learning, since two-dimensional vectors are not involved. Instead single values are used in the calculations. Despite all of this, however, the Perceptron learning algorithm is not suitable since its learning takes place in large jumps (Hagan et al 1996). This can result in the learning continually over-stepping the solution, resulting in a constantly fluctuating network/FRB. One solution to this problem is to incorporate an idea from the back-propagation algorithm (used to train multi-layer networks).  A learning rate is used to reduce the amount a weight vector is changed, thus giving the learning rule:

$$W^{n} = W^{(n-1)} + \propto(t-a)$$

Where $\propto$ is the learning rate

(Hagan et al 1996)

**Equation 2 Learning Rate parameter**

It can be seen that if $\propto$ is 1, then the learning will jump to the current (though not necessarily the best) solution with every presentation. If $\propto$ is small (0.1 for example) then the algorithm takes a much smaller step towards the current solution and is therefore less likely to be affected by one off results. However, small values of $\propto$ result in slow learning. A further modification to the rule is therefore applied. A momentum term acts as a low-pass filter. Essentially this incorporates the last weight change into the next weight change, resulting in a data presentation that is not leading in the current direction having less of an effect upon the learning. This makes the FRB more robust to exception data (data that does not comply with the general trend) presented to it. The momentum is defined as a fraction of the last weight change applied to the system, and is expressed as follows:

$\lambda\Delta W^{(n-1)}$

Where $\lambda$ governs the contribution of the momentum term

$\Delta W^{(n-1)}$ signifies the last weight change

The above momentum term is then added to the next weight change, giving the final learning rule:

$W^{n} = W^{(n-1)} + \propto(t-a) + \lambda\Delta W^{(n-1)}$

(Hagan et al 1996)

**Equation 3 FRB Learning with a Momentum Term**

If the last weight change was -5 (including the last momentum term), for example, and the current weight change, before the momentum term, was +3, then it can be seen that the current weight change is not in line with the previous trend, in that it is going in the opposite direction to the last learning value. The momentum term therefore adds -5 to the weight change, producing a change of -2. Thus, the weight is still changing in the direction of the previous trend, although to a lesser degree. If the next change, excluding momentum term, is again +3, then the weight change becomes +1 and the direction of learning has changed. However, if the initial trend is restored (-5) then the weight change is -3. Thus the momentum term reduces the effect of sudden changes and the system is robust (Hagen et al 1996, Masters 1993). Filtering out sudden changes is a requirement since if there are few then they are in direct conflict with the trend towards the correct solution (since the learning is generally progressing in one direction and is only being hindered by the sudden changes). If there are many sudden changes then there is no overall trend and hence the FRB will never find a solution or will only find a meaningless solution. If the latter situation were to arise then it would indicate that, contrary to Canter et al (1985), there is not a strong correlation between browsing patterns and types of students and tasks and therefore the application of the FRB, or any other system to fulfil the same process, would be pointless.

In order to increase the learning speed in the presence of some exception data the learning rule has two changeable parameters, $\propto$, the learning rate and $\lambda$, the momentum term. The momentum term allows the learning rate to be kept small, since extra learning is applied from the momentum term. Smaller learning rates reduce the chance of the optimal solution being "stepped over" (Masters 1993), whilst the momentum terms speeds learning up towards the solution, so long as the learning is heading towards a solution. Both a learning rate and a momentum term were employed in Hypernet for the reasons given above. A range of values for the learning rate and momentum term were determined by experimentation described below in section 7.7.

### 7.5.3 Expanded Fuzzy Rule Learning

Learning is increased by updating more than one fuzzy rule in response to an input. This is desirable since it reduces the number of student interactions required to train the FRB. The benefits of this are that the FRB can either become useful, in terms of aiding the student or providing the hypermedia researcher with browsing information more quickly (less examples required to train it), or the FRB can be expanded to more finely represent browsing patterns (the FRB has more rules). In the former case it is important to extract the browsing information as quickly as possible since the information does not currently exist.

Improving the speed of learning is feasible for this application of the FRB since the level input is likely to be highly related from one browsing pattern to another. The level input is strongly related in terms of a level 9 student being similar to a level 10 student. It is therefore reasonable to update similar rules together in response to a single input. For example if a browsing pattern has the level input set to 5 then the identical rules but with level set to 4 and 6 could also be updated, albeit to a lesser degree. If this relationship does not prove to exist in reality then, as stated previously, this can be determined by experimenting (using the same browsing data) with several configurations of FRB.

The function below steadily reduces the amount of learning applied to related rules, the further away they are from the original rule. This function provides a simple curve, which allows the learning to be reduced symmetrically about the current rule (rules equidistant, to either side of the current rule have the same amount of learning applied), similar rules have been applied to “creature learning” neural networks and Kohonen Self Organising [Feature] Map neural networks (Masters 1993, Kohonen 1989). The function is termed a Gaussian membership function and is used commonly in fuzzy logic, its use is described in (Jang 1997), the equation is as follows:

$$g = e^{\frac{1}{2}((x-c)/\delta)^2}$$

where c represents the centre and δ represents the width (termed bias) and e is the exponential constant. The width of the function determines by how much related rules learn, the wider the greater the spread of learning. (Jang 1997).

**Equation 4 Gaussian Membership Function**

Equation 4 provides a symmetrical distribution about the value x. This value is then multiplied with the learning value obtained in equation 2, which is a common process in fuzzy logic and is termed a fuzzy logic membership function (Jang 1997). Thus, the further away a rule is from the current rule, the smaller the value of *g* and the less learning it receives. The δ parameter allows this distribution to be altered. Diagram 7.5.3.A shows two values for δ, demonstrating that smaller values of δ give a sharper decrease in learning for related rules. Making δ smaller reduces the relationship between rules and thus gives the designer the ability to specify the strength of the relationship.

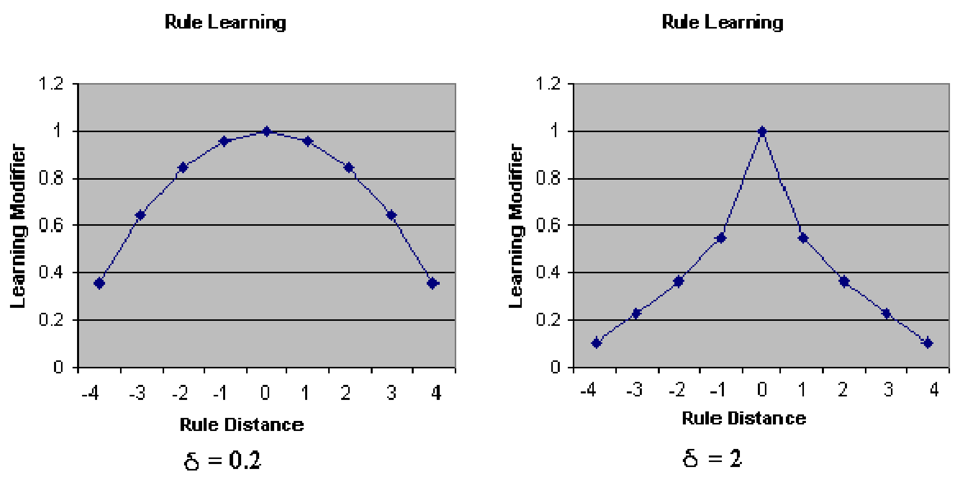


**Figure 7.5.3.A Learning Distribution with a Gaussian Function**

The enhanced learning rule is formed by multiplying the output of the original rule (Equation 3) with the learning modifier value, again a common fuzzy logic process (Jang 1997). This has the effect of reducing the learning for those rules on either side of the current rule, which is represented by the highest point on the above graph.

$$W^{n} = ((W^{(n-1)} + \propto(t-a) + \lambda\Delta W^{(n-1)}))\ \phi(x)$$

Where $\phi(x)$ is the learning modifier value for the rule x rules away from the current rule

**Equation 5 Enhanced Fuzzy Rule Learning**

The above equation must therefore be applied to all related fuzzy rules. For example, if there are ten levels of student and the current rule applies to a level 5 student, then all ten fuzzy rules (one for each level of student) are updated, where the value of x for the ten levels of student would be (level 1, x = -4), (level 2, x = -3), (level 3, x = -2), (level

4, x = -1) (level 5, x = 0), (level 6, x = 1), (level 7, x=-3), (level 8, x=-2), (level 9, x = -4), (level 10, x = -5).

## 7.6 Training the Fuzzy Rule Base

The method used to train the FRB involves the use of a Virtual Student Program (VSP) to produce browsing patterns that resemble real browsing patterns and to produce simulated student data, such as tasks and ability levels which are then related to the browsing patterns. The VSP generates similar browsing patterns and assigns similar task types and ability levels to it, although exceptions are also generated in order to test the robustness of the FRB. The task of the FRB is to map the outputs of the BPR neural network (the fuzzy browsing constructs) to student ability, as identified by the Tutorial Supervisor, and to tasks, as defined by the Tutorial Nodes. The Tutorial Supervisor therefore acts as the instructor to the FBR. This process would usually be conducted whilst the system is in use by real students. However, this was impractical whilst developing the paradigm, since this would require many hundreds of interactions (which could potentially be obtained using the Internet). The FRB was therefore trained independently of the Tutorial Supervisor. “Beneficial” values for each rule were set to a mid-value (50 for the range 0 to 99), indicating that all rules are initially described as being neither good nor bad. The use of neutral values is advised by Kosko (1997).

The simulated students attempt to mimic aspects of real students by providing a mapping between browsing patterns and ability levels and tasks. It is expected that similar patterns are likely to be used by similar abilities of student for similar tasks (Canter et al 1985). The VSP therefore, generates data that generally mimics this behaviour. However, exceptions to this mapping are provided in order to more accurately model real world data, where the relationship may be generally true but not always true. For example, a particular browsing pattern may be generally used by novice students for a particular task, but it may be that a particular expert happens to use it also, even though most experts do not. Therefore, in order to demonstrate the robustness of the FRB, the VSP also generates a certain number of exceptions to the general trend. These mappings may not necessarily correspond to real world situations, in that a certain browsing pattern may be set as expert level, when in reality it is generally used by novices. This is not important since it demonstrates that the FRB can map general trends and is able to adapt to exceptions. It does however mean that a FRB trained using simulated data is of no use to real students, since it has not learned real world mappings. This approach is therefore a proof of concept only, in that it shows that the mechanism is capable of identifying relationships between browsing patterns and types of (simulated) students. It does not show that such a relationship exists in reality, although research by Canter et al (1985) and to an extent that by Micarelli and Sciarrone (1998) suggests that it does. Appendix C provides more detail concerning the Virtual Student Program, which produces the simulated student data.

## 7.7 FRB Results

Results obtained using the VSP demonstrate that the FRB is able to identify browsing patterns and relate them to types of student. Several different mappings, in terms of linking browsing patterns to abilities and tasks, were used in separate experiments in order to prove that the results obtained were not an idiosyncrasy of one set of initial parameters.

The time taken to fully train the FRB depends on several factors. Firstly, it is unlikely that all rules in the rule base will be employed in the same proportion. Indeed it is likely to be the case that only a proportion of the rule-base will be activated, such as only those rules that correspond to useful browsing patterns become trained. The speed of training is therefore increased in relation to the distribution of utilised rules. Secondly, weight changes may occur in opposite directions. For example different presentations may result in opposite weight changes. The introduction of the momentum term was designed to reduce this problem. Furthermore, Higgins and Goodman (1992) suggest a method for removing redundant fuzzy rules, based upon a statistical test, so that the overall fuzzy logic rule-base learning and execution can be speeded up.  Each time a browsing pattern is presented to the FRB one of the rules undergoes some learning (as described above in section 7.5), or several closely-related rules if the expanded learning paradigm is employed (section 7.5.5 above). The expanded learning paradigm was considered useful, since there is a strong possibility that the relationship between fuzzy rules with the same browsing construct values (spike, ring, path, loop), but different student level values is strong. The separating of student ability into levels is a practical necessity; student ability in reality is a spectrum (Elsom-Cook 1989, Woods and Warren 1995).

### 7.7.1 Results Table

Table 8 below demonstrates the use of different parameters for learning rate, momentum and multiple-rule learning bias. For this implementation of the FRB, rules with the same browsing patterns but different levels may learn at the same time. The higher the bias value, the less distinction there is between rules with the same strategy but different levels. All experiments documented below used the same browsing data as generated by the VSP and used a rule base with 2560 rules (representing 4 constructs of ten levels).

Table 8 below shows various configurations of FRB. The iterations are the number of training examples required to train the FRB. The fewer the iterations the quicker the FRB can provide the mapping information to the hypermedia researcher and aid the student. However, it is not simply a case of selecting the FRB configuration that learns the quickest. The actual optimal configuration is dependent upon the mappings found in reality. Therefore, the table below can be used to guide experiments with real student data (as part of further research). For example, if it turns out that the linking between browsing patterns and students and tasks is strong then those configurations that have a large bias are the most useful, since the bigger the bias value, the more related rules will learn in response to one input. Smaller momentum terms and larger learning rates are the most useful when there is little exception data; for (an extreme) example, when all experts use one set of browsing patterns and all novices another. This is because a rule learns more in response to one example, which is acceptable only when a rule is generally moved in the same direction; otherwise large learning rates result in a rule-base that fluctuates and never settles towards a stable solution (the same is true of neural networks (Masters 1993)). In cases where there is a high degree of exception data then the momentum term becomes more useful, since it keeps the learning in the direction of the overall trend. Table 8 below can therefore be used to give an indication of the learning times required for various situations, such as there being a strong relationship between student levels (or not) and the presence of exception data (or not).

The number of iterations shown in the table below shows how many (simulated) browsing pattern presentations were required to move all the fuzzy rules in the rule-base by 10% in one direction. A movement of 10% is determined as statistically significant for a fuzzy rule-base (Kosko 1997). From this point the rules may move further away from their original (neutral) value, meaning that the rule is further enforced (or the certainty that it is correct is increased). This method of learning is similar to neural network learning, when it is never possible to say that the neural network is completely certain (Masters 1993, Kosko 1997). Instead it is necessary to set a threshold, where a value over the threshold denotes one state and a value below represents another. Although Kosko (1997) states that a movement of 10% in one direction is significant, if it occurs within an acceptable time frame for the problem, it can only be used here as a guide for future experimentation with real student data. This is because the trained FRB can not be tested for validity with real students unless it has been initially trained with real students.

Table 8 below shows several trial configurations of the FRB and the number of training examples (iterations) that were required to produce a movement of 10% from the initial neutral value. The configurations demonstrate the effect of altering the learning rate, momentum and bias parameters. The intention of the trials was to determine whether the FRB approach was able to accomplish the mapping of browsing patterns to student attributes given the assumption that they exist, to provide an indication as to how much example data is required and to provide an indication of which configurations may be the best to use for future experiments with real student data.

**Table 8 FRB Configurations**

| | Learning Rate | Momentum | Bias | Iterations |
|---|---|---|---|---|
| 1 | 0.1 | 0.5 | 2 | 424 |
| 2 | 0.2 | 0.5 | 2 | 286 |
| 3 | 0.3 | 0.5 | 2 | 249 |
| 4 | 0.4 | 0.5 | 2 | 199 |
| 5 | 0.5 | 0.5 | 2 | 168 |
| 6 | 0.6 | 0.5 | 2 | 129 |
| 7 | 0.7 | 0.5 | 2 | 159 |
| 8 | 0.8 | 0.5 | 2 | 139 |
| 9 | 0.9 | 0.5 | 2 | 110 |
| 10 | 1 | 0.5 | 2 | 102 |
| 11 | 0.5 | 0.1 | 2 | 259 |
| 12 | 0.5 | 0.2 | 2 | 186 |
| 13 | 0.5 | 0.3 | 2 | 250 |
| 14 | 0.5 | 0.4 | 2 | 173 |
| 15 | 0.5 | 0.5 | 2 | 197 |
| 16 | 0.5 | 0.6 | 2 | 101 |
| 17 | 0.5 | 0.7 | 2 | 151 |
| 18 | 0.5 | 0.8 | 2 | 159 |
| 19 | 0.5 | 0.9 | 2 | 96 |
| 20 | 0.5 | 1 | 2 | 103 |
| 21 | 0.5 | 0.5 | 0.2 | 466 |
| 23 | 1 | 1 | 2 | 63 |
| 24 | 0.1 | 0.1 | 2 | 790 |

### 7.7.2 Discussion

Table 8 above demonstrates that the FRB is able to adapt within a reasonable number of presentations. Experiments 1 to 10 demonstrate that an increasing learning rate reduces the time taken to train the rule base. However, as stated above, smaller learning rates are more desirable. Experiments 11 to 20 demonstrate that an increasing momentum terms also decreases the time taken to train the FRB, although to a lesser degree than the learning rate. This is because the momentum term keeps the learning in the direction of the overall trend. The bias term also has a strong effect on the learning speed of the FRB, large values greatly increase the speed of learning, but sacrifice some independence between novice browsers using the same pattern as expert browsers. However, when the FRB is trained with real student data it will be necessary to experiment with different values for the bias parameter, as stated previously this can be achieved by reusing browsing data. Experiment 21 shows that mid values for the learning rate and momentum and a low value for the bias term produces a FRB that requires a large number of training examples. Experiment 22 demonstrates that increasing the bias and hence applying more learning to related rules, drastically reduces the number of training examples required. Experiment 24 shows that large values of all three parameters produces a FRB that requires few training examples. However, when the FRB is employed with real student data, involving an unknown degree of exception data, sacrificing training (by changing the parameters) is likely to result in a FRB that misses the solution, or does not converge on a solution.

## 7.8 Linking Browsing Patterns to Tasks

### 7.8.1 Introduction

The FRB can be used to attempt to link browsing patterns, as identified by the BPR, to the undertaking of a particular task, such as finding a specific piece of information, or finding the relationship between concepts. A structure such as the FRB is required to perform this operation since the problem is fuzzy in nature. This linking information is formed by the FRB after it has been trained with real student browsing data. Once the FRB has been trained then it is able to offer an indication of what it is the current student is doing, based upon all of the previous student interactions it has been trained with. The FRB therefore functions as a tool for extracting this information, called for by Canter et al (1985). Similar tools have been developed, using neural networks, for performing the same operation with content-based browsing patterns (Micarelli and Sciarrone 1998); however, content-based browsing patterns are domain-dependent and so can not be used to extract information concerning the student without retraining for different domains.

Hypernet provides tasks for the student, via the tutorial nodes, that would not normally be present in an ordinary hypermedia system. Tutorial nodes furnish the student with a task to accomplish and are designed specifically for the use of the hypermedia domain. The domain author defines these tasks for particular purposes. For example, a task may be defined that encourages the student to browse the hierarchical structure of the domain and hence acquire structural knowledge of the domain. This task may induce a particular set of browsing patterns. Since the system is able to identify browsing patterns, and assuming the author of the tutorial node (task) provides information regarding the particular task that their tutorial is designed to induce, then, in principle, it is possible to link these patterns to the task.

Once this training has been completed to a sufficient level, then the FRB may be used with systems that do not employ tutorial nodes, such as general hypermedia system or WWW navigation. This can be achieved by employing a BPR to recognise a student’s browsing pattern and then using the FRB to extract the corresponding rule and all the information that was recorded within that rule whilst it was trained. For example, the student’s browsing patterns may have been used in the past by a number of students to locate specific information and may have been better suited to more advanced students. The information from the FRB, which is readable, is of general use to hypermedia researchers, because it is a readable record of experience the FRB has had with all the browsers it has encountered.

### 7.8.2 Mechanism

Once a tutorial node is visited by a student and a task is assigned (for example, "find the satellite which is the closest in size to The Moon"), then the output of the BPR is monitored and all changes are recorded. No further processing is undertaken until the student completes the tutorial, by re-invoking the tutorial node (by selecting a button on the browser). The complete record of browsing patterns between these two points in time therefore represents a complete browsing strategy employed by the student to complete the task.

The FRB takes the degree of success achieved by the student with the task. E.g. did the student perform well (their browsing pattern suited the task), or poorly? This information is obtained from the TS, in that the browsing pattern was induced by the task provided by a tutorial node and completes when the student reinvokes the tutorial node and is graded by the TS. The FRB then records within the rule for the current browsing patterns that it was used for the current task. Over time the FRB builds up the number of times the browsing pattern was used for each task that can be set by a tutorial node. This information can then be used in future when there is an absence of task data (because tutorial nodes are not being employed) to give an indication of what tasks the student might be trying to achieve based upon this recorded data.

Information about the task can be recorded in one of two ways. The simplest method is to record which tutorial nodes induced a particular browsing pattern. For example, the student is given a task by a tutorial node, the student then browses the hypermedia and the pattern is recorded and stored by Hypernet. Once the student achieves their goal they then re-invoke the tutorial node (by selecting a button on the Hypernet browser). The system can then grade the student, using the TS and process their browsing pattern using the BPR/FRB combination. This process involves the identification of the rule in the FRB that matches the student's browsing pattern used for the task induced by the tutorial node. The current task can then be recorded with the matching rule. Thus, after a period of interactions with many students, the FRB holds information linking browsing patterns to a number of tutorial nodes. The hypermedia researcher may then examine the browsing patterns used and the tasks stored within the tutorial and attempt to distinguish the relationships between the types of task. A second approach would be to encourage the domain author to define tasks, which are likely to induce different browsing patterns and then record this information within the tutorial node, thus moving the burden from the researcher. For example, the author may set the task “find out the name of Pluto’s Satellite (Moon)”. This task involves the student finding one node (“Charon” attached to “Pluto”) and reporting the result. The student is therefore likely to use a searching browsing pattern (searching for a specific piece of information). Another task might involve a more complex browsing pattern, for example “find out which satellite (Moon) is the biggest in the solar system”. This task requires a specific piece of information, but it is likely to require that the student search more of the domain. The task is therefore likely to induce a more complex search strategy, and hence more complex browsing pattern, than the former task. Other tasks might induce the student to browse the domain, e.g. “which satellite (moon) do you consider to be the most likely to harbour life?” or “are there any examples of major comet impacts?” Complex questions may not have a simple answer, and the domain author may therefore indicate which nodes the student should have visited in order to answer the question. If the author is encouraged to define tasks that are likely to involve different browsing patterns, such as the above, and then records this information within the tutorial node then the FRB can subsequently record which types of task have been used for a particular browsing pattern.

### 7.8.3 Interpreting the FRB task Rules

Once the FRB has been sufficiently trained and hence information is made available linking browsing patterns to tasks, then it becomes possible for a hypermedia system to infer of what tasks a particular student is undertaking by examining their browsing patterns only. This would provide the human tutor with an indication of what it is the student is trying to achieve. For tutoring systems, such as Hypernet, it may be desirable to identify what it is a student is trying to achieve, so that they do not have to be involved in a possibly distracting dialogue with the system. Such a situation may arise for expert students who are involved in their own discovery learning. However, a student may be an expert in a particular area of the domain, but not in others. If they enter a new domain they may then require the extra guidance offered by the system to lower level students, for example, reducing the complexity of the domain by offering less, but relevant, links. This mechanism offers a way to diagnose this problem without the need for a direct dialogue, since their browsing pattern may be sufficient to indicate that they are becoming lost or confused.

Furthermore, information linking browsing patterns to tasks is of use to hypermedia researchers and designers, since it may provide important information pertaining to structuring and designing hypermedia domains, in that it may be possible to design domains that encourage beneficial browsing. Again, research has indicated that different hypermedia interfaces tend to induce the student to use different browsing strategies (McAleese 1993), it would therefore be desirable to alter the presentation of information in order to produce the most beneficial browsing pattern, should it be known which browsing pattern is the most beneficial for a particular task.

## 7.9 Discussion

The FRB is premised on the existence of a link between different types of students, the tasks they are using the hypermedia for and the patterns that the students make as they use the hypermedia. It has been argued in this chapter and previously in chapter 5 that evidence for such a link is provided by Canter et al (1985) and Micarelli and Sciarrone (1996, 1996b, 1998). Given this, the issue becomes whether this information can be collected in a practical way and then used to aid the student in their navigation. The FRB was designed to enable the investigation of this issue.

The FRB approach is intended to derive information about the student, from their content-less browsing patterns, and then use this information at the hypermedia interface to aid the student. The construction of the FRB has been an exercise to attempt to determine whether a system can be built to record browsing information and link this to the other aspects of the student obtained by the system, such as student ability, and task types. The experiments, using simulated data, described above demonstrate that this is practical, although they do not show that such a link exists in reality or that to collect it would be useful. However, once this information has been recorded from real students it is anticipated that it might be useful for gaining a better understanding of how to design hypermedia systems to suit beneficial browsing. Furthermore, such browsing information may be useful for general internet navigation agents that are currently able only to extract information regarding a user's browsing interests from keywords and are thus unable to determine whether a user is interested in more specific information (and hence keyword extraction becomes useful) or not. For example, most web agents assume an increasing interest in information extracted from the student's current node, in that if they visit a node with a high preponderance of the word "Jupiter" then they are offered more nodes containing the word "Jupiter". However, it may be the case that the student is no longer interested in "Jupiter" and hence it is inappropriate to offer more nodes on the subject. Proponents of web agents acknowledge this drawback (Armstrong et al 1996). Browsing information may be of use in determining whether a student is interested in more detailed information or by contrast is interested in more general information.

A key issue in the choice of fuzzy-logic for the task of deriving information about browsing patterns was that fuzzy-logic provides a mechanism whereby the information stored within can be viewed, interpreted and manipulated (Kosko 1997). These attributes do not apply to neural networks, which are more powerful for pattern recognition tasks (and the current problem can be thought of as a pattern recognition task) than fuzzy logic systems. However, it was considered paramount that information within the FRB could be viewed and extracted so that hypermedia researchers could interpret it and therefore put it to use in other hypermedia systems.

The experimental work conducted as part of this research project and detailed above indicates that fuzzy logic does offer the potential to perform the task of mapping browsing patterns to student ability and the tasks for which they are using the hypermedia. However, the approach has a drawback in that the configuration of the FRB involves the adjustment of several parameters, each of which directly affects the same properties, namely the number of training presentations required to train the FRB and the FRB's ability to perform the mapping task. However, this problem may be reduced if real student data were to be employed since each of the parameters could be sensibly chosen (or at least the range could be narrowed down), since more information would then be known about the data (such as the degree of exception data and the variety of different browsing patterns). Hence, if the composition of the data is known to some degree then the parameters of the FRB can be chosen in line with the experimental trials, detailed above, that most closely match the composition of the real data. Furthermore, once real student browsing data has been recorded then it may be repeatedly applied to various configurations of FRB. This would be useful for identifying the number of rules (or granularity) required for the FRB. An indication of the degree of exception data would aid the choice of the value for the momentum term. Finally, the overall number of examples collected would aid the choice of the learning rate. A greater number of examples means that the learning rate can be reduced, since there is more data with which to train the FRB it is not necessary to take large steps during training. Smaller learning rates decrease the possibility of the optimal solution being "stepped over", larger learning rates are usually used only through necessity, when there is insufficient training data (Skapura 1996, Kosko 1997). However, the current research does indicate strongly that if the link between student attributes and browsing patterns exists, as Canter et al (1985) suggests, then the FRB is a useful system for investigating and recording that link for further research and further hypermedia systems.

Micarelli and Sciarrone (1998) have adopted a different approach to browsing patterns from that of the FRB in that they use previously recorded case histories of browsing patterns used successfully by students. When the current student requests help from Micarelli and Sciarrone's system an attempt is made to match the current student's partial browsing pattern to previously recorded complete browsing patterns of other students so that the missing elements from the current student's browsing pattern can be offered. The advantage of the FRB approach over that of Micarelli and Sciarrone is that the browsing patterns recorded by Micarelli and Sciarrone's system are not necessarily domain-independent. Micarelli and Sciarrone's approach is based upon previous students' content-based browsing and is therefore domain-dependent. However, Micarelli and Sciarrone's approach does have the advantage that it does not require the vast amount of training data that is required for the content-less approach adopted for the FRB. It should be noted that the FRB approach to browsing patterns and the approach used by Micarelli and Sciarrone are not mutually exclusive, however, the FRB approach was considered appropriate to investigate since it offers a greater potential to be domain-independent.

## 7.10 Conclusion

The FRB is designed to map browsing patterns (as identified by the BPR) to a student level and tasks being undertaken by the student. The design of the FRB makes it potentially useful for a number of applications where a neural network may be used, but for which it would be desirable to either examine the information inside the neural network or to manually modify the information inside the neural network to increase learning speed.

Initial results using simulated data demonstrate that the FRB can converge upon a solution within a practical number of iterations. However, the collection of real student browsing pattern data was beyond the scope of the current research project. Therefore, the FRB has not thus far been tested with a population of real students to prove conclusively that a link between browsing patterns and types of students and task types exists. However, research by Canter et al (1985) and Micarelli and Sciarrone (1998) does show that there is a correlation. However, Canter et al (1985) were unable to provide a practical mechanism for exploiting browsing information and Micarelli and Sciarrone (1998) use a different approach in that they are concerned with content-based browsing, which involves the exploitation of information within the nodes themselves. The FRB may be used as a data-collecting tool as part of Hypernet, in that it is able to extract relationships that may exist between browsing patterns and student types and task types, as students use Hypernet. After the FRB has converged upon a stable solution then the information inside the FRB may be examined and potentially used by hypermedia researcher to aid their design of new hypermedia systems and it may be used as part of a web-agent to provide high-level information about the student's information seeking goal.

# Chapter 8

# The Tutorial Supervisor

### 8.1 Introduction

This chapter is concerned with the practical aspects of neural network design, training and implementation, with specific reference to the neural network systems designed for the Tutorial Supervisor. The Tutorial Supervisor is used to extract the student's ability with the domain modelled by the hypermedia. This is useful in that the student is more likely to improve their knowledge of the domain if they are presented with tutorial material at or just beyond their ability (Bergeron et al 1989). Further, the presentation of hypermedia material can be targeted towards a particular ability of student. For example, offering less links to a novice helps reduce the complexity of the navigating the hypermedia and hence reduces the "cognitive loading" and "lost in hyperspace" associated with hypermedia (Conklin 1987, Theng and Thimbleby 1998). The TS is also used to provide information for the FRB so that the FRB can link content-less browsing patterns (as identified by the BPR) and student abilities (from the TS), this has been described in the previous chapter.

Chapter 4 introduced the Tutorial Supervisor as a black-box system and described how it interfaces and interacts with the rest of the student model. This chapter is concerned with the contents of the black box. The reasons why a neural network has been chosen are explained and the advantages that a neural network provides to the system are discussed. Finally, implications of an automatic assessment system are discussed.

A major advantage a neural network Tutorial Supervisor has over a rule-based alternative is that the neural network is able to adapt its evaluations according to various populations of students, with various student abilities using various domains, without the need to re-specify the rules that relate particular question levels to particular student ability levels (this is discussed further below). The neural network used for the Tutorial Supervisor is able to accomplish this task without any further interaction from the system designer. This is vital for a system that may potentially be employed for a large corpus of students, such as may be found on the World Wide Web, since such environments are not able to directly interrogate the student/user and so information about the student/user must be inferred indirectly (and not from a direct method such as tutorial nodes and a TS).

The utilisation of the Tutorial Supervisor provides a system with the facility to diagnose a student as being at a particular ability level and the system may then consequently tailor itself to that student. Specifically, the Tutorial Supervisor is able to adapt its ability levels to any range of typical student scores. For example, one domain may produce typical scores between 35% and 75%, therefore the Tutorial Supervisor adapts its levels to encompass this range. If the Tutorial Supervisor is then applied to a domain that typically provides scores of between 25% and 90%, for example, then it will readapt its levels accordingly, with no intervention from a human. This is potentially valuable since it removes the requirement for a human domain author from re-specifying student levels in relation to question and task levels.

## 8.2 Rationale for Using a Neural Network

A neural network has been chosen to form the Tutorial Supervisor in preference to a symbolic rule-based system chiefly because, unlike a rule-based system, it is domain independent. It is unlikely that, for example, a high level student produces results in the same range for every type of domain. Thus the rule "IF SCORE >70 THEN LEVEL 10" is only likely to apply to the domain that it was initially defined for. This is the reason why Bergeron et al (1989) use a neural network for their Tutorial Supervisor. Their neural network holds the rules that it has learnt from its training data (the first domain). It is then able to change its rules in response to new data (new domains), by retraining off-line. In this manner the neural network can adapt to misconceptions or inaccuracies in the original rules and adapt to new situations. This would be a difficult and time-consuming process for a symbolic rule-base system, since it would require the re-engineering of the rule-base by a knowledge engineer, in that the new rules would have to be identified and then encoded. In essence, the neural network is doing the job of the human rule designer. Designing such rules is not necessarily a simple matter, since it requires the human designer to examine many student interactions with various questions and tasks so that a valid grading of each question can be made (e.g. this question was answered well by novice students, it is therefore easy and can be presented to other novice students). The situation is further complicated by the possibility that different populations of students (students from different classes or tutorial groups) may have different previous knowledge of the domains and therefore the initial question gradings may not apply to them. It would therefore be helpful if an AI system could be employed to accomplish the task of dynamic question grading and therefore remove some burden from the author. However, Bergeron et al’s (1989) tutorial supervisor is unable to readapt to different domains without being manually provided with new training data and then retrained off-line. The remainder of this chapter describes the neural networks used for the Tutorial Supervisor found in Hypernet, which improves upon Bergeron et al’s (1989) design by allowing the automatic on-line adaptation to different domains.

The neural network architectures examined and implemented for this project can be separated into two classes, Multi-Layer Feed-Forward neural networks (MLFF) (Masters 1993), which was used to form the BPR and Kohonen Self-Organising Map Neural Networks (Kohonen 1989). The Kohonen neural network architecture was considered for the problem of grading students into abilities since it utilises a novel training approach which is useful in this case. The Kohonen neural network uses unsupervised training, which is useful in situations where it is not obvious or clear what is to be identified, or in situations where the network’s task may change over time (Hagen et al 1996). The Kohonen architecture could therefore in principle be used to allow the Tutorial Supervisor to adapt to different populations of students without human intervention.

The Tutorial Supervisor is an expansion of the Tutorial Supervisor designed by Bergeron et al (1989). Hypernet's TS improves on Bergeron’s original specification by allowing the neural network to adapt to both students and questions/tutorials as the system is in use. Bergeron et al’s (1989) system is unable to do this since they employ a supervised learning neural network. The inability to train during active use is a fundamental restriction of the supervised learning paradigm and it is for this reason that a neural network that utilises an unsupervised learning paradigm is used. This provides benefits to the student in that they are provided with questions that are targeted towards their ability.

## 8.3 Tutorial Supervisor Architecture

The same architectures were considered for the Tutorial Supervisor as for the Browsing Pattern Recogniser (Section 6.2), namely, the Multi-Layer Feed-Forward neural network and the Kohonen unsupervised learning neural network (Gurney 1997, Kohonen 1989). The MLFF was initially considered, since as described in the previous chapter, it has been mathematically proven that a MLFF is capable of performing any mapping function. It is therefore generally accepted that the standard MLFF architecture is the first architecture to consider (Master 1993). However, the Kohonen neural network offers an additional facility that was considered useful in this case, namely the ability to continually readapt its question gradings as it is used by students. It should be noted, however, that the Kohonen neural network is a mathematically weaker neural network than the MLFF. In practice this means that it takes longer to train it and it is less likely to perform the correct mapping when the input data is noisy (Hagen et al 1997). It is therefore important to establish whether the input data is noisy and that the Kohonen neural network does produce an acceptable solution in an acceptable period of time. This was the subject of experiments described later in this chapter.

The Kohonen neural network’s ability to learn without human interaction is useful, since the distinction between a novice student and an expert student, in terms of marks at tutorial nodes, may be small, or may vary significantly from domain to domain. For example, the majority of students may achieve marks between 50% and 60%, with a few results between 60% and 75% percent and a few between 40% and 50%. There are therefore two large ranges of numbers that occur infrequently (0-40 and 75-100). If a MLFF neural network were to be used to model the above problem then these mark ranges must be identified beforehand or the neural network’s outputs would have to be designed to produce student ability levels between 0 and 100, in order to accommodate a generic range of scores. Identifying scores beforehand is not likely to be practical since it would require the collection of a large amount of data (with no initial benefit for the student). Designing a generic neural network also introduces the following difficulties. The MLFF neural network must therefore have enough ability levels (outputs) to clearly demonstrate the distinction between students in these highly clustered areas, necessitating an increase in outputs for all areas to cover for all possibilities, even those that are unlikely. The increase in outputs renders the neural network more complex, resulting in a network that is more difficult to train. Further, and most crucially, once the trained MLFF neural network is used for different domains, then there is no direct correspondence between an ability level for one domain and an ability level for another. This is because an output of the MLFF does not correspond directly to a student ability level, since the student ability level may vary between domains. This can be seen in figure 8.3.A.

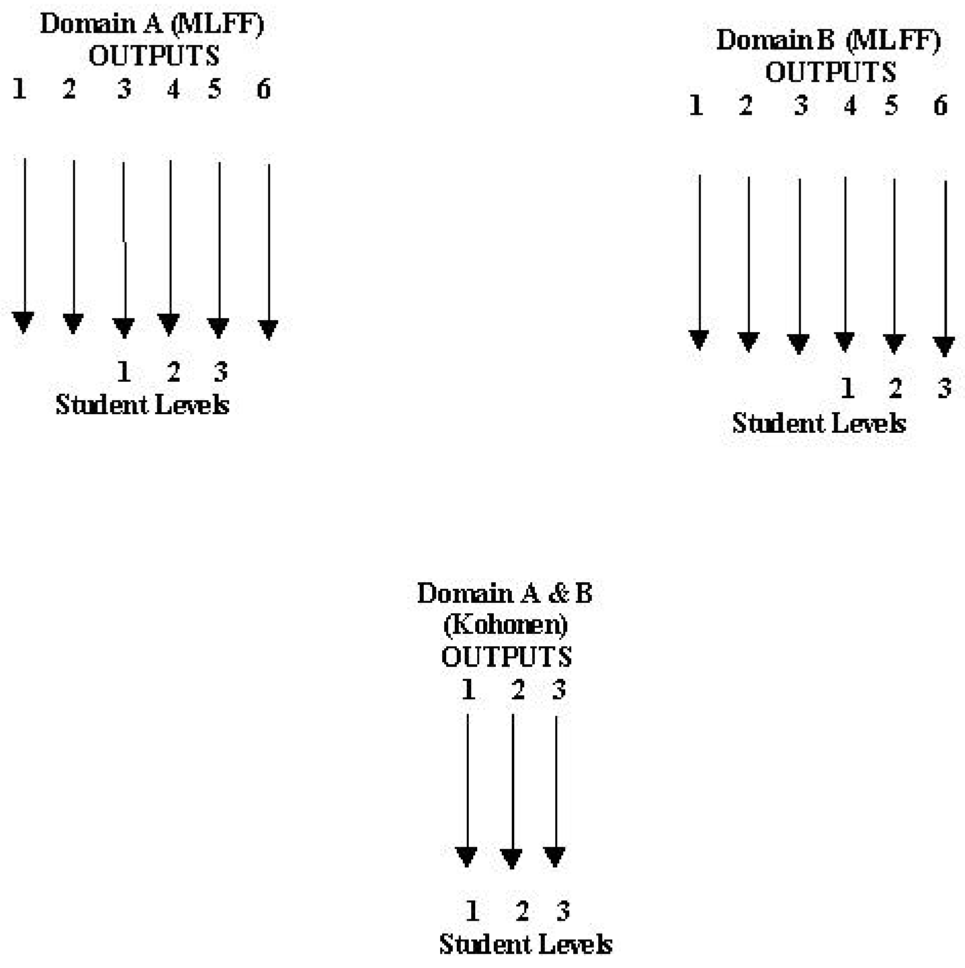


**Figure 8.3.A Using a MLFF or Kohonen neural network for the TS**

In figure 8.3.A Domain A (MLFF) and Domain B (MLFF) activate different outputs, since domain A and domain B have different ranges of scores. This results in different outputs of the MLFF neural network becoming active to represent each ability level. Since the outputs of the neural network are connected to the hypermedia manager and used to determine how many links to offer to a student, it becomes necessary to manually define which active output should be associated with each student ability level. By contrast, the Kohonen neural network automatically adjusts its outputs to match the ranges of student marks presented to it, since it uses unsupervised training. It therefore does not require any manual intervention and can be designed with fewer outputs, since each output directly corresponds to a student ability level. This advantage is considerable and warrants the investigation of the Kohonen architecture further irrespective of the literature claims that it is a weaker classifier than the MLFF.

A Kohonen network can solve the above problem by continually adapting to input stimuli whilst it is being used by students. This is because of the way a Kohonen network operates. A Kohonen network is given a number of outputs by the network designer, representing the number of categories that the network designer wishes the network to identify (the number of required student levels). It is left to the network itself to sort the input data into this number of categories, since it is not implied by the training data itself, as would be the case for a MLFF, in that a MLFF requires an example solution with its training data. If there are ten or more distinct patterns in the data then a correctly trained Kohonen will learn by itself to distinguish them (Kohonen 1989, Masters 1993, Gurney 1997). Therefore, it is irrelevant to the network if the actual values of the input data change; it will still attempt to separate them into the number of categories represented by the number of outputs that it has. The outputs of a Kohonen neural network therefore behave as fuzzy sets, whose boundaries may change over time. The Kohonen neural network is therefore simpler than the MLFF neural network for the TS in Hypernet, since the number of outputs can be kept small since is not necessary to design for all unlikely possibilities (the network will adapt to them if they occur).

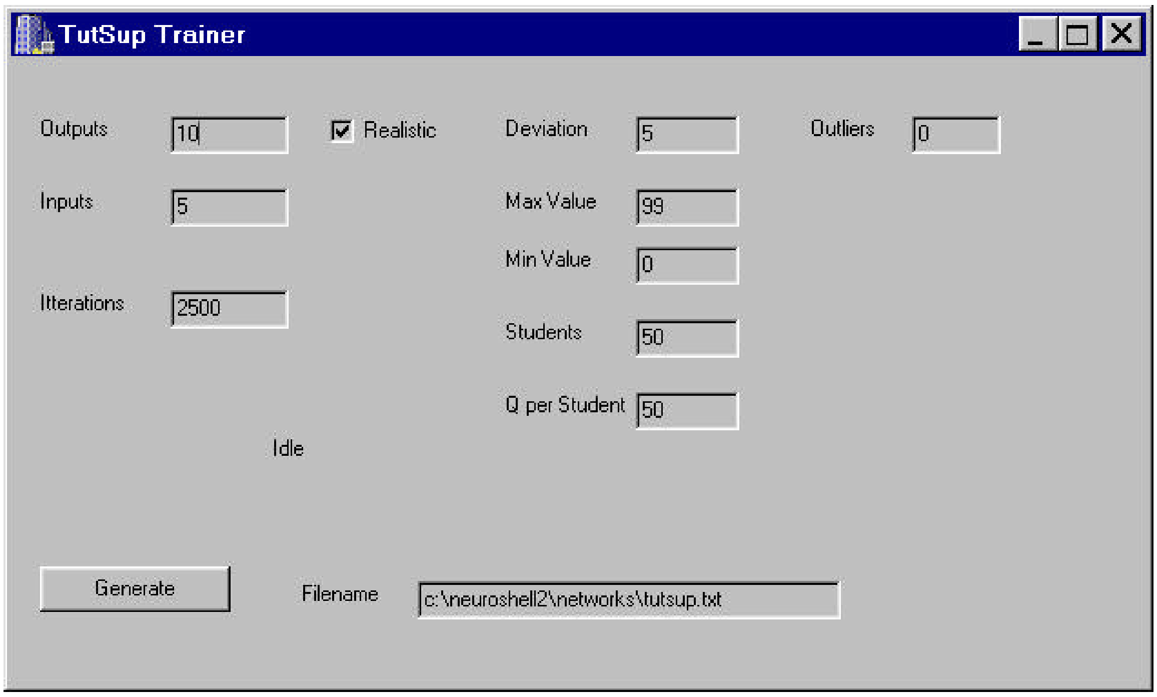
Note that the above situation means that the Kohonen neural network grades a student population according to all the populations it has encountered, since it is always learning, whereas a MLFF neural network always grades a population according to the population it was originally trained with. This situation may not be desirable in certain circumstances. For example, the system may be used by a final year students and then by first year students. It is expected that the final year students will be of a higher level than the first year students. However, since these populations of students use Hypernet separately the TS will attempt to adapt to them separately.  If, for example, the final year students use Hypernet before the first year students, then the TS will adapt to the final year students and be unable to grade the first year students, since they will be all grouped in the low levels. In practice, this situation is not likely to arise because the network requires hundreds of interactions to adapt fully. However, since Hypernet is designed to react to various levels of student and provide them with a certain number of links, it is inappropriate to allow the system to adapt to radically different populations of students (in terms of ability), since the number of links offered is defined by the domain author. For example, if the domain author defines an expert level question as a question that should be offered at a level expected to be a final year student, then it is unwise to allow the system to adapt to a population of first year students and hence still employ the expert level, offering the number of links defined by the domain author to be offered to final year students. It is therefore recommended that the Kohonen neural network only be allowed to continue to adapt to its population if the population using the system remains similar, or by periodically combining results from differing populations and training the neural network off-line. Note that this off-line training still requires no manual grading of students as would be the case with a MLFF neural network. The system can automatically gather student data and retrain periodically. Enabling and disabling the Kohonen neural network's ability to adapt can be accomplished by disabling the learning part of the Kohonen algorithm.

### 8.3.1 Training Data

The inputs to the Kohonen neural network must incorporate history data in order to make a more informed evaluation of the student and therefore avoid a restriction of Bergeron at al's (1989) neural network, namely reacting to a one off error (or success) from a student. History data can be used to prevent the TS from making snap judgements on the student. For example, if the student is generally performing well, but gets one question wrong, then if no history data is taken into account the TS is forced to make a decision based only upon the most recent presentation and the student is likely to drop a level. The student ability itself represents a degree of history data, in that if a student is regarded to be an expert student, then they must have performed well in the past. However, the direct incorporation of history data prevents a continual changing of levels based upon one interaction only. The incorporation of history data can be achieved by presenting a number of previous interactions with tutorial nodes to the neural network. Each time a new interaction is presented the previous interactions are shifted along the inputs to accommodate the new input and the oldest interaction is lost. Training data supplied to the neural network are figures that represent a percentage value of a student's interaction with a tutorial. For example, if the student achieved a 50% success level with a tutorial question, then it is this figure that is passed to the TS.

A program was devised to generate training data for the neural networks (Figure 8.3.1.A The Training Data Generator). The program generates simulated student scores based upon several parameters described below. The parameters allow the neural network designer to cluster actual student levels into groups. This allows the testing of the hypothesis that the Kohonen variant can adapt to different clustering of student results and hence different domains, since a network can be trained with data clustered in one way and then tested to ascertain whether it can adapt to data clustered in another way. Each line of input data presented represents the time-shifted data. The number of items on each line is variable in order to allow the design of neural networks with varying numbers of time series inputs, in order to determine experimentally which produce acceptable results.

The input parameters to the Training Data Generator are the number of time-shifted inputs, the number of outputs (student levels) and the number of test cases to be generated. Further inputs allow the tailoring of the actual data, including the minimum and maximum values to produce (to represent ranges of scores, for example most scores may be limited to between 30 and 70% say) and a deviation from the current value. The deviation limits the next value to be within the deviation from the current value. For example with a deviation of three and a current value (score from a tutorial) of fifty, then the next value will be either 47, 48, 49, 50, 51, 52 or 53. This gives a more realistic spread to the values, since in reality a student is not likely to exhibit extreme behaviour (10,99,2,78 etc). The data may be split into several virtual students by the student value, this simply generates a new value that is not dependent upon the deviation, in order to simulate a different student using the system. The program calculates the number of simulated questions per simulated student and displays the result in the “Q per Student” field. The Outliers (an item that does not fit with the current trend) field allows the over riding of the dependence upon the deviation value, for one value only. For example a student may perform poorly with one question only and then return to their original trend e.g. (45,47,38,46,50). This is necessary since in reality some students are likely to periodically perform in contrast to their current level. For example, they may make an uncharacteristic mistake (and get a lower score) or may have some background knowledge not gained within the system (and get a higher score).


TutSup Trainer

Outputs 10
[✓] Realistic
Deviation 5
Outliers 0
Inputs 5
Max Value 99
Min Value 0
Itterations 2500
Students 50
Q per Student 50
Idle
Generate
Filename c:\neuroshell2\networks\tutsup.txt

**Figure 8.3.1.A The Tutorial Supervisor Virtual Student Program**

An example output of the Tutorial Supervisor Training Program is:

```
43 46 46 46 46 46          5             new student
39 43 46 46 46 46          4
40 39 43 46 46 46          5
42 40 39 43 46 46          5
40 42 40 39 43 46          5
44 40 42 40 39 43          5
45 44 40 42 40 39          5
```

The first six values in any row represent the time-series input to the TS. The final value represents a guide student level based upon the mean value of all inputs and this figure may be used by the researcher to make a quick evaluation of the line of data. The "new student" indicates that the general trend in scores has been randomly reset to represent another student using the system. For example, when the VSP generates a score for a student, this score is between the minimum and maximum scores. The next score is then calculated as a random deviation value plus or minus the previous score, unless the VSP randomly signals an outlier score, in which case a completely random value is generated. After an outlier score has been generated then next score is again dependent upon the previous trend. This continues until a "new student" is generated, in which case a random value is generated, ignoring the previous trend (like an outlier) and all subsequent scores (with the exception of outliers) are based upon this score.

## 8.4 The Kohonen Tutorial Supervisor Experiment Design

The purpose of these experiments is to determine whether the Kohonen neural network is a viable and potentially useful architecture for the Tutorial Supervisor, in that it can map tutorial results onto student ability levels and that it can accomplish this automatically. This can be determined if the Kohonen TS can both generate a student level based upon the scores presented to it and continue to produce student levels if the clustering of scores changes overtime. The difficulty is that there are a number of parameters that must be selected for the experiments. Parameters (number of inputs, outputs, learning rates etc.) were chosen in line with guidelines proposed by Masters (1993) and Skapura (1996) and are discussed below.

The experimental Kohonen neural networks were defined in line with a strategy devised by Masters (1993). It is not usually possible to exhaustively test every possible combination of neural network parameters. Instead, it is advisable to:

- Define the number of inputs, as driven by the problem itself.

- Define the number of outputs, as driven by the problem.

- Use a standard number of epochs (number of training examples presented to the network), and increase only if a network does not converge on a solution.

- Use a standard training and test set. Change only if required specifically by the problem.

Specifically for this problem the following was undertaken in order determine the best configuration.

- The number of inputs was steadily increased to examine whether a successful network could be defined for instances where a large number of history cases are required.

- The number of outputs was increased, to test for cases where a large number of student levels are required.

The number of time shifted inputs is crucial to the grading of students using a Kohonen network, since it is not being provided with an expected level for the student. Instead it must work it out for itself. The inputs have equal priority, i.e. the network does not explicitly know that the first input, being more recent, is more important than the last output. Therefore, data presented to the neural network, both real and simulated, provides a temporal pattern, in that a value appears at one input and then travels along the inputs to the last input, after which it disappears. In order for the neural network to be able to take this into account two methods were employed for extracting training data from the simulated data. Original order extraction presents training material to the neural network in the original sequential order, so that the time-series element is preserved. The second method is to randomly extract the training data, so that each training item appears in isolation and is not related to the training item before or after.

A Kohonen is given a number of outputs that represent the number of categories that it must separate its input data into. It is possible to determine that a Kohonen has been trained to a sufficient level if it has activated all its outputs with the training data to the same degree that the patterns occur in the training data. This may be represented by a pie chart as the Kohonen network is training, each segment of the pie-chart being an output level for one of the outputs, making it possible to see how much each output has been activated and to see whether all outputs have been activated.

### 8.4.1 Experimental Method

The purpose of the experiments is to determine whether the Kohonen neural network architecture can be used as an automatic tutorial supervisor and further to determine the simplest and therefore most efficient neural network configuration to achieve this.

There may be a range of neural network configurations that are regarded as successful. However, the simplest neural network (in terms of size) requires less processing power to execute it and less training examples to train it.

Experimental parameters varied were:

• The number of inputs.

It is necessary to determine how many inputs were required to give a balanced level for the student, whilst maintaining as simple neural network as possible. This is not a straightforward matter, since the more inputs the neural network has the more history data is presented to it, which will directly affect its output. It is sensible to identify a range for the number of inputs that produce successful neural networks. This range of neural networks may then be employed in trials with real student data to determine which is/are the best. These trials were beyond the scope of the current project.

• The number of outputs

The number of outputs represents the number of student levels that the Tutorial Supervisor can grade a student into. The number of outputs also affects the complexity of the neural network (more complex networks, i.e. with more outputs, are less likely to converge on an acceptable solution and/or require more training). This parameter is therefore vital since it is not only driven by the necessity for a simple network design, but by the necessities of the Tutorial Supervisor itself (how many levels it can offer to the system as a whole). Ten students levels was defined as a reasonable number on the grounds that this provides enough distinction between a novice and an expert student, although the experiments were designed to identify a range (by changing the number of outputs), should there be a need for different amounts.

• The number of training epochs

This is how many data items are presented to the neural network during training. For complex networks (with a large number of input and output neurones) this is vital since a complex neural network may memorise its training data and lose its ability to generalise with other data, a phenomenon known as "over-fitting" (Masters 1993). For Kohonen neural networks it is possible to determine when all outputs have become activated to the desired degree and then stop the training. This can be achieved by examining the output level, by examining the pie-chart, and determining whether it is producing a similar value to that of the known (because it was generated by the VSP) composition of data.

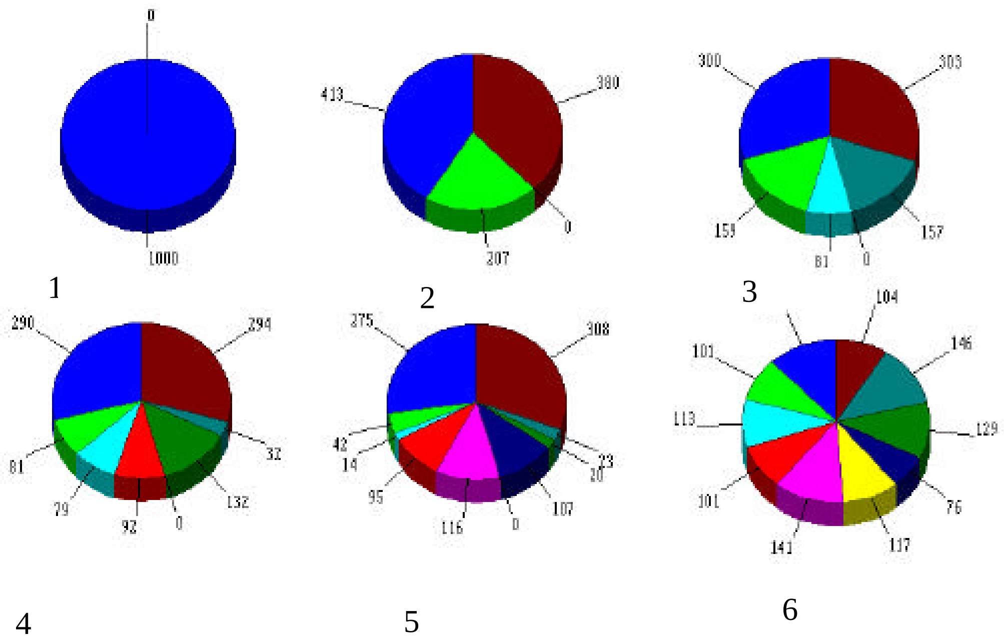
• The training set extraction method

Two types of extraction method were employed, random extraction and original order extraction. Since the data provided to the neural network represents time-series data it was considered important to present the data in its original order, so that the time element is not removed between examples. In order to test the significance of this a random extraction method was used as a control.

• The domain data type (limiting values to certain areas) (denoted by “Range Train” and “Range Test” in the following results table)

The neural networks were trained with various data clusters to determine to what degree they were able to differentiate between them. A neural network trained with data clustered in one area would then be tested with data clustered in another area to determine whether they were able to adapt to the new data. For example, a network may be trained with data that is clustered around 30-75%, it would then be retrained with data clustered in another region, 40-90% say. The columns in the results tables below that demonstrate this are "Range Train" and "Range Test". The "Range Train" column represents the range of scores that the neural network was trained with and the "Range Test" column represents the range of data that the neural network was tested with once it had been trained with the training data.. Different ranges in Range Test and Range Train can be used to determine whether the neural network can adapt to different domains where the range of scores is different.

## 8.5 Experiment Results



The pie charts show a single Kohonen neural network undergoing training. Each coloured segment in each pie chart represents an active output of the network. The relative size and attached figure represents how active that output is in comparison to other active outputs (the figure is the percentage of activity multiplied by 10). A network that has been trained with data that has equal amounts of examples for each category should produce active outputs of roughly equal size. Each pie chart represents a further 18 training examples, as more examples are presented more outputs become active and begin to form a more symmetrical pie chart.

**Figure 8.5.A Training a Kohonen**

In order to determine if the neural networks were successful or not it was necessary to train them with a known composition of data. For example, if ten categories of data of equal size were presented to a neural network then it should activate ten outputs to roughly the same degree. This was possible since the data was generated with the VSP. Figure 8.5.A shows a Kohonen neural network undergoing learning. Pie charts are a standard way for representing a Kohonen neural network (Kohonen 1989) and were generated with the neural network design package NeuroShell2, by Ward Systems. The pie chart represents the outputs of the Kohonen neural network. The number for each segment of the pie chart represents the number of presentations (to the network) that have been recognised as that particular category. The number of segments show how many categories the network has currently recognised, the more examples it sees the more categories it will attempt to separate them out into. The first pie chart shows a network that has just begun its training and is not activating any outputs. The second pie chart shows a network that is activating four of its ten outputs, although one of these outputs is very weak, as defined by its small size (and low corresponding size value). The subsequent pie charts show more outputs becoming active, until all ten are active (pie chart 5). At pie chart 5 all outputs are active but the network has not been trained sufficiently, this is readily apparent because all the pie chart slices are of different sizes but the training data consisted of categories of the same size. Finally chart 6 shows a fully trained network, in which all outputs are strongly activated. If a network is trained beyond this point, generally the output activations do not change further, this is because the network has fully learned to distinguish between patterns.

For Kohonen unsupervised learning architectures it is not possible to assign a definite success value, since the Kohonen architecture is not trained by supervised learning, in that it is not provided with a desired result and is not told whether it is right or wrong. This is not to say in all cases that a Kohonen neural network's results can not be graded against results obtained by another means. If results could be obtained by another means (i.e. a training set and test set can be constructed with results data, but then the trained neural network is used on its own) then a similar comparison can be made. In the case of the TS, it is not possible to generate accurate student levels with the generated input data. This is because of the special cases, such as the inclusion of history data that may contain outliers that renders the classification complex. The only way to judge the performance of the Kohonen in this case is to train the network with training data of a known composition, in terms of a statistical guide as to the ratio of patterns within it and to determine that the outputs are activated to a similar degree. This method was used for the experiments detailed below.

### 8.5.1 Success Criteria

The result tables below show the configurations of Kohonen neural networks that did and did not produce a successful result. Determining a successful result was based on an examination of the numerical data produced by the neural network and a comparison was made against the general pattern represented in figure 8.5.A. Success is denoted in the following results tables as a "Yes" in the success column for a successful neural network and a "No" if the neural network was not successful. The Kohonen architecture produces real number values, in most cases the output with the highest value is designated the “winner” and all other values are ignored (for example, the output representing level 6 produced the highest output activation, therefore the student is graded as level 6). This approach is not suitable for determining the success criteria of the Hypernet neural network, since an output may be designated the winner with a weak output (low real value), simply because all the other outputs are weaker. A network that produces a weak output is less likely to produce consistent results given similar data presentations (Skapura 1996). The relative sizes of the real values at each output can be interpreted as certainty factors. It is reasonable to select networks that are "certain" of their results (Masters 1993). Reasons for networks not producing high levels of certainty can be attributed to unclear input data (which would affect all networks), or insufficient training times (epochs). Once a network was defined as being "certain", in that it is outputting a similar value to the known composition of data (as generated by the VSP), then its actual outputs were examined to determine whether it was making a reasonable distinction. This was achieved firstly by ensuring that the outputs were active to a similar degree to the amount of each category in the input data. For example, since the composition of the training data is known (e.g. it is 25% level 6, 35 % level 5 etc.) then there should be a direct correlation with the relative outputs of the network. Secondly, exception data (the student performed poorly for one input when the rest were generally good, for example) was then tested with the networks to determine whether they were making a reasonable distinction. This was taken to be movement of one level for each item of exception data. This fits well with the pedagogy proposed by Bergeron et al (1989).

The results tables below show the various configurations of neural network that were applied to the problem of generating an efficient Tutorial Supervisor. A discussion of the table follows.

**Table 9 Altering the number of inputs and outputs**

| Network | Inputs | Outputs | Epochs | Range Train | Range Test | Success |
|---|---|---|---|---|---|---|
| 1 | 5 | 5 | 100 | 0-100 | 0-100 | Yes |
| 2 | 5 | 10 | 100 | 0-100 | 0-100 | Yes |
| 3 | 5 | 20 | 100 | 0-100 | 0-100 | Yes |
| 4 | 5 | 30 | 100 | 0-100 | 0-100 | No |
| 5 | 5 | 30 | 500 | 0-100 | 0-100 | No |
| 6 | 5 | 10 | 100 | 40-70 | 40-70 | Yes |
| 7 | 5 | 10 | 100 | 0-100 | 40-70 | Yes |
| 8 | 10 | 5 | 100 | 0-100 | 0-100 | Yes |
| 9 | 10 | 10 | 100 | 0-100 | 0-100 | Yes |
| 10 | 10 | 20 | 100 | 0-100 | 0-100 | Yes |
| 11 | 10 | 30 | 100 | 0-100 | 0-100 | Yes |
| 12 | 10 | 40 | 100 | 0-100 | 0-100 | No |
| 13 | 20 | 40 | 100 | 0-100 | 0-100 | No |
| 14 | 10 | 40 | 500 | 0-100 | 0-100 | No |

**Table 10 Range and Epoch Experiments**

| Network | Inputs | Outputs | Epochs | Range Train | Range Test | Success |
|---|---|---|---|---|---|---|
| 16 | 5 | 10 | 25 | 0-100 | 0-100 | No |
| 17 | 5 | 10 | 50 | 40-70 | 40-70 | Yes |
| 18 | 5 | 10 | 50 | 40-70 | 0-100 | No |
| 19 | 5 | 10 | 100 | 40-70 | 0-100 | Yes |
| 20 | 5 | 10 | 50 | 0-100 | 40-70 | No |

### 8.5.2 Discussion of Results

• Experiments 1,2 and 3 five inputs produced networks that identified a solution with between five and thirty outputs. Experiment 4 continues the trend of increasing the number of outputs; however this configuration produced a weak neural network. However, experiment 11, which has the same number of outputs as experiment 4, does produce as successful neural network. The difference between experiments 4 and 11 is that the number of inputs is increased to 10 from 5 in experiment 11. The likely reason as to why increasing the number of inputs produces a successful network is that there is enough information being presented to the network for it to make a distinction into the larger number of student levels. However, a potential difficulty could arise in that the range of data presented to the network might in reality not represent such a range of student levels. In this case the neural network is likely to be making other determinations to generate the range of outputs that are not consistent with the problem in hand. For example, the neural network may begin to identify certain exceptions to the current trend as levels in themselves. The problem of a neural network making unwanted or unforeseen determinations is a common problem. A classic example, often used to highlight the problem, was a system for identifying enemy tanks. The system worked well on the test data (data from the same population as the training data but not used in training), but failed totally with new data. It transpired that the training pictures of tanks were taken on a sunny day and the training pictures without tanks were taken on a cloudy day and the network has learned to distinguish weather (Skapura 1996). The problem of the tank system was a training data problem in that the problem would not have arisen if the training data had included pictures of tanks on both cloudy and sunny days. However, it does highlight the problem with connectionist systems that the designer is not in direct control of the task in hand.

- Experiments 4,5 and 12 reinforce the conclusion drawn above that a large discrepancy between the number of inputs and the number of outputs produces networks that do not produce acceptable results. Experiment 12 demonstrates that this is not dependent upon the number of inputs (since 10 are used) and experiment 5 demonstrates that training the network for longer makes no difference.

• Experiments 6 and 7 demonstrate that a network is capable of learning to produce the desired number of output levels for a limited data range and that it is able to adapt thereafter to data in other ranges. This is a crucial point in that it strongly indicates that the neural network will be able to adapt to different domains without manual intervention. Experiment 6 shows that the network can still identify the student levels with the student scores being limited to a smaller range. Further, experiment 7 shows that once a network has been trained with one range it can still adapt to another range.

• Experiments 8,9,10 and 11 demonstrate that larger network with 10 inputs work well. These experiments were mainly used to test the hypothesis that a large imbalance between the number of inputs and outputs produced untrainable networks (described above). It is doubtful whether this large number of inputs would be required for a practical tutorial supervisor, however. This is discussed further in the discussion section of this chapter (section 8.8).

• Experiments 12,13 and 14 all produce unacceptable networks, this is because there are too many levels for the network to produce discrete outputs. Since the data presented to the network falls in the range of 0-100, it is not reasonable to expect the network to classify categories that are only composed two or three numbers.

• Experiments 15 to 20 and experiment 5 demonstrate that the optimum number of training epochs is one hundred, since too few and the networks do not converge (16,18,20). Experiments 17 and 18 demonstrate that a network may learn to classify a data range in a low number of epochs but do not quickly adapt to a new data range.

Experiments conclude that a network will adapt to a new data range within 100 epochs, i.e. after one or more students have completed 100 questions or tutorials. Experiments not documented demonstrated that using more than 100 epochs did not affect training, for this problem at least. In practice it could be seen by examining the network outputs whilst they were training that they had converged upon a solution before the 100th epoch and that any further training did not change the network.

The results of the Kohonen neural network architecture were highly successful, a simple Kohonen neural network was able to separate its input data into the number of predefined student groups, i.e. they were able to activate all their outputs to the desired degree. A network trained on one set of data, data clustered into a certain range, was able to adapt quickly to data clustered differently. This is an important point, since this allows the network to adapt to various domains. It is this facility that the MLFF neural network cannot accomplish. A drawback of the Kohonen neural network architecture identified, however, was the difficulty in determining which output is associated with each student ability level. In other words, if there are ten outputs it is not necessarily the case that the first input represents level one and the second level two and so on. The Kohonen neural network is given the number of categories it should separate the input data into (the number of outputs) and it is left to the Kohonen neural network algorithm to determine which output belongs to which category. This is not accomplished in a rule-based fashion, but instead, the outputs tend to evolve towards a particular solution, dependent upon the initial random start weights and the possibly random order of examples. For most cases, this evolution of outputs did produce a graduation of output categories, i.e. level 2 is next to level 1 etc,. This can be explained by the fact that level 1 is more similar to level two than it is to level 10. However, level one sometimes appeared as the first input and sometimes as the last input. It was straightforward to determine which outputs were related to which category, since in the vast majority of cases the inputs occurred sequentially.

#### 8.5.2.1 The Number of Inputs

The number of inputs is the amount of time series data to be presented to the neural network. The practical limit for inputs is dependent upon the training data and the number of outputs. It was experimentally discovered that a large imbalance between inputs and outputs (many more outputs than inputs) produced networks that did not converge on a solution. Note that the reverse case (many more inputs than outputs) is always likely to produce a strong network, since it is being provided with a wealth of input and only asked to separate it into relatively few categories. However, these configurations were not desirable for this project for reasons described in section 8.4, namely too much history data puts too much weight on information that may be out of date.

These are issues related to the domain and not necessarily the network architecture itself. It is a key factor that the number of inputs reflects the correct solution to the problem and there must be sufficient inputs with which to make an accurate grading of the student. Therefore, it was necessary to test a range of input and output configurations that could be of use to several problem areas. This is necessitated by the fact that the number of time shifted inputs may be driven by the domain in question. Thus there may be many questions over a short period, or fewer questions over a short period. If the number of questions is small then it may not be desirable to provide a wealth of history data, since the oldest inputs may represent the student's state of knowledge an educationally long time ago. Similarly, if the student interacts with a large number of questions over a short period, then it may be desirable to provide the network with more time-series data, in order to give an adequate spread over time. This is a fundamental problem, since different domains are directly driving the number of required inputs to the neural network and hence negating the domain independence of the Tutorial Supervisor. A possible solution to this problem is to design a “jack-of all trades” neural network, in that it is able to deal with the maximum number of inputs and then to use only a selected few inputs for unique data items should the domain not require as much history information. This approach would however need further investigation with real student data to test its validity.

#### 8.5.2.2 The Number of Outputs

Changing the number of outputs did not affect the successful training of the neural network. Generally the more outputs the more complex the network and the longer it takes to train. Larger networks also require more training data, since the learning is spread over more neurones. The upshot of this is that the networks require more time to train and more processing power to execute.

The number of outputs represents the number of levels. Experimentally it was determined that twenty outputs were a practical limit for a neural network with between five and ten inputs, since it can be seen from the results table, experiments 5 and 12, that successful neural networks were rendered useless by applying more outputs (more than 20).

#### 8.5.2.3 The Number of Epochs

This is the number of times training items are presented to the neural network. The value of one hundred was found to produce a neural network that converged on a solution, in terms of all the outputs of a neural network becoming excited by at least one pattern within the training data. However, the fact that a neural network converged upon a solution does not necessarily mean that the solution is an acceptable one. The networks were examined to determine whether they were returning the correct number of levels presented to it in the input data, those networks termed successful did return the correct number of levels presented (as determined by examining if all the outputs had become active to the correct level).

#### 8.5.2.4 Data Extraction

The network produced good results even if the training data was not supplied to it in the original order. This indicates that the network was not acting as a memory; it was not remembering the previous input and using it to generate a new output. The network does not learn to prioritise its inputs. For example, a student who has given one bad answer and five good answers will be graded the same, no matter when the bad result occurred. This results in the student’s level being influenced by a bad result until such a result is no longer presented to the neural network. The effect is dependent upon the difference between the good results and the bad result. A large difference will make the student drop several levels and keep them at that level for an amount of time (the time being dependent upon the number of results being presented to the network and the time taken for the student to answer other tutorials with a higher degree of success). The consequence of this is that the number of inputs has a direct pedagogical influence, as well as significance in determining the neural network’s abilities. Fewer inputs and, regardless of the neural network’s abilities, the progression from one level to another will be much quicker than if there were more inputs. A constantly changing ability level may be distracting to the student.

### 8.5.2.5 Limiting values

The advantage of the Kohonen neural network unsupervised learning architecture is that it is able to continue learning whilst it is in use, without intervention from a person. As discussed earlier, this may be necessary if the range of answers from a student population falls within a limited area, since the limited areas may be different for various domains. The networks were therefore trained on data to mimic this situation. It was found that the networks could converge on a satisfactory solution for a clustering of data where the number of outputs was valid (a distinction can be made between one output level and another). For example, the networks were trained with a data set limited to values between forty and seventy percent as quickly as for data between zero and one hundred percent. If the data ranges were set to more restricted values (between 50% and 60% say), then a network with twenty output levels would not converge on a solution. This was to be expected, since there are only ten discrete possible values, while the network is trying to classify twenty clusters.

It also was found that a neural network could be trained to categorise data correctly over various cluster types, i.e. all the levels (outputs) were activated for a given cluster to the degree to which the categories were present within the training data. A neural network previously trained with one set of clustered data could be easily retrained with a differently clustered data set. These results demonstrate that in practice, the neural network can continually adapt to its student population.

Figure 8.5.2.5.A shows three pie charts representing the results from a Kohonen neural network with five inputs and ten outputs. The first pie chart shows a neural network that was trained with student data ranging from 0 to 100. The second pie chart shows the same neural network after it has been allowed to adapt to data in the range of 40 to 70. The middle pie chart shows the network before it has adapted to the new domain data. The two large chunks represent the outputs of the network that are now representing what was formally one level but are now in effect representing many levels. Therefore, in order for the TS to remain valid and useful it must adapt to the new domain. This adaptation takes place in the same number of epochs as the original training, i.e. 100 epochs. The size of the segments represents the activation of each output.

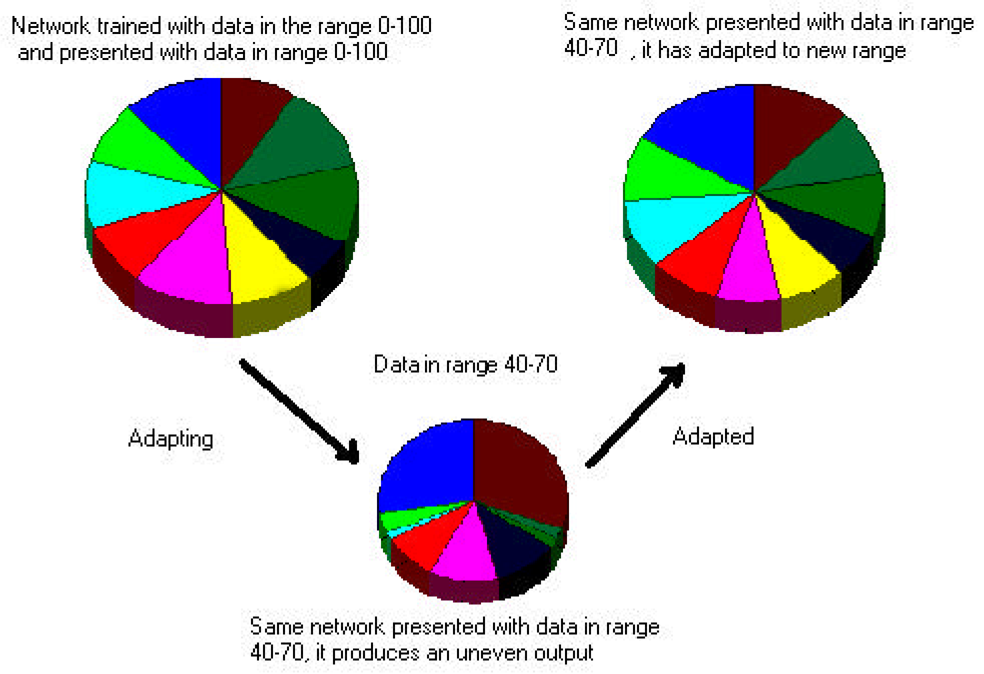

Network trained with data in the range 0-100 and presented with data in range 0-100
Same network presented with data in range 40-70 , it has adapted to new range
Data in range 40-70
Adapting
Adapted
Same network presented with data in range 40-70, it produces an uneven output

**Figure 8.5.2.5.A Range Data**

## 8.6 Re-grading Questions Using Fuzzy Logic

Each question level is generally presented to a student of the same level, or just below, a pedagogy used with success by Bergeron et al (1989), in that a level *x* student should be able, overall, to answer a level *x* question. A question may however, be graded incorrectly by the domain author. This can be determined by the system after a number of interactions with different students (a population of students who should be, generally, getting a question right are getting it wrong or vice versa). It is not suitable to immediately re-grade a question with respect to an interaction with one student, however. As has been discussed earlier, a student is a complex entity and it is difficult to formulate rules describing them accurately. In order to resolve this problem each question level is modelled as a fuzzy set. This allows a question's level to be adjusted slightly, within the level, without necessarily affecting the overall level (as presented to the student), thus there is a buffering effect and the question does not rapidly leap back and forth between levels. The use of fuzzy sets also provides a mechanism for allowing a question to belong to more than one question level set, providing a smoother transition between levels. This differs from Bergeron et al’s (1989) approach in that they collect data from the students and then periodically use it to update the training of the neural network. There is therefore a delay, which ensures that the question levels do not suddenly change, which could potentially result in the question level continually changing and thus be distracting to the students. However, the drawback is that this is a manual process that requires the direct intervention of the system designer. The process for regrading questions described below is achieved automatically, whilst still maintaining the delay between the question being presented to a student and changing its level.

Figure 8.6.A and 8.6.B show fuzzy sets overlapping with each other. The first diagram shows that a question with a value of *x* is graded as both levels two and three. The second diagram demonstrates that if the fuzzy sets are stretched, then a question with the same value of *x* has a membership of all three question levels. This is an appropriate situation since it may not be a simple matter to assign a question to one

**Figure 8.6.A Fuzzy Sets 1**

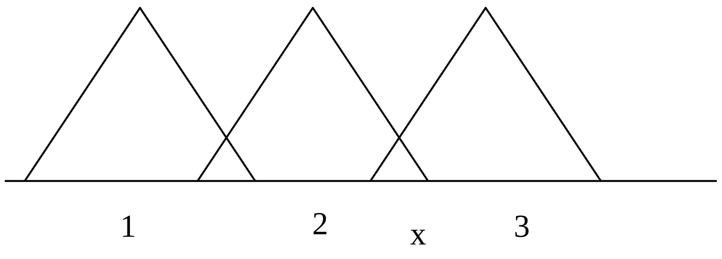
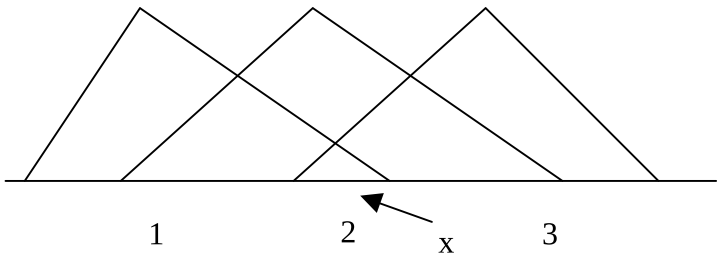


**Figure 8.6.B Fuzzy Sets 2**



particular level. One student may find a particular question more difficult than another student of the same level, due to some difference in previous knowledge. If a question is near the border of levels then it may appear as either level, i.e. it may be offered to students of both levels. The degree to which this multiplicity of questions occurs depends upon how the fuzzy sets are defined.

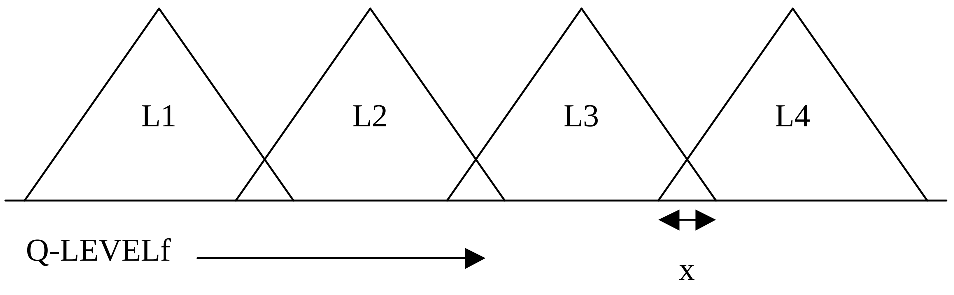
A question is re-graded by a population of students' interactions with the question being determined as incorrect by the TS, for the reasons described above. Such erroneous interactions cause the question ability to move within the fuzzy set until it crosses into a different fuzzy set.

A question's ability level is therefore only changed after a number of erroneous interactions with students, the actual number being dependent upon the size of the fuzzy set, the fuzzy set is therefore acting as a buffer. A simple fuzzy processor accomplishes question re-grading. The fuzzy processor compares the level of the current question and the student level output of the TS neural network. The buffer can be implemented using three fuzzy rules (Kosko 1996):

1 IF S_LEVEL > Q_LEVEL THEN Q_LEVELf = Q_LEVELf + 1

2 IF S_LEVEL < Q_LEVEL THEN Q_LEVELf = Q_LEVELf - 1

3 IF S_LEVEL = Q_LEVEL THEN Q_LEVELf = Q_LEVELf  (remain unchanged)



where

S_LEVEL is the level assigned to a student

Q_LEVEL is the level of the question, used to decide whether it is suitable for the student.

Q_LEVELf is the fuzzy membership number of the question.

**Figure 8.6.C Membership of the fuzzy sets**

Using the figure 8.6.C, a question may belong to one or two of four levels. If the question has a value of Q_LEVELf corresponding to x, then the question is regarded as both level three and level four. If however, as a result of interactions with several students, rule one is repeatedly fired, then the value of Q_LEVELf will increase and the question will become graded as level four only. Conversely, if rule two is repeatedly fired then the value of Q_LEVELf will decrease and the question will be graded as level3 only. Such changes may result in further increases or decreases in the value of Q_LEVELf, which may become any of the levels available. The utilisation of this fuzzy system ensures that the level of a question is not changed (in terms of presenting to students) in response to individual interactions with students. The fuzzy logic is able to distinguish overall trends and is therefore robust in the presence of exception data.

The changing of levels is dependent upon the size of the fuzzy sets, since the size of the fuzzy set directly affects how many interactions are required before a question migrates from one level to another. These start and end points also define whether a question can belong to more than one level or not. It is intuitively sensible to provide a small overlap of neighbouring fuzzy sets. This provides a simple buffer that will help the question to find its true grade. For example, if fuzzy sets were to be defined separately (or as a discrete set) then a question may be presented as a difficult level but become graded as an easier level after several interactions. Once this change has occurred then the question is no longer presented to the original class of students that graded it. This may result in the question becoming stuck at a particular level. If instead the question is presented to both the original class of student and the new class of student, then a smoother progression from one level to another may result and the extra information enables the question level to settle more easily. If the fuzzy sets are defined so that a question may belong to more than two levels, then the question level may not settle at all, as one level of student may push the value of Q_LEVEL one way and another level may push it in the opposite direction.

## 8.7 Results Conclusion

The Kohonen neural network proved to be a successful neural network architecture for the problem of grading students into ability levels. Most permutations of parameters produced neural networks that converged upon a solution. A fast and reliable neural network could be produced with between five and twenty inputs or outputs. It is possible to increase this number, but this is unlikely to be required, since it is not desirable to use information from too far in the past, since the network does not have any knowledge of time. It has been experimentally determined (as discussed in section 8.5.2) that the number of outputs should not rise above twenty. If more levels are required then the outputs may be combined to form fuzzy sets (explained in section 8.3.3).

Parameters for a generic Kohonen Tutorial Supervisor, i.e. one that will converge on a solution and provide a high degree of student grading for a variety of domains are suggested as the following:

• 5 Inputs – this has been found to provide the network with enough information with which to evaluate the student. It is suggested that an input filter, as described above is employed if more than this number of inputs is required.

• 10 Outputs – corresponding to ten student levels, if more levels are required then it is advised that thirty levels is the upper limit, beyond this the neural network becomes less likely to activate all of its outputs.

• Any form of data extraction may be used to form the training set, however a large number of examples are required to produce a fully trained network. One thousand student interactions provide enough examples. Note therefore, that for a network to be trained with real student data is probably impractical. However, since the Kohonen network is fully adaptable, it is suggested that a network be trained upon generated data and then allowed to adapt to real students. The generated data could be designed to represent realistic but uncomplicated situations. For example, training the network to model simple rules such as "If result in the range 40-50 THEN set student level to 5". The neural network is then able to adapt to any misconceptions present in these rules. Once a neural network has been trained it may be saved and replicated.

## 8.8 Discussion

The Tutorial Supervisor is intended to be an automatic system for gauging a student's abilities with tutorials. This imposes a restriction upon the kind of material that can be offered in a tutorial, since it must be suitable for automatic assessment. Automatic assessment effectively rules out assessments that cannot be graded in a relatively simple fashion. For example, it would not be possible, with current technology, to have an essay automatically assessed, since this would require a complex understanding of the essay on the part of the assessor. Automatic assessment is limited to tasks that can be broken down into elements that can then be individually marked. A tutorial may, for example, take the form of ten questions, each of which could be answered by the student and graded as right or wrong, or graded as containing relevant keywords. However, assessment is not the role of the TS, which is presented with completed assessments (results). Assessment is therefore limited to multiple-choice questions or identifying that the student has visited certain nodes (or a combination of both).

A key issue of concern regarding the TS is the number of student levels that the TS is to recognise and output. Each student level should have tutorial material generated for it; since it is important to target tutorial tasks at the student's ability, this is seen as being of more educational benefit than offering the same tutorials to all students and then assigning a student level based upon the grade that the student achieves (Bergeron 1989, Hartley 1993), although there is no technical reason why the latter could not be done. The number of student levels therefore may change between domains, since some domains may have a richer set of assessment questions than others (for a number of possible reasons). A possible conflict therefore could arise between the number of student levels that has been designed into the TS by the system designer and the number of student levels that are required by the current domain author. A possible solution to this problem is for the system designer to provide a TS that is capable of outputting a large number of student levels and then each domain author can allow it to adapt to their domains and ignore the inactive outputs from the TS that will naturally arise if there is not sufficient input student levels. The benefit of this approach is that one TS configuration could be used for many domains without the need for reconfiguration. However, the drawback is that some outputs of the TS will always remain inactive, although the experiments carried out as part of the research demonstrated that active outputs tend to cluster together and so are easily identifiable. A problem related to the number of outputs is the number of inputs.

The amount of history data presented to the TS directly affects the grading of the student, in that the more history data presented to the neural network, the greater the effect of previous results with tutorials. This is a similar situation to that of the number of student levels, in that it is possible to design a TS with a large number of inputs and then use only the required amount.  However, it is not a simple matter to determine how much history data to present to the neural network in order to aid the student the most. This issue is difficult to reconcile without extensive trials with real students and even if this were done it would still be unlikely that any real conclusions could be drawn since proving the effectiveness of educational systems is notoriously difficult in the educational field (Dillon and Gabbard 1998). The purpose of the TS here is to explore the technical issues relating to the feasibility of providing an automatic student grading system. Whether this facility is useful is open to educational debate. However it is likely to be the case that it will be useful should the correct set-up of the TS be achieved during trials with real students, since Bergeron et al (1989) found their TS to be useful.

Further issues arise concerning the adaptability of the neural network used for the TS. The neural network architecture used by Bergeron et al (1989) requires off-line training and is therefore under the control of the system designer. The drawback with this approach is that it requires the manual intervention of a person who can interpret the student interaction data with tutorials and determine whether it should be represented to the neural network. The advantage of the Kohonen neural network architecture is that it is able to train continually without any intervention from a human. However, there are situations where this adaptation is undesirable, most notably when different skill levels of students use the same domain at different times. For example, if a class of first year students use the system followed by a class of final year students. However, this is not a problem if the questions and tutorials have been adequately assigned a difficulty level, since the first year students will only be offered easier tutorials and so can be graded only as lower level students (although they can still progress if they continue to achieve success with the tutorial). Problems can arise only if both the student abilities and the question difficulties are unknown beforehand. This is because the TS acts as a bi-directional mapping device, in that if either the student abilities or the question difficulties are known beforehand then the TS can produce the unknown parameter. It is not, however, able to produce values when nothing is known beforehand. The TS's ability to re-grade questions automatically is an exploitation of this bi-directional mapping facility, in that the student ability can be changed in response to improving results and the question difficulty can be changed if a significant proportion of students who should get the question right in fact get it wrong.

Research into the TS has demonstrated that a fully adaptable system for automatically grading students is possible and practical. The approach of using an automatic tutorial supervisor has been practically justified by Bergeron et al (1989). However, their system requires manual periodic retraining which renders it unsuitable for a generic tutorial system, or a tutorial system that can be used without the need for reprogramming or otherwise rearranging the program code of the system.

## 8.9 Conclusion

This chapter has described the design and training of the neural networks for the Tutorial Supervisor. It was argued that the Kohonen neural network is the best architecture for this problem, because of its ability to adapt to various domains without the intervention of a human. Experimental designs for the Kohonen neural network were presented and discussed. The ability of the Tutorial Supervisor neural network to adapt to different domains and the ability of the Tutorial Supervisor's fuzzy processor to adapt to question difficulties were explained.

The discussion of the architecture of Hypernet in detail is now complete. The following chapter is concerned with how a domain for Hypernet can be authored.

# Chapter 9

# Domain Authoring

## 9.1 Introduction

This chapter is concerned with authoring hypermedia domains, with respect to aiding student/user navigation. The discussion is not limited to the Hypernet architecture, although Hypernet is used in places to motivate the discussion. The discussion may be relevant to all authors of hypermedia domains, including those designed for the expanding World Wide Web.

The overall system and the semantic network used for the automatic and dynamic linking system were described in chapter 3. The discussion is now continued, detailing how various domains may be constructed and structured with a semantic network. The suitability of various domains is discussed, in that certain types of domain may involve more effort on the part of the author than other domains.

It has been argued by Pereira et al (1991) that all hypermedia systems should have a semantic network (or similar structure) defined within them to aid both the author and the end user. Jonassen and Wang (1993) have further suggested that structures such as semantic networks provide the student with structural information that aids them in the retention of hypermedia material. Further studies have shown that complex information and arguments can be represented in semantic network-like web like structures, termed mind-maps or concept maps and that the use of such structures can aid the author of the structure to remember the information stored within them

(McAleese 1998, Holmes 1999). Given this, the chapter includes a discussion of the process of defining a semantic network, and the implications of this process.

Tutorial nodes are discussed in detail, notably their implications concerning the browsing behaviour they may induce the student to exhibit, and their relevance concerning providing the student with useful goals and feedback (in terms of informing them how well they did with the task) and providing the system with information about the student.

Hypermedia authors may find elements of the discussion useful when designing hypermedia systems for education, both for domains intended for use with Hypernet and hypermedia domains in general. The design of tutorial nodes and a semantic network involves extra design work for the author. However, in order for hypermedia to become a useful and usable medium for a tutoring system it is necessary for the author to remove as much of the cognitive burden from the student as possible. It may not necessarily be the case that the author is over burdened by the need to produce a semantic network and tutorial nodes, since they are being forced to think more structurally about their domain and may save time and effort in so doing. Further, the addition of a semantic network allows the use of automatic reasoning processes that may aid the author in ensuring that the domain conforms to the defined structure.

The latter part of the discussion concerns the "bottom-up" approach to domain authoring. This involves allowing paths through the hypermedia hyperspace to emerge naturally as either a student or an author browses the system. The implications of this approach are discussed, with particular reference to domains that do not have an adequate semantic structure, which it is argued includes many existing hypermedia domains.

## 9.2 Domain Types

The exemplar domain designed for the conception and implementation of Hypernet was an astronomy domain. This domain was chosen for the reason that it is highly structured and hierarchical, making it ideal for application to this system, since it allows the rapid design of a semantic network. However, the goal of the project as a whole is to devise a paradigm that is as generic in approach as possible, so that it can be applicable to a multitude of domains. In practice, however, it is likely to be the case that some domains are easier to describe as a semantic structure than others. This is because some domains do not exhibit an obvious and generally accepted structure. This could result in a new hypermedia problem similar to "Linkitis" (Conklin 1987), in that if an author is unsure how a domain may be structured then they may be tempted to force a structure on the domain that may not necessarily exist or be helpful. This may result in the student becoming confused and hence render the domain difficult to navigate, exacerbating the lost in hyperspace problem. It should be noted that the Browsing Pattern Recogniser, discussed in chapter 6, should identify this problem, since it will identify the wandering motions typically exhibited by lost students repeatedly visiting the same nodes. It is not an ideal situation, however, to allow the students to identify the shortcomings of a domain's design. It is suggested, therefore, that it is the author's responsibility to ensure that the domain is designed correctly before it is exposed to students. It is accepted that this is not a trivial task however.

The authoring process is a complex one and has often been overlooked by hypermedia designers (Theng and Thimbleby 1998, Littleford 1991, Conklin 1987). It is, however, fundamental to the student's learning needs that the domain is authored to a sufficient standard (Woodhead 1991). This is especially true for "fixed-link" systems (systems that do not provide dynamic links). In fixed-link systems it is vital that the author provides the correct links for every possible user of that hypermedia system. Bell (1995) argues that people are not good at writing in a non-linear fashion, therefore hypertext (hypermedia) documents tend to become linear and do not model the true (semantic) structure of the domain itself. Producing hypermedia systems that require a semantic network forces the domain author to consider the structure of the domain and hence is far more likely to result in a domain that provides an adequate structure for the student to navigate. Further, if dynamic links are employed the student is provided with links that more closely meet their requirements and so reduces their cognitive loading.

## 9.3 Defining A Domain

Here Hypernet is used as an example hypermedia tutoring system. The process of defining a domain for use in Hypernet involves the following steps:

- Structuring the domain into a semantic network (defining its structure) by providing typed links between class nodes (class nodes may also be used as hypermedia nodes). Discussed in previously in chapter 3 (c.f. section 3.6).

- Forming a semantic hypermedia domain by attaching instance nodes to the class nodes (instance nodes may be used as hypermedia nodes). Discussed in section 9.3.4.

• Storing suitable exercises within tutorial nodes, which are connected to the semantic hypermedia. Discussed in section 9.4.

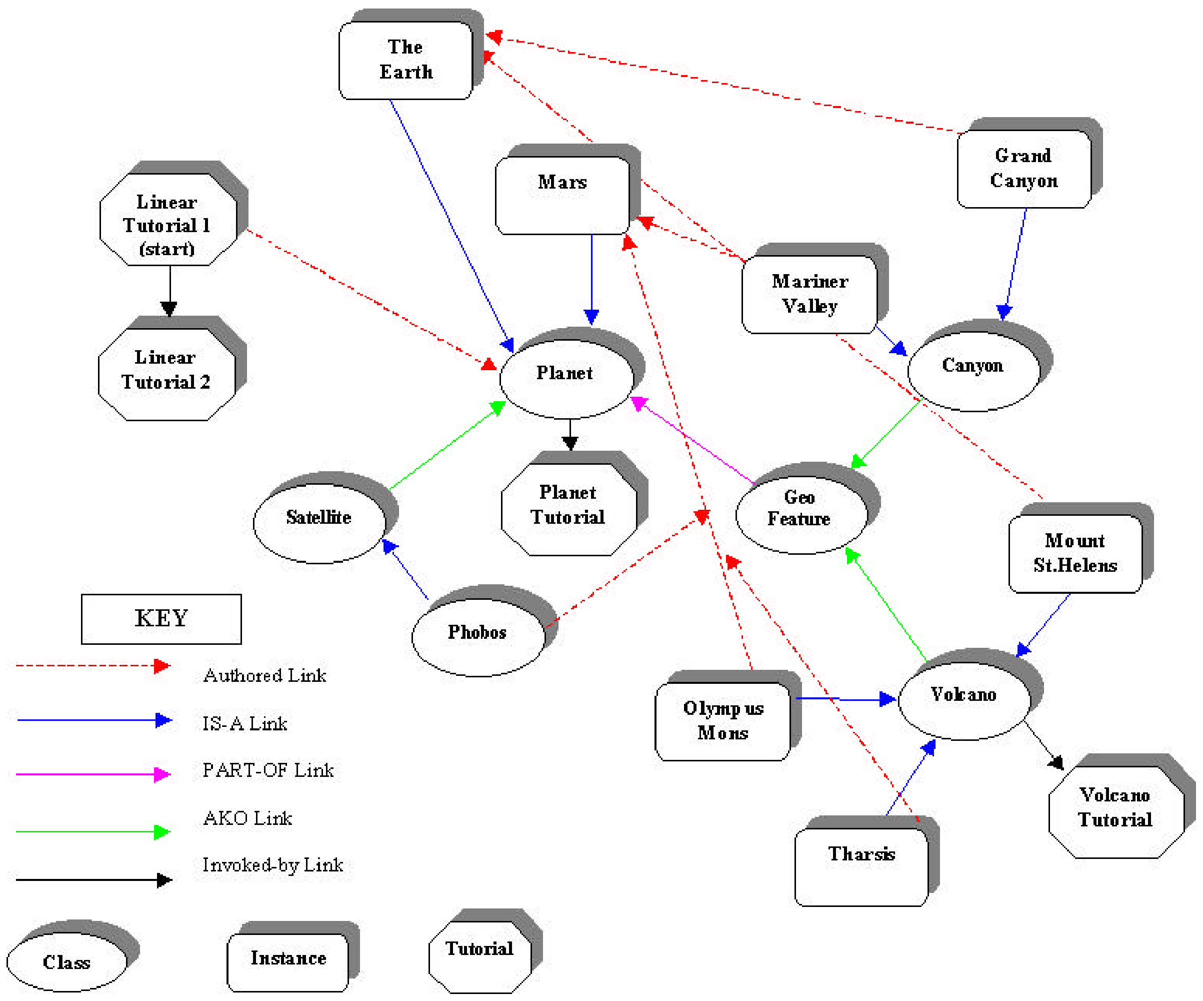


**Figure 9.3.A Astronomy Domain**

For a standard hypermedia system it would usually be the case that none of the above stages would be employed, since the majority of hypermedia systems remain unstructured (Dillon and Gabbard 1998). Instead, the author determines for every node in the domain how it connects to all of the other nodes, typically an extremely complex task. However, the application of the semantic network encourages the domain author to organise the domain in a more logical fashion. The final stage is unique to a computer based learning system, as opposed to a general browsing hypermedia system. Since it is difficult for a student to learn in an undirected environment (Jonassen and Wang 1993) it is appropriate for the domain author to require the student to accomplish some task, at least until they have become proficient enough in the domain to be allowed to browse freely. Further, Hartley (1993) argues that a student using a hypermedia-based tutoring system requires feedback from the system in order to monitor their progress and to maintain their interest.

A semantic network can be defined as a graphical representation relating concepts and information (Rich and Knight 1991, Bench-Capon 1990). A concept in a semantic network is defined as a node and labelled arcs (links) define the relationships between concepts. The nodes themselves do not necessarily contain information (other than the name or label for that node). Semantic networks that do contain information within nodes are more usually termed "frames", or "scripts" (Rich and Knight 1991). These may contain extra information that relates to the particular concept in question. The term “semantic network” has been used to define the structure under discussion here, in that the structure of the domain resembles a semantic network, although the contents of each node are not necessarily subject to the same structuring.

The example semantic network, shown in figure 9.3.A, describes part of a simple astronomical solar system domain. The oval nodes represent class nodes and form the overall structure of the domain. The square nodes represent the instance nodes that are subsequently attached to the class nodes. Each class node is therefore used as an anchoring point for instance nodes. Therefore every instance node in the hypermedia is connected directly to the class structure of the semantic network. For example, the instance node “Earth” is attached to the class node “Planet”, thus all “Planets” are logically collected around the “Planet” class node. Each instance and class node is visible to the student as a hypermedia node. It is therefore of potential use to the student and author to utilise the class nodes for presenting higher level information that relates to the nodes connected to it. For example, the instance nodes “Earth”, “Mars” and “Jupiter” are attached to the class node “Planet”. The instance nodes contain information that relates purely to the node itself (the “Earth” node contains information about the Earth). The class node “Planet” may therefore hold information, useful to the student, about the higher level “Planet” node (which may not have been present without the need for the semantic network) and may provide information as to why each node attached to it, should be attached to it. For example, it may hold contextual information such as what a planet is and the kind of planets that can be found, further such information is in a logical place and need not be “hidden” in an instance node. This information may not always be clear or obvious to the student. For example, the student may not have been previously aware that both "Olympus Mons" and "Mount St. Helens" are "Volcanoes". In many hypermedia systems the only way to discover this fact is to traverse a link that exists between the two and rely upon the domain author providing the information within both nodes that each is a volcano. If the student is forced to undergo this procedure in order to discover the relationship between two nodes then they are further exposed to the “risks” of navigation in a complex information structure. The semantic network however reduces the danger of the cognitive loading problem and navigation problem associated with the hypermedia paradigm, by providing a student with contextual information and by forcing the author to structure the domain more logically (Pereira et al 1991, Jonassen and Wang 1993).

### 9.3.1 Link Types

The semantic network paradigm is termed a weak structure (Rich and Knight 1991) since it is not in itself, a rigidly defined construct. It is possible for an author of semantic networks (a domain author for this system or an author in general) to invent their own link and node types, or to devise internal representations for the semantic network nodes. The automatic linking algorithm devised for Hypernet, however, uses a predefined set of link types.

This standardised set of link types, described fully in (Beynon-Davies et al 1994) are defined below as:

**ISA** – relates an object to a class, i.e. it defines an instance of a class. For example "Mars" ISA "Planet", "Mars" being the instance of the class "Planet". This link is unidirectional in that properties are inherited in one direction only. For example the "Mars" node inherits all properties from the "Planet" node by virtue of this link (such properties or attributes may include the fact that a Planet orbits the Sun, thus Mars orbits the Sun). However "Planet" does not inherit properties from "Mars", e.g. it does not follow that every Planet has a predominately Carbon Dioxide atmosphere because Mars does. If such a bi-directional relationship is required, it may be defined by the addition of an extra ISA link in the opposite direction. The IS-A relationship usually represents information towards the top of a hierarchy and therefore represents more general information (Tudhope and Taylor 1997).

**AKO** (a-kind-of) – relates a class to another class, or may define a subset. This link type demonstrates the relationship between classes. This type is fundamental in connecting a collection of class nodes together to form one semantic network. For example “Planet” AKO “Solar System Body”, meaning that a “Planet” is a kind of “Solar System Body” and that a “Planet” inherits all properties of the class “Solar System Body”. Tudhope and Taylor (1997) have referred to the AKO link as “narrower-than”, since it represents information at the bottom of a hierarchy which is usually more detailed or narrower in scope.

**PARTOF** (part-of) – represents how an object is composed of other objects, or inherits only part of the parent class. This link type demonstrates how a class may be associated with component parts. For example, “Geographical Feature” PARTOF "Planet". The part-of link implies that there is a relationship between instances of classes connected via a part-of link.

**HASA** (has-a) – relates an object to a property or attribute. This is not used to represent structural information (it is not used as a link type). Instead it may be used to represent knowledge within a class. For example, a “Planet has-a Diameter”. This may be used to ensure that an object conforms to a class exactly (i.e. the author is aided in deciding if a node belongs to a particular class by virtue of it having the same attributes). As noted by Beynon-Davies et al (1994), this link type also aids queries in that a search may search for exact matches, such as “Planets with diameters greater than 100”.

The above link types allow the structure of a domain to be represented. However, further link types may be required to represent course material. For example, it may be necessary to represent dependencies, demonstrated in figure 9.3.B, if one concept must be learned before another, or to represent counter arguments. Extra link types such as "dependent-on" could be used by hypermedia systems to determine whether a node is relevant to a particular student. If a student has not visited the node at one end of the link then the node at the other end is not relevant and may be hidden or greyed out in accordance with the hiding method discussed in Brusilovsky (1996). The "dependent-on" link can therefore be used to represent course information, where it is common to lead a student through topics that are dependent upon each other. Further link types may include "example-of", "counter-example-of", “supported-by” and “disputed-by”. Link types such as these provide a greater depth of information about a particular node. Therefore it may be necessary to separate a node into several nodes representing various levels of detail. It is arguable that some node types such as "counter-example-of" may not be beneficial to lower-level students, since they may contain contradictory or otherwise confusing information; therefore there may be a case to make such links available to only the higher level students. Further, in the area of concept mapping, which can be said to share some common attributes with hypermedia, there can be any number of author (of the concept map) definable link-types (McAleese 1998). In essence it does not matter whether the domain author uses a predefined set of link types or whether they use their own. However, in the case of the automatic linking algorithm presented in this thesis, it does not make any special provision for link types, instead they are treated as all the same. However, it may still be useful for the domain author to represent concepts held by the structure of the domain by using their own link types. Other researchers have attempted to produce automatic links differently, depending upon the link types that connect two remote nodes. Tudhope and Taylor (1997) calculate automatic links by giving a stronger relationship between nodes connected by AKO relationships than IS-A relationships.

The reason for this is that relationships at the bottom of hierarchies, represented by AKO relationships, are thought to be more closely related than those at the top, represented by IS-A relationships (Tudhope and Taylor 1997). The use of different strengths of relationships has not been explored as part of this research project, although it may prove to be beneficial as part of further work. Further, it may be possible to combine the information potentially available from the student's browsing pattern to dynamically alter the strength of the link types, in that if a user is determined, by virtue of their browsing pattern, to be interested in general information then the strength of the IS-A link type could be increased and vice versa.

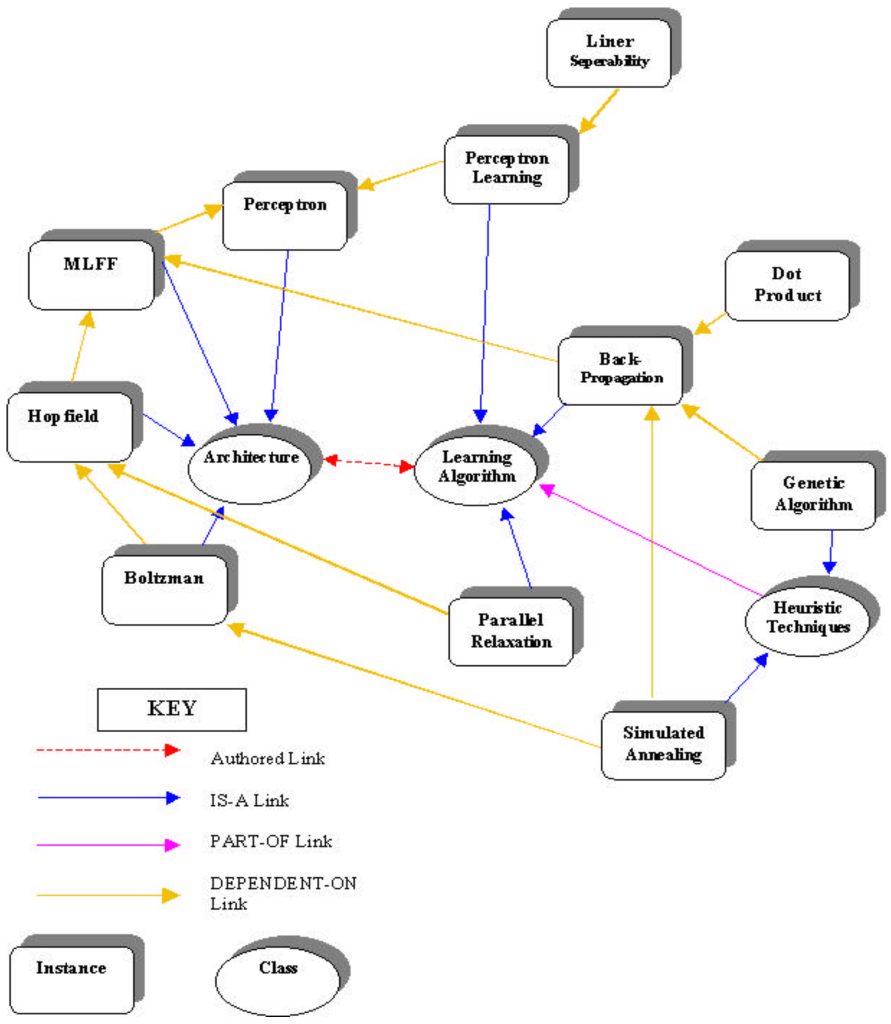


**Figure 9.3.B Representing Concept Dependencies**

Figure 9.3.B shows the dependencies that may be required for a neural network sub domain. The use of the "dependent-on" links ensures that the student visits certain nodes before others. For example, the student must visit the nodes concerning the "Perceptron" neural network before the "MLFF" neural network. In turn they must visit the "MLFF" node before the "Hopfield" node. This is because, in this case, it has been determined that the node "Perceptron" contains concepts that must be understood before the concepts contained in "MLFF" can be understood. The above example also demonstrates that the use of the "dependent-on" links "flattens out" the domain, in that the more dependent-on links there are the less navigational possibilities there are (at least at first) and hence the more linear the domain. The dependent-on links result in there being potentially less links to offer a student, until the student has visited the node at the other end of the dependent-on link. Determining whether a concept is understood could be achieved by the use of tutorial nodes, although Brusilovsky (1996) describes some Adaptive Hypermedia Systems that denote a concept as understood if the node containing it has been visited. In Hypernet, determining whether a concept was understood or not could be accomplished by attaching a dependency from a tutorial node, containing a tutorial on the concept, to the node that requires that the concept is understood, as demonstrated in figure 9.3.C.

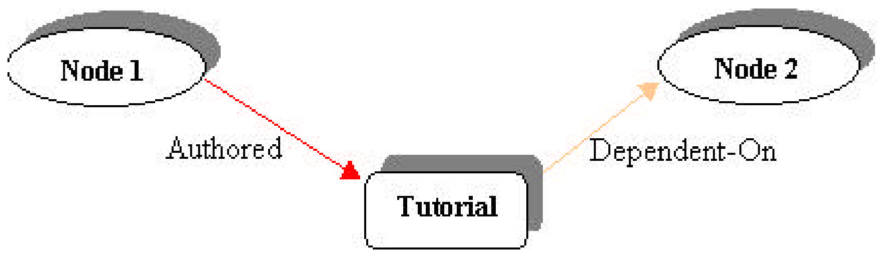


**Figure 9.3.C Using a Tutorial Node with dependencies**

Figure 9.3.C shows that in order for a student to visit "Node 2" they must have completed the tutorial, since it is dependent upon it. Once the tutorial has been completed it effectively disappears (for that student) and "Node 1" becomes connected to "Node 2" by the "Authored" link that originally connected "Node 1" to the tutorial node.

### 9.3.2 Automatic Reasoning to Aid the Author

The above three link types (ako, part-of and is-a), then, enable a domain to be defined as a hierarchical structure. The author is aided by this structure in that reasoning by the system ensures that a node is connected to the structure correctly. For example, given the simple domain below:

Domain Star_System

(Planet, Part-of, Star_System)

(Geo_feature, Part-of,Planet)

(Volcano,ako, Geo_Feature)

The domain author may define instance nodes as follows:

(Solar_system, Is-a, Star_System)

(Earth, Is-a, Planet)

(Mount St. Helens, Is-a, Volcano)

Using the hierarchical structure it can be reasoned that:

(Mount St. Helens, Is-a, Geo_Feature) since (Mount St. Helens, Is_a, Volcano, ako, Geo_Feature)

(Mount St. Helens, Part-of, Planet) since (Mount St. Helens, Is_a, Volcano, ako, Geo_Feature, Part-of, Planet)

The reasoning process therefore poses the question:

To which "Planet" does "Mount St. Helens" belong? (Mount St. Helens, Part-of, ?Planet)

The author would then respond "Earth" and a link (Mount St. Helens, Part-of, Earth) is created.

Therefore, the author is guided to provide some information based upon the hierarchical structure. This reasoning helps ensure that the link between instances of "Volcanoes" and instances of "Planet" always exists.

### 9.3.3 Node Types

Instance nodes in the hypermedia are assigned a type by the process of connecting them to a class node. This automatically clusters the node to its "relations". For example if the node "Pluto" is added to the domain, via the link "Pluto" ISA "Planet", then the node “Pluto” is automatically connected with all other nodes connected to the class node “Planet”. Thus the author is relieved of the responsibility of providing these highly related links and the student is assured that they exist. Thus, the ISA link type provides a set of highly related links (assuming other nodes of the same class have previously been authored). A further, larger, set of links may be automatically provided by virtue of the AKO link type, since class nodes are connected together via AKO link types into a hierarchical structure. A node connected to the class node “Volcano”, for example, inherits other nodes via the AKO link types that connect “Volcano” to other class nodes. For example, the “Volcano” node is a-kind-of “Geo-Feature”, it can be seen from diagram 9.3.A that a “Canyon” is also a-kind-of “Geo-Feature”. Therefore, a relationship exists between all “Volcanoes” and all “Canyons”, although this relationship is not as strong as the relationship that exists between all “Volcanoes”.

This relationship may be useful to the student in terms of placing the “Volcano” nodes in context (assuming the student is currently located at a “Planet” node) and in demonstrating information that is common to "Volcanoes" and "Canyons". For example, the student may be offered a link to a selection of “Canyon” nodes, or the “Canyon” class alone (depending upon their ability level and current interests, since a lower level student will not be presented with potentially confusing links). Such nodes may provide the student with extra information that may help them assimilate the knowledge they have learned about "Volcanoes". Jonassen and Wang (1993) argue that knowledge is stored in the human brain as semantic network-like structures termed "schemas". The theory is that in order for knowledge to be efficiently learned it must be connected to existing schema in the brain. It is therefore vital that information is presented to the student in relation to other information with which the student is already familiar. The student may gain an important insight into what a Volcano actually is by comparing and contrasting it to another related body (the Canyon class, in this case). Further, the student may be searching for some information of which the “Volcano” nodes are not necessarily part, they may therefore navigate the semantic network structure, using the link and node types as cues, to find nodes that more closely resemble their interests. This is, of course, the major advantage of hypermedia. However, the semantic network provides several helpful additions. These are the ability for the student to utilise the node and link types as navigational cues, the ability to for the student to examine exactly how two nodes in the hypermedia relate to each other (by virtue of the path connecting them through the semantic network) and finally by offering dynamic links that use information from the semantic network and information recorded from the student.

### 9.3.4 Adding Nodes to the Domain

Once a semantic network has been defined for the domain in terms of class nodes, then instance nodes may be added to it. The author must decide to which class a new node belongs, a class being represented by a class node (connected to other class nodes via ISA and/or AKO link types). If there is no such class node, then the semantic network has not been sufficiently defined for the domain or the new node does not belong to the domain. This is an important restriction, since there may be more than one author for the domain. It is possible that a domain may be initially defined by an author, but extra nodes are added to it subsequently, perhaps after students have begun using the hypermedia. The addition of the semantic network is an aid to all such authors, since it has the potential to help them to decide if a node should belong to the domain. An author who is unsure as to the legitimacy of a node’s relationship with the domain, may succumb to the “Linkitis” problem described above. The semantic structure aids such an author by providing them with a structure into which the node must fit. It is possible that the responsibility for deciding whether a node should be added to the domain lies with the original semantic network author, since it could be stated that if a node cannot be typed to an existing semantic network node, then it should not belong to the domain. This could be achieved by allowing only certain classes of author to modify or otherwise add to the semantic network. Thus, the integrity of the domain is protected to a far higher degree than would be the case with a hypermedia system without a semantic network.

Once a node has been correctly typed, it is connected to the semantic network. This immediately connects the new node to all the other nodes in the hypermedia system, via paths of varying lengths, thus the author need not specify any other direct links. It is likely however, that an author wishes to add a series of related nodes. For example, the author may wish to add the node "Pluto" to the domain. This could involve the addition of several new nodes, namely the node “Pluto” representing the planet itself and “Charon”, representing Pluto’s only satellite. The author therefore types “Pluto” is-a “Planet” and “Charon” is-a “Satellite”. Once “Charon” has been correctly typed, the author is required to connect it to a host “Planet”, since this is an explicit requirement of the “Satellite” type (a reasonable requirement since satellites do not exist in isolation). The original author of the semantic network encodes this requirement within the semantic node itself.

The semantic network is therefore a structure that connects the entire domain together. It is vital that the author connects the class nodes together in an accurate fashion. As discussed above, it may not be a clear cut decision whether to connect a node to a particular class node or not. The author may have to make a decision about connecting two class nodes together when the relationship between them is not a definite one. The author should detail this relationship within the class node so that any future author is clear about the defined relationship. It may not always be the case that such a one to one relationship can be defined, i.e. an instance node may not fit exactly into the semantic structure of the domain, but may still be valuable to the student studying the domain. In this situation the author should provide a suitable “non specific” semantic node. For example, the domain author may wish to add information about probe missions to various planets. The author may decide that it is not appropriate to add the semantic node “Probe Missions” as a-kind-of “Planet”. However, it may be appropriate to have information about the Voyager missions to Jupiter, with the "Jupiter" instance node. The author could therefore define the semantic node “Related Material” as part-of “Planet”. It is important that this facility is used sparingly, otherwise the benefits of structuring the domain could be lost.

## 9.4 Defining Tutorial Nodes

Tutorial nodes are employed by Hypernet as a means to aid the student’s retention of the domain contents. Reder (1985) suggests that a person retains knowledge more effectively if they are repeatedly exposed to it. Jonassen and Wang (1993) further suggest that information is retained by providing a context into which the information fits, thus aiding the student in determining where the information fits within their own knowledge representation structures (schema). In Hypernet this is aided by providing the students with tasks to complete, via the tutorial nodes.

Tutorial nodes may take the form of a simple question and answer session, or they may involve a more complex task that requires that the student visits several nodes. Such tasks may require the student to traverse the hypermedia looking for various pieces of information. Requiring the student to find and examine the relationship between concepts may also aid their retention of structural knowledge of the domain (Jonassen and Wang 1993). It is therefore paramount that tutorial nodes are designed and placed within the hypermedia in order to render this transfer of knowledge as efficient as possible.

The tutorial nodes are designed to remove the danger of passivity from the system, i.e. the student is not always left to wander through a hypermedia system, without a task or guidance, especially if they are a novice of the domain. It could be argued that a human teacher could present the student with a learning task before they are exposed to the hypermedia system. However, this would require the human teacher to supply the student with a complete task that would detail requirements for the student, as they become more proficient with the domain. Alternatively, it would require that the human teacher is constantly present, in order to gauge when the student requires a new more advanced task. The latter situation is not always possible, since the human teacher is an expensive resource and the need for ITSs has been driven by this fact (Woods and Warren 1995). The former situation is also not desirable, since providing the novice student with expert level tasks (for their future use), may be providing the student with a level of complexity that is not beneficial to them (Ridgeway 1989, Kinshuk and Patel 1997). The ideal situation therefore, is continually to provide the student with tasks that suit their current ability level (Elsom-Cook 1989). Tutorial nodes can be adapted to present different questions to different abilities of student.

This is similar to DeBra and Calvi's (1998) AHA (Adaptive Hypermedia Architecture) system.

Tutorial Nodes are embedded within the hypermedia, although they do not strictly form part of the semantic network, in that they do not represent the structure of the domain. A tutorial node may be triggered by a simple rule, defined by the domain author. For example:

**Rule 1** “Activate Tutorial Node 1 when the student enters the system”

The student is therefore given a task to complete as soon as they enter the system. It is advised that such a task should concentrate on providing the student with information about how to use the hypermedia tutoring system itself and not necessarily be related to the domain, since initially, a user who is new to a system expends most of their “cognitive power” learning how to use a new system (Preece et al 1995).

Other example rules are:

**Rule 2** “Activate Tutorial Node 2 when Tutorial Node 1 has been completed and the student has visited the “Planet” and “Satellite” semantic nodes and no other tutorial node is active.”

**Rule 3** “Activate Tutorial Node 3 when Tutorial Node 1 has been completed and the student has visited the “Earth” node and no other tutorial node is active.”

### 9.4.1 Linear and Non-Linear Tutorial Nodes

Tutorial nodes may exist in “linear hyperspace” and “non-linear hyperspace”. They may be designed to be encountered one after the other, guiding the student through the hypermedia (linear hyperspace). Alternatively, they may be activated by a student following their own interests into a particular area of the hypermedia (non-linear hyperspace). Linear hyperspace is similar to Trigg’s (1988) idea of paths authored into a hypermedia system. Trigg’s system, "TexNet", allows the author to specify default paths for the student to follow, in order to reduce the chances of them becoming lost. Such a method has been criticised by Jonassen and Grabinger (1990), for providing the student with a path that they may “blindly” follow, i.e. they are encouraged not to search the hyperspace. The same criticism can not however be levelled at the linear hyperspace tutorial nodes (LTN) employed by Hypernet, since they are not providing a node to node path. Instead the LTNs are providing the student with useful landmarks with which to orientate themselves, in that the student can navigate from one LTN to the next. They may therefore travel through the hypermedia in freedom, between such tutorial nodes. These nodes are suitable for novice students, since they are guided (as opposed to forcefully directed) by the tutorial nodes. Figure 9.3.A (shown on a previous page) shows two LTNs, labelled "Linear Tutorial 1" and "Linear Tutorial 2". For example, "Linear Tutorial 1" is activated when a student first uses the system. It may contain a simple tutorial demonstrating how the Hypernet interface operates. The link to the "Planet" class node is intended as the start link into the domain. Once the student has completed "Linear Tutorial 1" then "Linear Tutorial 2" is invoked. "Linear Tutorial 2" may provide the student with a simple browsing task to accomplish. The point is that the student always has a LTN task to complete, until all LTNs have been completed, or until they have reached an ability level that the author has determined can be the point where the student can indulge in their own discovery learning.

The non-linear hyperspace tutorial nodes are intended for more advanced students, since they do not provide the guidance that the LTNs provide. This is because the NLTNs do not appear in a predefined sequence. Instead, it is a matter of when the student uncovers them. They therefore do not offer the same landmark qualities as their linear counterparts. Such nodes are therefore more suited to students of higher levels, since the student may randomly encounter them and are not lead from one to the next.

The LTNs may be useful for lower level students and the NLTNs for higher level students. It may be desirable in some instances to use the two types of tutorial nodes interchangeably; for example, using LTNs for an expert level student who has entered an unvisited section of the domain.  It may also be desirable for the author to target certain tutorial nodes at certain levels of students, or to have slightly different tutorials, for each student level, present within the same tutorial node.  Figure 9.3.A (shown previously) shows two NLTNs, labelled "Planet Tutorial" and "Volcano Tutorial". These nodes are invoked when the node to which they are connected to via an "Invoked-by" link is selected. For example, the "Volcano Tutorial" is invoked when the "Volcano" class node is visited.

## 9.5 The Bottom-Up Approach to Authoring

### 9.5.1 Introduction

The "bottom-up" approach to authoring in effect provides the student population with a mechanism for structuring the domain. This is achieved by increasing the link weights slightly as a student moves from node to node. The result is that useful paths may emerge from the hypermedia domain over time. An analogy is the evolution of paths across a field. A person may find the best route from one point to another. As they travel, their feet make an impression in the grass. Other people then see this impression and naturally follow it, making the impression deeper. More than one path may exist; however, the best path will be used by more people, making intelligent decisions about which path, or parts of paths, is the best. Research has shown that paths emerge based upon two factors, the most direct route to a goal and the visibility of already available paths (Casti 1998). A path may not be the most direct route to a goal; however, it may involve less effort to follow parts of it or to combine it with other paths. In this way a complex network can evolve (Casti 1998). The analogy is not a direct one when applied to hypermedia, since a user/student cannot directly see their goal whilst using hypermedia. However, a similar emergence of paths and sub-paths may occur.

Many hypermedia researchers have discussed the virtues of paths within hypermedia (Halaz et al 1987, Jonassen and Grabinger 1990, Conklin 1987, Trigg 1988, Micarelli and Sciarrone 1996, 1998). Trigg’s "TexNet" system, for example, provides default paths for students to follow, although, as noted above, this approach has been severely criticised by Jonassen and Grabinger (1990). Their criticism is that providing default paths enables a student to blindly follow them and thus ignore the wealth of information present in the rest of the domain. The bottom-up approach to authoring does not seek to provide the student with such paths. Since each individual student has individual needs it is not appropriate to do so. Increasing the link weight slightly as a link is used slightly increases the possibility that another student will use the link, since the probability of it being offered to them is increased. These slight weight increases may build up over time until a once weak link may become strong. This follows the principles of connectionism and allows links that may not have been foreseen as useful by the author to emerge over time as the system is used.

### 9.5.2 Adapting the Links

Littleford (1991) advises that a hypermedia system should “take him [the student] where most other people have gone”, should the situation arise when the system is unable to offer navigation advice to the student based upon their interests or guidance from a human teacher. This situation will always arise if the system does not have "intelligence" embedded within the links (Rada 1990), since without such knowledge and further knowledge about the student/user, it is impossible to offer such help. The bottom-up paradigm may therefore be of use to many hypermedia domains that do not have any “intelligence” or structural knowledge encoded within them, e.g. the World Wide Web.

The bottom-up approach to authoring allows students’ movements to directly affect the link weights, stored within the hypermedia. Hypernet offers links to the student based upon weights associated with each link. The weights are therefore generated in accordance to the semantic network of the domain and are further modified in accordance with the student’s interests, as defined by their interactions with the semantic network. These weights are directly influenced by the author, in terms of the author defining the semantic network, which affects the links before the student encounters the domain. It has been suggested that in addition to the author defining links that the students should also be able to influence links, with respect to other students, since the students eventually become expert in the domain and, most importantly, know how to find information within the hypermedia (Brusilovsky 1996).

A mechanism has therefore been designed to allow this bottom-up authoring to take place. This form of authoring is termed "bottom-up", since it is achieved by modifying the links weights slightly as each student moves from one node to another. It is therefore viewing the domain from the lowest level, namely content-less link weights. It also takes into account "paths" that students may trace between several nodes, i.e. a path is a set of links that tend to be used by students in sequence. The system itself does not make any distinctions about the paths the student has taken, it simply modifies the link weight after each student follows a link. A path may therefore emerge from the domain, as a set of links whose weights have been increased, each link in the path is therefore more likely to be presented to the future student. The likelihood that the student follows the complete path or a section of it is therefore increased accordingly.

It is not appropriate to offer the student a complete path, since the path itself may not be obvious. This is because a path is not defined by one student moving through the hypermedia; this would result in a multitude of paths emerging that are suited to individual students only. The system therefore modifies the weights only slightly, so that repeated use of the same links by a number of students renders them more likely to be utilised by other students.

This bottom-up approach to authoring was inspired by the connectionist methodology. Neural networks learn by altering weights between neurones (nodes), forming a path between the inputs and the outputs (Gurney 1997). The weights in a supervised neural network are altered in response to an input and output pair (i.e. the neural network is presented with an input and its output is compared with the desired output). The weights are altered very slightly, at each presentation. This makes the neural network gradually learn how to map its inputs onto the desired output. It is undesirable to alter the weights a large amount for each input/output pair. For example, if the weights are altered so that for every input the weights are made to produce the correct output, then the neural network would only be able to model that last input/output pair that was presented to it. Neural network learning is designed to enable the neural network to model a complete set of input/output pairs. This can be achieved by altering the link weights by a small amount. These same arguments can be applied to the bottom-up approach to hypermedia authoring. If each time a student uses the hypermedia, their path is etched into the link weights, then the domain will rapidly become a profusion of paths, adding to and not alleviating the navigation problem. If however, several students tend to use the same paths (or part of several students' paths) then this path will have its weights increased more than paths that have not been used by as many students. The fact that a path has been utilised by several students indicates that it is perceived as a useful path and may be beneficial to other students. This is analogous to the evolution of paths across a field (Casti 1998). Such paths may not be obvious to the domain author, who approaches the domain from a totally different perspective, namely as an expert in the domain. The domain author may therefore overlook certain links and paths that may be beneficial to the student.

### 9.5.3 Altering Link Weights

There are complications with the bottom-up approach, however. It may not be appropriate to allow a link with a previously low link weight to have its weight increased beyond the point where it has become greater than a link that previously had a higher weight. For example, one link weight may be lower than another because it is lower in the domain hierarchy, as defined by the author. Through interactions with a student population, this link may become stronger than a link that is higher in the domain hierarchy. There is therefore a mismatch between what the author thinks should be a strong link and what the student population thinks should be a strong link. The author may decide that the above situation is not desirable and may therefore limit the students' authoring potential to ensure that the semantic hierarchy is not over written, by defining a upper limit to the change induced upon a link weight by the students. In any event, this problem is overcome to a large extent in Hypernet by presenting links representing the semantic network in a separate link window, however the student may be inclined to ignore these authored links in favour of the links that have had their weights increased via student interaction.

Determining the actual increase applied to the weights is a complex issue. It is important that the increase is only slight for each student that traverses the link, for the reasons discussed above. It may also be desirable to give certain students more authoring power than others; for example an expert student travelling along the links should perhaps be allowed to alter the weights more than a novice student because an expert student has more knowledge of the domain which could be used to guide the novice. This is similar to Micarelli and Sciarrone's (1998) idea of examining the lost student's partial browsing path, which represents a partial path to some solution, and then comparing it to complete browsing paths from successful students, so that a path to a solution can be offered to the currently lost student.

## 9.6 Discussion

This chapter has highlighted the need for the domain author to invest effort in producing a domain that is useful for the students. It has been argued by Woods and Warren (1995) and Ridgeway (1989) that in many cases the effort required to produce an ITS outweighs the benefits provided to the student. The key issue is therefore, does the effort of producing a hypermedia tutoring system outweigh the benefits provided by the system to the student? Again, the benefits to the student are difficult to verify, much less quantify. However, it is likely that a structured approach to authoring is better than an arbitrary approach (Jonassen and Wang 1993, Lowe and Hall 1999) and therefore the methods proposed here are a step in the right direction and warrant further study. However, it could be argued that in comparison to an ITS some of the effort required by the author has been reduced by reducing the benefit to the student, on the grounds that the use of domain-independent metrics that do not measure as many elements of the student's understanding as an ITS might. This may be true to an extent; however, ITSs have been criticised for over constraining students and for demanding too much of the author to make them practical (Woods and Warren 1995, Elsom-Cook 1989). In any case, the hypermedia tutoring paradigm is not intended to be a replacement for an ITS, which may also develop in the future. Instead it is intended to be a practical alternative to offering no student guidance. It is possible that an ITS could form part of a hypermedia tutoring system, so that the ITS could form a hypermedia node that the student could visit, and the results of the ITS could be used to gauge the student’s ability. It may be that a traditional ITS could be split up into several concepts that are to be taught, each concept could then form a node in the hypermedia. The ITS could be reformed into its original cluster of concepts by the use of “dependent-on” links, should it be necessary to do so for lower student levels.  A further issue, relating to defining the structure of a domain is the bottom-up approach to authoring.

The bottom-up approach to authoring has been discussed here. However it can be discussed as a concept only because it requires extensive trials with real student before the pathways can emerge and before a determination can be made as to whether these paths are of potential use to other students. However, it may be the case that such a system may become practical and useful over large browsing systems such as the WWW, since there is an increasing number of people using the WWW for a number of purposes, including education (Midouser et al 1999). Coupled with the likely increase in facilities, in terms of representation and programming facilities through the use of XML and Java, the WWW is likely to offer the potential to offer complex systems to remote users (Moody 1998, Lowe and Hall 1999).

A potential difficulty with the bottom-up approach is that it is difficult to determine when paths have emerged to a sufficient degree. This is a difficulty because it may be that if paths are continually “eroded” then the domain will not necessarily become a profusion of paths; rather there are likely to be only a few, most popular paths. This is in line with Casti’s (1998) description of the erosion of natural pathways, in that there is a “snow-balling” effect, whereby a path becomes more likely to be selected each time it actually is selected. It may be that a few paths are of use to many students;

however, it may also be that these few paths will limit the domain's diversity and so the system is likely to resemble Trigg's (1987) TexNet system, in that there will be default paths for the student to follow, albeit authored by students and not by the domain author. The idea of default paths, as stated earlier, has been severely criticised for use in education for providing the student with a path to follow blindly and thus causing them to ignore the rest of the domain (Jonassen and Grabinger 1990). Instead, it may be desirable to limit the authoring power of the students so that the erosion of paths stops, or slows down, at a point when as many useful and interesting paths as possible have emerged, although it is noted that without experimental trials it is difficult to say when this point will occur or whether it will be known. This is an important issue since it is difficult to say on an a priori basis what will happen, in terms of useful paths emerging, if students are given authoring power. However, it is likely to be the case that the most useful hypermedia structure for aiding the student has elements from the students and elements from the author, or expert. As such it may be necessary to specify a mechanism to allow this to occur, detailed below, although it may also be possible to present authored material to the student via a different mechanism than material authored by students, such as separate link boxes.

The above situation is similar to neural network learning, in that it is necessary to train a neural network to the point where it has learned only the generalities of its training data so that it can extrapolate what it has learned to other data that it has not been trained with (Skapura 1996). Therefore, there is an optimum point in the neural network learning cycle and it is desirable to stop learning at this point (although other factors can also contribute such as the amount and variation of training data). Therefore, it may be possible to employ elements of connectionism to the problem of reducing the effect of bottom-up learning. Employing a momentum term, for example, allows a link weight to degrade back to its original value, which may alleviate the problem of the domain becoming a few strong paths. It is suggested that the value of the momentum term may be determined by further research, in a similar fashion to the FRB (and explained previously in chapter 7). The range for the momentum term can, however, be narrowed since it is dependent upon the size of the domain and the number of students who are likely to use it, since there is a direct relationship between the likelihood of a node being visited (at an instance in time), the number of students using the domain and the number of nodes in the domain. For example, if there are a large number of nodes and only a few students then the probability of a node being visited at the next instance in time is low and the authoring power of the student may need to be increased in order for paths to emerge. The second parameter of concern is the amount by which to adjust the link weights as a student uses the link. It can be said on an *a priori* basis that it is necessary to set this parameter so that the weight adjustment is only slight, in order to avoid a multitude of paths emerging rapidly. Instead, paths should emerge slowly, so that non-useful paths do not obliterate or otherwise confuse the useful paths. This is because a path should be the result of a number of students using it and so indicating that a number of students found it useful and so future students may also find it useful.

The outcome of the bottom-up approach is similar to that of Micarelli and Sciarrone (1998), in that their system uses a case-based reasoning approach to compare a student's browsing pattern to date and then compare it to complete browsing paths (towards some goal) that other students have made. Once the closest match has been found the remainder of the path (from the other students) can be offered. Therefore, Micarelli and Sciarrone’s (1998) approach is attempting to offer previous students’ paths to the current student. The drawback of Micarelli and Sciarrone’s (1998) approach is that it requires the manual collection of student browsing information before any aid can be offered to the students. The bottom-up approach collects this data as the students use the system and it becomes more apparent to the students as the information builds up. The mechanism of the generation of such paths is dependent upon factors identified by Casti (1998). The key factors are the visibility of the already present paths (be they student authored or teacher authored) and the student’s goal. Therefore it is likely that different paths would emerge if students were not influenced by the presently forming paths, in the same way as walkers are influenced by increasingly eroded paths in a field, than if the students were not able to see the emerging paths. It therefore follows that the degree of visibility of emerging paths will also have a direct influence upon the continuing emergence of the paths in the hypermedia. The mechanism by which the students view the emerging paths is therefore paramount and it may not be a simple matter to decide how best to present emerging paths so as to encourage the most useful paths to emerge.

## 9.7 Conclusion

This chapter has discussed the implications of authoring a hypermedia system with structural knowledge embedded within it. The discussion has been aimed at hypermedia in general and may be of use to authors of any hypermedia domain or system. It has been argued that a semantic network is necessary for a true hypermedia system and that the authoring of such a construct is dependent upon the domain itself. Difficulties in defining a semantic network are dependent upon the structure of the domain in question, in that highly structured domains may lend themselves more readily to semantic network authoring than less well structured domains. The implications of Tutorial Nodes were discussed, with respect to providing the student with goals whilst browsing the hypermedia and it was argued that the location of tutorial nodes is key to ensuring that the student browse efficiently.

A second structural paradigm has also been discussed, namely the bottom-up approach to hypermedia authoring. This paradigm is based on connectionist ideas and allows both students as well as authors to affect the domain in a dynamic way. The hypermedia representation of the domain is able to adapt to a student population as it is being used, hence may better reflect the students' preferences.

# Appendix A

# Artificial Neural Networks

## A.1 Introduction

The purpose of this appendix is to introduce the technology of neural networks, since the body of the thesis deals with neural networks in some detail. A short overview of the subject is given followed by some design principles. For a more in-depth introduction the reader is referred to Gurney (1997).

## A.2 Introduction to Neural Networks

The early computer science researchers in the late forties and early fifties thought that operations such as understanding language would be relatively easy to perform in comparison to fast complex mathematics. They reasoned that since tasks that people found trivial such as understanding language, observing and tracking a moving ball with enough accuracy to catch it etc. whilst finding simple mathematics difficult to perform without aid, that a similar situation would arise with the new computer technology. It soon became apparent however that the reverse was true. Some fifty years later, speech recognition and, most importantly, understanding, to all intents and purposes does not exist and will not exist for the foreseeable future. One reason for this situation, although by no means the only, is that the design of a computer processing system does not resemble the brain processing system of biology. It must be remembered that the term "computer" would not have been used to refer to a brain fifty years ago as it might do today. The term actually derives from the task previously performed by human computers, people who would manually perform mathematical operations, usually for other people. Simple examples of this are logarithmic tables or sine tables. In most respects, the term computer is an accurate one, a modern computer is still either performing mathematic operation or moving numbers around in its memory, it is simply able to do it extremely rapidly.

The computer's inability to perform these tasks is simply because is it the wrong tool for the job; Rich and  Knight (1991) states that "neural network researchers contend that there is a mismatch between standard computer information processing technology and the technology used by the brain". Pioneering neural network researchers realised that if a computer were to solve problems that a human is good at then it should he designed to resemble the brain. The science of biology had already provided insights into the workings of the human brain. Today it is generally accepted that the basic mechanisms for the human brain are understood (the hardware). However, it is not well understood how these mechanisms combine to provide a truly remarkable device capable of dextrously controlling a body and providing the human mind and consciousness (the software).

## A.3 Foundations in Biology

Biological brains are neural networks, they are composed of approximately 10 billion neurones. Each neurone is composed of its inputs, called "dendrites" and its output, termed the "axon." Each axon splits into thousands of separate connections, which interface with other neurones or other tissues such as muscle, at a point called the synapse. The neurone operates by propagating and electrochemical signal from its input dendrites to its output axon, if the input signal is above a certain threshold. It is the neurone's ability to change its response to its inputs, which constitutes its ability to take part in the learning activity seen throughout the brain (Dimantaras and Jones 1996).

Donald Hebb provided the basis on which to build a neural network when he devised a theory of neurone behaviour (Hebb 1949). His theorem was simple and allowed Rosenblatt (1958) to devise his "Perception" neural network. The Perceptron is capable of learning any linearly separable task, however as Minsky and Papert (1969) pointed out, there are many problems that are not linearly separable, such as the simple XOR logic function. It is for this reason that other neural network architectures have been devised. Criticism at such a fundamental level halted neural network research in America for over a decade.

Computer based neural networks are more accurately termed Artificial Neural Networks since they are only emulating the behaviour of a natural neural network. The computer variety must harness the serial power of the modern microprocessor. Such an approach can not accurately model a fully parallel, natural neural network. An artificial neural network executes in cycles, the whole network of neurones are executed in series in response to some input. In biology this is not the case, neurones may respond to stimulus at any time, resulting in parts of the network seeming to separate out into sub networks. Artificial neural networks are usually designed to perform a specific task, this is mainly due to constraints imposed by the power of the, usually serial, computer on which the network is running.

Modern neural networks are a long way behind their biological counter parts. Estimates have been made that suggest that the most powerful artificial neural networks are only about as powerful as a quarter of a slug's brain (Skapura 1996), although he does note that computer science has taken around fifty years to reach a stage that took evolution millions of years.

## A.4 Artificial Neural Networks

An artificial neural network is comprised of an architecture and a learning algorithm. The architecture is the arrangement of the neurones within the network, i.e. how they are linked together. The learning algorithm is the program that each neurone runs every time the network is executed.

Most artificial neurones are a simple summing program. They take the sum of their input activations, then depending on their internal state, activate their output. The internal state changes in response to input activations over time, as well as output activations, i.e. the supervised learning paradigm dictates that a network must be informed whether or not it has produced an acceptable response. The neural network is judged on its ability to successfully produce a correct output given a certain set of inputs. An unsuccessful attempt induces a change in the neurones internal states.

Neural networks are practical and popular devices for solving problems that are intractable for traditional computer science approaches (Alexander 1989). Their applications can be broken down into the following areas.

### A.4.1 Classification

Perhaps the most common use for a neural network. Classification problems may result in the network making a simple binary decision or recognising a presented input into one, or more, of several classes. Example are detecting good or bad apples from a video image, or identifying enemy aircraft (Maren 1990).

### A.4.2 Autoassociation

A network is trained to produce its input at its output. Such a network can reproduce a filtered output from noisy data, or produce a complete output from a reduced input. Applications for this type of network include noise filters for telecommunications and data compression for video images.

### A.4.3 Time-Series Prediction

Neural networks can be trained to predict the next item in a sequence of data over time. Applications for this include predicting stock market prices and sporting scores.

### A.4.4 Function Approximation

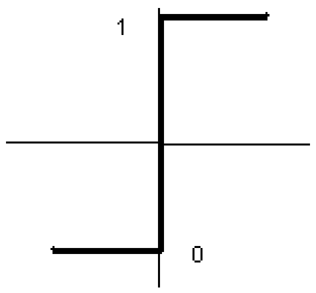
Strictly, every neural network fits into this category since it maps inputs onto outputs. However, in the narrow sense these networks are used to learn complex mathematical functions.

The neural networks in this project attempts to model students using browsing strategies and ability to answer questions, this is a classification problem. A review of relevant literature, most notably Maren 1990, Hagan et al 1991, Masters 1993) has stated conclusively that the multi-layer feed forward architecture (MLFF) is the best architecture for the vast majority of neural network applications. Theoretically, a MLFF is capable of learning anything that can be learned by a neural network (Maren et al 1990, Masters 1993), although this does not mean that it is necessarily the most efficient model to employ.

## A.5 Fundamentals of Neural Networks

### A.5.1 Activation Functions

The activation function determines the output of a neurone in a neural network. The original neural network, the Perceptron, has a simple threshold function as its output function (fig A.5.1.A) simply the output is either one or zero. However, there are advantages to utilising a differentiable output function, rather than a simple on/off output. The most common output function is the logistic sigmoid function (fig A.5.1.B).



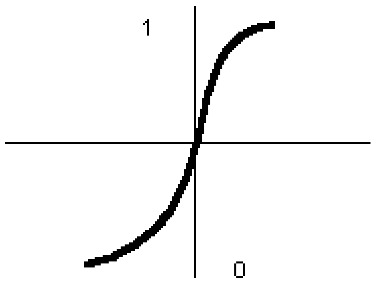


Figure A.5.1.A Threshold (stepwise) Function fig A.5.1.A Sigmoid function

The sigmoid activation function allows neurone to perform complex learning even when a small number of neurones are employed.

**A.5.2 Minima**

The back propagation algorithm attempts to reduce the error of the mathematical system represented by the neural network's weights and thus walk downhill to the optimum values for those weights. Unfortunately due to the mathematical complexity of even the simplest neural network, there are many minima, some deeper than others. The deepest minima is the ideal one and is dubbed the global minimum, the inferior minima are termed local minima. Local minima present a problem to gradient descent algorithms as the algorithm is invariably attracted to the nearest minima, which may not be the global one. The problem is exacerbated by the fact that two identical neural networks trained on the same training set for the same time will almost certainly produce different results, because one network will have reached one minimum while the other reaches a different one. This is due to the different random starting weights representing different starting locations on the mathematical landscape.

**A.5.3 Simulated Annealing**

Local minima may result in a non-optimally trained network, in many cases this may be acceptable. If it is not, then it is necessary retrain the network and repeat until suitable performance is found. However, there are facilities designed to increase the efficiency of the learning process, these include simulated annealing or a genetic algorithm (Masters 1993) to try and prevent the situation arising. Annealing is a metallurgic process, the usual random organisation of carbon atoms in steel is converted into a layered structure which results in a less brittle metal. The process involves heating the metal to a very high temperature and then cooling it very slowly. The atoms at high temperatures have a high energy, which causes them to vibrate, as the temperature decreases the vibration decreases and the settle into organised layers. This idea has been incorporated into the learning algorithms for neural networks, the temperature being the learning rate that is gradually reduced. The idea is that if the network settles into a local minima then it is "shaken" in an attempt to dislodge it from the minima. If the "temperature" is kept constant then the system will jump from one minima to another and never settle. If the ability to jump is steadily reduced then the trend will be towards the global minimum and once the global minimum is reached then the shaking will be insufficient to dislodge it.

## A.6 Neural Network Architectures

### A.6.1 The Perceptron

Neural Networks originated from work by Rosenblatt (1962); his Perceptron employs a simple threshold function (figure A.5.1.A) to activate its output from input stimulus, allowing it to output either a one or a zero. The Perceptron was effectively killed in 1969 by criticism raised by Minsky and Papert (1969), in effect this criticism also destroyed neural network research as a whole in America. Minsky and Papert's main criticism was that it could not solve non-linearly separable problem, i.e. a problem where a dividing line can not be drawn between inputs and outputs. They specifically cited the logic XOR function that they argued was such a simple and fundamental problem that the Perceptron's inability to solve it rendered it an academic curiosity only. What Minsky and Papert failed to realise was that this problem could be fixed by combining Perceptrons into a multi-layered Perceptron. This architecture is capable of solving non-linearly separable problems. Rosenblatt realised this but could not formulate a suitable learning algorithm, it wasn't until Rummelhart and McClelland (1986) published their back-propagation learning algorithm, that neural networks became popular again. Note that some books refer to a multi-layer Perceptron (MLP) and a Multi-Layer Feed-Forward neural networks interchangeably. This is not strictly accurate because a MLP still utilises the simple stepwise activation function.

### A.6.2 Adaline/Madaline

The Adaline neural network was developed in parallel to the Perceptron in the early sixties by Widrow and Hoff (1960). It is similar to the Perceptron except that it uses a linear transfer output function. The Adaline achieved success as an adaptive filter. The Madaline neural network, similarly to the MLP, is a combination of the smaller unit, with a more complex learning algorithm.

### A.6.3 The Hopfield Network

The Hopfield network (Hopfield 1982) was developed as a theory of memory. A Hopfield network can be used to store information that may be retrieved by supplying only a portion of the information at the network's input. The network is robust in that some of the neurones may be damaged (in some way) without, necessarily, impairing the stored memory. The operation of the Hopfield network is quite different from other networks in that the learning process is a highly iterative. Each Neurone has connections that are excitatory or inhibitory, a neurone has a state of either active or inactive. Its state is calculated based on the states of other neurones around it and its own state. Neurones are computed at random in order to finalise the state of the network.

### A.6.4 The Boltzman Machine

The Boltzman Machine is a Hopfield Network with the addition of simulated annealing (Sejnowski and Rosenberg 1987). The Hopfield network is not suitable for problem solving because it has a tendency to settle in local minima. The simulated annealing approach is analogous to shaking the network to dislodge it from a local minima and towards the global minima.

### A.6.5 The Multi-layer feed forward Network

The Multi-layer Feed Forward (MLFF) architecture is the backbone of modern neural networks. There are many variations, however in essence the architecture and learning algorithm are simple and fast (Masters 1993).

A MLFF network is comprised of several layers, an input layer, an output layer and one or more hidden layers. The input neurones are not actual neurones, as they perform no processing on the incoming data. The hidden and output neurones do perform a computation the result of which is “shaped” by the output function. The term feed forward (note: not the opposite of back propagation) relates to the fact that all neurones can only be connected uni-directionally to one or more neurones in the next layer (with the obvious exception of the output neurones). Each connection has a weight associated with it, this is added to the signal which passes along the connection. Weights can be positive, excitatory, and negative, inhibitory. The neurone calculates its activation by summing up its inputs and applying it to its activation function. Once the output of every neurone in a particular layer has been calculated then the next layer can be calculated since the outputs of the current layer form the inputs of the next. Once all the neurones have been processed then the output neurones finally present their results, which may or may not be as desired. If they are not then some learning is required.

The neural network gains its remarkable abilities because of the complex mathematical system that the combined neurones generate. One could argue that the connectionist approaches biggest asset is also its biggest drawback, since the complexity of them makes them almost impossible to describe or predict mathematically.

### A.6.6 Regression Networks

A regression network is a MLFF network with feedback connections, i.e. a layer may not only connect to the layer ahead of it, but it may connect to the previous layer also. This embodies the network with the ability to compare stimuli currently being received with a previous stimuli, enabling the network to compare current inputs with past inputs. Such networks are predominantly used for time series prediction problems, such as forecasting the stock market (Maren 1991).

## A.7 Supervised Learning

### A.7.1 The Back propagation Learning Algorithm

A distinction must be made between the architecture of a neural network and its learning algorithm, the above is a description of a particular architecture and how the neural network calculates its output from a set of inputs. The following is a description of the learning algorithm that is used to tune the network so that it becomes better at producing the desired output.

There are two forms of neural network training, supervised and unsupervised. All MLFF networks are trained by the supervised method. The supervised training method involves asking the neural network to perform its function and then showing it the correct answer so that it can learn from its mistakes.

The term back propagation does not sit comfortably with the term feed forward and is often confused; back propagation is in fact a mathematical technique for calculating errors in a complex mathematical system, such as a neural network. It is one of a number of gradient descent algorithms, which are inversely similar to more traditional Artificial Intelligence approaches such as gradient assent algorithms. Such algorithms map the function onto a three-dimensional surface, with low land valleys and up land hills. Depending on the problem the lower the point on the landscape the better the output of the function (this situation is reversed for gradient assent algorithms).

Supervised training involves sampling a representative sample of examples from the problem domain, this is then split into the training set and the validation set. The network is initialised by setting its weights to random values. The network then alters these weights in response only to the training set. The validation set is used to determine that the network is successful with examples from the same population but with which it has not been trained. The network is presented with a subset of the training set, one example at a time, after it has "seen" the whole subset then it updates its weights in response to the measure of error incurred by the network. This process, termed an epoch, is repeated until the network is sufficiently trained. The size of the subset is termed the epoch size and may be the same as the training set size, if it is not then it is important to choose each member of the subset randomly from the training set.

The measure of error is the mean square error of output activations; it is calculated by squaring the difference between the actual and target activations and then finding the average across all output neurones. It is possible to determine which way the weights must move in order to reduce the error by finding the partial derivative of the error. The size of the step taken in that direction is termed the learning rate, if too big a step is used the optimum weights may be stepped over, too small and the optimum weights may take an eternity to reach.

The biggest drawback of the traditional back propagation algorithm is that it is heavily influenced by the local gradient, it does not necessarily take a direct route. For example, if the global minimum was located at the end of a valley and the current position is at the side, above the valley, and then the back propagation algorithm would take a step in the direction of greatest gradient, over the valley side. Once it had discovered the other edge of the valley it would zig zag back and forth making small steps towards the goal. The final route taken can be thousands of time longer than the shortest route, and hence learning time is thousands of times longer.

The standard back propagation algorithm can be improved by adding a momentum term into the equation. This effectively filters out local gradients and allows the more global gradient to take influence.

The step size, or learning rate, of the standard back propagation algorithm is fixed, this leads to the algorithm "thrashing about" around the minima as the algorithm can not precisely find the bottom of two gradients. I.e. it steps down one, over the minimum and lands half way up the gradient at the other side. The Conjugate

Gradient method allows the learning algorithm to make progressively smaller steps as it approaches a minimum, thus it can reach a point close to the actual minimum very quickly.

## A.8 Unsupervised Learning

### A.8.1 The Kohonen Model

The Kohonen model proposed by Teuvo Kohonen (1989) does not attain the highs of practical application that have been achieved by the back propagation model. It is however worthy of further discussion for three reasons. Firstly, it is purported to work in a similar way to a natural brain, or at least much closer than a multi-layer feed forward network. Secondly, because of the first reason it is often mistook as being the defacto neural network. The final, and most compelling, reason is that the network utilises unsupervised training that makes it uniquely applicable to situations where the implementers are not sure what it is the neural network will be classifying.

The learning algorithm employed by the Kohonen network is sometimes termed ‘creature learning’ due to the similarities between it and the method thought to be employed by a natural brain. It alters its weights according to a competitive learning system; each neurone in the network competes for the right to adjust its weights. In some implementations only one neurone is allowed to alter its weights, in others, neighbouring neurones are also allowed to alter their weights, but to a lesser degree. The idea behind this is that if the correct output has `won`, perhaps by a narrow margin, then a weight increase at that neurone is likely to make the same result more probable, should similar circumstances arise. Thus behavioural patterns arise and strengthen as the network learns. The unsupervised learning paradigm bares resemblance to `creature learning`, in that a new born animal is essentially alone and must learn first to recognise and cope with its immediate surroundings before it can be engaged in supervised learning from a parent.

Some recent work conducted by cognitive scientists, physiologists and computer scientists has concluded that a natural brain is built up from several neural networks, the smallest are called the primal neural networks. These networks are extremely ancient in origin and probably form part of every animal brain. They are concerned with identifying beneficial behaviour, such as pushing or pulling, moving, touching etc. These simple actions are the prerequisites for any other form of learning and because of the ambiguous nature of what might or might not be beneficial, they are unsupervised.

Undoubtedly this network, or one of its progeny, will achieve increasing academic interest and practical applications. Computers are not good at making decisions for themselves, neural networks are remedying the situation to some extent. The Kohonen network, however, is capable of making discoveries for itself, which makes it almost unique in the entire field of computing.

## A.9 Neural Network Design Principles

The multi-layer feed forward network is comprised of an input layer, one or more hidden layers of neurones and an output layer of neurones. There are no theoretical guidelines that enable us to determine what architecture a particular problem requires.

The only course of action therefore is to experiment with several variations. Consulting the practical neural network literature (Maren 1990, Masters 1993, Diamantaras 1996, Cichocki and Unbehauen 1993) gives the following advice: In practical terms it is never necessary to have more than two hidden layers and uncommon to require more than one. Mathematical theory also unequivocally states that a two hidden layer neural network is capable of learning any problem capable of being learned by a neural network  (Masters 1993). This does not necessarily mean that a two hidden layer is the best architecture for the problem however, adding additional layers may make the network much slower in both operation and learning, as well as perhaps introducing the problem of over fitting (described below).

The number of neurones in the hidden layer is crucial to the performance of the neural network and is finely poised between ability and speed of learning. The more neurones the more capable the network but the longer it will take to train, more neurones also increases the likelihood that the neural network will memorise its training set (over fitting), rather than learning the parts of the training set that represent the population as a whole. Masters (1993) proposes a guideline for designing neural networks, given that most networks have more inputs than outputs, then the number of hidden neurones should be midway between the two, thus giving a triangular shape.

Selecting the number of hidden neurones is often a trial and error process, however it is better to underestimate than overestimate. The often reported problem of over fitting is really a symptom of having too many hidden neurones. The extra neurones give the network the ability to memorise the idiosyncratic details of the training set as well as the general details, thus they come to focus on the insignificant details and over fit the training set. Reducing the number of hidden neurones cuts down the network's memorising power and so, if trained properly, it will recognise the significant details only. Obviously too few hidden neurones will result in poor performance. However it is easier to determine that a network has too few neurones than too many, a network with too many may perform well until it is placed in its working environment (Masters 1993). Note that a myth has surrounded neural network training that suggests that a neural network can be over trained. This is the result of a neural network over fitting a training set because it has too many neurones or the training set is too small.

The problem of over fitting may also occur if the training set is too small or not representative of the population as a whole. The number of hidden neurones and the size of the training set in inexorably related, incorrectly defining either may result in over fitting or inability to learn.

From the above it can be seen that there are a significant number of variables, each of which can not be simply mathematically selected. Instead the network designer must experiment with many combinations using the heuristics described above in an attempt to attain the perfect network. Note that it is usually only an attempt as usually a near perfect network will suffice. However all this experimentation does present a very real problem to the designers of feed forward neural networks, simply because for each permutation of the network architecture involves completely retraining the network, which is the most time consuming aspect of  utilising the feed forward architecture.

The designer must carefully select the inputs for the neural network, too many and the network becomes more complex and thus slower to train and execute. However a rich set of inputs allows the network to make connections which may not be readily apparent to the human observer. There is no easy solution to this problem, again it is simply brute force in experimenting with different combinations.

### A.9.1 Summary of Neural Network Design

In summary the designer of a feed forward neural network should follow the following guidelines.

Use as few hidden neurones as possible.

Use as few hidden layers as possible.

Carefully select a representative training set from the domain.

Over training is a myth, so long as the training set is of the correct size and content.

Over fitting is a symptom of too many hidden neurones or a poor training set. Over fitting is almost always attributed to over training. This is when designers perceive that the network has been allowed to ruminate on its training data for so long that it has memorised the complete set and thus is unable to generalise when its meets examples from outside that set. This undoubtedly can happen, but it should not if the network has been designed properly. If the network has been supplied with too many neurones then it has the capacity to store not only the parts of the examples that make them unique but also the parts that do not. If only enough neurones are supplied to store the common factors then the neural network will by mathematical definition not be able to over write them with those parts that are not common. It is of course difficult to determine this ideal number. However if over fitting is encountered it simply means that the network should be redesigned with less neurones.

Carefully select inputs, too many and speed will be affected, too few and the network may not have enough information.

## A.10 Hybrid Systems

Various neural networks have been combined successfully with other neural network architectures and with other computer and statistical systems. However, breaking a problem down is exactly what a neural network does and, usually, it does a more than adequate job. There are times, however, when a problem should be broken down. Essentially a problem may be broken down if the single network is too slow, or when a problem dictates that it should be broken down.

# Chapter 10

# Conclusion and Further Work

## 10.1 Conclusion

This thesis has examined an approach to educational hypermedia that attempts to structure the domain in such a way so that a novel student model is able to extract information about the student, with the aim of using this information to offer the student navigational aid. The need for tutoring systems that can be applied to many domains without the need for extensive reprogramming has been discussed and it has been argued that the approach discussed in this thesis is a step in the direction to achieving this goal. The overall aim of the research project was to use the hypermedia paradigm as the basis of a tutoring system, so that issues pertaining to navigational aid could be explored. Navigational aids are useful since in order to render hypermedia a suitable educational medium it is necessary to reduce some of hypermedia's endemic problems, described by (Conklin 1987), by the addition of AI facilities (Brusilovsky 1996). The AI facilities discussed in the current research project are drawn specifically from the fields of symbolic AI and connectionist AI.

The thesis has argued that traditional ITSs have fundamental weaknesses as practical educational tools, since they are difficult and expensive to produce and have been the subject of serious criticism for overly constraining the student (Ridgeway 1989, Elsom-Cook 1989, Woods and Warren 1995, Kinshuk and Patel 1997). By contrast hypermedia systems have been criticised for not constraining a student enough, thus allowing them to get lost in a potentially complex maze of information (Theng and Thimbleby 1998). To combat these problems a hybrid paradigm has been discussed, consisting of a semantic hypermedia system with facilities to aid the student in their navigation.

It has been argued that a semantic structure is useful for most hypermedia systems, and helps to provide the student with structural information that may aid them in assimilating the knowledge that they have learned (Pereira et al 1991, Jonassen and Wang 1993). Such a structure also provides an essential navigational aid, since it forces the author to structure the domain hierarchically and logically and hence provides the student with a more meaningful structure to navigate. Further, the use of the semantic structure enables the use of an automatic linking system, that is able to produce links with a relevance factor (weight) based upon relationships within the semantic structure. Further, the addition of a student model allows the production of a dynamic linking system, since the weights can be changed in the light of information recorded about the student and stored in the student model.

The novel student model conceived and designed for this research project is based on domain independent metrics. The advantage is that the student model can be employed for a variety of domains with little modification to the system architecture (although the hypermedia content and tutorials will need to be changed), whereas a traditional student model would usually need to be completely reengineered (Kinshuk and Patel 1997). The disadvantage is that domain independent metrics are inevitably less suited to a given domain. It has been argued, however, that the student model does aid the student considerably, in comparison to no student model at all, and that this methodology is therefore potentially useful.

The purpose of the student model discussed within the thesis is to collect information about the student, without resorting to domain-dependent metrics. Once collect the information can then be used to tailor the hypermedia environment to suit that particular student at a particular instant in time. The student model that forms part of the educational hypermedia approach, discussed as part of the current research project, contains four sub-systems: the Student Experience Record (SER), the Tutorial Supervisor (TS), the Browsing Pattern recogniser (BPR) and the Fuzzy Rule Base (FRB). These sub-systems together collect and utilise information concerning the students' browsing habits and results with tutorials contained within the hypermedia.

The SER is a record of the student's interactions with the hypermedia (termed content-based browsing), so that the links can be tailored towards the student's interests. This can be achieved by recording the classes of nodes that the student has previously visited and then giving other nodes of the same class a higher priority, so the likelihood of them being offered to the student is increased. This has the effect of bringing the most relevant nodes, in terms of the student's recent browsing habits, towards the top of a list of nodes that are previously ordered in relevance order, according to the semantic network alone. This system is similar to keyword based web-agents (Armstrong et al 1995), although it is potentially more powerful since the hypermedia domain is highly structured. However, a possible drawback of this method is that there is a danger of a "snow-balling" effect, in that once a student demonstrates an interest in a particular area of the domain then there is an increased likelihood that the student will be offered further material in that area, when they may be no longer interested. In order to combat this problem the area of content-less browsing patterns was examined, since content-less browsing patterns provide the potential to identify whether a student requires more specific information of not. This issue was investigated by using a neural network device called the Browsing Pattern Recogniser.

The Browsing Pattern Recogniser is a neural network device for recognising a student's content-less browsing patterns through the hypermedia, identified previously by Canter et al (1985). The BPR is an attempt to provide the mechanism, called for by Canter et al (1985), for automatically identifying content-less browsing patterns so that other elements of a browsing system can utilise the information to aid the student. Thus, potentially useful information can be extracted from the student by examining their content-less browsing pattern alone. This is of potential use for situations where it is not possible to employ a tutorial system often enough to provide a useful grading of the student, or where there is no tutorial system at all, such as World Wide Web browsing systems. The BPR was trained using browsing data designed to mimic the browsing patterns identified by Canter et al (1985) and was able to successfully identify further browsing patterns. Once the browsing patterns have been identified, the FRB is used to link the browsing patterns to a student ability, as identified by the TS, and to the types of task that the student was using the hypermedia for, from the tutorial nodes.

The FRB is a hybrid fuzzy logic-neural network system that is used to ascribe meaning to browsing patterns collected by the BPR. The FRB acts like a neural network, in that it classifies data where there are a potentially large number of examples upon which to train it. The difference is that the FRB is internally interpretable. The FRB may be set manually with initial values (like a standard rule-base) and may then adapt to any misconceptions in the original rules whilst it is in use. The human researcher may also interpret the FRB, once it is fully trained, and the rules that have formed during training may be manually examined and altered if necessary. The interpretability of the FRB is of use since it enables the hypermedia researcher to examine how various browsing patterns are employed by various students. This information may then be used to better design hypermedia systems that encourage beneficial browsing. In practical experiments with simulated browsing data the FRB was able to learn quickly enough to warrant further investigation and successfully accomplished the task of mapping simulated content-less browsing patterns onto student abilities. The FRB may be used in conjunction with the BPR and TS and real student browsing data to identify potentially useful information regarding content-less browsing patterns, the types of students that use them and the types of task to which they are put.

The Tutorial Supervisor is used to provide a direct measure of the student’s ability by measuring the student's performance with tutorial nodes. A tutorial node engages the student in a dialogue, either as a task or as a question and answer session. The TS uses a particular type of neural network, a Kohonen, the advantage of which is that it is able to readapt its output grading of the student to different ranges of marks that may arise with different domains, without the need for human intervention. It is also able to adapt to potentially erroneous ratings of question difficulty, applied by the question designer. For example, if an author defines a question as easy, but subsequent interactions with a population of students determines that the question is in fact difficult such a question will be dynamically re-graded by the TS. The following section details how these sub-systems may be improved and utilised as part of further research.

## 10.2 Further Work

The current research project has identified many areas that could be further investigated. The major area of further research is the use of the facilities discussed within this thesis with real students. It is anticipated that this will enable further research into the other areas discussed below.

### 10.2.1 Collecting Student Browsing Data

This research project has proposed a novel tutoring system, based upon the addition of several AI facilities to the standard hypermedia paradigm. These systems are based upon connectionist technologies that require extensive training data from the domain to which they are to be applied. Obtaining such data was not possible within the confines of the current research, since the amount of data required would necessitate the collection of thousands of interactions with the hypermedia system. An alternative approach was therefore undertaken, namely simulating the large amount of data required. It is unlikely that enough student data could be obtained from classes of local students using the system. It is therefore proposed that the system be utilised over the Internet as a distance learning system. The utilisation of the system by a large number of students over a prolonged amount of time should provide enough information to thoroughly examine the real life application of the system. Such information when utilised with the FRB may also provide the as yet unknown information linking browsing patterns to student abilities and the tasks that they use the hypermedia for.

### 10.2.2 Utilisation with the World Wide Web

Increasing numbers of students and increasing use of the World Wide Web (WWW) has driven the need for generic tutoring systems (DeBra and Calvi 1998, Midouser et al 1999). This thesis has explored issues relating to a hypermedia-based tutoring system that could be expanded to cover delivery over the WWW. The WWW has been severely criticised for not offering sufficient linking facilities and methods for structuring nodes (Lowe and Hall 1999). However, it is likely to become more complex with the introduction of the Extensible Mark-up Language (XML), which is expected to replace the Hypertext Mark-up Language (HTML) as the standard language of the WWW (Moody 1998). XML coupled with the continuing development of the JAVA programming language is likely to enhance the WWW and therefore open up opportunities for software engineers to provide further functionality at the server end that can produce extra functionality at the client end (Moody 1998, Lowe and Hall 1999). The likely result is that complex programs could be executed on an Internet server but be viewable on the client Internet browser on the user's computer. Such a client/server based Internet system has already been developed. Hyper-G is such an Internet-based system that operates independent of the WWW (Maurer 1996). Hyperwave is a WWW based version of Hyper-G, which offers the expanded functionality over the native WWW. However, delivery is provided by WWW servers and is viewable from WWW browsers. Hyperwave demonstrates the feasibility of the approach of using an Internet server to hold the intelligent facilities (Fuzzy Rule Base, Tutorial Supervisor and Browsing Pattern Recogniser) and the domain for the students to access at the student's remote location. The upshot of this is that there are likely to be a greater number of students that have access to the system (Callaghan and Hand 1996). Therefore, there is a greater probability that enough information will be extracted to draw conclusions concerning how browsing patterns relate to students and the types of task that the student is using the hypermedia for.

### 10.2.3 Expansions to the Browsing Pattern Recogniser

#### 10.2.3.1 Expanded Hyperspace Representations

The Browsing Pattern Recogniser was designed to recognise browsing patterns composed of the browsing constructs previously identified by Canter et al (1985). These constructs do not take into account all possible information about the student's browsing habits. For example, extra information could be gathered from the student, concerning the types of nodes that they have visited and the time that the student has spent at each node. Since a large number of students would be required to examine browsing patterns and hypermedia learning (Jonassen and Wang 1993, Dillon and Gabbard 1998) a browsing pattern generator, called the Virtual Student Program (VSP) was utilised. The drawback of the VSP is that it can model only previously identified browsing constructs. It would be unsuitable to model link types and the time a student spends at a node using the VSP since no information is currently available concerning these items, instead only information regarding the student moving to previously visited or unvisited nodes is taken into account. However, as indicated above, further research could gather the required large amount of student browsing data by allowing users of the Internet to use the system. Browsing data would then be recorded and experiments could be carried out off-line using the data. Once real student browsing data is available then the BPR could be redesigned to incorporate extra input parameters, such as time spent at each node and various link types. The pros and cons of using extra information have been discussed in chapter 6.

#### 10.2.3.2 The Kohonen Neural Network for the BPR

The Kohonen neural network, which was used for the TS but not the BPR, uses an unsupervised learning neural network. An unsupervised learning neural network is potentially able to pick out new browsing constructs, without the need for them to be previously identified. The Kohonen model was not employed for the BPR since there was not sufficient browsing data from real students to pick out new constructs. The MLFF neural network was therefore a better paradigm to choose. However, further research may enable the Kohonen network to identify new constructs using a richer set of input parameters from the hypermedia such as those input parameters described above.

The Kohonen neural network requires that the designer provide the number of categories to separate the presented data into. This is a direct requirement since the number of categories is the number of outputs the neural network has. For example, the designer may decide beforehand that there is a new browsing construct and hence design the Kohonen neural network with five outputs (spike, path, ring, loop and the new construct to be identified). If there are indeed five constructs then the network will separate the input into five categories. Note that it may not be the case that the original categories still exist exactly as before. It may be, for example, that two slightly different types of spike exist. This is dependent upon the “strength” of the new construct. If it is as distinct as the previously defined constructs then it will quickly become encoded within the network, pushing the new spike aside. It may be that even more constructs exist, the designer could then design a new Kohonen network with six inputs and try the experiment again. Note that if real student browsing data has been recorded then it can be reapplied in many experiments. It is a requirement of the designer/researcher to decide whether the categories the Kohonen has produced are valid. For example, a network that produces an output for the new spike, may be rejected since the network will always attempt to activate all of its outputs and it may produce categories that are not necessarily useful. It may be that there is not sufficient difference between the old type of spike and the new to make their separate evaluation useful. If a Kohonen neural network is designed with more outputs than there are categories in the real input data then it is likely to smear the actual number of categories over the number of outputs. This is an important restriction of the Kohonen neural network (Masters 1993), although it can be overcome by extra experimentation (testing with various numbers of outputs).

### 10.2.4 Experiments with the Bottom-Up Approach

The bottom-up approach, as discussed in chapter 9, requires the collection of real student browsing data. However, once this has been achieved it may be possible to use the collected data repeatedly with a number of configurations of the bottom-up approach. This approach allows experimentation without adversely affecting the student, since the data has already been collected. For example, all the link weights could be removed from the system. In this case the system would be essentially a standard hypermedia system. Trials with students would then be used to allow the domain to be authored from the bottom-up. The resulting structure could then be compared with the semantic network for the domain. It is unlikely that there will be a match since the top-down approach involves an expert engineering the domain, whereas the bottom-up approach involves the system being “engineered” by novice to expert students. However, since it is the intention of the system to facilitate the student’s progress from novice to expert, such bottom-up information may be extremely useful, in that it may highlight elements of the domain where there is a fundamental mismatch between the author and students' understanding. Such information may enlighten the domain author resulting in them producing better hypermedia domains in the future. Indeed such information may provide general guidelines for authoring hypermedia domains.

This approach may also provide a useful mechanism for organising a hypermedia system without a semantic network. Jonassen and Wang (1993) note that “an appropriate method for structuring hypertext is to mirror the semantic network of an experienced or knowledgeable performer or expert”. They are referring to the schema stored within the expert’s brain. Thus, the domain expert already has a semantic network-like structure stored within their brain and it would be highly desirable to transfer some of this structure into the computer. The problem is that the expert is not always capable of elucidating this structure (Jonassen and Wang 1993). It may be possible to extract this structure by allowing an expert or author to browse in an unstructured domain. Their browsing habits can then be used to define the structure of the domain.

#### 10.2.4.1 The Bottom-Up Mechanism

The mechanism for providing the bottom-up approach operated by allowing the system to modify link weights according to parameters determined by experimental research. These parameters are:

| | |
|---|---|
| The Momentum Term | (the effect of the current weight change) |
| The Initial Weight Modifier | (initial weight change value, to be reduced over time) |

The Maximum Weight Increase (to stop changing weights after a period)

Weight Decrease Modifier (to decrease weights of links no longer used)

Since this paradigm has been designed for use with any hypermedia system, including systems without a semantic network (or similar structure), an extra construct could be designed to exist with the hypermedia domain. This construct is a web, mirroring the hypermedia, a node for a node and a link for a link together with a control mechanism. The web is a structure that contains the link weights, since an already designed hypermedia system/domain may not have the ability to store the link weights. The control system must be integrated into the hypermedia system, in terms of being executed every time the student moves from one node to another. The student accesses the hypermedia via the control mechanism. The student/user is therefore navigating the web and not the hypermedia itself. This allows the control structure to modify link weights within the web. The control structure therefore controls the interaction between the student and the hypermedia. A similar approach has been used for WWW browsing aids and agents (Micarelli and Sciarrone 1998, Armstrong et al 1994).

The control system requests domain nodes from the underlying system, which are then displayed to the student/user. When the student moves to another node, the control system is again called. It modifies the link weights in the web according to the predefined terms described above.

This method requires some reengineering of the original hypermedia system. It requires that the linking information from the hypermedia is copied or transferred to the web construct and requires that the hypermedia system relinquish control to the control system. This situation is essentially a compromise between using the old system alone and completely redesigning the old hypermedia system. It is beneficial to provide a mechanism that can be utilised and developed using existing systems and domains, since this cuts down on development costs, a view argued for in Woods and Warren (1995).

#### 10.2.4.2 Testing the Approach

Testing the bottom-up approach is complex, since it requires a multitude of student/users or domain experts, in order for paths to emerge. In the future, it is likely that the increasing use of the Internet, most notably the World Wide Web (Callaghan and Hand 1996), will provide a medium where there are enough users of a hypermedia system for this approach to become viable. The separation of the control mechanism and the web structure from the hypermedia system itself renders this paradigm suitable for integration into the World Wide Web, since the control mechanism can be installed upon a web server, from where it can monitor student movements within the web server.

Since this approach is novel it must be tested rigorously in order to determine its actual merits. However, the combination of parameters (momentum, weight modifier, maximum weight and weight decrease modifier) renders the required number of test cases extremely large. A strategy must therefore be defined to reduce the number of test cases. This can be achieved by recording a population of students'/users' interactions with a hypermedia domain, in terms of their movements from one node to another. This information can then be used, off-line, in a series of test cases. The researcher must then determine which emergent paths are useful or otherwise interesting.

The drawback of this method is that it does not take into account cumulative effects of weight increases. If the weights were to be updated when a student/user moved from one node to another, then the next time that the link were used (by the same person or another person) its weight would have increased and it would appear as a stronger link. This would increase the probability of the link being selected again. Hence there is a “snowballing” effect such that each time a link is used the probability of it being used again is increased and the increase in probability each time is increased (the effect is exponential). This effectively removes the evolution element, since there is no distinction between a “fit” path and an “unfit” path. The result is a statistical structure, demonstrating which paths happen to have been used more than others. However, research into paths has indicated that the snowballing effect is only a small factor in the evolution of paths (Casti 1998).

A compromise would be to record student/user movement data in short chunks. After each chunk has been recorded the researcher could apply the various parameters and select the best parameters. The web structure could then be updated with the resulting weights. The process might then be repeated several times.

It should be noted that this problem is greatly reduced for hypermedia systems that do not employ any other structural information. In such cases, the actual value of the link weight is not important. Where a structure is provided there is already a hierarchy of link weights, some of which are stronger than others. For example, when should a link weight be increased above that of a strong structural link, such as between "The Earth and Planet"? There is no easy answer to this question; however, Hypernet overcomes it by specifying such structural links in a separate link box. The relative differences between weights are the only important factor, since there are no other predefined weights based upon the structure. Hence, the weight modifier is unimportant.

## 10.3 Summary

In sum, this research project contributes to knowledge an approach for organising hypermedia into a logical and hierarchical structure, to aid both the author and the student. This has been achieved by the use of a formalised semantic-network-like structure of class nodes and instance nodes, each of which may contain hypermedia information. This domain structure is then augmented with tutorial nodes to provide the student with tasks to accomplish whilst using the hypermedia. Using the semantic hypermedia approach as a basis, a further contribution to knowledge is made by providing a student model to obtain information about individual students without recourse to domain dependent measures. This was achieved by employing a combination of symbolic AI, connectionist (neural network) and fuzzy logic methods.

This research project has explored some of the issues concerning the use of a hypermedia tutoring system that measures only those elements of the student that can be measured for every domain. Specifically, it has explored a structured approach to hypermedia and argued that such an approach benefits both the author and the student in that links can be automatically generated and then further tailored towards the student’s interests, which are in turn derived from the student browsing the formalised structure. Secondly, the research has identified, discussed and specified a mechanism for identifying and extracting information from browsing patterns, which have been previously identified as offering potentially useful information about the student (Canter et al 1985). Finally, the research has concluded that practical trials, using the student model described in this thesis, as part of further research may provide insights into how various students use browsing patterns, which may in turn provide an insight into how to design hypermedia systems to better suit beneficial browsing.

# APPENDIX B

# Virtual Student Program

## B.1 Introduction

This appendix contains development details of the Virtual Student Programs, which were used in the absence of real student browsing data. The body of the thesis covers details of the use of the VSPs, whereas this appendix is concerned with the inner workings of the VSP.

The Virtual Student Program (VSP) is designed to mimic the behaviour of real students in order to train the Browsing Pattern Recogniser (BPR) and Fuzzy Rule Base (FRB). The BPR is a neural network system and the FRB is a fuzzy logic processor. Both systems require extensive interactions with students in order for their capabilities to emerge. The VSP allows these facilities to be tested without the need for interactions with real students (which are difficult to obtain over a short period). They also allow the BPR and FRB to be trained before they interact with real students, thus increasing the learning speed. This is a property of the BPR and FRB, in that they are able to readapt to real students, whether they have initially been trained with the VSP or not.  A second VSP was devised to train the Tutorial Supervisor, this is described with the Tutorial Supervisor in chapter 8.

## B.2 Rationale

A major drawback of novel hypermedia systems, as noted by Jonassen (1991) and Dillon and Gabbard (1998), is the difficulty in obtaining empirical data concerning their usefulness or otherwise. A true hypermedia system is designed to organise a large corpus of knowledge, if such a system is then to be used as part of a tutoring system it requires extensive trials with students to accurately evaluate it. This is true of this system, in that many of the facilities require extensive interaction with students in order for the emergent abilities of the AI facilities to become apparent. The practical upshot of this is that it was not feasible to train the neural networks with real students. However, it is argued that it is not necessary to prove the contention that the AI facilities proposed can model real world students, although this would be desirable. It is argued that it is important to prove that the proposed facilities can adapt to a set of situations of which the general population of students is a member. If the neural networks can be successfully trained to accomplish this, then they can also be trained to successfully identify real students. To this end the neural networks were trained with "Virtual Students", a program that can mimic the behaviour of students using hypermedia systems. The bases of the movements simulated by the Virtual Student Program (VSP) are those low-level browsing constructs identified by Canter et al (1985). Training the neural network to recognise the low-level constructs allows the system to identify higher level browsing patterns that have not been previously identified by researchers. The VSP allows the neural network to be trained to recognise the well-understood low-level browsing constructs identified by Canter et al (1985). Linking these low-level constructs to higher level browsing patterns is accomplished by the Fuzzy Rule Base (FRB). The FRB was also trained using a different Virtual Student Program for the same reasons as those defined above.

The difference between the two paradigms is that the neural network undergoes training to recognise the low-level browsing constructs. Once it has been trained, it is a fixed unit, in that no further training takes place. However, if the neural network has been trained properly then it will be able to adapt to novel situations. For example, a neural network may be trained to recognise handwriting, if trained properly it should be able to recognise various peoples handwriting, provided they use the same letters it was trained to recognise. If a person uses new symbols and letters then the network would have to be retrained in order to recognise them. Therefore, the neural network is trained to recognise low-level browsing constructs, the training process does rely upon the assumption that Canter et al (1985) did identify all the possible low-level constructs. However, chapter 10 details further research which may be useful for identifying browsing constructs that have not been previously identified. The FRB, by contrast, is able to learn whilst the system is in active use. This is because the higher level browsing patterns are more complex in nature and it is therefore difficult to identify all of them. The system must therefore be imbued with the ability to adapt to any misconceptions in the original assumptions made about the linking of low-level browsing constructs to high-level browsing patterns.

## B.3 VSP for training the Browsing Pattern Recogniser

The VSP involves generating a student level for each browsing pattern produced. It is vital that this synthetic mapping between the browsing constructs and ability level generated by the VSP is constant between similar browsing patterns. The problem arises because the browsing patterns as generated by the VSP are random patterns based upon percentage values of browsing constructs as supplied by the system designer. This restriction renders the VSP, modified to train the FBR, a complex program in terms of its design.  The VSP generates a level for the generated browsing pattern, based upon the inputted percentage levels for each construct. This mapping between construct percentage and level is defined by a mathematical function, the actual results being unimportant so long as a constant mapping is produced. Note that it is possible to produce several browsing patterns for the same inputted construct percentages, for example:

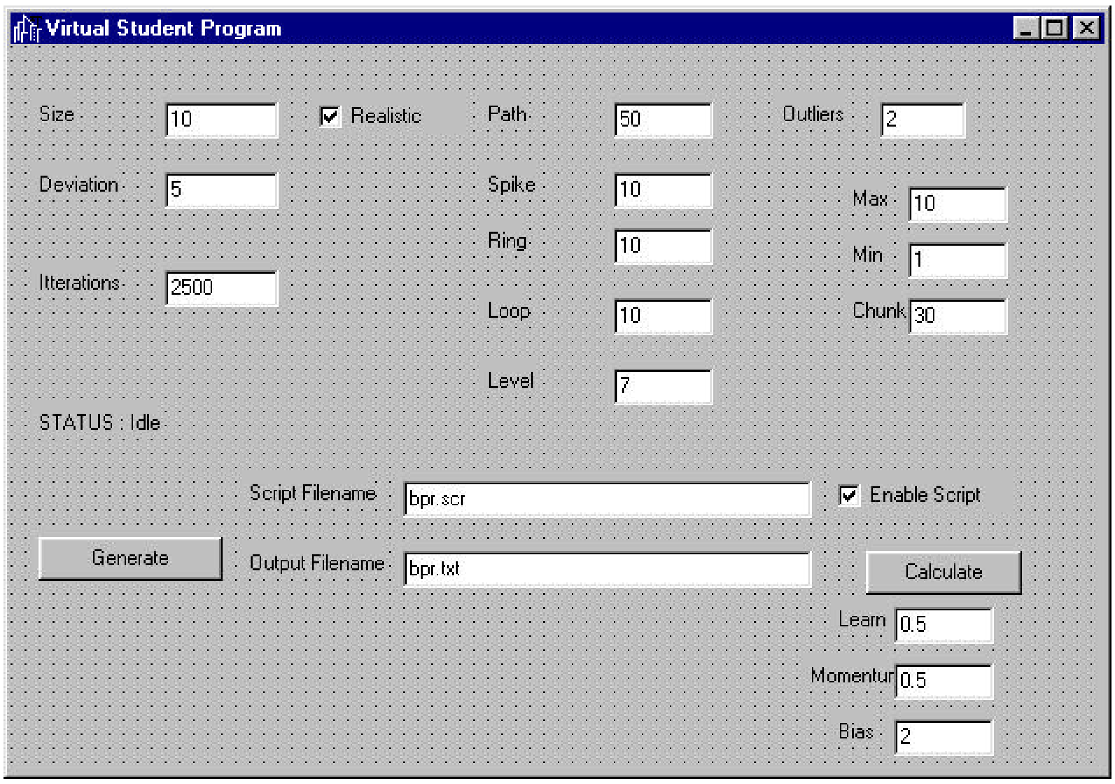
10% spike 50% path 30% ring 10% loop

The BPR neural network is trained to produce the same outputs as the inputted percentage constructs, via the simulated browsing patterns, i.e. the neural network is not directly presented with the percentage values (supplied by the human trainer). The VSP generates a random browsing pattern based on the inputted percentages. This process can generate several browsing patterns based upon the same inputted levels. However, since the neural network is trained to output the construct percentages this is not significant, i.e. the neural network does not model the exact pattern, this is the role of the FBR.

The VSPs generate patterns by simulating the visited flags for an imagined trip through the hypermedia, based upon the desired (by the human trainer) percentage values for each browsing construct. It is important to ensure that the constructs are not broken up to such a degree that they no longer represent their original construct. For example, if the inputted value for the spike construct was set to fifty-percent, then fifty-percent of the resulting browsing pattern will be composed of the spike browsing construct. It may, however, be desirable to split the construct over the pattern as a whole, rather than having the spike occur as one long spike somewhere within the pattern. The breaking-up process may be limited by a preset to a certain value (i.e. a construct can not be less than this value). If this limiting factor were not included then the individual constructs might be fragmented to such a degree that they no longer reflect their intended construct. For example, a ring is essentially a loop finishing (or crossing) its own starting point, this could conceivably be broken down into a path and a one step spike.

It is noted that generating these synthetic browsing patterns, based only upon a random but constant relationship, does not take into account the reason why a particular pattern, or construct is used. The information provided by the VSP is essentially meaningless, whereas browsing patterns used by students have a particular purpose. The VSPs are one-dimensional in that the browsing patterns generated do not relate in any meaningful way to each other. In reality, this is likely not to be the case. It is probable that some browsing patterns group together. For example, there may be several types of "exploring". These patterns may be superficially similar, but be composed of the low-level patterns differently. The VSP does not make this distinction, each pattern it generates does not relate to any other pattern. However, since the application of the VSPs is to prove that the connectionist systems can adapt to a given situation, it can be extrapolated that if they can adapt to a synthetic situation, they can adapt to the real situation. Since the FRB is not an interconnected system, as a neural network is, this distinction is not important. Each browsing pattern has its own rule, which is updated independently. These rules are modified in order to produce a meaningful and differentiable output for browsing patterns that differ only in the actual values of the individual constructs. If, in reality two or more browsing patterns exist that use the same fuzzy rule, but have vastly different beneficial values (one is deemed good, another bad), then the application of one browsing pattern will move the rule one way, whilst the other moves it the other way. In practice this is unlikely to happen, since browsing patterns that occupy the same fuzzy rule are extremely closely related. This is because in order for a browsing pattern to qualify for a rule, it must have individual construct values that are greater than partition level activation (described in chapter 7) in order to qualify for the rule. Since this problem can only occur for complex browsing patterns (ones that involve the use of multiple constructs) the likely highest value of each individual construct must be much lower than one hundred (since all construct values add up to one hundred percent). The range of values that must produce such differing desirability (in terms of the browsing patterns perceived desirability) is small.

### B.3.1 Operation of the Virtual Student Program



Size 10
[✓ Realistic]
Path 50
Outliers 2
Deviation 5
Spike 10
Max 10
Ring 10
Min 1
Itterations 2500
Loop 10
Chunk 30
Level 7
STATUS : Idle
Script Filename bpr.scr
[✓ Enable Script]
Output Filename bpr.txt
Learn 0.5
Momentur 0.5
Bias 2

**Figure B.3.1.A The Virtual Student Program Interface**

The VSPs generate a text file that may be used as a training file for the BPR or FRB. The neural networks were developed using a neural network shell (NeuroShell 2 by Ward Systems), this application allows the use of a comma-delimited file as a training file for the neural network. The file is imported into the shell as a spreadsheet, each field of which may be directed, or not, to an input of the neural network. This allows the name of the construct created by the VSP to be ignored when training the BPR neural network, although it may be used for manual interpretation of the pattern.

The generation of browsing patterns is a simple process and is discussed first. The program takes as inputs following:

Inputs :

The number of browsing constructs in the generated browsing pattern
The percentage values for each of the constructs (must add up to 100%)

Outputs:

A readable text file containing comma delimited browsing data. The file is readable since blank fields are inserted which identify each individual construct, these fields are ignored by the BPR.

The VSP iterates round a loop, until it has produced a pattern of the desired size. Each time around the loop a random number determines, according to the input percentage levels, which browsing construct to generate. Note that the resulting browsing pattern may not have exactly the same ratio of constructs, however, the actual ratio is stored in the output file.

The output file takes the following form:

```
Spike, 1,0,0,  10,10,10,10,10,10,10,10,10,10,0,1,2,3,4,5,6,7,8,9,
Ring,  0,1,0,  10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,0,
Spike, 1,0,0,  10,10,10,10,10,10,10,10,10,10,0,1,2,3,4,5,6,7,8,9,
Path,                                                                   0,0,1,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,1
0,10,
Spike, 1,0,0,  10,10,10,10,10,10,10,10,0,1,2,3,4,5,6,7,
Ring,                                                                   0,1,0,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,0,
Spike, 1,0,0,  10,10,10,10,10,10,10,10,0,1,2,3,4,5,6,7,
Path,                                                                   0,0,1,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
Path,                                                                   0,0,1,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,1
0,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
Spike, 1,0,0,  10,10,10,10,10,10,10,10,10,0,1,2,3,4,5,6,7,8,
```

The first field is for readability only and is not used by the BPR neural network. The following four fields represent the desired output the neural network should produce, for example a one value for the second field indicates that the construct on the line is a spike. When the neural network undergoes training, it is these fields that the back-propagation algorithm uses to alter the network's weights. The final fields represent the construct itself. A value of ten represents a node that has not previously been visited (within the defined timeframe), a value of zero represents a node that has been visited very recently. Intermediate values represent different degrees of "newness".

The VSP generates a random browsing pattern based upon the percentage levels for each construct, the collection of constructs (browsing pattern) is then assigned a level. Note that if two or more browsing patterns are generated with the same vales, they will be slightly different, although they will be assigned the same level. In reality is likely that two different students may adopt the same browsing pattern although they will appear to be slightly different. This is exactly why neural networks and fuzzy logic are a better paradigm to employ than symbolic artificial intelligence, the fuzzy nature of the data. Since the VSP is generating different, but similar browsing patterns for the same data set, it is able to train the FRB in a realistic fashion.

In order to automate the process of training the BPR, the VSP may be called repeatedly with differing values. This enables the VSP to generate a large training set for the BPR. This is useful since the BPR requires many of examples to become fully trained.

**B.3.2 Scripting Language**

The scripting language allows the designer to simulate complete sessions with a hypermedia system. This may be done in a realistic fashion, i.e. the student browses, then switches to search then back to browsing etc. Alternatively, may be done in an ordered fashion, i.e. ten browsing patterns, ten searching patterns etc. It is not, however, necessary to provide the FRB with a realistic simulation of the student using the hypermedia at the top most level. It is sufficient to provide the FRB with realistic individual browsing patterns. This is due to the fact that the FRB learns by updating its rules one at a time. The order these patterns occur is immaterial. Generating realistic simulations may however, be helpful to demonstrate the fully trained rule base.

An example script for the VSP is detailed below:

```
constructs 50,10,10,30
level 7
output 10
constructs 20,30,40,10
level 4
output 10
end
```

A simple interpreter program scans each line in the input file and interprets each command as follows:

constructs n.n.n.n  The individual construct levels are set to the desired values for spike, ring, loop and path constructs.

level n, sets the current simulated student level to the value n. It is vital that the same construct values always produce similar student levels.

Output n, produces n random browsing patterns, concatenated together. Note that these patterns are produced randomly, they will therefore not be identical.

## B.4 Program Listing

The VSP has been described in detail, however for completeness the program code is also included. The program was written using Borland (Inprise) Builder 3, a Visual C++ development environment.

```
//Virtual Student Program
//vsp.cpp
//Duncan Mullier
//last modified 3/96

//---------------------------------------------------------------------------
#include <vcl\vcl.h>
#pragma hdrstop
#include <stdlib.h>
#include <stdio.h>
#include <math.h>


#include "vspcode.h"
//---------------------------------------------------------------------------
```

```
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
        Questions->Text = Itt->Text.ToInt() /  Chunks->Text.ToInt();
    Questions->Repaint();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
int last [100];
int current=40;
int newstudent = 0;
int outlierflag=0;
int percent=0;
FILE *fp;

        //init
    for (int i = 0; i < 100; last[i++]=0);
    last[0] = 45;

    //sections are used to restart a student level, as if another student is now using
the system
    int sectionSize =  Itt->Text.ToInt() /  Chunks->Text.ToInt();
    Questions->Text = sectionSize;
    Questions->Repaint();

        Message->Caption = "Processing";
        Message->Repaint();

    //open text file
    fp = fopen((char *) FileName->Text.c_str(),"wt");
    if (!fp)
               return; //file error

        for  (int  i  =  0;  i  <  Itt->Text.ToInt();  i++)  //loop  for  the  number  of
iterations defined by the user
    {
         //reset for a new student?
         if (!(i % sectionSize))
         {
               newstudent=1;         //set flag so we can write "new student" into the
text file at the end of a data line
            current = random(100);   //completely new question percentage

            //check bounds
               if (current > Max->Text.ToInt())
                       current = current % Max->Text.ToInt();

               if (current < Min->Text.ToInt())
                       current = last[0];//current % Min->Text.ToInt();

            for (int x=0; x < 100; last[x++] = current); //reset

         }
         else
               newstudent = 0;

         if (Realistic->Checked) //link next random value to last random value (slight
change only)
         {

            if (outlierflag)
               current=last[1]; //go back to the old pattern if the last one was an
outlier

               int change = Changed->Text.ToInt();
            change = random(change);
            if (random(2)) //toss a coin
               current = last[0] + change;
            else
               current = last[0] - change;

               //check for outlier
            if (Outlier->Text.ToInt())
```

```
                      percent   =   100   /   Outlier->Text.ToInt();   //calculete   the
probability of an outlier as a percentage
            else
               percent = 0;

            if (!random(percent-1) && percent)
            {
               outlierflag=1;
               current = random(100); //new value
            }
            else
               outlierflag=0;
         }
         else
               current = random(100); //upto and including 99

         //threshold
         if (current > Max->Text.ToInt() && !outlierflag)
               current = current % Max->Text.ToInt();

         if (current < Min->Text.ToInt())
               current = last[0];//current % Min->Text.ToInt();

         fprintf(fp,"%2d ",current); //most recent data item

         last[0] = current;

         //calculate student level (no history data used)

    }
    fclose(fp);
    Message->Caption = "Idle";
       Message->Repaint();
}
//---------------------------------------------------------------------------

void __fastcall TForm1::Button2Click(TObject *Sender)
{
#define LEVELS 10
#define RULEBASESIZE 16*LEVELS
#define INITIALVALUE 50
#define THRESHOLD 10

#define MAXITT 10000 //before giving up on frb
double distTable[LEVELS];
double learn,pwr,max,val,a,b;
int itt=0,flag=0,bpr,rule;
double lastlearn=0; //for mementum
//open data file
FILE *fp=fopen("bpr.txt","rt");
    if (!fp)
        return;
//initialise rule base
double frb[RULEBASESIZE];
    for( int i = 0; i < RULEBASESIZE; i++)
        frb[i]=INITIALVALUE;


//calculate the learning (the hard bit)
//read in the value representing the t-a vector   Wn = W(n-1) + a(t-a)
 itt=0;
 while(itt++ < MAXITT)
 {

    fscanf(fp,"%2d ",&bpr);

    //randomly select a rule to learn
    rule = random(RULEBASESIZE);

    //W * learning rate
    learn = (bpr * Learn->Text.ToDouble());
    //calculate momentum term  lDW(n-1)
    learn += Momentum->Text.ToDouble() * lastlearn;
    lastlearn = learn;
    //calculate the learning for related rules

    pwr = Bias->Text.ToDouble();
    max = LEVELS/2;
    val=0;

    for(int i =0; i<=LEVELS; i++)
    {
        val = i-max;
        a = (pow(max,pwr));
        b = (pow(abs(val),pwr));
        distTable[i] =  (a-b)/a;
    }
    //apply to frb
    for(int i=0; i<=LEVELS; i++)
    {
        frb[rule + (i - LEVELS/2)] = frb[rule + (i - LEVELS/2)] + (distTable[i] *
learn); //i.e. middle entry gets full learning
    }

    //check the rulebase to see if it has fully trained
    flag=0;
    for (int i=0; i <RULEBASESIZE; i++)
        if  ((frb[i]  >  INITIALVALUE  -  THRESHOLD)  &&  (frb[i]  <  INITIALVALUE  +
THRESHOLD)) //rule not learned
        {
            flag=1;
            break;
        }
    if (!flag) //frb fully trained
    {
        fclose(fp);
        Message->Caption = itt;
        //write results to results file
        FILE *fp2 = fopen("results.txt","at");
        FILE *fp3 = fopen("results.csv","at");
        if (fp2 && fp2)
        {
            fprintf(fp2,"Learning Rate = %lf, Momentum = %lf, Bias = %lf, Result =
%d\n",Learn->Text.ToDouble(),Momentum->Text.ToDouble(),Bias->Text.ToDouble(),itt);
            fprintf(fp3,"%lf,%lf,%lf,%d\n",Learn->Text.ToDouble(),Momentum-
>Text.ToDouble(),Bias->Text.ToDouble(),itt);
            fclose(fp2);
            fclose(fp3);
        }
        return;
    }
 }//while
//finish
    fclose(fp);
}
//---------------------------------------------------------------------------
```

# Appendix C

# Prototype Hypermedia System Program

## C.1 Introduction

This appendix contains the program code for the prototype hypermedia system. The system is split into two parts. The first part creates the automatic links and is written using Knowledge Pro for Windows (KPWIN) by Knowledge Garden. The second part is the graphical browser, which is written in C++ using the visual environment of Borland (Inprise) Builder 3. Initially the browser was also written in KPWin, however it was rewritten in C++ to improve performance.

## C.2 Dynamic Links

Below is a selection of code from the solar system knowledge base. Note that a topic in Knowledge Pro is a class node. Knowledge Pro stores knowledge constructs in a similar fashion to LISP, namely as a long list, which may be processed in various ways. The list may be processed up and down to examine the hierarchies and inheritance characteristics of a particular class instance. All of the classes/types defined below are children of the type "node", which has been previously defined (i.e. every class in the system is a child of the all-encompassing "node" class). Within this definition exist functions for connecting nodes onto the hypermedia (connect_links()) and for sorting the links into relevance order (sort_weights()). Each class that is defined as the type "node" therefore inherits these functions which are applied when a new node is added to the system.

The domain is stored as a large, nested linked list, resembling a fractal. The fractal characteristics, namely repeated patterns at greater depths, is due to the fact that each link to a node is represented as the complete linked node, including its links. In practice, this is cumbersome and so instead of storing a copy of each occurrence of a node every time it appears in a link, a pointer to the node is stored instead. Below is a representation of a simple domain using Knowledge Pro.

```
(
(solarsystem,(has-a,
(planet,(has-a,
(satellite,(has-a,
(geographical-feature,
((Mare Tranquilis,geographical-feature,(The Moon,part-of,(....)),
(Mount Etna,geographical-feature,(The Earth,part-of,(....))))))
)
```

The above ungainly structure is a direct representation of how the domain is stored, the parts designated (....) represent another branch of the fractal tree, showing repeated detail. In practice, the domain can be designed simply as a semantic network. The computer then takes care of the representation and even the construction of such a construct is simple for a programming language. The benefit of such a structure is that the tree of nodes can be traversed through the various hierarchies. It is a relatively simple matter for a program to step through each node, following the links to see how one node relates to another. It can also be seen why this is a highly processor-intensive process. Since the structure is highly recursive it is time consuming to traverse once, the linking system must traverse this structure once for every node in the domain. This is the reason why a system such as Strath Tutor (Kibby-Mays, 1991) responds to students slowly, since it processes its dynamic linking system when the student asks for a link.

In Knowledge Pro the nodes are defined as follows:

```
topic picture.
  im_a(node).
end.

topic planet.
  im_a(node).
end. (*planet*)

topic satellite.
  im_a(planet).
partof(planet).
  general='Satellites are bodies which orbit planets. They may be large lumps of rock to
planet sized spheres.'.
end. (*satellite*)

topic planet_part.
  im_a(node).
partof(planet).
end. (*planet_part*)
```

The above defines four node types that will be employed by the semantic network. The class/topic “node” has been previously defined and contains the links for each node as a linked list as well as program code for sorting the links. The function im_a() allows one class/topic to inherit from another, this inheritance continues up the chain of topics. It is therefore unnecessary to include the line im_a(node) for the definition of 'satellite', since it has inherited this property from its parent, Planet. Note that the function im_a() corresponds directly to the more traditionally named is_a described in the semantic network section.

The topic satellite contains some general information that serves two purposes. For the author it gives a reminder of what the node is for and what it should connect to. Secondly the information may be presented to the student, giving them some high level information or contextual information about the node that is connected to this type node. Note that there is no reason why this convention could not be expanded to cover many areas, for example different levels of text which could be presented to different levels of user, or text that could be presented depending on which nodes the student has previously visited, i.e. the context with which the student is using the node.

Below is the definition of a class instance, the first node is the head node, which has been given the type of head_node as a convenient point of entry into the system, i.e. the student starts at this node, from which point they will see links to some of the objects that are found within the solar system. They may then delve deeper into the hierarchy of planets, asteroids and comets.

```
(* K N O W L E D G E   B A S E *)

topic 'The Solar System'(newlinks).
im_a(node).
  :node_type is text.
  :content='Head node.'.
  :direct_links=((100,planet,relative),(100,comet,relative),(100,asteroid,relative)).
  'The Solar System'=union(?node_type,?content,connect_links(?newlinks)).
end. (*the solar sys*)
```

The above defines the entry node to the hypermedia, i.e. a node that represents the whole hypermedia domain, at the highest level. This makes a convenient point for the automatic link generator to start, as well as providing a convenient starting point for the student to start exploring the domain. The link connects it to the planet type node. Note that the link is encoded in the program code as an 'is_a' link, whereas in figure C.7.1.A the link is of type 'ako' (a kind of). This is purely a restriction of the programming language employed. It makes little difference, however, since the mechanism of inheritance is identical in both cases and is purely to differentiate between hierarchies in classes and hierarchies in objects (instances of classes). This link provides access to the rest of the domain, in that everything in this example is either a planet or some child type of planet (satellite). A more complex domain could contain additional links at this point, for example a link to type 'Near Earth Objects'. The node is added to the "Solar System" by the union function which executes the connect_links() function. This function connects the links previously generated by the automatic link generator. The number 100 represents a link weight. Since these are direct links (i.e. connected by the author) they are given the highest possible weight. Generated links are given a lower weight.

Below is a knowledge base containing information about the Earth and the Moon (the Terran system).

```
(* T E R R A N   S Y S T E M*)
topic 'The Earth'(newlinks).
im_a(planet).
  :node_type is external_text.
  :content='earth.txt'.
  :direct_links=((100,'Picture of The Earth',picture),(90,'The Moon',relative)).
  'The Earth'=union(?node_type,?content,connect_links(?newlinks)).
end.

topic 'Picture of The Earth'(newlinks).
im_a(planet).
  :node_type is bmp_picture.
  :content='mars1.bmp'.
  :direct_links=((100,'The Earth',parent)).
  'Picture of The Earth'=union(?node_type,?content,connect_links(?newlinks)).
end. (*pic of earth*)


topic 'The Moon'(newlinks).
im_a(satellite).
  :node_type is external_text.
  :content='moon.txt'.
  :direct_links=((100,'Picture of the Moon',picture),(90,'The Earth',relative)).
  'The Moon'=union(?node_type,?content,connect_links(?newlinks)).
end.

topic 'Picture of The Moon'(newlinks).
im_a(picture).
  :node_type is bmp_picture.
  :content='moon.bmp'.
  :direct_links=((100,'The Moon',parent)).
  'Picture of The Moon'=union(?node_type,?content,connect_links(?newlinks)).
end. (*pic of moon*)
```

Each node is defined (as a topic) and linked into the rest of the domain via the union operator. The names of nodes represent pointers to the node itself (along with all of its links). The above example shows direct links with two weights. It is argued that nodes representing certain media types, such as pictures, are really just an extension of the current node that have been separated from the current node for reasons of clarity. Therefore, these nodes have been given a slightly higher value than other direct links that are not so closely related to the node. In the above example it is clear that a picture of the Earth is more closely related to the Earth than the Moon node, even though they have both been provided by the author. It is important to remember that all generated links must have weights below all the direct links. The burden on the author is not necessarily increased, since there is a simple rule, in that if a node is a media type (i.e. graphic, animation, sound etc.) then they should be thought of as part of the node and given the highest link weight.

### C.2.1 The Link Generator Algorithm

The Link Generator traverses the list of nodes (described above) and determines the relationships, through the semantic network, between nodes that are not directly connected. Each relationship is given a weight according to its strength (relating to the number of intermediate semantic network nodes between the two nodes in question).

The linking algorithm involves two processes, the first connects nodes that are indirectly connected by one intermediate node (e.g. Phobos to Mars to Deimos). The second process connects nodes that are related through the hierarchical structure (e.g. Phobos to The Moon, Deimos, Ganymede, Io, Europa, Callisto, Titan, etc.). Note that this second process does not make a distinction between a link from Phobos to Deimos and a link between Phobos and Titan. The relationship between Phobos and Deimos is stronger than the relationship between Phobos and Titan, by virtue of the fact that Phobos and Deimos belong to the same planet, namely Mars. This is the reason for the first, the direct links, process. The algorithm therefore assigns a higher weight to those calculated by the direct links algorithm. The Hierarchical (structure) links are calculated by examining the class hierarchy of the current node and attaching all nodes that belong to the same hierarchy, the next class in the hierarchy is then examined and those nodes that belong to this class are connected also, albeit with a diminished link weight. This process continues until the complete hierarchy has been processed.

The automatic link generator algorithm traverses each node in the hypermedia, finding and connecting its indirect links and finding and connecting its hierarchical links. This is highly iterative and takes considerable processor power. The algorithm is separated into three parts below, the first part shows the loop into which the two other parts fit.

### C.2.1.1 Generate Links Algorithm Main Loop

```
1 Get the first node in the domain and make it the current node

2 Get the links for the current node

3 Do the Direct Links Process (see section C.8.2.2)

4 Do the Hierarchical Links Process (see section C.8.2.3)

5 Make the current node the next node in the domain (go to the next node)

6 If there are no more nodes left then finish otherwise go to step 2
```

The above process connects to a node at a time, applying the direct links and hierarchical links processes for that node, until all nodes have been processed.

### C.2.1.2 The Direct Links Algorithm

```
1 Get the Node connected to the current node by a direct link

2 Get all nodes linked to this node

3 Make a link from current node to all of these nodes at the direct weight
```

```
  if the node is not the current node and not an excluded node type

4 if no more direct links finish otherwise go to step 1
```

The above algorithm examines all the nodes connected via a direct link to a parent node and connects all other nodes that are also connected via a direct link. For example the planet Jupiter may have nodes connected to it, via direct links, called Io, Europa, Ganymede, Callisto and "Non Galilean Satellites". The above algorithm makes connections between all of these nodes (e.g. Io to Ganymede etc.). All of these links are given an equal weight, the direct links weight, which is the highest weight for a generated link. The algorithm does not reconnect the current node (no Phobos to Phobos link) and ignores links of a certain type. For this project links to multimedia types have been excluded from the linking process, since it is argued that a link to a picture of video of the Moon is only relevant when the student is at the Moon node itself, thus student who is intent upon examining "exciting multi media nodes" (Hartley 1993) only is not aided by the automatic linking system.

The above process would therefore create the following links:

Io - Europa
Io - Ganymede
Io - Callisto
Io - Non Galilean

Europa - Io
Europa - Ganymede
Europa - Callisto
Europa - None Galilean

Ganymede - Io
Ganymede - Europa
Ganymede - Callisto
Ganymede - Non Galilean

Callisto - Io
Callisto - Europa
Callisto - Ganymede
Callisto - Non Galilean

Non Galilean - Io
Non Galilean - Europa
Non Galilean - Ganymede
Non Galilean - Callisto

Therefore the author is relieved from generating these links and the student is provided with links of a strong relationship. This example provides four new links at each of the nodes involved in the process.

### C.2.1.3 The Hierarchical Links Algorithm

This algorithm reads the class structure generated by the system when the system is authored. The class structure, classes, is a linked list containing sub lists for each class in the domain. For example, if the domain contains two classes, Planet and Satellite, for which instances have been defined called planets and satellites respectively, then the classes list is composed as follows:

classes = (Planet,planets,Satellite,satellites)

where planets = (Earth, Mars)
and satellites = (Moon, Phobos, Deimos)

The classes list therefore provides access to those nodes contained within the domain class structure. The algorithm below generates thus list.

```
1 Get the current Node’s class

2 Get the inheritance structure for the current node

3 Get the current node’s class from the classes structure

4 Make a link with the current weight to all node instances contained in the class list

5 Reduce the current weight (to distinguish levels of hierarchy)

6 Change the current class to the next in the inheritance structure

7 If there are no hierarchies left finish, otherwise go to step 3
```

The above algorithm gets the hierarchy structure for the current node. Since all nodes in the domain are connected to a class node (they are instances of a class) and class nodes may inherit from other class nodes, the system can gain the complete hierarchical structure by virtue of the implementation language. Other object orientated languages, such a C++, that provide class/hierarchy structures also provide this facility of inheritance. The algorithm examines each class in the hierarchy structure and links all members of that class to the current node, the further down the hierarchy, the lower the weight of the link. This process is extremely iterative, since the whole domain must be examined for every node in the domain, therefore the addition of a new node increases the processor power required to complete this process exponentially. This is the reason why this process is completed off line, so as not to interfere with the student’s learning process.

### C.2.2 The Sort Links Algorithm

Once the Automatic Link Generator has processed each node then the new links for each node must be combined and added to the domain. The links for each node are represented by a linked list. All the linked lists are connected together to form one large linked list. A process is then applied to sort these links into weight order. This is a simple sort algorithm that compares two adjacent node weights and swaps them if they are not in the correct order. The process is repeated over the whole list again and again until the links are in the correct order, this is termed a bubble sort and is a standard computer science algorithm.

### C.2.3 Link Hierarchy Explanation

The student is provided with a set of links, from which they may select. They are then transported to a new node and a new set of links. The presentation of the links corresponds with the strength of their semantic relationship. It does not matter whether this is via the most relevant being at the top of a list or in separate boxes. It matters only that a distinction can be made by the student between a link that is strongly related to the current node and a link that is related less strongly. This aids the student in making their selection, since they are able to determine which links most closely relate to their current node. However it might not be readily apparent why such a link should be offered. For this reason a facility has been included to explain to the student why the link has been offered to them. This is done by presenting the student, upon request, with a description of how the current node connects, via the semantic network, with the node offered. For example, if a student were presented with a link to Phobos whilst at The Moon node, they may be unclear as to what Phobos actually is and hence unsure as to its suitability to their current task. The system may therefore present them with the information that “The Moon is a Satellite belonging to (has-a link) the Earth and Phobos is a Satellite belonging to Mars”, (since the author connects “The Moon” and “Phobos” to “Satellite”) hence the student is able to make a decision before selecting the node and exacerbating the potential “lost in Hyperspace” problem (discussed in chapter 2).

## C.3 Knowledge Pro Program for Generating Links

```
(* knowledge structure *)
(*#include desscrob.src
*)
picpath='d:\winapps\kpwinpp\bmp'.
txtpath='d:\winapps\kpwinpp\text'.
incfile='d:\winapps\kpwinpp\source\solarsys.kb'.

knowl ( ).
Topic 'knowl'.



(* E V E N T  H A N D L E R *)
(* =================== *)
  topic winevent(info,event,handle).
        do(?event).

         topic list_select_event.
          node_name=get_list_box(?linkwin).
         (* generate_links(?node_name). NOT IF LINKS ARE LOADED *)

          current=?node_name().  (* must run the topic to get at the local info *)
set_display_window(?editwin).
          node_type=element(?current,1).
          content=element(?current,2).
          links=rest(rest(?current)).
text('#ecurrent                    node==',?node_name,'#ncurrent                     node
content=',?content,?node_type,'#ncurrent node links=',?links).
          if ?node_type=external_text then
            change_directory(?txtpath) and
            node_text=read(?content) and
            close(?content).

(* graphics *)

         ( if ?node_type=bmp_picture then
            change_directory(?picpath) and
            display_bmp(?content,?node_name)).


(* set links in link window *)
          link=[].
          count=0.
         ( while ?count<list_length(?links) then
            count=?count+1 and
           text('#n#nlink',element(element(?links,?count),2)) and
           link=union(?link,element(element(?links,?count),2))    ) . (*endwhile*)
text('#nnew links=',?link,?count).
         set_text(?linkwin,?link).

(* display text_node or external_text_node *)
       (   if ?node_type=text or ?node_type=external_text then
            nodewin  =  window(  WindowEvent,  5,  8,  72,  20,  ?node_name,  [  thickframe,
siblings,  child,  visible,  vertscroll,  horzscroll,  maximizebox,  titlebar,  minimizebox,
controlmenu],?win ,black,white ) and
            attach_icon(?nodewin,?txtico)     ).   (*endif*)
            text(?node_text).

        end. (*resize_event*)

        topic resize_event.
       bitmap(?pic,0,0,element(?info,1),element(?info,2)).
        end. (*resize_event*)

  end. (*winevent*)

topic display_bmp(filename,name).
(* external_text_nodes a bmp into memory, makes a window the correct size, then displays
the bmp *)

               pic=load_bitmap(?filename).
               bmp_info=bitmap_info(?pic).
               width=element(?bmp_info,1).
               height=element(?bmp_info,2).
               picwin  =  window(winevent    ,  30,  2,  ?width/2,  ?height/2,  ?name  ,  [
minimizebox,thickframe, siblings,child, visible, , ],?win,white,black,resize_event).
               attach_icon(?picwin,?gfxico).
               bitmap(?pic,0,0,?width/2, ?height/2).
               update_window(?picwin).
end. (*display_bmp*)


(* G E N E R A T E  L I N K S *)

topic generate_links(node).
(*  uses  the  semantic  structure  of  the  knowledge  base  to  produced  additional  weighted
links *)
      set_display_window(?editwin).
text('#eallnodes=',?all_nodes).
update_window(?editwin).
```

```
      current=?node().
      node_type=element(?current,1).
      node_content=element(?current,2).

(* get Direct Local Links i.e. Mars to Phobos, Mars to Deimos *)
          links=rest(rest(?current)).

(* get Indirect Local Links i.e. Phobos to Deimos via Mars *)

 count=0.
  gen_links=[].
       while ?count < list_length(?links) then
          count=?count+1 and
          next_node=element(?links,?count) and
          next_name=element(?next_node,2) and
          next_type=element(?next_node,3) and
          next_links=rest(rest(??next_name)) and

          if ?next_type <> picture then
            (* this node is alreay linked so we must examine the nodes linked to it *)
            count2=0.
            while ?count2 < list_length(?next_links) then
              count2=?count2+1 and
              if      element(element(?next_links,?count2),3)      <>      picture      and
element(element(?next_links,?count2),2) <> ?node
                then next_links=change_weight(?next_links,?count2,2) and
                        gen_links=combine(?gen_links,[element(?next_links,?count2)]).


(* get hierarchical links *)

   node_class=class(?node).
   count=0.
   weight=40. (* strongest hierarchical link *)
   hier_links=[].
   (while ?count<list_length(?node_class)
     then count=?count+1 and
     pos=where(?classes,element(?node_class,?count))+1 and
     hier=element(?classes,?pos) and (* hier now points to the instance of a class ,
stored in the list [class,instance,c,i etc.] *)
     node_links=??hier and  (* node links is the list of all node of the current class *)
     count2=0 and

     (while ?count2 < list_length(?node_links) (* inner while does the hiers for each
link of the current node *)
       then count2=?count2+1 and
       hier=element(?node_links,?count2) and
       (if ?hier=?node then  (* if the indirect link is to the curent node then ignore it
*)
         count2=?count2+1 and
         hier=element(?node_links,?count2)    ) (*endif*)  and
       hier_link=[?weight,?hier,hierarchy] and (* make up all the link attributes *)
     hier_links=union(?hier_links,[?hier_link])     ) and (* add on all members of the
instance/class *)  (* end inner while *)

     weight=?weight/2  ) . (* end outer while while *)

  gen_links=union(?gen_links,?hier_links).

(*text('#eFULL LINKS=',?gen_links).*)





do (?node,?gen_links). (* for some reason the topic will ignore the params unless do is
used, anyway, this add on the new links*)


(*      close_window(?genwin).*)
```

```
end. (*generate links*)

topic change_weight(links,pos,divisor).
(*<links>   is   a   list   of   3   element   lists,   each   sub   list   being   a   link
[<weight>,<node>,<type>]
   <pos> points at the link sub list we are dealing with
   <divisor> is used to reduce the current weight by dividing it with this *)
  link=element(?links,?pos).
  link=replace_elements(?link,1,element(?link,1)/?divisor). (* update weight *)

  links=replace_elements(?links,?pos,[?link]). (* put it back in list of links *)
  change_weight=?links.

end. (*change weight*)


(* F I L E  I N P U T  /  O U T P U T *)
#include \winapps\kpwinpp\source\save.kb

(* M A K E  L I N K  F I L E *)

topic make_link_file().
set_display_window(?genwin).
text('#n#n#n',?all_nodes).
mk_count=0.
(while ?mk_count<list_length(?all_nodes) then
  mk_count=?mk_count+1 and
  set_display_window(?genwin) and
  text('#eG E N E R A T I N G ---  #s',element(?all_nodes,?mk_count),'#n') and
  update_window(?genwin) and
  generate_links(element(?all_nodes,?mk_count))
 ). (* end while *)
text(' #e#nW R I T I N G').
write_links(?all_nodes).

end. (* makelinkfile*)



(* K N O W L E D G E  B A S E *)
#include \winapps\kpwinpp\source\solarsys.kb

topic remember_links.
(* read in previously generated links *)
read_links().  (* get all generated links so it doesnt have to work them out again*)
lend. (*remember_links*)

(* hypertext engine *)
(*generate_links('Mars').*)
text('#n#n#nnodes=',class(?node)).
  current='The Solar System'.
  links=?current().
    text('#nNode Name :#s',?current).

    set_text(?linkwin,?links).

  end. (*knowl*)
```

## C.4 Hypernet Interface Link Control

```
//HYPERNET Link Control
```

```
//this is the most powerful unit
```

```
//text and graphics are subordinate to this
```

```
//this passes filenames to text and graphics
```

```
//allnavigation is handled here
```

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include <dos.h>//for gettime
#include <alloc.h> //for malloc
#include <stdio.h>
#include "linkcontrol.h"

#include "textcontrol.h"
#include "graphicscontrol.h"
#include "popupgfx.h"
#include "common.h"
#include "main.h"

//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TLinks *Links;
TMDIChild *GfxChild;
int childActive=0; //use to say whether a pop up graphic is on the screen
//TForm1 *LinkControl;

#define DIRECT 0
#define STRUCTURE 1
#define RELATED 2
#define POPUP 3
#define TUTORIAL 4
#define NULL 0

#define DECREASE 0.01     //each time a node is visited knock this off
#define VISITVALUE 1.0

typedef struct
{
    char node[255];
    unsigned char hour;
    unsigned char minute;
    unsigned char second;
    int visited; //number of times visited
    double bpr;
    char *next;    //pointer to next
}hist;

hist historyStruct = {"HEAD",0,0,0,0,0,NULL};

 //TStringList *HistoryList; //maintain a string list so it can be quickly searched to
see if a node has been previously visited


char HistoryFilename[255] = HISTORYFILENAME;


char currentNode[255];
char currentClass[255]; //keep a recod of the current instance node's class
char class_visited[255][SERSIZE];
int class_visited_times[SERSIZE];
int sersize=0;
char nodes_visited[255][SERSIZE]; //textual record of nodes
int nodes_visited_times[SERSIZE]; //how many times has it been visited
int totalbrowse=0; //total number of movements
int newbrowse=0; //only increemented for navigations to previously unvisited nodes
int direct=0, structure=0, related=0,pop=0,tutorial=0; //link counts
int no_links; //how many to offer
int backClick=0; //record the number of times back is pressed
time starttime; //record the time when the system was first entered
//---------------------------------------------------------------------------
__fastcall TLinks::TLinks(TComponent* Owner)
    : TForm(Owner)
{
    historyPtr=maxPtr=0;
 processLinkFile(STARTNODENAME);  //start the first node
 Gfx1->processGfx(STARTNODENAME);
    TextNode->processText(STARTNODENAME,75);
    //HistoryList = new(TStringList);

 //initialise SER
 totalbrowse=1;
 for(int i=0;i<SERSIZE;i++)
 {
    nodes_visited_times[i]=0;
    strcpy(nodes_visited[i],"");
 }

 //initialise clock
    gettime(&starttime);

}
//---------------------------------------------------------------------------
void __fastcall TLinks::FormCreate(TObject *Sender)
{
    Top = 325;
    Left = 10;
}
//---------------------------------------------------------------------------

char *TLinks::parseLinks(char *memory, char *filename)
{

    char links[LINES][3][LINESIZE];

    FILE *fp;
    static char node_name[255];
     //fifty links of three 50 characters
    int i =0, j=0, k=0;
    direct=0, structure=0, related=0,pop=0,tutorial=0;

    memset(links,0,LINES*LINESIZE*3); //clear link array
    if (fp=fopen(filename,"rt"))
    {
        //get node name
        char c=0;
        while(c != '\n')
        {
            node_name[i++] = c = fgetc(fp);
        }
        node_name[--i]=0; //finish name
        //get links  ignoring blank lines
        strcpy(currentNode,node_name);
        i=j=k=0;
        int flag=0,serflag=0;
        while(!feof(fp))
        {
           c = fgetc(fp);
           if (c == ',')
           {
            c = fgetc(fp);
            k++;
            i=0;
            flag=0;
           }

           if (c>=' ' && c != '(' && c != ')')
           {
                if (flag || c!= ' ')
                    links[j][k][i++] = c;
                flag=1;
           }
           //if a new line all n its own, ignore it
            if (c == '\n' && flag)
            {
                links[j][k][i++] = 0;
                j++;
                i=flag=k=0;
            }

            if (c == ')') //close link so read next char to see what type it is
            {
                c=fgetc(fp);
                if (c == 'd') //direct link
                {
                    strcpy(directLinks[direct][0],links[j][0]);
                    strcpy(directLinks[direct][1],links[j][1]);
                    strcpy(directLinks[direct++][2],links[j][2]);
                    linktypes[j] = DIRECT;
                }
                else if (c == 's') //structure link
                {
                //the first structure link is the current nodes class
                    if (/*j==0 &&*/ (compareLink(links[j][1],"is-a")==0) && !serflag)
//first structure node?  and current node is an instance
                    {
                        strcpy(currentClass,links[j][0]); //get class and record
                        addSER(currentClass);
                        serflag=1;
                    }
                    strcpy(structureLinks[structure][0],links[j][0]);
                    strcpy(structureLinks[structure][1],links[j][1]);
                    strcpy(structureLinks[structure++][2],links[j][2]);
                    linktypes[j] = STRUCTURE;
                }
                else if (c == 'r') //related material
                {
                    strcpy(relatedLinks[related][0],links[j][0]);
                    strcpy(relatedLinks[related][1],links[j][1]);
                    strcpy(relatedLinks[related++][2],links[j][2]);
                    linktypes[j] = RELATED;
                }
                else if (c == 'g')//popup
                {
                    strcpy(popup[pop][0],links[j][0]);
                    strcpy(popup[pop][1],links[j][1]);
                    strcpy(popup[pop++][2],links[j][2]);
                    linktypes[j] = POPUP;
                }
                else if (c == 't')//tutorial node
                {
                    strcpy(tutorialNode[tutorial][0],links[j][0]);
                    strcpy(tutorialNode[tutorial][1],links[j][1]);
                    strcpy(tutorialNode[tutorial++][2],links[j][2]);
                    //open and display the tutorial node
                    FILE *fp;
                    char filename[255*10];
                    strcpy(filename,PATH);
                    strcat(filename,links[j][2]);
                    strcat(filename,".ltn");
                    if (fp=fopen(filename,"rt"))
                    {
                        strcpy(filename,""); //might as well use this variable
                        int loop=0;
                        char letter;
                        do
                        {
                            filename[loop++] = letter = fgetc(fp);

                        }while(letter != '/'); //stop at comment field
                        filename[--loop]=0;
                        fclose(fp);
                        MainForm->TutorialNodeText->Caption = filename;
                        MainForm->TutorialNodeText->Repaint();

                    }
                    MainForm->Panel1->Enabled = true;
                    MainForm->Panel1->Visible = true;
                    linktypes[j] = TUTORIAL;
                }

            }

        }//endwhile
        links[++j][0][0]=0; //end of links
        fclose(fp);
        ///sortstring(relatedLinks,related-1);
    }
    else
      Links->Caption = "No links available";
    //copy array
    memcpy(memory, links, LINES*LINESIZE*3);
    //parse the link types
    #define IS_A 0
    #define PART_OF 1
    #define AKO 2
    #define HAS_A 3
    return node_name;
}
//---------------------------------------------------------------------------

int parseLink(char link[255])
{
//reads a link string such as (Planet, part-of, Star System)
//seperates it into its three parts
}
int TLinks::processLinkFile(char *nodename)
{
//open the link file and set the link boxes
//take into account the student level
 char *node_name;//[255];
 char linkstr[255],filename[255];
 static int lastLink=0;
 int flag=0;
 int interest_disp=0;
 int student_level = MainForm->Level->Text.ToInt();

 int dispstruct=0,disprelated=0; //counters for how many currently displayed

 //sort out statistics since this is called every time someone moves
 totalbrowse++; //one more node visited;
 if (newnode(nodename,newbrowse,1))
    addnode(nodename,newbrowse++); //add it to the list

 //sort out how many links to offer a particular student
 switch(student_level)
 {
    case 1:
        no_links = LEVEL1;
        break;
    case 2:
        no_links = LEVEL2;
        break;
    case 3:
        no_links = LEVEL3;
    case 4:
        no_links = LEVEL4;
        break;
    case 5:
        no_links = LEVEL5;
        break;
    case 6:
        no_links = LEVEL6;
        break;
    case 7:
        no_links = LEVEL7;
        break;
    case 8:
        no_links = LEVEL8;
        break;
    case 9:
        no_links = LEVEL9;
        break;
    case 10:
        no_links = LEVEL10;
        break;
    default:
        no_links = LEVEL10;
        break;
 }


 //reset link boxes
 if(childActive)
 {
 //   delete GfxChild;

    Gfx2->Height=0;
    Gfx2->Caption="No popups";
    Gfx2->SendToBack();
    Gfx1->BringToFront();
    //Gfx2->Enabled=false;
    childActive=0;
 }
 DirectLinks->Clear();
 StructureLinks->Clear();
 RelatedLinks->Clear();
 RelatedLinks->Sorted=false;
 strcpy(filename,PATH);
 strcat(filename,nodename);
 strcat(filename,".link");

 //links take the form
 //links[link number][link component (head, links, tail)]
     node_name = parseLinks((char*) links, filename);

     int loop=0;
     ///direct=0,structure=0,related=0;
     while (links[loop][0][0])
     {
        strcpy(linkstr,links[loop][0]);
        //first structure link is the current class
     /*   if (loop==0 && linktypes[0] == STRUCTURE)
        {
                addSER(linkstr);
        }
      */
      // int blink=0;
        char temp[255];

        //if the node has been previously visited mark it as so


        if (linktypes[loop] == DIRECT)
        {
            if (!linkExists(linkstr))
             strcpy(linkstr,markLink(linkstr,MISSING));
            if (!newnode(linkstr,newbrowse,0))
                strcpy(linkstr,markLink(linkstr,VISITED));

            DirectLinks->Items->Add(/*Strings[direct++] = */linkstr);

        }
        else if (linktypes[loop] == STRUCTURE  && dispstruct++<no_links)
         {
         //the first structure link is the type of the current node so record it in the
SER

            if (!linkExists(linkstr))
```

```
                    strcpy(linkstr,markLink(linkstr,MISSING));
              if (!newnode(linkstr,newbrowse,0))
                  strcpy(linkstr,markLink(linkstr,VISITED));

              StructureLinks->Items->Add(linkstr);

           }
          else if  (linktypes[loop] == RELATED /*&& disprelated++<no_links*/)
               {

                  char linkClass[255];
                  char interest[255];
                  strcpy(interest,getSERInterest());
                  memset(linkClass,0,255);
                  strcpy(linkClass,links[loop][2]);

          //link annotations
           flag=0;

          //check for broken links
                  if (!linkExists(linkstr))
                  {
                      strcpy(linkstr,markLink(linkstr,MISSING));
                      flag=1;
                  }


                  //if it has previously been visited and it is the same class
                  if (compareLink(linkClass,currentClass)==0 &&
!newnode(linkstr,newbrowse,0) && !flag)
                  {
                      strcpy(linkstr,markLink(linkstr,VISITED));
                      flag=1;
                  }

                  //if the related material link is of a different class to the current
node
                  if(strcmpi(linkClass,currentClass)!=0 && !flag)
                  {
                      if (strcmpi(linkClass,interest)==0 && interest_disp<MAXINTEREST) //is
this new class the current interest
                      {
                          strcpy(linkstr,markLink(linkstr,INTEREST));
                          flag=1;
                          interest_disp++;
                      }
                  }

                  //if it is the same class
                  if (compareLink(linkClass,currentClass)==0 && !flag) //same as current
class
                  {
                          strcpy(linkstr,markLink(linkstr,UNVISITED));
                          flag=1;
                  }
                  //if it has been visited but it is a different class
                  if (compareLink(linkClass,currentClass)!=0 &&
!newnode(linkstr,newbrowse,0) && !flag)
                  {
                      strcpy(linkstr,markLink(linkstr,VISBUTDIF));
                      flag=1;
                  }



                  //otherwise it hasn't been visited and it is a different class
                  if(!flag)
                  {
                      strcpy(linkstr,markLink(linkstr,DIFFERENT));
                      flag=1;
                  }
```

```
                /*

                     else {
                         strcpy(linkstr,markLink(linkstr,DIFFERENT));
                         flag=1;
                     }
                 }
                 else if (!linkExists(linkstr))
                   strcpy(linkstr,markLink(linkstr,MISSING));
                 if (!newnode(linkstr,newbrowse,0))
                 {//it has been visited but is it the current class?
                     if(compareLink(linkClass,currentClass)==0)
                         strcpy(linkstr,markLink(linkstr,VISITED));
                     else
                         strcpy(linkstr,markLink(linkstr,VISBUTDIF));
                 }
                 else if (!flag)
                     strcpy(linkstr,markLink(linkstr,UNVISITED));
                  */
                 RelatedLinks->Items->Add(linkstr);
              }
        else if (linktypes[loop] ==POPUP)
        {
        //pop up graphics

            Gfx2->Enabled=true;
            //Gfx2->Height=300;
            Gfx1->SendToBack();
            Gfx2->BringToFront();
           /* GfxChild = new TMDIChild(Application);
               GfxChild->Caption = linkstr; */
            childActive=1;
            Gfx2->popUpGfx(linkstr);

        }

        if (loop > lastLink)
            lastLink=loop; //keep a record of the last entty in the box so they can be
cleared
        loop++;
     }
         //reduce the number of links according to student level
    RelatedLinks->Sorted=true;
    if(related>no_links) //need to remove some?
    {
        //AnsiString s;
        int count = RelatedLinks->Items->Count;
        for(int i=related-1; i>=no_links-1; i--)
        {

            //for debuggings = RelatedLinks->Items->Strings[i];
            RelatedLinks->Items->Delete(i);
        }
    }

    RelatedLinks->Repaint();
     return 1;
}


void __fastcall TLinks::DirectLinksClick(TObject *Sender)
{
char linkfile[255];

    if (Navigate->Checked==false)
    {
        getLinkInfo(directLinks[DirectLinks->ItemIndex][0]);
        return;
    }
    addToHistory(directLinks[DirectLinks->ItemIndex][0]);
    Links->Caption = directLinks[DirectLinks->ItemIndex][0]; //title of link box
    strcpy(linkfile,directLinks[DirectLinks->ItemIndex][0]);


    Gfx1->processGfx(directLinks[DirectLinks->ItemIndex][0]);
    TextNode->processText(directLinks[DirectLinks->ItemIndex][0],75);
    Links->processLinkFile(linkfile);
}
//---------------------------------------------------------------------------

void __fastcall TLinks::StructureLinksClick(TObject *Sender)
{
char linkfile[255];
     if (Navigate->Checked==false)
    {
        getLinkInfo(structureLinks[StructureLinks->ItemIndex][0]);
        return;
    }
    addToHistory(structureLinks[StructureLinks->ItemIndex][0]);
    Links->Caption = structureLinks[StructureLinks->ItemIndex][0]; //title of link box
    strcpy(linkfile,structureLinks[StructureLinks->ItemIndex][0]);


    Gfx1->processGfx(structureLinks[StructureLinks->ItemIndex][0]);
    TextNode->processText(structureLinks[StructureLinks->ItemIndex][0],75);
    Links->processLinkFile(linkfile);
}
//---------------------------------------------------------------------------

void __fastcall TLinks::RelatedLinksClick(TObject *Sender)
{
char linkfile[255];
AnsiString s;
  s = RelatedLinks->Items->Strings[RelatedLinks->ItemIndex];
 if (Navigate->Checked==false)
    {
        getLinkInfo(s.c_str()/*relatedLinks[RelatedLinks->ItemIndex][0]*/);
        return;
    }

    //Links->Caption = relatedLinks[RelatedLinks->ItemIndex]; //title of link box
    strcpy(linkfile,relatedLinks[RelatedLinks->ItemIndex][0]);
    //s = RelatedLinks->Items->Strings[RelatedLinks->ItemIndex];
    Links->Caption = s.c_str();
    strcpy(linkfile,s.c_str());
    removeAnnotation(linkfile);
    addToHistory(/*relatedLinks[RelatedLinks->ItemIndex][0]*/linkfile);
   /*
    if (linkfile[0] == ANNOTATED)
    {
        int i=0;
        while(linkfile[i])
        {//get rid of the annotation for the linkfilename
            linkfile[i] = linkfile[i+3];
            i++;
        }
    }
     */

    Gfx1->processGfx(linkfile/*relatedLinks[RelatedLinks->ItemIndex][0]*/);
    TextNode->processText(linkfile/*relatedLinks[RelatedLinks->ItemIndex][0]*/,75);
    Links->processLinkFile(linkfile);
}
//---------------------------------------------------------------------------

void __fastcall TLinks::processBackClick(void)
{
    char linkfile[255];

    backClick++;
    strcpy(linkfile,getHistory());  //get the last node in the history
    recordInHistoryFile(linkfile,"BACK");
    Links->Caption = linkfile; //title of link box


    Gfx1->processGfx(linkfile);
    TextNode->processText(linkfile,75);
    Links->processLinkFile(linkfile);
}

void __fastcall TLinks::processForwardClick(void)
{
    char linkfile[255];

    strcpy(linkfile,getHistoryForwards());  //get the last node in the history
    recordInHistoryFile(linkfile,"FORWARD");
    Links->Caption = linkfile; //title of link box


    Gfx1->processGfx(linkfile);
    TextNode->processText(linkfile,75);
    Links->processLinkFile(linkfile);
}

void TLinks::addToHistory(char *link)
{
//enable back button
///HistoryList->Add(link);
    recordInHistoryFile(link,"LINK");
    strcpy(history[historyPtr++],link); //insert history item
    if (historyPtr > MAXHISTORY)
        historyPtr=0;
    if (maxPtr++ > MAXHISTORY)
        maxPtr=MAXHISTORY;

}

void TLinks::recordInHistoryFile(char *link, char *specialMessage)
{
time systi,store;
static time lastti={0,0,0,0};
hist *nextNode, *lastnode;
unsigned char hour, minute, second;
int flag=0, visited = 0;
double bpr=0;
static int running=0;

while(running); //this code might already be executing wait for it to finish
running=1;
    //get system time
    gettime(&systi);
    memcpy(&store,&systi,sizeof(time));
    //knock off time system started and last time
   // subtractTime(&systi,&starttime);
    subtractTime(&systi,&lastti);

    //update history structure

    //search the list for the current node (has it been previously visited?)

    nextNode = &historyStruct; //get node
    flag=0;
    do
    {
        if (!strcmp(nextNode->node,link)) //is this the current node?
        {
        //if so extract the time it was last visited
            hour = nextNode->hour;
            minute = nextNode->minute;
            second = nextNode->second;
            visited = (nextNode->visited++);
            bpr = (nextNode->bpr);
            nextNode->bpr = VISITVALUE;
            flag=1;
            break;
        }

        lastnode = nextNode;
    }while((nextNode = (hist *) nextNode->next) != NULL); //go until end of list

    //process the list in full ow, but this time decrease the bpr value
    nextNode = &historyStruct; //get node
    do
    {
        nextNode->bpr-=DECREASE;
        if (nextNode->bpr < 0)
            nextNode->bpr = 0;
    }while((nextNode = (hist *) nextNode->next) != NULL); //go until end of list


    //if the node wasn't found then a new entry must be made for it
    //otherwsie the old entry is updated
    if (!flag)
    {
        hist *newhistory = (hist *) malloc(sizeof(hist));
        if (newhistory)
        {
            lastnode->next = (char *) newhistory; //point last node in list at new entry
            strcpy((char *)newhistory->node,(char *)link); //fill in name
            //fill in time
            ///gettime(&systi);
            newhistory->hour = systi.ti_hour;
            newhistory->minute = systi.ti_min;
            newhistory->second = systi.ti_sec;
            newhistory->visited = 0;
            newhistory->bpr = VISITVALUE;
            newhistory->next = NULL;

        }

    }

    //log all navigation
    FILE *historyFile;
  ///  strcpy(historyFile, HISTORYFILENAME);
    char filename[255];
    strcpy(filename,PATH);
    strcat(filename,HistoryFilename);
    if (historyFile=fopen(filename,"at"))
    {

    //finish off the last entry by writing out the time spent at the node
    fprintf(historyFile,"\t%2d:%2d:%2d",systi.ti_hour,systi.ti_min,systi.ti_sec);
    //now write out the firts half of the data for this node
    fprintf(historyFile,"\n%20s\t%d\t%lf\t%d\t%8s",link,visited,bpr,flag,specialMessage);
        //write system time followed by node name followed by last visited time

//fprintf(historyFile,"\n%d\t%2d:%2d:%2d\t%20s\t%d\t%lf",flag,lastti.ti_hour,lastti.ti_mi
n,lastti.ti_sec,link,visited,bpr);


///fprintf(historyFile,"\n%d\t%2d:%2d:%2d\t%20s\t%d\t%lf",flag,systi.ti_hour,systi.ti_min
,systi.ti_sec,link,visited,bpr);
        /*if (!flag)
            fprintf(historyFile,"\n
%2d:%2d:%2d\t\t00:00:00\t\t%s",systi.ti_hour,systi.ti_min,systi.ti_sec,link);
        else
            fprintf(historyFile,"\n
%2d:%2d:%2d\t\t%2d:%2d:%2d\t\t%s",systi.ti_hour,systi.ti_min,systi.ti_sec,hour,minute,sec
ond,link);
        */
        //if (strlen(specialMessage))
          //  fprintf(historyFile,"\t%s",specialMessage);
        fclose(historyFile);
    }
    lastti.ti_hour = store.ti_hour;
    lastti.ti_min = store.ti_min;
    lastti.ti_sec = store.ti_sec;
    running=0;
}

char *TLinks::getHistory(void)
{
    //enable forward button

    if (historyPtr < 1)
    {
        historyPtr=1;
    }
    if (backClick<2)
        --historyPtr;
    return history[--historyPtr];
}

char *TLinks::getHistoryForwards(void)
{
       return history[++historyPtr];
}

bool TLinks::linkExists(char *link)
{
    char filename[255];
    FILE *fp;

    strcpy(filename,PATH);
    strcat(filename,link);
    strcat(filename,".link");

    fp = fopen(filename,"rt");
    if (fp)
    {
        fclose(fp);
        return true;    //file exists
    }
    else
        return false; //file does not exist

}

char *TLinks::markLink(char *link, char* mark)
//if link file is missing mark link as missing on link box
{
    static char missing[255];
    strcpy(missing,mark);
    strcat(missing,link);
    return missing; //result is a string with MISSING appended on the front
}

void TLinks::getLinkInfo(char *link)
{
//write into TutorialTextnode
//read which link has been double clicked and the show its link file
    char text[2048],linkfile[255],ch;
    int loop=0,lines=0;
    FILE *fp;


//if link is annotated remove annotation to get at filename
    if (link[0] == ANNOTATED)
    {
        removeAnnotation(link);
    }
        strcpy(linkfile,PATH);
      // strcat(linkfile,directLinks[DirectLinks->ItemIndex][0]);
```

```
        strcat(linkfile,link);

       strcat(linkfile,".link");
       fp = fopen(linkfile,"rt");
       if (fp)
       {
           while(!feof(fp))
           {
            ch = fgetc(fp);
            if (ch==0)
                break;
            if ((ch >=32 && ch<='z') || ch=='\n' || ch=='\t')
            //if ((ch>' ' && ch<'z') || ch=='\n')
             text[loop++] = ch;
             if (ch=='\n')
                lines++;

           }
           text[loop]=0;
           //while(!feof(fp) && loop<2048);
           fclose(fp);
           //calculate size of panel
           MainForm->Panel1->Height = lines * 15;
           MainForm->Panel1->Enabled = true;
           MainForm->TutorialNodeText->Caption = text;
           MainForm->Panel1->Visible = true;
           MainForm->TutorialNodeText->Repaint();

       }
       else
        MainForm->TutorialNodeText->Caption = "No link information available";

}
//---------------------------------------------------------------------------
bool TLinks::newnode(char *node, int browse, bool set)
{
//browse is the number of new nodes the student has visited
//if set is true it will update the visited number if the node has been visited
//returns true if node has not already been visited else false
    for(int i=0; i<browse; i++)
    {
        if(strcmpi(nodes_visited[i],node) == 0) //remember 0 means it is ==
        {
            if(set)
                nodes_visited_times[i++]; //increase count
            return false; //not a new node
        }

    }
    //it is a new node so add it to the list and report new node

    return true;

}

void TLinks::addnode(char *node, int browse)
{
//adds a node to the visited list
    strcpy(nodes_visited[browse],node); //put the name in
    nodes_visited_times[browse]=1; //it has now been visited
}

void TLinks::addSER(char *node)
{
//add a class node into the SER if it is already there increment its counter
 for(int i=0; i<sersize; i++)
    {
        if(strcmpi(class_visited[i],node) == 0) //remember 0 means it is ==
        { //if it is already there increment its counter
            class_visited_times[i]++; //increase count
            return;
        }

    }
    strcpy(class_visited[sersize],node); //add the node in
    class_visited_times[sersize++]=1; //visited once
    //obviously could make this faster by sorting and binary searching
}

char *TLinks::getSERInterest(void)
{
//return the student's current favourite topic
    int biggest = 0;
    static char interest[255];
    strcpy(interest,"none");

    for(int i=0; i<sersize; i++)
    {
        if (class_visited_times[i] > biggest)
        {
            biggest = class_visited_times[i];
            strcpy(interest,class_visited[i]);
        }
    }
    return interest;
}

void TLinks::sortstring(char list[LINES][3][LINESIZE],int items)
{
//straight bubble sort of an array of strings
char temp[255],left[255],right[255];
    for(int i=0;i<items;i++)
    {
        for(int j=0;j<items;j++)
        {
            strcpy(left,list[j][0]);
            strcpy(right,list[j+1][0]);
            if (strcmpi(right,left)<0)
            {
                strcpy(temp,list[j+1][0]);
                strcpy(list[j+1][0],list[j][0]);
                strcpy(list[j][0],temp); //sorted left

                strcpy(temp,list[j+1][1]);
                strcpy(list[j+1][1],list[j][1]);
                strcpy(list[j][1],temp); //sorted  type

                strcpy(temp,list[j+1][1]);
                strcpy(list[j+1][1],list[j][1]);
                strcpy(list[j][1],temp); //sorted  right
            }
        }
    }
}

void __fastcall TLinks::dumpStatistics(void)
{
    char filename[255];
    FILE *fp;

    //calculate stats
    bpr.executeBPR(&spike,&path,&ring,&loop);
    strcpy(filename,PATH);
    strcat(filename,STATSFILE);

    //calculate fuzzy rule
    #define PARTITION .25
    double frbspike = spike/PARTITION;
    double frbring =  loop/PARTITION;
    double frbloop =  ring/PARTITION;
    double frbpath =  path/PARTITION;
    int frblevel = MainForm->Level->Text.ToInt();
    fp = fopen(filename,"wb");
    if (fp)
    {
        fprintf(fp,"Browser Statistics\n==============\n");
        fprintf(fp,"\nTotal nodes visited : %d",totalbrowse);
        fprintf(fp,"\nNew nodes visited : %d",newbrowse);
        fprintf(fp,"\n\nIndices data\n--------\n");
        fprintf(fp,"\ntotal/new = %f ",(float)totalbrowse/newbrowse);
        fprintf(fp,"\nPath = %.2lf\nRing = %.2lf\nLoop = %.2lf\nSpike =
%.2lf\n\n",path,ring,loop,spike);
        fprintf(fp,"\nFuzzy rule =
%d,%d,%d,%d,%d",frblevel,(int)frbspike,(int)frbring,(int)frbloop,(int)frbpath);
        fprintf(fp,"\n\nContent-based\n----------------\n");
        fprintf(fp,"\nClasses visited : %d",sersize);
        fprintf(fp,"\nInstances/classes = %f",(float)sersize/(totalbrowse-sersize));
        fprintf(fp,"\n\nClass List\n-------------");

        for(int i=0; i<sersize; i++)
        {
            fprintf(fp,"\n%s visited %d times",class_visited[i],class_visited_times[i]);
        }
        fclose(fp);

        TextNode->processText(STATSFILE,-1);
    }
}

void TLinks::removeAnnotation(char *linkname)
{
 static char link[255];
 strcpy(link,linkname);
    if (link[0] == ANNOTATED)
    {
        int i=0;
        while(link[i])
        {//get rid of the annotation for the linkfilename
            link[i] = link[i+3];
            i++;
        }
    }
    strcpy(linkname,link);
}

bool TLinks::compareLink(char *link1, char *link2)
{//strcmp takes into account spaces this doesn't
    int i=0,j=0;
    char a,b;
    do
    {
      a=tolower(link1[i]);
      b=tolower(link2[j]);
      if (a==' ')
      {
       i++;
       break;
      }
      if (b==' ')
      {
       j++;
       break;
      }
      if (a !=b)
        return true;
    }
    while(link1[i++] !=0 && link2[j++] !=0);
    return false;
}

void TLinks::subtractTime(time *time1, time *time2)
```

```
{//subtract one time from another and return the result by reference in time1

    //subtract seconds
    if ((time1->ti_sec - time2->ti_sec) < 0) //need to borrow from min?
    {
        time1->ti_min--;
        time1->ti_sec+=60; //add on a minutes worth of seconds
    }
    time1->ti_sec-=time2->ti_sec; //now subtract
    //subtract minutes
    if ((time1->ti_min - time2->ti_min) < 0) //need to borrow from min?
    {
        time1->ti_hour--;
        time1->ti_min+=60; //add on an hours worth of minutes
    }
    time1->ti_min-=time2->ti_min; //now subtract
    //subtract hours
    time1->ti_hour-=time2->ti_hour;
}
```

# Appendix D

# Browsing Experiments

## D.1 Introduction

Contained within this appendix is the data that relates to the browsing experiments described and discussed in chapter 5. The reader is referred to chapter 5 section 6 for a description of the experiment.

## D.2 Overview of Subjects

### D.2.1 Subject 1

Browser Statistics
Spike 0.4
Path 0.5
Ring 0.7
Loop 0.1
Total nodes visited: 119
New nodes visited: 65
Total/new = 1.830769
Classes visited: 8
Instances/classes = 0.072072
Planet visited 28 times
Satellite visited 24 times
Volcano visited 12 times
Geographical Feature visited 5 times
Spacecraft visited 3 times
Canyon visited 3 times
Asteroid visited 7 times
Comet visited 3 times

### D.2.2 Subject 2

Browser Statistics
Spike 0.3
Path 0.6
Ring 0.7
Loop 0.3
Total nodes visited: 77
New nodes visited: 51
Total/new = 1.509804
Classes visited: 8
Instances/classes = 0.115942
Planet visited 10 times
Satellite visited 14 times
Spacecraft visited 2 times
Geographical Feature visited 6 times
Volcano visited 14 times
Canyon visited 2 times
Asteroid visited 6 times
Comet visited 5 times

### D.2.3 Subject 3

Browser Statistics
Spike 0.3
Path 0.7
Ring 0.5
Loop 0.2
Total nodes visited: 126
New nodes visited: 58
Total/new = 2.172414
Classes visited: 7
Instances/classes = 0.058824
Satellite visited 46 times
Volcano visited 15 times
Spacecraft visited 8 times
Geographical Feature visited 3 times
Asteroid visited 9 times
Comet visited 5 times

### D.2.4 Subject 4

Browser Statistics
Spike 0.3
Path 0.6
Ring 0.4
Loop 0.3
Total nodes visited: 103
New nodes visited: 63
Classes visited: 8
Instances/classes = 0.084211
Planet visited 16 times
Satellite visited 24 times
Volcano visited 13 times
Spacecraft visited 1 times
Geographical Feature visited 5 times
Canyon visited 3 times
Comet visited 7 times
Asteroid visited 10 times

## D.2.4 Subject 5

Browser Statistics
Spike 0.5
Path 0.6
Ring 0.4
Loop 0.2
Total nodes visited: 66
New nodes visited: 32
Total/new = 2.062500
Classes visited: 5
Instances/classes = 0.081967
Planet visited 11 times
Satellite visited 23 times
Spacecraft visited 8 times
Volcano visited 11 times
Geographical Feature visited 1 times

## D.2.5 Subject 6

Browser Statistics
Spike 0.8
Path 0.4
Ring 0.6
Loop 0.1
Total nodes visited: 183
New nodes visited: 55
Classes visited: 8
Instances/classes = 0.045714
Planet visited 39 times
Satellite visited 38 times
Volcano visited 21 times
Spacecraft visited 3 times
Canyon visited 1 times
Comet visited 4 times
Asteroid visited 30 times
Geographical Feature visited 11 times

## D.2.7 Subject 7

Browser Statistics
Spike 0.8
Path 0.2
Ring 0.7
Loop 0.1
Total nodes visited: 127
New nodes visited: 36
Total/new = 3.527778
Classes visited: 6
Instances/classes = 0.049587
Planet visited 29 times
Volcano visited 16 times
Geographical Feature visited 8 times
Satellite visited 55 times
Canyon visited 1 times
Spacecraft visited 6 times

## D.2.8 Subject 8

Browser Statistics
Spike 0.3
Path 0.6
Ring 0.5
Loop 0.0
Total nodes visited: 130
New nodes visited: 40
Total/new = 3.250000
Classes visited: 5
Instances/classes = 0.040000
Planet visited 31 times
Satellite visited 42 times
Volcano visited 5 times
Comet visited 3 times
Asteroid visited 20 times

## D.2.9 Subject 9

Browser Statistics
Spike 0.3
Path 0.7
Ring 0.1
Loop 0.0
Total nodes visited: 95
New nodes visited: 44
Total/new = 2.159091
Classes visited: 6
Instances/classes = 0.067416
Planet visited 26 times
Satellite visited 12 times
Spacecraft visited 4 times
Volcano visited 25 times
Geographical Feature visited 7 times
Canyon visited 3 times

## D.2.10 Subject 10

Browser Statistics
Spike 0.2
Path 0.6
Ring 0.4
Loop 0.1
Total nodes visited: 158
New nodes visited: 44
Total/new = 3.590909
Classes visited: 7
Instances/classes = 0.046358
Asteroid visited 14 times
Comet visited 3 times
Planet visited 47 times
Satellite visited 65 times
Volcano visited 15 times
Geographical Feature visited 1 times
Spacecraft visited 4 times

## D.2.11 Subject 11

Browser Statistics
Spike 0.7
Path 0.2
Ring 0.6
Loop 0.1
Total nodes visited: 166
New nodes visited: 45

Comet visited 4 time

**Subject 11 continued**
Total/new = 3.688889
Classes visited: 6
Instances/classes = 0.037500
Planet visited 40 times
Satellite visited 29 times
Volcano visited 10 times
Geographical Feature visited 18 times
Canyon visited 3 times

## D.3 Browsing Data

| ITEM | NODE | C | BPR | V | CLICK | TIME | Comment |
|---|---|---|---|---|---|---|---|
| 1 | | Subject 1 : expert | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | Solar System | 0 | 0 | 0 | LINK | 0:02:12 | Reads the initial instructions |
| 5 | Planet | 0 | 0 | 0 | LINK | 0:00:33 | |
| 6 | Jupiter | 0 | 0 | 0 | LINK | 0:00:38 | |
| 7 | Ganymede | 0 | 0 | 0 | LINK | 0:00:45 | Goes straight to answer 1 |
| 8 | Ganymede | 0 | 1 | 1 | BACK | 0:00:01 | |
| 9 | Jupiter | 0 | 0.98 | 1 | BACK | 0:00:09 | |
| 10 | Planet | 0 | 0.96 | 1 | LINK | 0:00:12 | |
| 11 | Saturn | 0 | 0 | 0 | LINK | 0:00:22 | |
| 12 | Titan | 0 | 0 | 0 | LINK | 0:00:24 | Goes to answer 2 via hierarchy |
| 13 | Ganymede | 1 | 0.95 | 1 | LINK | 0:02:22 | |
| 14 | Isis | 0 | 0 | 0 | LINK | 0:00:59 | |
| 15 | Volcano | 0 | 0 | 0 | LINK | 0:00:43 | |
| 16 | Asteria | 0 | 0 | 0 | LINK | 0:01:18 | |
| 17 | Inverness Corona | 0 | 0 | 0 | LINK | 0:01:06 | |
| 18 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:13 | |
| 19 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:17 | |
| 20 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:23 | |
| 21 | Volcano | 0 | 0.95 | 1 | LINK | 0:00:21 | |
| 22 | Versuvius | 0 | 0 | 0 | LINK | 0:00:08 | |
| 23 | Versuvius | 0 | 1 | 1 | BACK | 0:00:00 | |
| 24 | Volcano | 1 | 0.97 | 1 | BACK | 0:00:14 | |
| 25 | Montes Haemus | 0 | 0 | 0 | LINK | 0:01:12 | |
| 26 | Moon | 0 | 0 | 0 | LINK | 0:00:27 | |
| 27 | Satellite | 0 | 0 | 0 | LINK | 0:00:15 | |
| 28 | Europa | 0 | 0 | 0 | LINK | 0:01:28 | |
| 29 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:24 | |
| 30 | Spacecraft | 0 | 0 | 0 | LINK | 0:00:08 | |
| 31 | Galileo | 0 | 0 | 0 | LINK | 0:05:34 | Spends a lot of time here, node is quite short |
| 32 | Europa | 0 | 0.97 | 1 | LINK | 0:00:01 | |
| 33 | Jupiter | 1 | 0.76 | 1 | LINK | 0:00:06 | |
| 34 | Planet | 1 | 0.76 | 1 | LINK | 0:00:08 | |
| 35 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:03 | |
| 36 | Canyon | 0 | 0 | 0 | LINK | 0:00:09 | |
| 37 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:18 | |
| 38 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:31 | |
| 39 | Noctis | 0 | 0 | 0 | LINK | 0:00:31 | |
| 40 | Canyon | 0 | 0.97 | 1 | LINK | 0:00:02 | |
| 41 | Geographical Feature | 0 | 0.95 | 1 | LINK | 0:00:04 | |
| 42 | Planet | 2 | 0.92 | 1 | LINK | 0:00:09 | |
| 43 | Vulcan | 0 | 0 | 0 | LINK | 0:00:18 | |
| 44 | Vulcan | 0 | 1 | 1 | BACK | 0:00:00 | |
| 45 | Planet | 3 | 0.97 | 1 | BACK | 0:00:43 | |
| 46 | Uranus | 0 | 0 | 0 | LINK | 0:00:40 | |
| 47 | Neptune | 0 | 0 | 0 | LINK | 0:00:41 | |
| 48 | Uranus | 0 | 0.99 | 1 | LINK | 0:00:20 | |
| 49 | Saturn | 0 | 0.63 | 1 | LINK | 0:00:23 | |
| 50 | Jupiter | 2 | 0.83 | 1 | LINK | 0:00:20 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 51 | Planet | 4 | 0.94 | 1 | LINK | 0:00:08 | |
| 52 | Pluto | 0 | 0 | 0 | LINK | 0:00:31 | |
| 53 | Charon | 0 | 0 | 0 | LINK | 0:00:57 | |
| 54 | Pluto | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 55 | Planet | 5 | 0.96 | 1 | LINK | 0:00:03 | |
| 56 | Solar System | 0 | 0.49 | 1 | LINK | 0:00:05 | |
| 57 | Comet | 0 | 0 | 0 | LINK | 0:00:09 | |
| 58 | Solar System | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 59 | Asteroid | 0 | 0 | 0 | LINK | 0:00:05 | |
| 60 | Kuiper | 0 | 0 | 0 | LINK | 0:00:58 | |
| 61 | Oort | 0 | 0 | 0 | LINK | 0:00:47 | |
| 62 | Atlas | 0 | 0 | 0 | LINK | 0:00:06 | |
| 63 | Eros | 0 | 0 | 0 | LINK | 0:00:11 | |
| 64 | Asteroid | 0 | 0.96 | 1 | LINK | 0:00:09 | |
| 65 | Trojan | 0 | 0 | 0 | LINK | 0:00:03 | |
| 66 | Hally's | 0 | 0 | 0 | LINK | 0:00:35 | |
| 67 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:08 | |
| 68 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:08 | |
| 69 | Comet | 0 | 0.89 | 1 | LINK | 0:00:07 | |
| 70 | Solar System | 2 | 0.88 | 1 | LINK | 0:00:04 | |
| 71 | Planet | 6 | 0.84 | 1 | LINK | 0:00:06 | |
| 72 | Mercury | 0 | 0 | 0 | LINK | 1:196:52 | |
| 73 | Planet | 7 | 0.98 | 1 | LINK | 0:00:04 | |
| 74 | Venus | 0 | 0 | 0 | LINK | 0:00:05 | |
| 75 | Asteria | 0 | 0.42 | 1 | LINK | 0:00:03 | |
| 76 | Venus | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 77 | Planet | 8 | 0.96 | 1 | LINK | 0:00:08 | |
| 78 | Satellite | 0 | 0.5 | 1 | LINK | 0:00:08 | |
| 79 | Iapatus | 0 | 0 | 0 | LINK | 0:00:46 | |
| 80 | Saturn | 1 | 0.69 | 1 | LINK | 0:00:07 | |
| 81 | Janus | 0 | 0 | 0 | LINK | 0:01:22 | |
| 82 | Encelydus | 0 | 0 | 0 | LINK | 0:00:44 | |
| 83 | Saturn | 2 | 0.97 | 1 | LINK | 0:00:01 | |
| 84 | Planet | 9 | 0.93 | 1 | LINK | 0:00:05 | |
| 85 | Jupiter | 3 | 0.65 | 1 | LINK | 0:00:03 | |
| 86 | Callisto | 0 | 0 | 0 | LINK | 0:01:26 | |
| 87 | Io | 0 | 0 | 0 | LINK | 0:00:31 | |
| 88 | Calisto | 0 | 0 | 0 | LINK | 0:00:04 | |
| 89 | Calisto | 0 | 1 | 1 | BACK | 0:00:00 | |
| 90 | Io | 0 | 0.98 | 1 | BACK | 0:00:02 | |
| 91 | Satellite | 1 | 0.87 | 1 | LINK | 0:00:08 | |
| 92 | Planet | 10 | 0.92 | 1 | LINK | 0:00:08 | |
| 93 | Neptune | 0 | 0.55 | 1 | LINK | 0:00:03 | |
| 94 | Triton | 0 | 0 | 0 | LINK | 0:01:12 | |
| 95 | Neried | 0 | 0 | 0 | LINK | 0:00:08 | |
| 96 | Neptune | 1 | 0.97 | 1 | LINK | 0:00:04 | |
| 97 | Planet | 11 | 0.95 | 1 | LINK | 0:00:02 | |
| 98 | Uranus | 1 | 0.5 | 1 | LINK | 0:00:03 | |
| 99 | Miranda | 0 | 0 | 0 | LINK | 0:01:09 | |
| 100 | Ariel | 0 | 0 | 0 | LINK | 0:00:19 | |
| 101 | Umbriel | 0 | 0 | 0 | LINK | 0:00:43 | |
| 102 | Titania | 0 | 0 | 0 | LINK | 0:00:39 | |
| 103 | Oberon | 0 | 0 | 0 | LINK | 0:00:16 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 104 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:05 | |
| 105 | Uranus | 2 | 0.93 | 1 | LINK | 0:00:04 | |
| 106 | Planet | 12 | 0.91 | 1 | LINK | 0:00:07 | |
| 107 | Pluto | 1 | 0.47 | 1 | LINK | 0:00:02 | |
| 108 | Planet | 13 | 0.98 | 1 | LINK | 0:00:01 | |
| 109 | Mars | 0 | 0 | 0 | LINK | 0:00:07 | |
| 110 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:09 | |
| 111 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:09 | |
| 112 | Mars | 0 | 0.98 | 1 | LINK | 0:00:03 | |
| 113 | Deimos | 0 | 0 | 0 | LINK | 0:00:26 | |
| 114 | Phobos | 0 | 0 | 0 | LINK | 0:00:22 | |
| 115 | Mars | 1 | 0.97 | 1 | LINK | 0:00:02 | |
| 116 | Planet | 14 | 0.92 | 1 | LINK | 0:00:01 | |
| 117 | Solar System | 3 | 0.53 | 1 | LINK | 0:00:02 | |
| 118 | Star System | 0 | 0 | 0 | LINK | 0:00:07 | |
| 119 | Star | 0 | 0 | 0 | LINK | 0:01:10 | |
| 120 | Star | 0 | 1 | 1 | BACK | 0:00:01 | |
| 121 | Star System | 0 | 0.98 | 1 | BACK | | Finds the way out |
| 122 | | | | | | | |
| 123 | | | | | | | |
| 124 | Subject 2 : expert : complete | | | | | | |
| 125 | | | | | | | |
| 126 | | | | | | | |
| 127 | Solar System | 0 | 0 | 0 | LINK | 0:00:56 | Reads instructions |
| 128 | Planet | 0 | 0 | 0 | LINK | 0:00:27 | Spends a fair amount of time at initial nodes |
| 129 | Jupiter | 0 | 0 | 0 | LINK | 0:00:33 | |
| 130 | Ganymede | 0 | 0 | 0 | LINK | 0:00:47 | Goes straight to answer 1 |
| 131 | Jupiter | 0 | 0.99 | 1 | LINK | 0:00:11 | |
| 132 | Planet | 0 | 0.97 | 1 | LINK | 0:00:03 | |
| 133 | Saturn | 0 | 0 | 0 | LINK | 0:00:04 | |
| 134 | Titan | 0 | 0 | 0 | LINK | 0:00:19 | Goes to answer 2 via hierarchy |
| 135 | Ariel | 0 | 0 | 0 | LINK | 0:00:50 | From now on doesn't use many hierarchical links |
| 136 | Charon | 0 | 0 | 0 | LINK | 0:00:17 | |
| 137 | Callisto | 0 | 0 | 0 | LINK | 0:00:27 | |
| 138 | Europa | 0 | 0 | 0 | LINK | 0:00:55 | |
| 139 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:44 | |
| 140 | Galileo | 0 | 0 | 0 | LINK | 0:00:26 | |
| 141 | Europa | 0 | 0.98 | 1 | LINK | 0:00:15 | |
| 142 | Encelydus | 0 | 0 | 0 | LINK | 0:01:16 | |
| 143 | Saturn | 0 | 0.91 | 1 | LINK | 0:00:13 | |
| 144 | Neptune | 0 | 0 | 0 | LINK | 0:00:12 | |
| 145 | Triton | 0 | 0 | 0 | LINK | 0:00:10 | |
| 146 | Satellite | 0 | 0 | 0 | LINK | 0:00:19 | |
| 147 | Io | 0 | 0 | 0 | LINK | 0:00:19 | |
| 148 | Jupiter | 1 | 0.83 | 1 | LINK | 0:00:13 | |
| 149 | Planet | 1 | 0.83 | 1 | LINK | 0:00:05 | |
| 150 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:03 | A new interest here? |
| 151 | Volcano | 0 | 0 | 0 | LINK | 0:00:05 | |
| 152 | Asteria | 0 | 0 | 0 | LINK | 0:00:23 | |
| 153 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:13 | |
| 154 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:03 | |
| 155 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:08 | |
| 156 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:02 | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 157 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:05 |  |
| 158 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:13 |  |
| 159 | Versuvius | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 160 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:12 |  |
| 161 | Volcano | 0 | 0.91 | 1 | LINK | 0:00:03 |  |
| 162 | Isis | 0 | 0 | 0 | LINK | 0:00:20 |  |
| 163 | Mount Etna | 0 | 0.93 | 1 | LINK | 0:00:04 | Starts revisiting nodes |
| 164 | Asteria | 0 | 0.89 | 1 | LINK | 0:00:06 |  |
| 165 | Volcano | 1 | 0.96 | 1 | LINK | 0:00:06 |  |
| 166 | Montes Haemus | 0 | 0.89 | 1 | LINK | 0:00:56 |  |
| 167 | Geographical Feature | 0 | 0.84 | 1 | LINK | 0:00:02 |  |
| 168 | Volcano | 2 | 0.97 | 1 | LINK | 0:00:06 |  |
| 169 | Eruption | 0 | 0 | 0 | LINK | 0:00:12 |  |
| 170 | Volcano | 3 | 0.98 | 1 | LINK | 0:00:01 |  |
| 171 | Geographical Feature | 1 | 0.96 | 1 | LINK | 0:00:02 |  |
| 172 | Canyon | 0 | 0 | 0 | LINK | 0:00:04 |  |
| 173 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:14 |  |
| 174 | Noctis | 0 | 0 | 0 | LINK | 0:00:12 |  |
| 175 | Geographical Feature | 2 | 0.96 | 1 | LINK | 0:00:04 |  |
| 176 | Planet | 2 | 0.73 | 1 | LINK | 0:00:02 |  |
| 177 | Solar System | 0 | 0.51 | 1 | LINK | 0:00:06 |  |
| 178 | Asteroid | 0 | 0 | 0 | LINK | 0:00:49 | Starts spending more time at each node |
| 179 | Kuiper | 0 | 0 | 0 | LINK | 0:01:46 |  |
| 180 | Oort | 0 | 0 | 0 | LINK | 0:00:31 |  |
| 181 | Atlas | 0 | 0 | 0 | LINK | 0:00:25 |  |
| 182 | Eros | 0 | 0 | 0 | LINK | 0:00:39 |  |
| 183 | Trojan | 0 | 0 | 0 | LINK | 0:00:34 |  |
| 184 | Hally's | 0 | 0 | 0 | LINK | 0:00:23 |  |
| 185 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:33 |  |
| 186 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:50 |  |
| 187 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:43 |  |
| 188 | Shoemaker-Levey | 0 | 0.99 | 1 | LINK | 0:00:02 |  |
| 189 | Comet | 0 | 0 | 0 | LINK | 0:00:01 |  |
| 190 | Solar System | 1 | 0.87 | 1 | LINK | 0:00:01 |  |
| 191 | Planet | 3 | 0.85 | 1 | LINK | 0:00:02 |  |
| 192 | Jupiter | 2 | 0.56 | 1 | LINK | 0:00:20 |  |
| 193 | Uranus | 0 | 0 | 0 | LINK | 0:00:06 |  |
| 194 | Miranda | 0 | 0 | 0 | LINK | 0:00:36 |  |
| 195 | Umbriel | 0 | 0 | 0 | LINK | 0:00:13 |  |
| 196 | Titania | 0 | 0 | 0 | LINK | 0:00:12 |  |
| 197 | Oberon | 0 | 0 | 0 | LINK | 0:00:16 |  |
| 198 | Satellite | 0 | 0.49 | 1 | LINK | 0:00:01 |  |
| 199 | Planet | 4 | 0.92 | 1 | LINK | 0:00:15 |  |
| 200 | Solar System | 2 | 0.9 | 1 | LINK | 0:00:02 |  |
| 201 | Star System | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 202 | Planet | 5 | 0.97 | 1 | LINK |  |  |
| 203 |  |  |  |  |  |  |  |
| 204 | Subject 3 : expert : complete |  |  |  |  |  |  |
| 205 |  |  |  |  |  |  |  |
| 206 |  | 0 | 0 | 0 | BACK | 0:00:00 | Identifies a bug in the system |
| 207 |  | 0 | 1 | 1 | BACK | 0:00:01 | By pressing Back straight away! |
| 208 |  | 1 | 0.99 | 1 | BACK | 0:00:00 |  |
| 209 |  | 2 | 0.99 | 1 | BACK | 0:00:01 |  |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 210 | | 3 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 211 | | 4 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 212 | | 5 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 213 | | 6 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 214 | | 7 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 215 | | 8 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 216 | | 9 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 217 | | 10 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 218 | | 11 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 219 | | 12 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 220 | | 13 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 221 | | 14 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 222 | | 15 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 223 | | 16 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 224 | | 17 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 225 | | 18 | 0.99 | 1 | BACK | 0:00:01 | |
| 226 | | 19 | 0.99 | 1 | BACK | 0:00:00 | |
| 227 | | 20 | 0.99 | 1 | BACK | 0:00:00 | |
| 228 | | 21 | 0.99 | 1 | BACK | 0:00:00 | |
| 229 | | 22 | 0.99 | 1 | BACK | 0:00:01 | |
| 230 | | 23 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 231 | | 24 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 232 | | 25 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 233 | | 26 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 234 | | 27 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 235 | | 28 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 236 | | 29 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 237 | | 30 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 238 | | 31 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 239 | | 32 | 0.99 | 1 | FORWARD | 0:00:02 | |
| 240 | | 33 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 241 | | 34 | 0.99 | 1 | FORWARD | 0:00:01 | |
| 242 | | 35 | 0.99 | 1 | FORWARD | 0:00:00 | |
| 243 | | 36 | 0.99 | 1 | FORWARD | | |
| 244 | Planet | 0 | 0 | 0 | LINK | 0:00:17 | Restarts the system? |
| 245 | Jupiter | 0 | 0 | 0 | LINK | 0:00:22 | |
| 246 | Ganymede | 0 | 0 | 0 | LINK | 0:00:28 | Straight to answer 1 |
| 247 | Satellite | 0 | 0 | 0 | LINK | 0:00:45 | |
| 248 | Titan | 0 | 0 | 0 | LINK | 0:00:40 | Straight to answer 2 |
| 249 | Ariel | 0 | 0 | 0 | LINK | 0:00:25 | Now reads a lot of the nodes |
| 250 | Callisto | 0 | 0 | 0 | LINK | 0:00:21 | |
| 251 | Charon | 0 | 0 | 0 | LINK | 0:00:43 | |
| 252 | Deimos | 0 | 0 | 0 | LINK | 0:00:12 | |
| 253 | Encelydus | 0 | 0 | 0 | LINK | 0:00:30 | |
| 254 | Europa | 0 | 0 | 0 | LINK | 0:01:11 | |
| 255 | Iapatus | 0 | 0 | 0 | LINK | 0:00:31 | |
| 256 | Io | 0 | 0 | 0 | LINK | 0:00:51 | |
| 257 | Janus | 0 | 0 | 0 | LINK | 0:00:34 | |
| 258 | Saturn | 0 | 0 | 0 | LINK | 0:00:25 | |
| 259 | Moon | 0 | 0 | 0 | LINK | 0:00:14 | |
| 260 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:34 | |
| 261 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:15 | |
| 262 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:14 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 263 | Earth | 0 | 0 | 0 | LINK | 0:01:19 | |
| 264 | Deimos | 0 | 0.88 | 1 | LINK | 0:00:14 | |
| 265 | Europa | 0 | 0.89 | 1 | LINK | 0:00:12 | |
| 266 | Janus | 0 | 0.91 | 1 | LINK | 0:00:20 | |
| 267 | Europa | 1 | 0.98 | 1 | LINK | 0:00:04 | |
| 268 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:13 | |
| 269 | Galileo | 0 | 0 | 0 | LINK | 0:00:21 | |
| 270 | Spacecraft | 0 | 0 | 0 | LINK | 0:00:13 | |
| 271 | Voyager | 0 | 0 | 0 | LINK | 0:00:13 | |
| 272 | Uranus | 0 | 0 | 0 | LINK | 0:00:35 | |
| 273 | Miranda | 0 | 0 | 0 | LINK | 0:00:55 | |
| 274 | Umbriel | 0 | 0 | 0 | LINK | 0:01:02 | |
| 275 | Titania | 0 | 0 | 0 | LINK | 0:01:11 | |
| 276 | Oberon | 0 | 0 | 0 | LINK | 0:00:41 | |
| 277 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:16 | |
| 278 | Uranus | 0 | 0.95 | 1 | LINK | 0:00:07 | |
| 279 | Mars | 0 | 0 | 0 | LINK | 0:00:13 | |
| 280 | Phobos | 0 | 0 | 0 | LINK | 0:00:13 | |
| 281 | Ganymede | 0 | 0.65 | 1 | LINK | 0:00:05 | |
| 282 | Isis | 0 | 0 | 0 | LINK | 0:00:18 | Finished with Satellites? |
| 283 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:06 | |
| 284 | Volcano | 0 | 0 | 0 | LINK | 0:00:11 | Medium time spent at nodes |
| 285 | Geographical Feature | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 286 | Canyon | 0 | 0 | 0 | LINK | 0:00:01 | |
| 287 | Geographical Feature | 1 | 0.98 | 1 | LINK | 0:00:08 | |
| 288 | Versuvius | 0 | 0 | 0 | LINK | 0:00:03 | |
| 289 | Eruption | 0 | 0 | 0 | LINK | 0:00:04 | |
| 290 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:04 | |
| 291 | Asteria | 0 | 0 | 0 | LINK | 0:00:05 | |
| 292 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:13 | |
| 293 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:06 | |
| 294 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:25 | |
| 295 | Geographical Feature | 2 | 0.92 | 1 | LINK | 0:00:04 | |
| 296 | Planet | 1 | 0.45 | 0 | LINK | 0:00:02 | |
| 297 | Jupiter | 0 | 0.37 | 1 | LINK | 0:00:00 | Begins revisiting a lot of nodes |
| 298 | Ganymede | 1 | 0.72 | 1 | LINK | 0:00:10 | However, the bpr value is quite low |
| 299 | Satellite | 0 | 0.37 | 1 | LINK | 0:00:03 | indicating that it is a while since |
| 300 | Titan | 0 | 0.37 | 1 | LINK | 0:00:01 | they were last visited |
| 301 | Ariel | 0 | 0.37 | 1 | LINK | 0:00:01 | |
| 302 | Callisto | 0 | 0.37 | 1 | LINK | 0:00:01 | Just navigating though |
| 303 | Charon | 0 | 0.37 | 1 | LINK | 0:00:01 | |
| 304 | Deimos | 1 | 0.49 | 1 | LINK | 0:00:02 | |
| 305 | Encelydus | 0 | 0.37 | 1 | LINK | 0:00:01 | |
| 306 | Europa | 2 | 0.5 | 1 | LINK | 0:00:22 | Rereads? |
| 307 | Iapatus | 0 | 0.37 | 1 | LINK | 0:00:12 | |
| 308 | Io | 0 | 0.37 | 1 | LINK | 0:00:32 | |
| 309 | Janus | 1 | 0.46 | 1 | LINK | 0:00:11 | |
| 310 | Saturn | 0 | 0.37 | 1 | LINK | 0:00:13 | |
| 311 | Montes Haemus | 0 | 0.39 | 1 | LINK | 0:00:01 | Short visits again |
| 312 | Mount Saint Helens | 0 | 0.39 | 1 | LINK | 0:00:01 | Are they lost? |
| 313 | Mount Rainier | 0 | 0.39 | 1 | LINK | 0:00:01 | |
| 314 | Earth | 0 | 0.39 | 1 | LINK | 0:00:02 | |
| 315 | Deimos | 2 | 0.89 | 1 | LINK | 0:00:02 | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 316 | Europa | 3 | 0.9 | 1 | LINK | 0:00:03 |  |
| 317 | Janus | 2 | 0.92 | 1 | LINK | 0:00:01 | Very quick visits |
| 318 | Europa | 4 | 0.98 | 1 | LINK | 0:00:01 | Lost? Frustrated? |
| 319 | Europa Orbiter | 0 | 0.39 | 1 | LINK | 0:00:01 | Perhaps run out of related links |
| 320 | Galileo | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 321 | Spacecraft | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 322 | Voyager | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 323 | Uranus | 1 | 0.44 | 1 | LINK | 0:00:01 |  |
| 324 | Miranda | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 325 | Umbriel | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 326 | Titania | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 327 | Oberon | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 328 | Uranus Minor Satellites | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 329 | Uranus | 2 | 0.94 | 1 | LINK | 0:00:01 |  |
| 330 | Mars | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 331 | Phobos | 0 | 0.39 | 1 | LINK | 0:00:01 |  |
| 332 | Ganymede | 2 | 0.66 | 1 | LINK | 0:00:02 |  |
| 333 | Isis | 0 | 0.39 | 1 | LINK | 0:00:04 |  |
| 334 | Geographical Feature | 3 | 0.5 | 1 | LINK | 0:00:03 |  |
| 335 | Volcano | 0 | 0.39 | 1 | LINK | 0:00:02 |  |
| 336 | Planet | 0 | 0 | 1 | LINK | 0:00:02 |  |
| 337 | Satellite | 1 | 0.62 | 1 | LINK | 0:00:01 |  |
| 338 | Planet | 1 | 0.98 | 1 | LINK | 0:00:01 |  |
| 339 | Solar System | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 340 | Asteroid | 0 | 0 | 0 | LINK | 0:00:10 | Finds a new area |
| 341 | Kuiper | 0 | 0 | 0 | LINK | 0:00:34 | and begins reading |
| 342 | Oort | 0 | 0 | 0 | LINK | 0:00:32 |  |
| 343 | Atlas | 0 | 0 | 0 | LINK | 0:00:38 |  |
| 344 | Eros | 0 | 0 | 0 | LINK | 0:00:42 |  |
| 345 | Trojan | 0 | 0 | 0 | LINK | 0:00:41 |  |
| 346 | Hally's | 0 | 0 | 0 | LINK | 0:00:33 |  |
| 347 | Eros | 0 | 0.98 | 1 | LINK | 0:00:01 |  |
| 348 | Hally's | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 349 | Trojan | 0 | 0.97 | 1 | LINK | 0:00:12 |  |
| 350 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:01:07 |  |
| 351 | Eros | 1 | 0.96 | 1 | LINK | 0:00:02 |  |
| 352 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:33 |  |
| 353 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:24 |  |
| 354 | Comet | 0 | 0 | 0 | LINK | 0:00:13 |  |
| 355 | Solar System | 0 | 0.85 | 1 | LINK | 0:00:04 | Exits the system |
| 356 | Star System | 0 | 0 | 0 | LINK |  |  |
| 357 |  |  |  |  |  |  |  |
| 358 | Subject 4 : Intermediate : complete |  |  |  |  |  |  |
| 359 |  |  |  |  |  |  |  |
| 360 |  |  |  |  |  |  |  |
| 361 |  |  |  |  |  |  |  |
| 362 | Solar System | 0 | 0 | 0 | LINK | 0:00:38 |  |
| 363 | Planet | 0 | 0 | 0 | LINK | 0:00:33 |  |
| 364 | Jupiter | 0 | 0 | 0 | LINK | 0:00:34 |  |
| 365 | Ganymede | 0 | 0 | 0 | LINK | 0:01:24 | Goes straight to answer 1 |
| 366 | Isis | 0 | 0 | 0 | LINK | 0:00:09 |  |
| 367 | Ganymede | 0 | 0.98 | 1 | BACK | 0:00:04 |  |
| 368 | Io | 0 | 0 | 0 | LINK | 0:00:12 |  |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 369 | Europa | 0 | 0 | 0 | LINK | 0:00:38 | |
| 370 | Calisto | 0 | 1 | 1 | BACK | 0:00:31 | |
| 371 | Europa | 0 | 0.98 | 1 | BACK | 0:00:06 | |
| 372 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:16 | |
| 373 | Europa | 1 | 0.98 | 1 | LINK | 0:00:20 | |
| 374 | Jupiter | 0 | 0.89 | 1 | LINK | 0:00:19 | |
| 375 | Planet | 0 | 0.87 | 1 | LINK | 0:00:06 | |
| 376 | Saturn | 0 | 0 | 0 | LINK | 0:00:14 | |
| 377 | Titan | 0 | 0 | 0 | LINK | 0:00:36 | Takes longer to find answer 2 |
| 378 | Janus | 0 | 0 | 0 | LINK | 0:00:32 | Starts off with some fairly tight |
| 379 | Iapatus | 0 | 0 | 0 | LINK | 0:00:32 | rings |
| 380 | Encelydus | 0 | 0 | 0 | LINK | 0:00:51 | |
| 381 | Saturn Minor Satellites | 0 | 0 | 0 | LINK | 0:00:16 | |
| 382 | Satellite | 0 | 0 | 0 | LINK | 0:00:02 | |
| 383 | Planet | 1 | 0.92 | 1 | LINK | 0:00:03 | |
| 384 | Uranus | 0 | 0 | 0 | LINK | 0:00:42 | |
| 385 | Miranda | 0 | 0 | 0 | LINK | 0:00:01 | |
| 386 | Inverness Corona | 0 | 1 | 1 | BACK | 0:00:11 | |
| 387 | Miranda | 0 | 0.98 | 1 | BACK | 0:00:24 | |
| 388 | Ariel | 0 | 0 | 0 | LINK | 0:00:41 | |
| 389 | Umbriel | 0 | 0 | 0 | LINK | 0:01:02 | |
| 390 | Titania | 0 | 0 | 0 | LINK | 0:00:01 | |
| 391 | Oberon | 0 | 0 | 0 | LINK | 0:00:01 | |
| 392 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:10 | |
| 393 | Uranus | 0 | 0.91 | 1 | LINK | 0:00:07 | |
| 394 | Planet | 2 | 0.88 | 1 | LINK | 0:00:02 | |
| 395 | Neptune | 0 | 0 | 0 | LINK | 0:00:52 | Spends an intermittent time at nodes |
| 396 | Triton | 0 | 0 | 0 | LINK | 0:00:44 | |
| 397 | Neried | 0 | 0 | 0 | LINK | 0:00:25 | |
| 398 | Neptune | 0 | 0.98 | 1 | LINK | 0:00:06 | |
| 399 | Planet | 3 | 0.95 | 1 | LINK | 0:00:10 | |
| 400 | Pluto | 0 | 0 | 0 | LINK | 0:00:03 | |
| 401 | Charon | 0 | 0 | 0 | LINK | 0:00:51 | |
| 402 | Satellite | 0 | 0.8 | 1 | LINK | 0:00:03 | |
| 403 | Moon | 0 | 0 | 0 | LINK | 0:00:16 | |
| 404 | Montes Haemus | 0 | 0 | 0 | LINK | 0:01:22 | Interest taken by direct link |
| 405 | Asteria | 0 | 0 | 0 | LINK | 0:00:23 | |
| 406 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:12 | |
| 407 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:11 | |
| 408 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:05 | |
| 409 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:08 | |
| 410 | Versuvius | 0 | 0 | 0 | LINK | 0:00:03 | |
| 411 | Volcano | 0 | 0 | 0 | LINK | 0:00:03 | |
| 412 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:07 | |
| 413 | Volcano | 0 | 0.99 | 1 | LINK | 0:00:08 | Sister area |
| 414 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:02 | |
| 415 | Canyon | 0 | 0 | 0 | LINK | 0:00:02 | |
| 416 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:08 | |
| 417 | Noctis | 0 | 0 | 0 | LINK | 0:00:06 | |
| 418 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:04 | |
| 419 | Canyon | 0 | 0.97 | 1 | LINK | 0:00:01 | |
| 420 | Geographical Feature | 0 | 0.95 | 1 | LINK | 0:00:02 | |
| 421 | Volcano | 1 | 0.92 | 1 | LINK | 0:00:01 | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 422 | Geographical Feature | 1 | 0.98 | 1 | LINK | 0:00:03 |  |
| 423 | Planet | 4 | 0.76 | 1 | LINK | 0:00:01 |  |
| 424 | Mercury | 0 | 0 | 0 | LINK | 0:00:06 |  |
| 425 | Phobos | 0 | 0 | 0 | LINK | 0:04:29 |  |
| 426 | Deimos | 0 | 0 | 0 | LINK | 0:00:06 |  |
| 427 | Mars | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 428 | Planet | 5 | 0.95 | 1 | LINK | 0:00:02 | Starting to revisit |
| 429 | Venus | 0 | 0 | 0 | LINK | 0:00:04 |  |
| 430 | Asteria | 0 | 0.76 | 1 | LINK | 0:00:02 |  |
| 431 | Venus | 0 | 0.99 | 1 | LINK | 0:00:03 |  |
| 432 | Planet | 6 | 0.96 | 1 | LINK | 0:00:01 |  |
| 433 | Earth | 0 | 0 | 0 | LINK | 0:00:23 |  |
| 434 | Planet | 7 | 0.98 | 1 | LINK | 0:00:08 |  |
| 435 | Solar System | 0 | 0.25 | 1 | LINK | 0:00:03 |  |
| 436 | Comet | 0 | 0 | 0 | LINK | 0:00:12 | Finds new area again |
| 437 | Hally's | 0 | 0 | 0 | LINK | 0:00:32 |  |
| 438 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:21 |  |
| 439 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:51 |  |
| 440 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:31 |  |
| 441 | Eros | 0 | 0 | 0 | LINK | 0:00:14 |  |
| 442 | Atlas | 0 | 0 | 0 | LINK | 0:00:11 |  |
| 443 | Kuiper | 0 | 0 | 0 | LINK | 0:02:01 |  |
| 444 | Oort | 0 | 0 | 0 | LINK | 0:01:11 |  |
| 445 | Trojan | 0 | 0 | 0 | LINK | 0:00:31 |  |
| 446 | Hally's | 0 | 0.92 | 1 | LINK | 0:00:52 | Some quick revisits |
| 447 | Trojan | 0 | 0.99 | 1 | LINK | 0:00:00 |  |
| 448 | Hyutaki | 0 | 0.91 | 1 | LINK | 0:00:01 |  |
| 449 | Kuiper | 0 | 0.95 | 1 | LINK | 0:00:01 |  |
| 450 | Oort | 0 | 0.95 | 1 | LINK | 0:00:02 |  |
| 451 | Asteroid | 0 | 0 | 0 | LINK | 0:00:34 |  |
| 452 | Eros | 0 | 0.9 | 1 | LINK | 0:00:02 |  |
| 453 | Hally's | 1 | 0.93 | 1 | LINK | 0:00:01 |  |
| 454 | Comet | 0 | 0.83 | 1 | LINK | 0:00:01 |  |
| 455 | Solar System | 1 | 0.8 | 1 | LINK | 0:00:03 |  |
| 456 | Planet | 8 | 0.78 | 1 | LINK | 0:00:01 |  |
| 457 | Satellite | 1 | 0.45 | 1 | LINK | 0:00:06 |  |
| 458 | Planet | 9 | 0.98 | 1 | LINK | 0:00:01 |  |
| 459 | Solar System | 2 | 0.96 | 1 | LINK | 0:00:02 |  |
| 460 | Star System | 0 | 0 | 0 | LINK |  | Leaves the system |
| 461 |  |  |  |  |  |  |  |
| 462 |  |  |  |  |  |  |  |
| 463 | Subject 5 : Intermediate : Complete |  |  |  |  |  |  |
| 464 |  |  |  |  |  |  |  |
| 465 | Solar System | 0 | 0 | 0 | LINK | 0:00:54 |  |
| 466 | Planet | 0 | 0 | 0 | LINK | 0:00:16 |  |
| 467 | Jupiter | 0 | 0 | 0 | LINK | 0:00:13 |  |
| 468 | Ganymede | 0 | 0 | 0 | LINK | 0:01:02 | Goes straight answer 1 |
| 469 | Satellite | 0 | 0 | 0 | LINK | 0:00:16 |  |
| 470 | Titan | 0 | 0 | 0 | LINK | 0:00:16 | Goes straight to answer 2 |
| 471 | Saturn | 0 | 0 | 0 | LINK | 0:00:32 |  |
| 472 | Janus | 0 | 0 | 0 | LINK | 0:00:13 |  |
| 473 | Iapatus | 0 | 0 | 0 | LINK | 0:00:17 |  |
| 474 | Saturn | 0 | 0.98 | 1 | LINK | 0:00:05 |  |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 475 | Jupiter | 0 | 0.93 | 1 | LINK | 0:00:03 | |
| 476 | Europa | 0 | 0 | 0 | LINK | 0:00:56 | |
| 477 | Calisto | 0 | 0 | 0 | LINK | 0:00:13 | |
| 478 | Europa | 0 | 0.99 | 1 | BACK | 0:00:05 | |
| 479 | Jupiter | 2 | 0.99 | 1 | BACK | 0:00:03 | |
| 480 | Europa | 1 | 0.97 | 1 | LINK | 0:00:01 | |
| 481 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:36 | Interest caught? |
| 482 | Galileo | 0 | 0 | 0 | LINK | 0:00:24 | |
| 483 | Spacecraft | 0 | 0 | 0 | LINK | 0:00:12 | |
| 484 | Voyager | 0 | 1 | 1 | BACK | 0:01:03 | |
| 485 | Spacecraft | 0 | 0.98 | 1 | BACK | 0:01:07 | |
| 486 | Pioneer | 0 | 0 | 0 | LINK | 0:01:03 | |
| 487 | Pioneer | 0 | 1 | 1 | BACK | 0:00:00 | Uses BACK quite a lot |
| 488 | Galileo | 0 | 0.94 | 1 | BACK | 0:01:08 | |
| 489 | Europa Orbiter | 0 | 0.92 | 1 | BACK | 0:00:01 | |
| 490 | Europa | 2 | 0.89 | 1 | BACK | 0:00:04 | Back to satellites |
| 491 | Jupiter | 3 | 0.87 | 1 | BACK | 0:00:05 | |
| 492 | Saturn | 1 | 0.8 | 1 | LINK | 0:00:48 | |
| 493 | Minor Satellite Saturn | 0 | 0 | 0 | LINK | 0:00:02 | |
| 494 | Minor Satellite Saturn | 0 | 1 | 1 | BACK | 0:00:11 | |
| 495 | Saturn | 2 | 0.97 | 1 | BACK | 0:00:03 | |
| 496 | Planet | 0 | 0.69 | 1 | LINK | 0:00:03 | |
| 497 | Uranus | 0 | 0 | 0 | LINK | 0:00:22 | Explores satellites based on planet |
| 498 | Miranda | 0 | 0 | 0 | LINK | 0:00:34 | |
| 499 | Ariel | 0 | 0 | 0 | LINK | 0:00:12 | |
| 500 | Umbriel | 0 | 0 | 0 | LINK | 0:00:51 | |
| 501 | Titania | 0 | 0 | 0 | LINK | 0:00:31 | |
| 502 | Oberon | 0 | 0 | 0 | LINK | 0:00:11 | |
| 503 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:52 | |
| 504 | Uranus Minor Satellites | 0 | 1 | 1 | BACK | 0:00:01 | |
| 505 | Oberon | 0 | 0.98 | 1 | BACK | 0:00:00 | |
| 506 | Titania | 0 | 0.96 | 1 | BACK | 0:00:01 | |
| 507 | Umbriel | 0 | 0.94 | 1 | BACK | 0:00:02 | |
| 508 | Ariel | 0 | 0.92 | 1 | BACK | 0:00:00 | |
| 509 | Miranda | 0 | 0.9 | 1 | BACK | 0:00:03 | |
| 510 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:11 | New interest but gets there via |
| 511 | Asteria | 0 | 0 | 0 | LINK | 0:00:14 | a direct link |
| 512 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:43 | |
| 513 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:21 | |
| 514 | Mount Etna | 0 | 1 | 1 | BACK | 0:00:01 | |
| 515 | Montes Haemus | 0 | 0.98 | 1 | BACK | 0:00:03 | |
| 516 | Asteria | 0 | 0.96 | 1 | BACK | 0:00:13 | |
| 517 | Montes Haemus | 1 | 0.98 | 1 | FORWARD | 0:00:00 | |
| 518 | Mount Etna | 1 | 0.96 | 1 | FORWARD | 0:00:01 | |
| 519 | Oberon | 1 | 0.86 | 1 | FORWARD | 0:00:00 | |
| 520 | Uranus Minor Satellites | 1 | 0.84 | 1 | FORWARD | 0:00:02 | |
| 521 | Oberon | 2 | 0.98 | 1 | BACK | 0:00:01 | |
| 522 | Mount Etna | 2 | 0.96 | 1 | BACK | 0:00:03 | |
| 523 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:10 | |
| 524 | Volcano | 0 | 0 | 0 | LINK | 0:00:07 | |
| 525 | Planet | 1 | 0.71 | 1 | LINK | 0:00:01 | |
| 526 | Solar System | 0 | 0.38 | 1 | LINK | 0:00:02 | |
| 527 | Star System | 0 | 0 | 0 | LINK | | Missed quite a lot of the domain |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 528 | | | | | | | |
| 529 | Subject 6 : Novice : Complete | | | | | | |
| 530 | | | | | | | |
| 531 | | | | | | | |
| 532 | Solar System | 0 | 0 | 0 | LINK | 0:00:17 | |
| 533 | Planet | 0 | 0 | 0 | LINK | 0:00:12 | |
| 534 | Earth | 0 | 0 | 0 | LINK | 0:03:07 | |
| 535 | Moon | 0 | 0 | 0 | LINK | 0:04:15 | |
| 536 | Ariel | 0 | 0 | 0 | LINK | 0:00:23 | |
| 537 | Moon | 0 | 0.99 | 1 | LINK | 0:00:12 | Spikey browsing |
| 538 | Ariel | 0 | 0.99 | 1 | LINK | 0:00:12 | Familiarisation? |
| 539 | Moon | 1 | 0.98 | 1 | LINK | 0:00:13 | |
| 540 | Ariel | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 541 | Callisto | 0 | 0 | 0 | LINK | 0:00:21 | |
| 542 | Charon | 0 | 0 | 0 | LINK | 0:00:12 | |
| 543 | Moon | 2 | 0.96 | 1 | LINK | 0:00:02 | Is Moon the base? |
| 544 | Deimos | 0 | 0 | 0 | LINK | 0:00:22 | |
| 545 | Encelydus | 0 | 0 | 0 | LINK | 0:00:29 | |
| 546 | Deimos | 0 | 0.99 | 1 | LINK | 0:00:51 | |
| 547 | Moon | 3 | 0.96 | 1 | LINK | 0:00:27 | base |
| 548 | Earth | 0 | 0.87 | 1 | LINK | 0:00:04 | |
| 549 | Planet | 0 | 0.85 | 1 | LINK | 0:00:09 | |
| 550 | Mercury | 0 | 0 | 0 | LINK | 0:01:05 | |
| 551 | Moon | 4 | 0.96 | 1 | LINK | 0:00:13 | base |
| 552 | Earth | 1 | 0.96 | 1 | LINK | 0:00:03 | |
| 553 | Planet | 1 | 0.96 | 1 | LINK | 0:00:03 | Spikes or tight rings |
| 554 | Venus | 0 | 0 | 0 | LINK | 0:02:04 | Searching for a satellite via planets? |
| 555 | Asteria | 0 | 0 | 0 | LINK | 0:00:12 | |
| 556 | Venus | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 557 | Earth | 2 | 0.95 | 1 | LINK | 0:00:02 | Base? |
| 558 | Planet | 2 | 0.95 | 1 | LINK | 0:00:03 | |
| 559 | Mars | 0 | 0 | 0 | LINK | 0:05:06 | |
| 560 | Deimos | 1 | 0.86 | 1 | LINK | 0:00:03 | |
| 561 | Mars | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 562 | Phobos | 0 | 0 | 0 | LINK | 0:01:03 | |
| 563 | Mars | 1 | 0.98 | 1 | LINK | 0:00:19 | |
| 564 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:24 | |
| 565 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:12 | |
| 566 | Olympus Mons | 0 | 0.99 | 1 | LINK | 0:00:12 | |
| 567 | Mars | 2 | 0.96 | 1 | LINK | 0:00:04 | |
| 568 | Earth | 3 | 0.89 | 1 | LINK | 0:00:03 | base |
| 569 | Moon | 5 | 0.82 | 1 | LINK | 0:00:02 | |
| 570 | Earth | 4 | 0.98 | 1 | LINK | 0:00:02 | |
| 571 | Planet | 3 | 0.87 | 1 | LINK | 0:00:08 | |
| 572 | Jupiter | 0 | 0 | 0 | LINK | 0:01:12 | |
| 573 | Io | 0 | 0 | 0 | LINK | 0:00:16 | |
| 574 | Jupiter | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 575 | Europa | 0 | 0 | 0 | LINK | 0:00:34 | |
| 576 | Jupiter | 1 | 0.98 | 1 | LINK | 0:00:04 | |
| 577 | Ganymede | 0 | 0 | 0 | LINK | 0:03:22 | Finds answer 1 |
| 578 | Jupiter | 2 | 0.98 | 1 | LINK | 0:00:04 | Continues same strategy |
| 579 | Galileo | 0 | 0 | 0 | LINK | 0:00:12 | |
| 580 | Europa | 0 | 0.96 | 1 | LINK | 0:00:12 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 581 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:04 | |
| 582 | Europa | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 583 | Europa Orbiter | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 584 | Europa | 2 | 0.98 | 1 | LINK | 0:00:07 | |
| 585 | Jupiter | 3 | 0.93 | 1 | LINK | 0:00:03 | |
| 586 | Earth | 5 | 0.84 | 1 | LINK | 0:00:02 | |
| 587 | Planet | 4 | 0.84 | 1 | LINK | 0:00:05 | |
| 588 | Saturn | 0 | 0 | 0 | LINK | 0:02:07 | |
| 589 | Titan | 0 | 0 | 0 | LINK | 0:01:06 | Finds answer 2 |
| 590 | Janus | 0 | 0 | 0 | LINK | 0:01:07 | More or less continues the |
| 591 | Saturn | 0 | 0.98 | 1 | LINK | 0:00:05 | same strategy after the task is |
| 592 | Earth | 6 | 0.94 | 1 | LINK | 0:00:05 | complete |
| 593 | Planet | 5 | 0.94 | 1 | LINK | 0:00:09 | |
| 594 | Uranus | 0 | 0 | 0 | LINK | 0:00:56 | |
| 595 | Miranda | 0 | 0 | 0 | LINK | 0:00:25 | |
| 596 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:28 | Interest not taken by a different class |
| 597 | Miranda | 0 | 0.99 | 1 | LINK | 0:00:11 | |
| 598 | Umbriel | 0 | 0 | 0 | LINK | 0:00:13 | |
| 599 | Titania | 0 | 0 | 0 | LINK | 0:00:32 | |
| 600 | Oberon | 0 | 0 | 0 | LINK | 0:01:01 | |
| 601 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:44 | |
| 602 | Uranus | 0 | 0.93 | 1 | LINK | 0:00:02 | |
| 603 | Planet | 6 | 0.9 | 1 | LINK | 0:00:47 | |
| 604 | Neptune | 0 | 0 | 0 | LINK | 0:03:12 | |
| 605 | Triton | 0 | 0 | 0 | LINK | 0:02:04 | |
| 606 | Neried | 0 | 0 | 0 | LINK | 0:00:09 | |
| 607 | Neptune | 0 | 0.98 | 1 | LINK | 0:00:03 | |
| 608 | Planet | 7 | 0.95 | 1 | LINK | 0:00:07 | |
| 609 | Pluto | 0 | 0 | 0 | LINK | 0:05:02 | |
| 610 | Charon | 0 | 0.33 | 1 | LINK | 0:03:02 | |
| 611 | Pluto | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 612 | Charon | 1 | 0.98 | 1 | LINK | 0:00:01 | That's it for satellites |
| 613 | Pluto | 1 | 0.98 | 1 | LINK | 0:00:04 | |
| 614 | Planet | 8 | 0.94 | 1 | LINK | 0:00:06 | |
| 615 | Earth | 7 | 0.77 | 1 | LINK | 0:00:04 | |
| 616 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:23 | This time subject realises that |
| 617 | Earth | 8 | 0.98 | 1 | LINK | 0:00:03 | there are no more satellites |
| 618 | Versuvius | 0 | 0 | 0 | LINK | 0:01:04 | and interest changes |
| 619 | Earth | 9 | 0.98 | 1 | LINK | 0:00:02 | |
| 620 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:53 | |
| 621 | Earth | 10 | 0.98 | 1 | LINK | 0:00:02 | |
| 622 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:05:02 | Spends a long time (distracted?) |
| 623 | Earth | 11 | 0.98 | 1 | LINK | 0:00:02 | |
| 624 | Mount Rainier | 0 | 0 | 0 | LINK | 0:05:01 | Spends a long time again |
| 625 | Earth | 12 | 0.98 | 1 | LINK | 0:00:04 | |
| 626 | Moon | 6 | 0.43 | 1 | LINK | 0:00:02 | |
| 627 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:23 | |
| 628 | Moon | 7 | 0.98 | 1 | LINK | 0:00:11 | |
| 629 | Earth | 13 | 0.96 | 1 | LINK | 0:00:02 | |
| 630 | Planet | 9 | 0.84 | 1 | LINK | 0:00:15 | |
| 631 | Solar System | 0 | 0.02 | 1 | LINK | 0:00:03 | |
| 632 | Planet | 10 | 0.98 | 1 | LINK | 0:00:02 | |
| 633 | Solar System | 1 | 0.98 | 1 | LINK | 0:00:03 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 634 | Comet | 0 | 0 | 0 | LINK | 0:00:03 | New interest |
| 635 | Hally's | 0 | 0 | 0 | LINK | 0:06:31 | |
| 636 | Comet | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 637 | Solar System | 2 | 0.96 | 1 | LINK | 0:00:11 | |
| 638 | Asteroid | 0 | 0 | 0 | LINK | 0:03:02 | |
| 639 | Eros | 0 | 0 | 0 | LINK | 0:02:02 | |
| 640 | Asteroid | 0 | 0.99 | 1 | LINK | 0:00:32 | |
| 641 | Trojan | 0 | 0 | 0 | LINK | 0:00:02 | |
| 642 | Asteroid | 1 | 0.98 | 1 | LINK | 0:00:01 | |
| 643 | Atlas | 0 | 0 | 0 | LINK | 0:00:02 | |
| 644 | Asteroid | 2 | 0.98 | 1 | LINK | 0:00:01 | |
| 645 | Kuiper | 0 | 0 | 0 | LINK | 0:00:02 | |
| 646 | Oort | 0 | 0 | 0 | LINK | 0:00:02 | |
| 647 | Trojan | 0 | 0.95 | 1 | LINK | 0:00:00 | |
| 648 | Oort | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 649 | Trojan | 1 | 0.98 | 1 | LINK | 0:00:01 | |
| 650 | Atlas | 0 | 0.94 | 1 | LINK | 0:00:01 | |
| 651 | Kuiper | 0 | 0.95 | 1 | LINK | 0:00:01 | |
| 652 | Asteroid | 3 | 0.92 | 1 | LINK | 0:00:07 | |
| 653 | Atlas | 1 | 0.97 | 1 | LINK | 0:00:02 | |
| 654 | Kuiper | 1 | 0.97 | 1 | LINK | 0:00:01 | Very spikey, lost? |
| 655 | Eros | 0 | 0.85 | 1 | LINK | 0:00:00 | Running out of related links? |
| 656 | Trojan | 2 | 0.93 | 1 | LINK | 0:00:01 | |
| 657 | Kuiper | 2 | 0.97 | 1 | LINK | 0:00:01 | |
| 658 | Eros | 1 | 0.97 | 1 | LINK | 0:00:01 | |
| 659 | Trojan | 3 | 0.97 | 1 | LINK | 0:00:00 | |
| 660 | Kuiper | 3 | 0.97 | 1 | LINK | 0:00:01 | |
| 661 | Asteroid | 4 | 0.91 | 1 | LINK | 0:00:01 | |
| 662 | Atlas | 2 | 0.91 | 1 | LINK | 0:00:08 | |
| 663 | Oort | 1 | 0.85 | 1 | LINK | 0:00:02 | |
| 664 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:02 | |
| 665 | Oort | 2 | 0.98 | 1 | LINK | 0:00:01 | |
| 666 | Hyutaki | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 667 | Comet | 1 | 0.69 | 1 | LINK | 0:00:02 | |
| 668 | Solar System | 3 | 0.69 | 1 | LINK | 0:00:07 | |
| 669 | Planet | 11 | 0.63 | 1 | LINK | 0:00:04 | |
| 670 | Solar System | 4 | 0.98 | 1 | LINK | 0:00:03 | |
| 671 | Comet | 2 | 0.96 | 1 | LINK | 0:00:01 | |
| 672 | Solar System | 5 | 0.98 | 1 | LINK | 0:00:01 | |
| 673 | Planet | 12 | 0.96 | 1 | LINK | 0:01:10 | |
| 674 | Earth | 14 | 0.55 | 1 | LINK | 0:00:02 | |
| 675 | Versuvius | 0 | 0.44 | 1 | LINK | 0:00:07 | Back to Volcanoes |
| 676 | Volcano | 0 | 0 | 0 | LINK | 0:00:04 | But this is the first time the class |
| 677 | Asteria | 0 | 0 | 1 | LINK | 0:00:03 | is visited |
| 678 | Volcano | 0 | 0.99 | 1 | LINK | 0:00:04 | |
| 679 | Versuvius | 1 | 0.96 | 1 | LINK | 0:00:02 | |
| 680 | Volcano | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 681 | Mount Etna | 0 | 0.4 | 1 | LINK | 0:00:01 | |
| 682 | Volcano | 2 | 0.98 | 1 | LINK | 0:00:02 | |
| 683 | Mount Saint Helens | 0 | 0.4 | 1 | LINK | 0:00:01 | Strategy looks different |
| 684 | Volcano | 3 | 0.98 | 1 | LINK | 0:00:02 | More circling around the class |
| 685 | Mount Rainier | 0 | 0.4 | 1 | LINK | 0:00:01 | very short time at node |
| 686 | Volcano | 4 | 0.98 | 1 | LINK | 0:00:01 | Lost? Bored? Searching for exit? |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 687 | Montes Haemus | 0 | 0.41 | 1 | LINK | 0:00:03 | |
| 688 | Volcano | 5 | 0.98 | 1 | LINK | 0:00:02 | |
| 689 | Montes Haemus | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 690 | Volcano | 6 | 0.98 | 1 | LINK | 0:00:01 | |
| 691 | Olympus Mons | 1 | 0 | 1 | LINK | 0:00:02 | |
| 692 | Volcano | 7 | 0.98 | 1 | LINK | 0:00:03 | |
| 693 | Tharsis Volcano Group | 0 | 0 | 1 | LINK | 0:00:03 | |
| 694 | Volcano | 8 | 0.98 | 1 | LINK | 0:00:02 | |
| 695 | Isis | 0 | 0 | 0 | LINK | 0:00:02 | |
| 696 | Volcano | 9 | 0.98 | 1 | LINK | 0:00:05 | |
| 697 | Planet | 13 | 0.76 | 1 | LINK | 0:00:02 | |
| 698 | Earth | 15 | 0.76 | 1 | LINK | 0:00:02 | Familiar place |
| 699 | Moon | 8 | 0.29 | 1 | LINK | 0:00:03 | |
| 700 | Earth | 16 | 0.98 | 1 | LINK | 0:00:03 | |
| 701 | Planet | 14 | 0.96 | 1 | LINK | 0:00:53 | |
| 702 | Solar System | 6 | 0.7 | 1 | LINK | 0:00:09 | |
| 703 | Planet | 15 | 0.98 | 1 | LINK | 0:00:01 | |
| 704 | Solar System | 7 | 0.98 | 1 | LINK | 0:00:04 | |
| 705 | Comet | 3 | 0.66 | 1 | LINK | 0:00:01 | |
| 706 | Solar System | 8 | 0.98 | 1 | LINK | 0:00:03 | |
| 707 | Asteroid | 5 | 0.54 | 1 | LINK | 0:00:02 | |
| 708 | Asteroid | 6 | 0.99 | 1 | LINK | 0:00:03 | |
| 709 | Kuiper | 4 | 0.51 | 1 | LINK | 0:00:02 | |
| 710 | Hyutaki | 1 | 0.56 | 1 | LINK | 0:00:01 | |
| 711 | Comet | 4 | 0.94 | 1 | LINK | 0:00:02 | |
| 712 | Solar System | 9 | 0.94 | 1 | LINK | 0:00:02 | Finds Exit |
| 713 | Star System | 0 | 0 | 0 | LINK | | |
| 714 | | | | | | | |
| 715 | | | | | | | |
| 716 | Subject 7 : Novice : Complete | | | | | | |
| 717 | | | | | | | |
| 718 | Solar System | 0 | 0 | 0 | LINK | 0:04:04 | |
| 719 | Planet | 0 | 0 | 0 | LINK | 0:01:01 | |
| 720 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:22 | |
| 721 | Planet | 0 | 0.99 | 1 | LINK | 0:00:12 | |
| 722 | Venus | 0 | 0 | 0 | LINK | 0:00:24 | |
| 723 | Asteria | 0 | 0 | 0 | LINK | 0:01:27 | |
| 724 | Venus | 0 | 0.99 | 1 | LINK | 0:00:04 | |
| 725 | Asteria | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 726 | Volcano | 0 | 0 | 0 | LINK | 0:00:02 | |
| 727 | Asteria | 1 | 0.98 | 1 | LINK | 0:00:27 | |
| 728 | Volcano | 0 | 0.99 | 1 | LINK | 0:00:06 | |
| 729 | Asteria | 2 | 0.98 | 1 | LINK | 0:00:03 | |
| 730 | Volcano | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 731 | Versuvius | 0 | 0 | 0 | LINK | 0:00:02 | |
| 732 | Earth | 0 | 0 | 0 | LINK | 0:01:15 | Familiar? |
| 733 | Moon | 0 | 0 | 0 | LINK | 0:03:06 | |
| 734 | Earth | 0 | 0.99 | 1 | LINK | 0:02:01 | |
| 735 | Moon | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 736 | Earth | 1 | 0.98 | 1 | LINK | 0:00:15 | |
| 737 | Moon | 1 | 0.98 | 1 | LINK | 0:00:07 | |
| 738 | Montes Haemus | 0 | 0 | 0 | LINK | 0:01:02 | |
| 739 | Moon | 2 | 0.98 | 1 | LINK | 0:00:02 | Base node |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 740 | Earth | 2 | 0.96 | 1 | LINK | 0:00:09 | |
| 741 | Moon | 3 | 0.98 | 1 | LINK | 0:00:09 | |
| 742 | Montes Haemus | 0 | 0.97 | 1 | LINK | 0:00:03 | |
| 743 | Volcano | 2 | 0.87 | 1 | LINK | 0:00:10 | Wandered into a new area |
| 744 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:55 | |
| 745 | Earth | 3 | 0.95 | 1 | LINK | 0:00:05 | |
| 746 | Versuvius | 0 | 0.86 | 1 | LINK | 0:00:04 | |
| 747 | Earth | 4 | 0.98 | 1 | LINK | 0:00:05 | |
| 748 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:22 | |
| 749 | Earth | 5 | 0.98 | 1 | LINK | 0:00:04 | |
| 750 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:47 | |
| 751 | Earth | 6 | 0.98 | 1 | LINK | 0:00:06 | |
| 752 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:45 | |
| 753 | Earth | 7 | 0.98 | 1 | LINK | 0:00:04 | Familiar node and back on track |
| 754 | Moon | 4 | 0.87 | 1 | LINK | 0:01:38 | Did this at the start |
| 755 | Ariel | 0 | 0 | 0 | LINK | 0:02:02 | |
| 756 | Moon | 5 | 0.98 | 1 | LINK | 0:00:01 | Base node |
| 757 | Ariel | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 758 | Moon | 6 | 0.98 | 1 | LINK | 0:00:00 | |
| 759 | Ariel | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 760 | Callisto | 0 | 0 | 0 | LINK | 0:01:01 | |
| 761 | Moon | 7 | 0.97 | 1 | LINK | 0:00:02 | |
| 762 | Charon | 0 | 0 | 0 | LINK | 0:00:53 | |
| 763 | Moon | 8 | 0.98 | 1 | LINK | 0:00:01 | |
| 764 | Deimos | 0 | 0 | 0 | LINK | 0:00:11 | |
| 765 | Encelydus | 0 | 0 | 0 | LINK | 0:00:03 | |
| 766 | Deimos | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 767 | Moon | 9 | 0.96 | 1 | LINK | 0:00:06 | |
| 768 | Earth | 8 | 0.85 | 1 | LINK | 0:00:08 | |
| 769 | Venus | 1 | 0.55 | 1 | LINK | 0:00:04 | |
| 770 | Jupiter | 0 | 0 | 0 | LINK | 0:03:03 | |
| 771 | Io | 0 | 0 | 0 | LINK | 0:02:23 | |
| 772 | Jupiter | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 773 | Earth | 9 | 0.95 | 1 | LINK | 0:00:02 | |
| 774 | Jupiter | 1 | 0.98 | 1 | LINK | 0:00:03 | |
| 775 | Earth | 10 | 0.98 | 1 | LINK | 0:00:00 | |
| 776 | Jupiter | 2 | 0.98 | 1 | LINK | 0:00:13 | |
| 777 | Europa | 0 | 0 | 0 | LINK | 0:00:59 | |
| 778 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:22 | |
| 779 | Galileo | 0 | 0 | 0 | LINK | 0:00:41 | |
| 780 | Europa Orbiter | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 781 | Galileo | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 782 | Europa Orbiter | 1 | 0.98 | 1 | LINK | 0:00:01 | |
| 783 | Galileo | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 784 | Europa | 0 | 0.94 | 1 | LINK | 0:00:01 | |
| 785 | Jupiter | 3 | 0.91 | 1 | LINK | 0:05:03 | |
| 786 | Io | 0 | 0.86 | 1 | LINK | 0:00:02 | |
| 787 | Jupiter | 4 | 0.98 | 1 | LINK | 0:00:02 | |
| 788 | Ganymede | 0 | 0 | 0 | LINK | 0:03:46 | Finds answer 1 |
| 789 | Jupiter | 5 | 0.98 | 1 | LINK | 0:00:02 | |
| 790 | Ganymede | 0 | 0.99 | 1 | LINK | 0:02:23 | Spends a lot of time at it again |
| 791 | Io | 1 | 0.95 | 1 | LINK | 0:00:03 | |
| 792 | Ganymede | 1 | 0.98 | 1 | LINK | 0:00:02 | Getting bored |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 793 | Io | 2 | 0.98 | 1 | LINK | 0:00:01 | |
| 794 | Ganymede | 2 | 0.98 | 1 | LINK | 0:00:19 | |
| 795 | Europa | 1 | 0.89 | 1 | LINK | 0:00:02 | |
| 796 | Ganymede | 3 | 0.98 | 1 | LINK | 0:00:02 | |
| 797 | Europa | 2 | 0.98 | 1 | LINK | 0:00:01 | |
| 798 | Ganymede | 4 | 0.98 | 1 | LINK | 0:00:01 | |
| 799 | Europa | 3 | 0.98 | 1 | LINK | 0:00:05 | |
| 800 | Ganymede | 5 | 0.98 | 1 | LINK | 0:00:02 | A lot of searching around Ganymede |
| 801 | Io | 3 | 0.92 | 1 | LINK | 0:00:03 | |
| 802 | Ariel | 2 | 0.57 | 1 | LINK | 0:00:03 | |
| 803 | Uranus | 0 | 0 | 0 | LINK | 0:00:43 | |
| 804 | Miranda | 0 | 0 | 0 | LINK | 0:00:19 | |
| 805 | Uranus | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 806 | Miranda | 0 | 0.99 | 1 | LINK | 0:00:07 | |
| 807 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:13 | |
| 808 | Volcano | 3 | 0.35 | 1 | LINK | 0:00:04 | |
| 809 | Asteria | 3 | 0.2 | 1 | LINK | 0:00:02 | |
| 810 | Venus | 2 | 0.59 | 1 | LINK | 0:00:04 | |
| 811 | Earth | 11 | 0.64 | 1 | LINK | 3:04:23 | A very long time! |
| 812 | Moon | 10 | 0.55 | 1 | LINK | 0:00:22 | Not much reading after this |
| 813 | Charon | 0 | 0.5 | 1 | LINK | 0:00:07 | |
| 814 | Deimos | 1 | 0.52 | 1 | LINK | 0:00:01 | |
| 815 | Europa | 4 | 0.84 | 1 | LINK | 0:00:01 | |
| 816 | Moon | 11 | 0.96 | 1 | LINK | 0:00:01 | |
| 817 | Ganymede | 6 | 0.83 | 1 | LINK | 0:00:01 | |
| 818 | Iapatus | 0 | 0 | 0 | LINK | 0:00:00 | |
| 819 | Ganymede | 7 | 0.98 | 1 | LINK | 0:00:04 | |
| 820 | Iapatus | 0 | 0.99 | 1 | LINK | 0:00:06 | |
| 821 | Europa | 5 | 0.94 | 1 | LINK | 0:00:04 | |
| 822 | Satellite | 0 | 0 | 0 | LINK | 0:00:06 | |
| 823 | Moon | 12 | 0.93 | 1 | LINK | 0:00:07 | |
| 824 | Earth | 12 | 0.87 | 1 | LINK | 0:00:06 | |
| 825 | Deimos | 2 | 0.89 | 1 | LINK | 0:00:06 | |
| 826 | Mars | 0 | 0 | 0 | LINK | 0:00:19 | |
| 827 | Phobos | 0 | 0 | 0 | LINK | 0:00:03 | |
| 828 | Ganymede | 8 | 0.91 | 1 | LINK | 0:00:02 | |
| 829 | Iapatus | 1 | 0.91 | 1 | LINK | 0:00:02 | |
| 830 | Ganymede | 9 | 0.98 | 1 | LINK | 0:00:04 | |
| 831 | Isis | 0 | 0 | 0 | LINK | 0:00:07 | |
| 832 | Volcano | 4 | 0.76 | 1 | LINK | 0:00:06 | |
| 833 | Mount Rainier | 0 | 0.18 | 1 | LINK | 0:00:33 | |
| 834 | Eruption | 0 | 0 | 0 | LINK | 0:00:11 | |
| 835 | Volcano | 5 | 0.97 | 1 | LINK | 0:00:02 | |
| 836 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:01 | |
| 837 | Volcano | 6 | 0.98 | 1 | LINK | 0:00:01 | |
| 838 | Geographical Feature | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 839 | Planet | 1 | 0 | 1 | LINK | 0:00:03 | |
| 840 | Geographical Feature | 1 | 0.98 | 1 | LINK | 0:00:04 | |
| 841 | Planet | 2 | 0.98 | 1 | LINK | 0:00:03 | |
| 842 | Solar System | 1 | 0 | 1 | LINK | 0:00:04 | |
| 843 | Star System | 0 | 0 | 0 | LINK | | |
| 844 | | | | | | | Answer 2 not found |
| 845 | Subject 8 : Novice | | | | | | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 846 |  |  |  |  |  |  |  |
| 847 |  |  |  | ## |  |  |  |
| 848 | Solar System | 0 | 0 | 0 | LINK | 0:00:25 | Spends a while around the first 2 nodes |
| 849 | Star System | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 850 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:06 |  |
| 851 | Star System | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 852 | Solar System | 1 | 0.98 | 1 | LINK | 0:02:38 |  |
| 853 | Star System | 1 | 0.98 | 1 | LINK | 0:00:03 |  |
| 854 | Solar System | 2 | 0.98 | 1 | LINK | 0:00:13 |  |
| 855 | Planet | 0 | 0 | 0 | LINK | 0:00:05 |  |
| 856 | Satellite | 0 | 0 | 0 | LINK | 0:00:32 |  |
| 857 | Moon | 0 | 0 | 0 | LINK | 0:00:54 |  |
| 858 | Earth | 0 | 0 | 0 | LINK | 0:00:29 |  |
| 859 | Moon | 0 | 0.99 | 1 | BACK | 0:00:04 |  |
| 860 | Satellite | 0 | 0.97 | 1 | BACK | 0:02:28 |  |
| 861 | Moon | 1 | 0.98 | 1 | LINK | 0:00:40 |  |
| 862 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:10 |  |
| 863 | Moon | 2 | 0.98 | 1 | LINK | 0:00:13 |  |
| 864 | Satellite | 1 | 0.96 | 1 | LINK | 0:00:10 | Spends a while around Moon |
| 865 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:10 |  |
| 866 | Satellite | 2 | 0.98 | 1 | LINK | 0:00:12 |  |
| 867 | Phobos | 0 | 0 | 0 | LINK | 0:01:08 |  |
| 868 | Phobos | 0 | 1 | 1 | BACK | 0:00:03 |  |
| 869 | Satellite | 3 | 0.97 | 1 | BACK | 0:00:05 |  |
| 870 | Deimos | 0 | 0 | 0 | LINK | 0:01:17 |  |
| 871 | Deimos | 0 | 1 | 1 | BACK | 0:00:02 |  |
| 872 | Geographical Feature | 0 | 0.94 | 1 | BACK | 0:00:08 |  |
| 873 | Satellite | 4 | 0.96 | 1 | BACK | 0:00:09 |  |
| 874 | Io | 0 | 0 | 0 | LINK | 0:01:23 |  |
| 875 | Satellite | 5 | 0.98 | 1 | LINK | 0:00:05 |  |
| 876 | Europa | 0 | 0 | 0 | LINK | 0:00:45 |  |
| 877 | Satellite | 6 | 0.98 | 1 | LINK | 0:00:08 |  |
| 878 | Ganymede | 0 | 0 | 0 | LINK | 0:01:47 | Finds answer 1 |
| 879 | Satellite | 7 | 0.98 | 1 | LINK | 0:00:05 |  |
| 880 | Calisto | 0 | 0 | 0 | LINK | 0:00:16 |  |
| 881 | Calisto | 0 | 1 | 1 | BACK | 0:00:02 |  |
| 882 | Satellite | 8 | 0.97 | 1 | BACK | 0:00:45 |  |
| 883 | Europa | 0 | 0.94 | 1 | LINK | 0:03:31 |  |
| 884 | Satellite | 9 | 0.98 | 1 | LINK | 0:00:03 |  |
| 885 | Io | 0 | 0.9 | 1 | LINK | 0:03:10 |  |
| 886 | Satellite | 10 | 0.98 | 1 | LINK | 0:00:07 |  |
| 887 | Phobos | 1 | 0.81 | 1 | LINK | 0:00:18 |  |
| 888 | Satellite | 11 | 0.98 | 1 | LINK | 0:00:08 |  |
| 889 | Moon | 3 | 0.74 | 1 | LINK | 0:01:47 | Spends a while at Moon again |
| 890 | Satellite | 12 | 0.98 | 1 | LINK | 0:00:05 |  |
| 891 | Phobos | 2 | 0.96 | 1 | LINK | 0:01:29 |  |
| 892 | Deimos | 1 | 0.79 | 1 | LINK | 0:00:22 |  |
| 893 | Satellite | 13 | 0.97 | 1 | LINK | 0:00:06 |  |
| 894 | Moon | 4 | 0.95 | 1 | LINK | 0:00:04 |  |
| 895 | Earth | 0 | 0.64 | 1 | LINK | 0:00:15 |  |
| 896 | Earth | 1 | 0.99 | 1 | BACK | 0:00:01 |  |
| 897 | Moon | 5 | 0.97 | 1 | BACK | 0:00:03 |  |
| 898 | Satellite | 14 | 0.95 | 1 | LINK | 0:00:13 |  |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 899 | Io | 1 | 0.86 | 1 | LINK | 0:00:20 | |
| 900 | Europa | 1 | 0.83 | 1 | LINK | 0:01:35 | A lot of users find this node interesting |
| 901 | Io | 2 | 0.98 | 1 | LINK | 0:01:35 | |
| 902 | Satellite | 15 | 0.96 | 1 | LINK | 0:00:35 | |
| 903 | Titan | 0 | 0 | 0 | LINK | 0:01:02 | Finds answer 2 |
| 904 | Saturn | 0 | 0 | 0 | LINK | 0:00:17 | |
| 905 | Titan | 0 | 0.99 | 1 | LINK | 0:00:10 | |
| 906 | Ariel | 0 | 0 | 0 | LINK | 0:00:17 | |
| 907 | Uranus | 0 | 0 | 0 | LINK | 0:00:07 | |
| 908 | Planet | 0 | 0.48 | 1 | LINK | 0:00:08 | |
| 909 | Solar System | 3 | 0.45 | 1 | LINK | 0:00:07 | |
| 910 | Star System | 2 | 0.43 | 1 | LINK | 0:00:24 | |
| 911 | Comet | 0 | 0 | 0 | LINK | 0:00:23 | |
| 912 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:08 | |
| 913 | Comet | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 914 | Hally's | 0 | 0 | 0 | LINK | 0:00:06 | |
| 915 | Comet | 1 | 0.98 | 1 | LINK | 0:00:05 | |
| 916 | Solar System | 4 | 0.93 | 1 | LINK | 0:00:02 | |
| 917 | Asteroid | 0 | 0 | 0 | LINK | 0:00:50 | |
| 918 | Eros | 0 | 0 | 0 | LINK | 0:00:06 | |
| 919 | Asteroid | 0 | 0.99 | 1 | LINK | 0:00:04 | |
| 920 | Trojan | 0 | 0 | 0 | LINK | 0:00:07 | |
| 921 | Asteroid | 1 | 0.98 | 1 | LINK | 0:00:03 | |
| 922 | Atlas | 0 | 0 | 0 | LINK | 0:00:07 | |
| 923 | Atlas | 0 | 1 | 1 | BACK | 0:00:02 | |
| 924 | Atlas | 1 | 0.99 | 1 | BACK | 0:00:01 | |
| 925 | Atlas | 2 | 0.99 | 1 | BACK | 0:00:01 | |
| 926 | Atlas | 3 | 0.99 | 1 | BACK | 0:00:00 | |
| 927 | Atlas | 4 | 0.99 | 1 | BACK | 0:00:02 | |
| 928 | Atlas | 5 | 0.99 | 1 | BACK | 0:00:02 | |
| 929 | Atlas | 6 | 0.99 | 1 | BACK | 0:00:02 | |
| 930 | Atlas | 7 | 0.99 | 1 | BACK | 0:00:02 | |
| 931 | Atlas | 8 | 0.99 | 1 | BACK | 0:00:03 | |
| 932 | Atlas | 9 | 0.99 | 1 | BACK | 0:00:05 | Possible bug in the system! |
| 933 | Asteroid | 2 | 0.88 | 1 | LINK | 0:00:02 | |
| 934 | Kuiper | 0 | 0 | 0 | LINK | 0:00:09 | |
| 935 | Trojan | 0 | 0.86 | 1 | LINK | 0:00:02 | |
| 936 | Eros | 0 | 0.83 | 1 | LINK | 0:00:05 | |
| 937 | Hally's | 0 | 0.78 | 1 | LINK | 0:00:02 | |
| 938 | Comet | 2 | 0.77 | 1 | LINK | 0:00:03 | |
| 939 | Solar System | 5 | 0.77 | 1 | LINK | 0:00:03 | |
| 940 | Planet | 1 | 0.68 | 1 | LINK | 0:00:05 | |
| 941 | Neptune | 0 | 0 | 0 | LINK | 0:00:08 | |
| 942 | Neried | 0 | 0 | 0 | LINK | 0:00:16 | |
| 943 | Triton | 0 | 0 | 0 | LINK | 0:00:11 | |
| 944 | Neptune | 0 | 0.98 | 1 | LINK | 0:00:07 | |
| 945 | Planet | 2 | 0.95 | 1 | LINK | 0:00:06 | |
| 946 | Uranus | 0 | 0.62 | 1 | LINK | 0:00:14 | |
| 947 | Miranda | 0 | 0 | 0 | LINK | 0:00:07 | |
| 948 | Umbriel | 0 | 0 | 0 | LINK | 0:00:05 | |
| 949 | Oberon | 0 | 0 | 0 | LINK | 0:00:05 | Seems to want to continue |
| 950 | Uranus Minor Satellites | 0 | 0 | 0 | LINK | 0:00:03 | an interest in satellites |
| 951 | Oberon | 0 | 0.99 | 1 | LINK | 0:00:01 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 952 | Uranus Minor Satellites | 0 | 0.99 | 1 | LINK | 0:00:12 | |
| 953 | Uranus Minor Satellites | 1 | 0.99 | 1 | BACK | 0:00:01 | |
| 954 | Oberon | 1 | 0.97 | 1 | BACK | 0:00:02 | |
| 955 | Uranus Minor Satellites | 2 | 0.98 | 1 | BACK | 0:00:01 | |
| 956 | Oberon | 2 | 0.98 | 1 | BACK | 0:00:00 | |
| 957 | Umbriel | 0 | 0.92 | 1 | BACK | 0:00:01 | |
| 958 | Miranda | 0 | 0.9 | 1 | BACK | 0:00:01 | |
| 959 | Uranus | 1 | 0.87 | 1 | BACK | 0:00:00 | |
| 960 | Planet | 3 | 0.85 | 1 | BACK | 0:00:12 | |
| 961 | Pluto | 0 | 0 | 0 | LINK | 0:00:16 | |
| 962 | Charon | 0 | 0 | 0 | LINK | 0:00:51 | |
| 963 | Satellite | 16 | 0.39 | 1 | LINK | 0:00:10 | |
| 964 | Geographical Feature | 1 | 0.08 | 1 | LINK | 0:00:06 | |
| 965 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:16 | Brief look at volcanoes |
| 966 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:09 | |
| 967 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:08 | |
| 968 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:15 | |
| 969 | Earth | 2 | 0.27 | 1 | LINK | 0:00:03 | Back to familiar territory? |
| 970 | Moon | 6 | 0.27 | 1 | LINK | 0:00:05 | |
| 971 | Satellite | 17 | 0.92 | 1 | LINK | 0:00:13 | |
| 972 | Moon | 7 | 0.98 | 1 | LINK | 0:00:06 | |
| 973 | Satellite | 18 | 0.98 | 1 | LINK | 0:00:03 | |
| 974 | Planet | 4 | 0.86 | 1 | LINK | 0:00:02 | |
| 975 | Solar System | 6 | 0.64 | 1 | LINK | 0:00:06 | |
| 976 | Star System | 3 | 0.34 | 1 | LINK | | |
| 977 | | | | | | | |
| 978 | Subject 9 : Novice | | | | | | |
| 979 | | | | | | | |
| 980 | | ### | | | | | |
| 981 | Solar System | 0 | 0 | 0 | LINK | 0:00:55 | |
| 982 | Star System | 0 | 0 | 0 | LINK | 0:00:41 | |
| 983 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:03 | Brief skip around the first 2 nodes |
| 984 | Planet | 0 | 0 | 0 | LINK | 0:00:21 | Takes off quickly |
| 985 | Mercury | 0 | 0 | 0 | LINK | 0:00:23 | |
| 986 | Planet | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 987 | Mercury | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 988 | Planet | 1 | 0.98 | 1 | LINK | 0:00:04 | |
| 989 | Venus | 0 | 0 | 0 | LINK | 0:00:02 | |
| 990 | Earth | 0 | 0 | 0 | LINK | 0:00:51 | |
| 991 | Venus | 0 | 0.99 | 1 | LINK | 0:00:21 | |
| 992 | Jupiter | 0 | 0 | 0 | LINK | 0:00:56 | |
| 993 | Mars | 0 | 0 | 0 | LINK | 0:00:26 | |
| 994 | Jupiter | 0 | 0.99 | 1 | LINK | 0:00:21 | |
| 995 | Neptune | 0 | 0 | 0 | LINK | 0:00:34 | |
| 996 | Pluto | 0 | 0 | 0 | LINK | 0:00:12 | |
| 997 | Pluto | 0 | 1 | 1 | LINK | 0:00:01 | |
| 998 | Pluto | 1 | 0.99 | 1 | LINK | 0:00:01 | |
| 999 | Mercury | 1 | 0.88 | 1 | LINK | 0:00:01 | |
| 1000 | Mars | 0 | 0.94 | 1 | LINK | 0:00:01 | |
| 1001 | Saturn | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1002 | Pluto | 2 | 0.96 | 1 | LINK | 0:00:01 | |
| 1003 | Neptune | 0 | 0.93 | 1 | LINK | 0:00:00 | |
| 1004 | Mercury | 2 | 0.95 | 1 | LINK | 0:00:01 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1005 | Mars | 1 | 0.95 | 1 | LINK | 0:00:01 | |
| 1006 | Jupiter | 1 | 0.88 | 1 | LINK | 0:00:01 | |
| 1007 | Io | 0 | 0 | 0 | LINK | 0:00:12 | |
| 1008 | Europa | 0 | 0 | 0 | LINK | 0:00:13 | |
| 1009 | Jupiter | 2 | 0.97 | 1 | LINK | 0:00:01 | |
| 1010 | Io | 0 | 0.98 | 1 | LINK | 0:00:02 | |
| 1011 | Europa | 0 | 0.98 | 1 | LINK | 0:00:01 | |
| 1012 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1013 | Galileo | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1014 | Spacecraft | 0 | 0 | 0 | LINK | 0:00:32 | |
| 1015 | Voyager | 0 | 0 | 0 | LINK | 0:00:44 | |
| 1016 | Uranus | 0 | 0 | 0 | LINK | 0:00:12 | |
| 1017 | Miranda | 0 | 0 | 0 | LINK | 0:00:15 | |
| 1018 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:04 | Quite a long way away from the |
| 1019 | Volcano | 0 | 0 | 0 | LINK | 0:00:42 | start, seems unconcerned |
| 1020 | Asteria | 0 | 0 | 0 | LINK | 0:00:03 | and is spending less time at nodes |
| 1021 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:12 | Well away from task now |
| 1022 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:41 | |
| 1023 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:34 | |
| 1024 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:56 | |
| 1025 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:09 | |
| 1026 | Versuvius | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1027 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:08 | |
| 1028 | Versuvius | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 1029 | Mount Saint Helens | 0 | 0.96 | 1 | LINK | 0:00:18 | |
| 1030 | Mount Rainier | 0 | 0.94 | 1 | LINK | 0:00:03 | |
| 1031 | Mount Etna | 0 | 0.92 | 1 | LINK | 0:00:01 | |
| 1032 | Montes Haemus | 0 | 0.9 | 1 | LINK | 0:00:02 | |
| 1033 | Moon | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1034 | Earth | 0 | 0.57 | 1 | LINK | 0:00:04 | |
| 1035 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1036 | Canyon | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1037 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1038 | Canyon | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 1039 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1040 | Noctis | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1041 | Canyon | 1 | 0.97 | 1 | LINK | 0:00:01 | |
| 1042 | Geographical Feature | 0 | 0.96 | 1 | LINK | 0:00:01 | |
| 1043 | Volcano | 0 | 0.77 | 1 | LINK | 0:00:04 | |
| 1044 | Mount Etna | 1 | 0.87 | 1 | LINK | 0:00:04 | |
| 1045 | Eruption | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1046 | Versuvius | 1 | 0.82 | 1 | LINK | 0:00:02 | |
| 1047 | Eruption | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 1048 | Mount Saint Helens | 1 | 0.81 | 1 | LINK | 0:00:03 | |
| 1049 | Mount Rainier | 1 | 0.81 | 1 | LINK | 0:00:00 | |
| 1050 | Olympus Mons | 0 | 0.76 | 1 | LINK | 0:00:01 | |
| 1051 | Montes Haemus | 1 | 0.81 | 1 | LINK | 0:00:01 | |
| 1052 | Geographical Feature | 1 | 0.9 | 1 | LINK | 0:00:02 | |
| 1053 | Mount Saint Helens | 2 | 0.95 | 1 | LINK | 0:00:01 | |
| 1054 | Earth | 1 | 0.8 | 1 | LINK | 0:00:04 | |
| 1055 | Versuvius | 2 | 0.91 | 1 | LINK | 0:00:02 | |
| 1056 | Volcano | 1 | 0.87 | 1 | LINK | 0:00:02 | |
| 1057 | Mount Rainier | 2 | 0.92 | 1 | LINK | 0:00:04 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1058 | Volcano | 2 | 0.98 | 1 | LINK | 0:00:01 | |
| 1059 | Planet | 2 | 0.29 | 1 | LINK | 0:00:01 | |
| 1060 | Satellite | 0 | 0 | 0 | LINK | 0:00:12 | This nodes seems to have reminded them |
| 1061 | Phobos | 0 | 0 | 0 | LINK | 0:00:22 | of the task in hand |
| 1062 | Deimos | 0 | 0 | 0 | LINK | 0:00:28 | |
| 1063 | Ariel | 0 | 0 | 0 | LINK | 0:00:12 | |
| 1064 | Callisto | 0 | 0 | 0 | LINK | 0:00:14 | |
| 1065 | Encelydus | 0 | 0 | 0 | LINK | 0:00:13 | |
| 1066 | Titan | 0 | 0 | 0 | LINK | 0:00:50 | Found answer 2 but has more or less |
| 1067 | Satellite | 0 | 0.94 | 1 | LINK | 0:00:01 | Stumbled over it? |
| 1068 | Planet | 3 | 0.91 | 1 | LINK | 0:00:01 | |
| 1069 | Solar System | 1 | 0.14 | 1 | LINK | 0:00:01 | |
| 1070 | Planet | 4 | 0.98 | 1 | LINK | 0:00:01 | |
| 1071 | Solar System | 2 | 0.98 | 1 | LINK | 0:00:00 | |
| 1072 | Planet | 5 | 0.98 | 1 | LINK | 0:00:03 | |
| 1073 | Solar System | 3 | 0.98 | 1 | LINK | 0:00:03 | |
| 1074 | Star System | 0 | 0.09 | 1 | LINK | | No further interest after finding an answer |
| 1075 | | | | | | | |
| 1076 | Subject 10 : Novice | | | | | | |
| 1077 | | | | | | | |
| 1078 | | | | | | | |
| 1079 | Asteroid | 0 | 0 | 0 | LINK | 0:00:02 | Seem to have the very start missing! |
| 1080 | Asteroid | 0 | 1 | 1 | LINK | 0:00:05 | As a result is off track for a while |
| 1081 | Eros | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1082 | Trojan | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1083 | Kuiper | 0 | 0 | 0 | LINK | 0:00:00 | |
| 1084 | Atlas | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1085 | Oort | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1086 | Trojan | 0 | 0.97 | 1 | LINK | 0:00:01 | |
| 1087 | Eros | 0 | 0.95 | 1 | LINK | 0:00:01 | |
| 1088 | Atlas | 0 | 0.97 | 1 | LINK | 0:00:00 | |
| 1089 | Kuiper | 0 | 0.95 | 1 | LINK | 0:00:08 | |
| 1090 | Oort | 0 | 0.96 | 1 | LINK | 0:00:01 | |
| 1091 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1092 | Oort | 1 | 0.98 | 1 | LINK | 0:00:01 | |
| 1093 | Hally's | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1094 | Trojan | 1 | 0.92 | 1 | LINK | 0:00:02 | |
| 1095 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1096 | Comet | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1097 | Solar System | 0 | 0 | 0 | LINK | 0:00:01 | This looks like the entry point |
| 1098 | Planet | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1099 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:11 | |
| 1100 | Planet | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 1101 | Earth | 0 | 0 | 0 | LINK | 0:00:08 | |
| 1102 | Moon | 0 | 0 | 0 | LINK | 0:00:08 | Finds a familiar node |
| 1103 | Earth | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 1104 | Moon | 0 | 0.99 | 1 | LINK | 0:00:06 | |
| 1105 | Satellite | 0 | 0 | 0 | LINK | 0:00:11 | |
| 1106 | Phobos | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1107 | Deimos | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1108 | Phobos | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 1109 | Deimos | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 1110 | Mars | 0 | 0 | 0 | LINK | 0:00:04 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1111 | Deimos | 1 | 0.98 | 1 | LINK | 0:00:03 | |
| 1112 | Mars | 0 | 0.99 | 1 | LINK | 0:00:04 | |
| 1113 | Deimos | 2 | 0.98 | 1 | BACK | 0:00:01 | |
| 1114 | Mars | 1 | 0.98 | 1 | BACK | 0:00:01 | |
| 1115 | Deimos | 3 | 0.98 | 1 | BACK | 0:00:01 | |
| 1116 | Phobos | 1 | 0.92 | 1 | BACK | 0:00:01 | |
| 1117 | Deimos | 4 | 0.98 | 1 | BACK | 0:00:00 | |
| 1118 | Phobos | 2 | 0.98 | 1 | BACK | 0:00:01 | |
| 1119 | Satellite | 0 | 0.87 | 1 | BACK | 0:00:02 | |
| 1120 | Moon | 1 | 0.84 | 1 | BACK | 0:00:06 | |
| 1121 | Earth | 1 | 0.82 | 1 | LINK | 0:00:28 | |
| 1122 | Saturn | 0 | 0 | 0 | LINK | 0:00:03 | related links |
| 1123 | Pluto | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1124 | Mercury | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1125 | Saturn | 0 | 0.98 | 1 | LINK | 0:00:01 | |
| 1126 | Pluto | 0 | 0.98 | 1 | LINK | 0:00:02 | |
| 1127 | Jupiter | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1128 | Neptune | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1129 | Saturn | 1 | 0.96 | 1 | LINK | 0:00:06 | |
| 1130 | Pluto | 1 | 0.96 | 1 | LINK | 0:00:03 | |
| 1131 | Planet | 1 | 0.69 | 1 | LINK | 0:00:03 | |
| 1132 | Mercury | 0 | 0.93 | 1 | LINK | 0:00:02 | |
| 1133 | Saturn | 2 | 0.96 | 1 | LINK | 0:00:01 | |
| 1134 | Pluto | 2 | 0.96 | 1 | LINK | 0:00:02 | Exploring all the related links on offer? |
| 1135 | Pluto | 3 | 0.99 | 1 | BACK | 0:00:00 | |
| 1136 | Saturn | 3 | 0.97 | 1 | BACK | 0:00:01 | |
| 1137 | Mercury | 1 | 0.95 | 1 | BACK | 0:00:01 | |
| 1138 | Planet | 2 | 0.93 | 1 | BACK | 0:00:00 | |
| 1139 | Pluto | 4 | 0.96 | 1 | BACK | 0:00:01 | |
| 1140 | Saturn | 4 | 0.96 | 1 | BACK | 0:00:00 | |
| 1141 | Neptune | 0 | 0.88 | 1 | BACK | 0:00:01 | |
| 1142 | Jupiter | 0 | 0.86 | 1 | BACK | 0:00:05 | |
| 1143 | Earth | 2 | 0.78 | 1 | LINK | 0:00:56 | Clicks back until a familiar node is found |
| 1144 | Planet | 3 | 0.94 | 1 | LINK | 0:00:03 | |
| 1145 | Satellite | 1 | 0.74 | 1 | LINK | 0:00:04 | |
| 1146 | Moon | 2 | 0.74 | 1 | LINK | 0:00:04 | |
| 1147 | Earth | 3 | 0.96 | 1 | LINK | 0:00:03 | Back to Earth again |
| 1148 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:07 | |
| 1149 | Volcano | 0 | 0 | 0 | LINK | 0:00:11 | Wandering off track |
| 1150 | Versuvius | 0 | 0 | 0 | LINK | 0:00:14 | |
| 1151 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1152 | Asteria | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1153 | Versuvius | 0 | 0.98 | 1 | LINK | 0:00:00 | |
| 1154 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1155 | Mount Etna | 0 | 0.94 | 1 | LINK | 0:00:01 | |
| 1156 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1157 | Versuvius | 1 | 0.96 | 1 | LINK | 0:00:01 | |
| 1158 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1159 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1160 | Mount Saint Helens | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 1161 | Inverness Corona | 0 | 0.91 | 1 | LINK | 0:00:02 | |
| 1162 | Inverness Corona | 1 | 0.99 | 1 | BACK | 0:00:01 | |
| 1163 | Mount Saint Helens | 1 | 0.97 | 1 | BACK | 0:00:02 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1164 | Earth | 4 | 0.83 | 1 | LINK | 0:00:17 | Back until familiar |
| 1165 | Moon | 3 | 0.81 | 1 | LINK | 0:00:09 | |
| 1166 | Satellite | 2 | 0.79 | 1 | LINK | 0:00:08 | |
| 1167 | Io | 0 | 0 | 0 | LINK | 0:00:07 | |
| 1168 | Ariel | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1169 | Phobos | 3 | 0.49 | 1 | LINK | 0:00:00 | |
| 1170 | Charon | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1171 | Callisto | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1172 | Iapatus | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1173 | Titan | 0 | 0 | 0 | LINK | 0:00:03 | Finds one of the answers but hasn't been there |
| 1174 | Charon | 0 | 0.97 | 1 | LINK | 0:00:01 | long enough to realise it |
| 1175 | Iapatus | 0 | 0.98 | 1 | LINK | 0:00:02 | |
| 1176 | Titan | 0 | 0.98 | 1 | LINK | 0:00:01 | and again |
| 1177 | Janus | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1178 | Saturn | 5 | 0.62 | 1 | LINK | 0:00:07 | |
| 1179 | Pluto | 5 | 0.6 | 1 | LINK | 0:00:01 | |
| 1180 | Pluto | 6 | 0.99 | 1 | LINK | 0:00:01 | |
| 1181 | Pluto | 7 | 0.99 | 1 | LINK | 0:00:00 | |
| 1182 | Neptune | 1 | 0.59 | 1 | LINK | 0:00:03 | |
| 1183 | Neptune | 2 | 0.99 | 1 | BACK | 0:00:00 | lots of backs |
| 1184 | Pluto | 8 | 0.97 | 1 | BACK | 0:00:00 | |
| 1185 | Pluto | 9 | 0.99 | 1 | BACK | 0:00:01 | |
| 1186 | Pluto | 10 | 0.99 | 1 | BACK | 0:00:01 | |
| 1187 | Saturn | 6 | 0.91 | 1 | BACK | 0:00:06 | |
| 1188 | Janus | 0 | 0.9 | 1 | BACK | 0:00:00 | |
| 1189 | Titan | 1 | 0.87 | 1 | BACK | 0:00:01 | |
| 1190 | Iapatus | 1 | 0.85 | 1 | BACK | 0:00:00 | |
| 1191 | Charon | 1 | 0.83 | 1 | BACK | 0:00:01 | |
| 1192 | Titan | 2 | 0.97 | 1 | BACK | 0:00:01 | |
| 1193 | Iapatus | 2 | 0.97 | 1 | BACK | 0:00:00 | |
| 1194 | Callisto | 0 | 0.78 | 1 | BACK | 0:00:01 | |
| 1195 | Charon | 2 | 0.96 | 1 | BACK | 0:00:00 | |
| 1196 | Phobos | 4 | 0.73 | 1 | BACK | 0:00:00 | |
| 1197 | Ariel | 0 | 0.72 | 1 | BACK | 0:00:01 | |
| 1198 | Io | 0 | 0.7 | 1 | BACK | 0:00:01 | |
| 1199 | Io | 1 | 0.99 | 1 | BACK | 0:00:00 | |
| 1200 | Io | 2 | 0.99 | 1 | BACK | 0:00:01 | |
| 1201 | Io | 3 | 0.99 | 1 | BACK | 0:00:00 | |
| 1202 | Io | 4 | 0.99 | 1 | BACK | 0:00:03 | |
| 1203 | Io | 5 | 0.99 | 1 | BACK | 0:00:01 | |
| 1204 | Charon | 3 | 0.91 | 1 | LINK | 0:00:04 | quite high spikes |
| 1205 | Callisto | 1 | 0.89 | 1 | LINK | 0:00:01 | |
| 1206 | Charon | 4 | 0.98 | 1 | LINK | 0:00:03 | |
| 1207 | Moon | 4 | 0.58 | 1 | LINK | 0:00:20 | |
| 1208 | Earth | 5 | 0.56 | 1 | LINK | 0:00:22 | |
| 1209 | Moon | 5 | 0.98 | 1 | LINK | 0:00:21 | |
| 1210 | Io | 6 | 0.93 | 1 | LINK | 0:00:02 | |
| 1211 | Iapatus | 3 | 0.82 | 1 | LINK | 0:00:01 | |
| 1212 | Deimos | 5 | 0.05 | 1 | LINK | 0:00:01 | |
| 1213 | Io | 7 | 0.97 | 1 | LINK | 0:00:05 | |
| 1214 | Deimos | 6 | 0.98 | 1 | LINK | 0:00:02 | |
| 1215 | Iapatus | 4 | 0.96 | 1 | LINK | 0:00:08 | |
| 1216 | Satellite | 3 | 0.5 | 1 | LINK | 0:00:04 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1217 | Moon | 6 | 0.92 | 1 | LINK | 0:00:01 | |
| 1218 | Earth | 6 | 0.9 | 1 | LINK | 0:00:01 | |
| 1219 | Moon | 7 | 0.98 | 1 | LINK | 0:00:39 | |
| 1220 | Io | 8 | 0.93 | 1 | LINK | 0:00:01 | |
| 1221 | Janus | 1 | 0.67 | 1 | LINK | 0:00:02 | |
| 1222 | Satellite | 4 | 0.94 | 1 | LINK | 0:00:07 | |
| 1223 | Io | 9 | 0.97 | 1 | LINK | 0:00:01 | |
| 1224 | Europa | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1225 | Europa Orbiter | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1226 | Galileo | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1227 | Europa Orbiter | 0 | 0.99 | 1 | LINK | 0:00:01 | |
| 1228 | Galileo | 0 | 0.99 | 1 | LINK | 0:00:05 | |
| 1229 | Europa | 0 | 0.96 | 1 | LINK | 0:00:01 | |
| 1230 | Jupiter | 1 | 0.12 | 1 | LINK | 0:00:02 | |
| 1231 | Ganymede | 0 | 0 | 0 | LINK | 0:00:02 | Answer 1 but again can't have read it |
| 1232 | Io | 10 | 0.91 | 1 | LINK | 0:00:03 | |
| 1233 | Janus | 2 | 0.88 | 1 | LINK | 0:00:01 | |
| 1234 | Europa | 1 | 0.95 | 1 | LINK | 0:00:00 | |
| 1235 | Moon | 8 | 0.84 | 1 | LINK | 10:38:53 | A long time |
| 1236 | Solar System | 0 | 0 | 0 | LINK | 0:00:11 | Near the start |
| 1237 | Star System | 0 | 0 | 0 | LINK | 0:00:07 | |
| 1238 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:09 | |
| 1239 | Planet | 0 | 0 | 0 | LINK | 0:00:10 | |
| 1240 | Solar System | 1 | 0.98 | 1 | BACK | 0:00:01 | |
| 1241 | Star System | 0 | 0.97 | 1 | BACK | 0:00:05 | |
| 1242 | Planet | 0 | 0.98 | 1 | LINK | 0:00:02 | |
| 1243 | Earth | 0 | 0 | 0 | LINK | 0:00:35 | |
| 1244 | Moon | 0 | 0 | 0 | LINK | 0:00:34 | |
| 1245 | Moon | 0 | 1 | 1 | BACK | 0:00:06 | |
| 1246 | | 0 | 0 | 0 | FORWARD | 0:00:02 | |
| 1247 | Moon | 1 | 0.98 | 1 | BACK | 0:00:07 | |
| 1248 | Satellite | 0 | 0 | 0 | LINK | 0:00:35 | |
| 1249 | Phobos | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1250 | Moon | 2 | 0.97 | 1 | LINK | 0:00:06 | |
| 1251 | Earth | 0 | 0.93 | 1 | LINK | 0:00:08 | |
| 1252 | Phobos | 0 | 0.98 | 1 | LINK | 0:00:07 | |
| 1253 | Mars | 0 | 0 | 0 | LINK | 0:00:20 | |
| 1254 | Deimos | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1255 | Moon | 3 | 0.95 | 1 | LINK | 0:00:44 | |
| 1256 | Satellite | 0 | 0.93 | 1 | LINK | 0:00:06 | |
| 1257 | Io | 0 | 0 | 0 | LINK | 0:00:07 | |
| 1258 | Moon | 4 | 0.97 | 1 | LINK | 0:00:04 | |
| 1259 | Satellite | 1 | 0.97 | 1 | LINK | 0:00:02 | |
| 1260 | Io | 0 | 0.98 | 1 | LINK | 0:00:01 | |
| 1261 | Moon | 5 | 0.97 | 1 | LINK | 0:00:27 | |
| 1262 | Ariel | 0 | 0 | 0 | LINK | 0:00:09 | |
| 1263 | Moon | 6 | 0.98 | 1 | LINK | 0:00:08 | |
| 1264 | Satellite | 2 | 0.95 | 1 | LINK | 0:00:10 | |
| 1265 | Europa | 0 | 0 | 0 | LINK | 0:00:10 | |
| 1266 | Satellite | 3 | 0.98 | 1 | LINK | 0:00:05 | |
| 1267 | Moon | 7 | 0.96 | 1 | LINK | 0:00:04 | |
| 1268 | Charon | 0 | 0 | 0 | LINK | 0:00:14 | |
| 1269 | Satellite | 4 | 0.97 | 1 | LINK | 0:00:03 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1270 | Ganymede | 0 | 0 | 0 | LINK | 0:00:29 | spends a while at answer 1 this time |
| 1271 | Jupiter | 0 | 0 | 0 | LINK | 0:00:06 | |
| 1272 | Moon | 8 | 0.95 | 1 | LINK | 0:00:03 | |
| 1273 | Satellite | 5 | 0.96 | 1 | LINK | 0:00:14 | |
| 1274 | Calisto | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1275 | Calisto | 0 | 1 | 1 | BACK | 0:00:00 | |
| 1276 | Satellite | 6 | 0.97 | 1 | BACK | 0:00:09 | |
| 1277 | Titan | 0 | 0 | 0 | LINK | 0:00:09 | |
| 1278 | Saturn | 0 | 0 | 0 | LINK | 0:02:01 | |
| 1279 | Moon | 9 | 0.93 | 1 | LINK | 0:00:03 | |
| 1280 | Satellite | 7 | 0.96 | 1 | LINK | 0:00:02 | |
| 1281 | Planet | 1 | 0.61 | 1 | LINK | 0:00:06 | |
| 1282 | Earth | 1 | 0.69 | 1 | LINK | 0:00:02 | |
| 1283 | Planet | 2 | 0.98 | 1 | LINK | 0:00:03 | |
| 1284 | Venus | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1285 | Planet | 3 | 0.98 | 1 | LINK | 0:00:03 | |
| 1286 | Mercury | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1287 | Planet | 4 | 0.98 | 1 | LINK | 0:00:04 | |
| 1288 | Mars | 0 | 0.66 | 1 | LINK | 0:00:02 | |
| 1289 | Planet | 5 | 0.98 | 1 | LINK | 0:00:05 | |
| 1290 | Satellite | 8 | 0.9 | 1 | LINK | 0:00:02 | |
| 1291 | Planet | 6 | 0.98 | 1 | LINK | 0:00:02 | |
| 1292 | Satellite | 9 | 0.98 | 1 | LINK | 0:00:02 | |
| 1293 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1294 | Planet | 7 | 0.97 | 1 | LINK | 0:00:58 | |
| 1295 | Geographical Feature | 0 | 0.99 | 1 | LINK | 0:00:04 | |
| 1296 | Volcano | 0 | 0 | 0 | LINK | 0:00:08 | |
| 1297 | Geographical Feature | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 1298 | Planet | 8 | 0.96 | 1 | LINK | 0:00:03 | |
| 1299 | Earth | 2 | 0.83 | 1 | LINK | 0:00:05 | |
| 1300 | Planet | 9 | 0.98 | 1 | LINK | 0:00:02 | |
| 1301 | Geographical Feature | 2 | 0.96 | 1 | LINK | 0:00:03 | |
| 1302 | Volcano | 0 | 0.95 | 1 | LINK | 0:00:02 | |
| 1303 | Planet | 10 | 0.97 | 1 | LINK | 0:00:01 | |
| 1304 | Earth | 3 | 0.95 | 1 | LINK | 0:00:02 | |
| 1305 | Planet | 11 | 0.98 | 1 | LINK | 0:00:02 | |
| 1306 | Geographical Feature | 3 | 0.95 | 1 | LINK | 0:00:03 | |
| 1307 | Volcano | 1 | 0.95 | 1 | LINK | 0:00:03 | |
| 1308 | Asteria | 0 | 0 | 0 | LINK | 0:00:09 | |
| 1309 | Geographical Feature | 4 | 0.97 | 1 | LINK | 0:00:13 | |
| 1310 | Volcano | 2 | 0.97 | 1 | LINK | 0:00:02 | |
| 1311 | Versuvius | 0 | 0 | 0 | LINK | 0:00:14 | |
| 1312 | Earth | 4 | 0.92 | 1 | LINK | 0:02:56 | a very long time here |
| 1313 | Planet | 12 | 0.92 | 1 | LINK | 0:00:03 | |
| 1314 | Geographical Feature | 5 | 0.95 | 1 | LINK | 0:00:02 | |
| 1315 | Volcano | 3 | 0.95 | 1 | LINK | 0:00:01 | |
| 1316 | Mount Etna | 0 | 0 | 0 | LINK | 0:03:24 | |
| 1317 | Volcano | 4 | 0.98 | 1 | LINK | 0:00:03 | |
| 1318 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:24 | |
| 1319 | Earth | 5 | 0.93 | 1 | LINK | 0:00:02 | |
| 1320 | Planet | 13 | 0.93 | 1 | LINK | 0:00:02 | |
| 1321 | Geographical Feature | 6 | 0.93 | 1 | LINK | 0:00:02 | |
| 1322 | Volcano | 5 | 0.95 | 1 | LINK | 0:00:02 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1323 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:36 | Interested in Volcanoes? |
| 1324 | Volcano | 6 | 0.98 | 1 | LINK | 0:00:02 | |
| 1325 | Montes Haemus | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1326 | Moon | 10 | 0.53 | 1 | LINK | 0:00:10 | |
| 1327 | Satellite | 10 | 0.65 | 1 | LINK | 0:00:02 | |
| 1328 | Geographical Feature | 7 | 0.93 | 1 | LINK | 0:00:02 | |
| 1329 | Volcano | 7 | 0.95 | 1 | LINK | 0:00:02 | |
| 1330 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1331 | Tharsis Volcano Group | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1332 | Mars | 1 | 0.56 | 1 | LINK | 0:00:01 | |
| 1333 | Planet | 14 | 0.87 | 1 | LINK | 0:00:02 | |
| 1334 | Geographical Feature | 8 | 0.94 | 1 | LINK | 0:00:05 | |
| 1335 | Volcano | 8 | 0.94 | 1 | LINK | 0:00:05 | |
| 1336 | Isis | 0 | 0 | 0 | LINK | 0:00:18 | |
| 1337 | Ganymede | 0 | 0.34 | 1 | LINK | 1:19:05 | Must have realised this time |
| 1338 | Ganymede | 1 | 0.99 | 1 | BACK | 0:00:07 | |
| 1339 | Isis | 0 | 0.98 | 1 | BACK | 0:00:25 | |
| 1340 | Volcano | 9 | 0.95 | 1 | LINK | 0:00:05 | |
| 1341 | Inverness Corona | 0 | 0 | 0 | LINK | 0:00:41 | |
| 1342 | Geographical Feature | 9 | 0.92 | 1 | LINK | 0:00:02 | |
| 1343 | Canyon | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1344 | Planet | 15 | 0.89 | 1 | LINK | 0:00:03 | |
| 1345 | Geographical Feature | 10 | 0.97 | 1 | LINK | 0:00:02 | |
| 1346 | Canyon | 0 | 0.98 | 1 | LINK | 0:00:03 | |
| 1347 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1348 | Earth | 6 | 0.71 | 1 | LINK | 0:00:01 | |
| 1349 | Planet | 16 | 0.95 | 1 | LINK | 0:00:01 | |
| 1350 | Geographical Feature | 11 | 0.95 | 1 | LINK | 0:00:01 | |
| 1351 | Canyon | 1 | 0.95 | 1 | LINK | 0:00:02 | |
| 1352 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:03 | |
| 1353 | Canyon | 2 | 0.98 | 1 | LINK | 0:00:03 | |
| 1354 | Noctis | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1355 | Geographical Feature | 12 | 0.95 | 1 | LINK | 0:00:03 | |
| 1356 | Planet | 17 | 0.93 | 1 | LINK | 0:00:06 | |
| 1357 | Uranus | 0 | 0 | 0 | LINK | 0:00:48 | |
| 1358 | Miranda | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1359 | Satellite | 11 | 0.68 | 1 | LINK | 0:00:02 | |
| 1360 | Planet | 18 | 0.96 | 1 | LINK | 0:00:01 | |
| 1361 | Neptune | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1362 | Planet | 19 | 0.98 | 1 | LINK | 0:00:06 | |
| 1363 | Solar System | 2 | 0 | 1 | LINK | 0:00:01 | |
| 1364 | Planet | 20 | 0.98 | 1 | LINK | 0:00:01 | |
| 1365 | Solar System | 3 | 0.98 | 1 | LINK | 0:00:02 | |
| 1366 | Comet | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1367 | Hally's | 0 | 0 | 0 | LINK | 0:00:05 | |
| 1368 | Comet | 0 | 0.99 | 1 | LINK | 0:00:02 | |
| 1369 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1370 | Comet | 1 | 0.98 | 1 | LINK | 0:00:01 | |
| 1371 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:01 | |
| 1372 | Comet | 2 | 0.98 | 1 | LINK | 0:00:02 | |
| 1373 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:02 | |
| 1374 | Comet | 3 | 0.98 | 1 | LINK | 0:00:01 | |
| 1375 | Solar System | 4 | 0.9 | 1 | LINK | 0:00:02 | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 1376 | Planet | 21 | 0.88 | 1 | LINK | 0:00:44 |  |
| 1377 | Solar System | 5 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1378 | Asteroid | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 1379 | Eros | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 1380 | Asteroid | 0 | 0.99 | 1 | LINK | 0:00:03 |  |
| 1381 | Trojan | 0 | 0 | 0 | LINK | 0:00:01 |  |
| 1382 | Asteroid | 1 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1383 | Atlas | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 1384 | Asteroid | 2 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1385 | Kuiper | 0 | 0 | 0 | LINK | 0:00:01 |  |
| 1386 | Asteroid | 3 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1387 | Oort | 0 | 0 | 0 | LINK | 0:00:04 |  |
| 1388 | Asteroid | 4 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1389 | Oort | 0 | 0.99 | 1 | LINK | 0:00:03 |  |
| 1390 | Hyutaki | 0 | 0.82 | 1 | LINK | 0:00:02 |  |
| 1391 | Comet | 4 | 0.83 | 1 | LINK | 0:00:02 |  |
| 1392 | Solar System | 6 | 0.85 | 1 | LINK | 0:00:05 | Finished |
| 1393 | Star System | 1 | 0 | 1 | LINK |  |  |
| 1394 |  |  |  |  |  |  |  |
| 1395 | Subject 11 : Novice |  |  |  |  |  |  |
| 1396 |  |  |  |  |  |  |  |
| 1397 | Solar System | 0 | 0 | 0 | LINK | 0:01:01 | Spends some time at the first 2 nodes |
| 1398 | Star System | 0 | 0 | 0 | LINK | 0:02:18 |  |
| 1399 | Solar System | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 1400 | Star System | 0 | 0.99 | 1 | LINK | 0:00:02 |  |
| 1401 | Solar System | 1 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1402 | Planet | 0 | 0 | 0 | LINK | 0:00:26 |  |
| 1403 | Solar System | 2 | 0.98 | 1 | LINK | 0:02:27 |  |
| 1404 | Planet | 0 | 0.99 | 1 | LINK | 0:00:02 | Spends a long time around Earth, Moon |
| 1405 | Earth | 0 | 0 | 0 | LINK | 0:03:06 | Perhaps reluctant to move away? |
| 1406 | Planet | 1 | 0.98 | 1 | LINK | 0:00:03 |  |
| 1407 | Earth | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 1408 | Moon | 0 | 0 | 0 | LINK | 0:04:32 |  |
| 1409 | Earth | 1 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1410 | Planet | 2 | 0.96 | 1 | LINK | 0:00:02 |  |
| 1411 | Earth | 2 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1412 | Moon | 0 | 0.97 | 1 | LINK | 0:00:45 |  |
| 1413 | Earth | 3 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1414 | Moon | 1 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1415 | Satellite | 0 | 0 | 0 | LINK | 0:01:32 |  |
| 1416 | Moon | 2 | 0.98 | 1 | LINK | 0:00:03 |  |
| 1417 | Satellite | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 1418 | Phobos | 0 | 0 | 0 | LINK | 0:01:03 |  |
| 1419 | Satellite | 1 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1420 | Deimos | 0 | 0 | 0 | LINK | 0:00:32 |  |
| 1421 | Satellite | 2 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1422 | Moon | 3 | 0.94 | 1 | LINK | 0:00:01 | Keeps coming back here |
| 1423 | Earth | 4 | 0.9 | 1 | LINK | 0:00:02 |  |
| 1424 | Planet | 3 | 0.86 | 1 | LINK | 0:00:01 |  |
| 1425 | Solar System | 3 | 0.78 | 1 | LINK | 0:00:03 |  |
| 1426 | Planet | 4 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1427 | Earth | 5 | 0.96 | 1 | LINK | 0:00:02 |  |
| 1428 | Moon | 4 | 0.94 | 1 | LINK | 0:00:01 |  |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 1429 | Satellite | 3 | 0.92 | 1 | LINK | 0:01:20 |  |
| 1430 | Io | 0 | 0 | 0 | LINK | 0:00:56 |  |
| 1431 | Moon | 5 | 0.97 | 1 | LINK | 0:00:02 | Now picking a related link as a shortcut back |
| 1432 | Satellite | 4 | 0.97 | 1 | LINK | 0:00:03 |  |
| 1433 | Europa | 0 | 0 | 0 | LINK | 0:02:11 |  |
| 1434 | Satellite | 5 | 0.98 | 1 | LINK | 0:00:03 |  |
| 1435 | Ganymede | 0 | 0 | 0 | LINK | 0:04:45 | Finds the answer, quite quickly for such |
| 1436 | Jupiter | 0 | 0 | 0 | LINK | 0:02:26 | A repetitive search (ripple?) |
| 1437 | Callisto | 0 | 0 | 0 | LINK | 0:00:46 |  |
| 1438 | Moon | 6 | 0.93 | 1 | LINK | 0:00:01 | Still using this as a land mark |
| 1439 | Satellite | 6 | 0.95 | 1 | LINK | 0:00:32 |  |
| 1440 | Planet | 5 | 0.86 | 1 | LINK | 0:00:07 |  |
| 1441 | Mercury | 0 | 0 | 0 | LINK | 0:00:12 |  |
| 1442 | Planet | 6 | 0.98 | 1 | LINK | 0:00:02 | Wandered away from Satellites |
| 1443 | Venus | 0 | 0 | 0 | LINK | 0:00:52 |  |
| 1444 | Planet | 7 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1445 | Mars | 0 | 0 | 0 | LINK | 0:04:58 |  |
| 1446 | Olympus Mons | 0 | 0 | 0 | LINK | 0:00:11 | Not caught by Volcanoes |
| 1447 | Mars | 0 | 0.99 | 1 | LINK | 0:00:02 |  |
| 1448 | Planet | 8 | 0.96 | 1 | LINK | 0:00:02 |  |
| 1449 | Mars | 1 | 0.98 | 1 | LINK | 0:00:17 |  |
| 1450 | Planet | 9 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1451 | Jupiter | 0 | 0.86 | 1 | LINK | 0:00:03 |  |
| 1452 | Planet | 10 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1453 | Saturn | 0 | 0 | 0 | LINK | 0:01:28 | Using Saturn as a landmark for these satellites? |
| 1454 | Janus | 0 | 0 | 0 | LINK | 0:00:44 |  |
| 1455 | Saturn | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 1456 | Titan | 0 | 0 | 0 | LINK | 0:00:58 | Answer 2 found by searching |
| 1457 | Iapatus | 0 | 0 | 0 | LINK | 0:00:17 |  |
| 1458 | Encelydus | 0 | 0 | 0 | LINK | 0:00:06 |  |
| 1459 | Saturn Minor Satellites | 0 | 0 | 0 | LINK | 0:00:04 |  |
| 1460 | Saturn | 1 | 0.95 | 1 | LINK | 0:00:03 |  |
| 1461 | Planet | 11 | 0.91 | 1 | LINK | 0:00:02 |  |
| 1462 | Uranus | 0 | 0 | 0 | LINK | 0:00:13 | Now seems comfortable with moving away from Moon/Earth |
| 1463 | Uranus | 0 | 1 | 1 | BACK | 0:00:00 | and is using the Planet as a base |
| 1464 | Planet | 12 | 0.97 | 1 | BACK | 0:00:02 |  |
| 1465 | Saturn | 2 | 0.95 | 1 | BACK | 0:00:01 |  |
| 1466 | Saturn Minor Satellites | 0 | 0.94 | 1 | BACK | 0:00:01 |  |
| 1467 | Encelydus | 0 | 0.92 | 1 | BACK | 0:00:00 |  |
| 1468 | Iapatus | 0 | 0.9 | 1 | BACK | 0:00:01 |  |
| 1469 | Titan | 0 | 0.88 | 1 | BACK | 0:00:02 |  |
| 1470 | Saturn | 3 | 0.95 | 1 | BACK | 0:00:03 |  |
| 1471 | Janus | 0 | 0.84 | 1 | BACK | 0:00:01 |  |
| 1472 | Saturn | 4 | 0.98 | 1 | BACK | 0:00:01 |  |
| 1473 | Planet | 13 | 0.91 | 1 | BACK | 0:00:04 |  |
| 1474 | Neptune | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 1475 | Triton | 0 | 0 | 0 | LINK | 0:00:05 |  |
| 1476 | Neried | 0 | 0 | 0 | LINK | 0:00:07 |  |
| 1477 | Neried | 0 | 1 | 1 | BACK | 0:00:00 |  |
| 1478 | Triton | 0 | 0.98 | 1 | BACK | 0:00:02 |  |
| 1479 | Neptune | 0 | 0.96 | 1 | BACK | 0:00:05 |  |
| 1480 | Earth | 6 | 0.47 | 1 | LINK | 0:00:26 | Back to a familiar place |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1481 | Planet | 14 | 0.92 | 1 | LINK | 0:00:04 | |
| 1482 | Geographical Feature | 0 | 0 | 0 | LINK | 0:00:02 | Now somewhere new |
| 1483 | Planet | 15 | 0.98 | 1 | LINK | 0:00:00 | Seems to want to make sure |
| 1484 | Geographical Feature | 0 | 0.99 | 1 | LINK | 0:00:01 | that they can get back |
| 1485 | Planet | 16 | 0.98 | 1 | LINK | 0:00:01 | |
| 1486 | Geographical Feature | 1 | 0.98 | 1 | LINK | 0:00:02 | |
| 1487 | Volcano | 0 | 0 | 0 | LINK | 0:00:04 | |
| 1488 | Planet | 17 | 0.97 | 1 | LINK | 0:00:02 | |
| 1489 | Geographical Feature | 2 | 0.97 | 1 | LINK | 0:00:01 | |
| 1490 | Volcano | 0 | 0.98 | 1 | LINK | 0:00:01 | |
| 1491 | Planet | 18 | 0.97 | 1 | LINK | 0:00:01 | |
| 1492 | Geographical Feature | 3 | 0.97 | 1 | LINK | 0:00:02 | |
| 1493 | Volcano | 1 | 0.97 | 1 | LINK | 0:00:02 | |
| 1494 | Asteria | 0 | 0 | 0 | LINK | 0:00:17 | |
| 1495 | Venus | 0 | 0.49 | 1 | LINK | 0:00:02 | Finally gets to an instance |
| 1496 | Planet | 19 | 0.95 | 1 | LINK | 0:00:16 | but then goes back round again |
| 1497 | Geographical Feature | 4 | 0.95 | 1 | LINK | 0:00:01 | |
| 1498 | Volcano | 2 | 0.95 | 1 | LINK | 0:00:02 | |
| 1499 | Versuvius | 0 | 0 | 0 | LINK | 0:02:18 | longest read for a while |
| 1500 | Earth | 7 | 0.8 | 1 | LINK | 0:00:02 | |
| 1501 | Planet | 20 | 0.95 | 1 | LINK | 0:00:01 | |
| 1502 | Geographical Feature | 5 | 0.95 | 1 | LINK | 0:00:01 | Using geo as a landmark? |
| 1503 | Volcano | 3 | 0.95 | 1 | LINK | 0:00:01 | |
| 1504 | Mount Etna | 0 | 0 | 0 | LINK | 0:00:32 | |
| 1505 | Earth | 8 | 0.95 | 1 | LINK | 0:00:01 | |
| 1506 | Planet | 21 | 0.95 | 1 | LINK | 0:00:02 | |
| 1507 | Geographical Feature | 6 | 0.95 | 1 | LINK | 0:00:01 | |
| 1508 | Volcano | 4 | 0.95 | 1 | LINK | 0:00:02 | |
| 1509 | Mount Saint Helens | 0 | 0 | 0 | LINK | 0:00:56 | |
| 1510 | Earth | 9 | 0.95 | 1 | LINK | 0:00:01 | |
| 1511 | Planet | 22 | 0.95 | 1 | LINK | 0:00:02 | |
| 1512 | Geographical Feature | 7 | 0.95 | 1 | LINK | 0:00:01 | |
| 1513 | Volcano | 5 | 0.95 | 1 | LINK | 0:00:02 | |
| 1514 | Mount Rainier | 0 | 0 | 0 | LINK | 0:00:41 | |
| 1515 | Earth | 10 | 0.95 | 1 | LINK | 0:00:01 | |
| 1516 | Planet | 23 | 0.95 | 1 | LINK | 0:00:02 | |
| 1517 | Geographical Feature | 8 | 0.95 | 1 | LINK | 0:00:04 | |
| 1518 | Olympus Mons | 0 | 0.29 | 1 | LINK | 0:00:05 | |
| 1519 | Olympus Mons | 1 | 0.99 | 1 | BACK | 0:00:01 | |
| 1520 | Geographical Feature | 9 | 0.97 | 1 | BACK | 0:00:52 | |
| 1521 | Volcano | 6 | 0.92 | 1 | LINK | 0:00:03 | |
| 1522 | Montes Haemus | 0 | 0 | 0 | LINK | 0:02:34 | |
| 1523 | Moon | 7 | 0.15 | 1 | LINK | 0:00:02 | |
| 1524 | Montes Haemus | 0 | 0.99 | 1 | LINK | 0:00:03 | |
| 1525 | Volcano | 7 | 0.96 | 1 | LINK | 0:00:05 | |
| 1526 | Geographical Feature | 10 | 0.94 | 1 | LINK | 0:00:01 | |
| 1527 | Volcano | 8 | 0.98 | 1 | LINK | 0:00:01 | |
| 1528 | Geographical Feature | 11 | 0.98 | 1 | LINK | 0:00:01 | |
| 1529 | Canyon | 0 | 0 | 0 | LINK | 0:00:01 | Away from Volcanoes |
| 1530 | Bryce Canyon | 0 | 0 | 0 | LINK | 0:00:02 | Not very interested in reading anymore |
| 1531 | Earth | 11 | 0.84 | 1 | LINK | 0:00:01 | |
| 1532 | Planet | 24 | 0.84 | 1 | LINK | 0:00:01 | |
| 1533 | Geographical Feature | 12 | 0.95 | 1 | LINK | 0:00:01 | |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 1534 | Canyon | 0 | 0.96 | 1 | LINK | 0:00:02 |  |
| 1535 | Marianis Valles | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 1536 | Canyon | 1 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1537 | Noctis | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 1538 | Canyon | 2 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1539 | Geographical Feature | 13 | 0.94 | 1 | LINK | 0:00:00 |  |
| 1540 | Canyon | 3 | 0.98 | 1 | LINK | 0:00:05 |  |
| 1541 | Canyon | 4 | 0.99 | 1 | BACK | 0:00:00 |  |
| 1542 | Geographical Feature | 14 | 0.97 | 1 | BACK | 0:01:13 |  |
| 1543 | Volcano | 9 | 0.84 | 1 | LINK | 0:00:02 |  |
| 1544 | Geographical Feature | 15 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1545 | Canyon | 5 | 0.96 | 1 | LINK | 0:00:02 |  |
| 1546 | Geographical Feature | 16 | 0.98 | 1 | LINK | 0:00:02 | Still using the same search pattern |
| 1547 | Planet | 25 | 0.85 | 1 | LINK | 0:00:02 |  |
| 1548 | Satellite | 7 | 0 | 1 | LINK | 0:00:03 |  |
| 1549 | Planet | 26 | 0.98 | 1 | LINK | 0:00:14 |  |
| 1550 | Solar System | 4 | 0 | 1 | LINK | 0:00:02 |  |
| 1551 | Comet | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 1552 | Hally's | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 1553 | Comet | 0 | 0.99 | 1 | LINK | 0:00:01 |  |
| 1554 | Swift-Tuttle | 0 | 0 | 0 | LINK | 0:00:01 |  |
| 1555 | Comet | 1 | 0.98 | 1 | LINK | 0:00:01 |  |
| 1556 | Hyutaki | 0 | 0 | 0 | LINK | 0:00:02 |  |
| 1557 | Comet | 2 | 0.98 | 1 | LINK | 0:00:00 |  |
| 1558 | Shoemaker-Levey | 0 | 0 | 0 | LINK | 0:00:03 |  |
| 1559 | Comet | 3 | 0.98 | 1 | LINK | 0:00:02 |  |
| 1560 | Solar System | 5 | 0.9 | 1 | LINK | 0:00:04 |  |
| 1561 | Star System | 1 | 0 | 1 | LINK |  | finds way out |

## APPENDIX E

## Glossary

**Automatic Links**

Automatic links are links generated by Hypernet. Nodes indirectly connected through the semantic structure of the domain may be connected directly and automatically. The length of the pathway through the semantic network represents the strength of the link (weight). This weight may be adjusted in accordance with the student model as the student uses the system, producing *dynamic links.* Automatic links are discussed in chapter 3.

**BPR**

The Browsing Pattern Recogniser. A sub-system of the student model that is used to determine *content-less browsing patterns.* The BPR is the subject of chapter 6.

**Browsing Pattern**

See *Content-less Browsing Pattern.*

**Browsing Strategy**

A Browsing Strategy is a method employed by a hypermedia user for their particular information needs. Examples include "Scanning" and "Exploring". A browsing strategy induces the student to generate a *content-less browsing pattern*. Browsing strategies are discussed in chapter 5.

**Content-based Browsing**

Content-based browsing is the student's interactions with the semantic hypermedia, in that a record may be kept of the class nodes that the student has browsed and is hence interested in. Content-based browsing s discussed in chapter 4.

**Content-less Browsing Pattern**

A content-less browsing pattern is a pattern that the student generates as they move from one node to another. Different patterns are produced by different *browsing strategies*, in that different patterns emerge as the student uses the hypermedia for a different task. Content-less browsing patterns are discussed in chapter 5.

**Class Node**

A node in the semantic hypermedia hierarchy that represents class, onto which *instance nodes* may be attached. For example, "Planet" is a class node onto which the instance nodes "Earth" and "Mars" may be attached.

**Dynamic Links**

Dynamic links are *automatic links* that have had their weight adjusted as the student uses Hypernet, in accordance to the student's *content-based browsing*. Dynamic links are discussed in chapter 4.

**FRB**

The Fuzzy Rule Base. A sub-system of the student model that is used to collect information for a population of students and form a link between the *content-less browsing pattern* and student abilities and the types of task that they are using the hypermedia for. The FRB is the subject of chapter 7.

**Instance Node**

A node in the semantic hypermedia hierarchy that represents the instance of a particular class. See *class node*.

**LTN**

Linear Tutorial Node. A LTN is invoked either when the student first uses the system or by the completion of a previous LTN. LTNs contrast with *NLTNs* in that they are sequential and are not uncovered by a student browsing the hypermedia. LTNs are discussed in chapter 9.

**NLTN**

Non-Liner Tutorial Node. These tutorial nodes may be embedded within the hypermedia for the student to uncover. For example a NLTN may be attached to the *class node* "Planet", so that it becomes active when the student encounters the class node "Planet". See *LTN*. NLTN are discussed in chapter 9.

**SER**

The Student Experience Record. A sub-system of the student model that is used to record the student's content-based browsing patterns with the semantic hypermedia. This information is utilised to infer the student's interests which in turn enables the *dynamic linking* system to tailor the links presented towards the student's interests. The SER is discussed in chapter 4.

**TS**

The Tutorial Supervisor. A sub-system of the student model that is used to grade the student into an ability level based upon the student's interactions with *tutorial nodes*. The TS is the subject of chapter 8.

**VSP**

Virtual Student Program. The VSP was used to generate test and training data for the *BPR, TS and FRB* in the absence of real student data. The VSP is discussed in Appendix B.

## References

Alexander I, 1989, Why neural computing? A personal view. In **Journal of Information Technology**, Vol 4, No 2, June 1989, pp 108-111

Armstrong R, Freitag D, Joachims T, Mitchell T, 1995, WebWatcher: A Learning Apprentice for the World Wide Web. In **1995 AAA1 Spring Symposium on Information Gathering in Distributed Environments**, Stanford March 1995. Available from www.cs.cmu.edu/~dayne

Bailey D, Thompson D, 1990, Developing neural network applications. In **AI Expert**, Vol 5, No 9, Sept 1990, pp 34-41

Barlow R, Barlow J 1989 Expert Systems and Hypertext. In **Knowledge Engineering Review 3,** pp 285-301

Bates M J, The design of browsing berrypicking techniques for the on-line search interface. In **Online Review,** Vol 13, No. 5, 1989, pp 407-431

Battiti R, Serra R 1991, Neural Networks for Intelligent Tutoring Systems. In **Artificial Neural Networks: Proceedings of ICANN-91**; North-Holland, Amsterdam, Netherlands, 1991; 2 vol. xix+1819 pp1731-1734 vol.2. Kohonen T (ed).

Beer M, Diaper D 1991  Reading and Writing Documents using header Record Expertext. In **AISB-Quarterly**, No 77, Summer 1991, p.22-28

Bell G, 1995, Knowledge Modelling and Adaptive Hypertext. In **Proceedings of the University of Nottingham Psychology Department, Post Graduate Conference**, 1995. Available from www.psyc.nott.ac.uk/~grb/papers

[DJM1]

Bench-Capon I, 1990, Academic Semantic Networks. In **Knowledge Representation ; An Approach to AI,** Academic Press; ISBN 0120864401

[DJM2]

Bergeron B, Morse A, Greenes R, 1989, A Generic Neural Network Based Tutorial Supervisor for C.A.I.; In **14th Annual Symposium on Computer Applications in Medical Care**. IEEE Publishing, pp 435-439

Bernstein M, Bolter J D, Joyce M,  Mylonas E. 1991 Architectures for Volatile Hypertext. In **Proceedings of ACM Hypertext'91**, pp 243-260

Beynon-Davies P, Tudhope D, Taylor D, Jones C, 1994, A Semantic Approach to Knowledge-Based Hypermedia Systems. In **Information and Software Technology,** 36(6), pp 323-329

Biennier F, Guivarch M, Pinon J 1990 Browsing in Hyperdocuments with the Assistance of a Neural Network. In **Hypertext: Concepts, Systems and Applications INRIA 90 Proceedings**. pp 288-297. Cambridge Press.

Bloom C P, 1995; Roadblocks to Successful ITS Authoring in Industry. In **Proceedings of AI-ED-95 Workshop on Authoring Shells for Intelligent Tutoring Systems**, Washington, DC, 16 August 1995, pp 1-6.

Bossley K, Mills D, Brown M, Harris C,1994, Neurofuzzy high-dimensional modelling. In **Taylor J G (ed), Neural Networks**, pp 297-332. Alfred Waller Ltd.

Boyle C D B, Snell J R, 1989, Intelligent Navigation for Semi-structured Hypertext Documents. In **UK Hypertext** 1989, pp 28-42.

**[DJM3]**

Brusilovsky P, Pesin L 1994 ISIS-Tutor: An Adaptive Hypertext Learning Environment. In **JCKBSE'94, Japanese Symposium on knowledge-based software engineering**, Pereslavl-Zalesski, Russia, May 10-13, 1994, pp 83-87.

Brusilovsky P. 1995, Integrating Hypermedia and Intelligent Tutoring Technologies: From Systems to Authoring Tools. In **Proceedings of AI-ED-95 Workshop on Authoring Shells for Intelligent Tutoring Systems**, Washington, DC, 16 August 1995, pp 1-8.

Brusilovsky P 1996, Adaptive Hypermedia: an Attempt to Analyze and Generalize. In **Multimedia, Hypermedia and Virtual Reality, Lecture Notes in Computer Science, Vol 1077**, pp 288-304, Springer-Verlag.

Brusilovsky P, Swartz E, Weber G 1996a, A Tool for Developing Adaptive Electronic Textbooks on the World Wide Web, In **Proceedings of WebNet'96 – World Conference of the Web Society,** June 1996, Boston MA, pp 64-69, AACE 1996.[DJM4]

Bush V, 1945, As We May Think. In **The Atlantic Monthly**, July 1945.**[DJM5]**

Callaghan M and Hand C, 1996, Presentation and Representation of Implicit Knowledge in the World Wide Web. In **Workshop on Knowledge Media for Improving Organizational Expertise - Impacts of new methods and enabling technologies**, at International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland, 30-31 October 1996. Available from www.cms.dmu.ac.uk/~cph/Publications.

Canter D, Rivers R, Storrs G, 1985, Characterising User Navigation through Complex Data Structures. In **Behaviour and Information Technology**; Volume 4, Issues 2, pp 93-102, 1985 .

Carbonell J R, 1970, AI in CAI: The Artificial Intelligence Approach to Computer Based Instruction. In **International Journal of Man-Machine Studies**, 1970, No 12, pp 259-282.

Cartledge L, Pitkow J, Characterising Browsing Strategies in the World Wide Web. In **Computer Networks and Open Systems**, Vol 27, No 6, April 1995; p.1065-73.

Casti J, 1998, Easy Does It. In **The New Scientist**, 9 May 1998, pp 44-47, Reed Business Information.

Caudill M, 1988, Network paradigm selection guidelines for application development. In **Proceedings of the Fourth Annual Artificial Intelligence and Advanced Computer Technology Conference.** Tower Conference Management, Glen Ellyn, IL, USA, 1988, 540 pp 298-302.

Chen C H 1996, Fuzzy logic and neural network handbook, McGraw-Hill.

Chignell M, Nordhausen B, Valdez F, Waterworth J, 1991, The HEFTI Model of Text to Hypertext Conversion. In **Hypermedia**, Vol.3, No.3, 1991, pp 187-205.

Chiu C, Norcio A F, Petrucci K E 1991, Using Neural Networks and Expert Systems to Model Users in an OO Environment, In **Proceedings of the 1991 IEEE Conference on Systems, Man and Cybernetics,** 1991, Vol 3, pp 1943-1948, IEEE Publishing.

Cichocki A, Unbehauen R, 1993, Neural Networks for Optimisation and Signal Processing. John Wiley & Son Ltd. ISBN: 0471930105

Conklin J 1987, Hypertext: An Introduction and Survey. In **IEEE Computer,** September 1987, pp 17-41.

Debevec M, Rajko S, Donlagic D, 1994, Adaptive bar implementation and ergonomics. In **Infomatica: Journal of Computing and Infomatics** 18, pp 357-366.

DeBra P, Calvi L, 1998, AHA: a Generic Adaptive Hypermedia System. In **Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Hypertext'98**, Pittsburgh, USA 1998, Web-base proceedings available from http://wwwis.win.tue.nl/ah98/contents.html.

Denning P, 1999, Talking back to the machine: Association for Computing Machinery. Springer Verlag; ISBN 0387984135

Diamantaras K I, Kung S Y, 1996, Principal Component Neural Networks, John Wiley & Sons, ISBN: 0471054364

Dillon A, 1990  Designing the human computer interface to hypermedia applications. In **Hypermedia for Learning**, D.Jonassen (ed), pp 57-65. Springer-Verlag.

Dillon A, Gabbard R, 1998, Hypermedia as an Educational Technology. In **Review of Educational Research**; Vol. 68, No.3, pp. 322-349.

Elsom-Cook M, 1989, Guided discovery tutoring and bounded user modelling. In **Artificial Intelligence and Human Learning**; J. Self (ed), pp 65-170. Chapman & Hall. ISBN 041230130X.

Frost R A, 1982, Binary-Relational Storage Structures. In **The Computer Journal**. 25(3), pp 358-367.

Gallant S I, 1988,  Connectionist Expert Systems. In **Communications of the ACM**, Volume 31 Number 2, pp  152-169, 1988.

Gurney K 1997, An Introduction to Neural Networks, UCL Press; ISBN: 1857286731

Hagan M T, Bemurth H, Beale M, 1996, Neural Network Design, Pws Publishing Co; ISBN: 0534943322

Halaz F G, Moran T P, Trigg T H, 19 1987, NoteCards in a nutshell. In **SIGCHI-Bulletin**, Special Issue, 1987, pp 45-52

Hammond N, 1993, Learning with Hypertext: Problems, principals and prospects. In **HYPERTEXT a psychological perspective**, McKnight C, Dillon A, Richardson J (eds), pp 51-70. Ellis Horwood Ltd. ASIN: 0134416430.

Hammwohner R, Kuhlen R, 1994, Semantic Control of Open Hypertext Systems by Typed Objects. In **Journal of Information Science**, Vol 20, No 3, pp 175-184, 1994.

Hartley J R, 1993, Interacting with multimedia. In **University Computing** No. 15, pp 129-36 ,1993.

Hebb D O, 1949, The Organisation of Behaviour, Wiley, New York; ASIN: 0471367273

Henze N, Wolfgang N, Wolpers N, Modeling Constructivist Teaching Functionality and Structure in the KBS Hyperbook System. In **Proceedings of Computer Suppoort for Collaborative Learning (CSCL) 1999.** Pp223-231.

Higgins C, Goodman R, 1992, Learning fuzzy rule-based neural networks for function approximation. In **Proceedings of the International Joint Conference on Neural Networks, Baltimore,** Vol 1, pp 251-256.

Hinton G E, Sejnowski T J, 1986, Learning and Relearning in Boltzman Machines. In **Parallel Distributed Processing**, ed. D E Rumelhart, J L McClelland, and the PDP Research Group, pp282-317. Cambridge, MA: MIT Press.

Holmes B, 1999, Beyond Words, In **The New Scientist,** No.2194, July 10, 1999, pp 32-37, Reed Business Information.

Honey P, Mumford A, 1992, The Manual of Learning Styles, 3rd Edition, Maidenhead.

Hook K, Karlgren J, Waern A, Dahlback N, Jansson C, Karlgren H, Lemaire B, 1996, A glass box approach to adaptive hypermedia. In **User Models and User Adated Interaction,** Vol 6, No 2-3**,** pp157-84, July 1996.

Hopfield J, 1982, Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In **Proceedings of the National Academy of Sciences USA,** Vol 79 No 8, pp 2554-2558, 1982.

Jacobson M, 1994, Issues in hypertext and hypermedia research: Towards a framework for linking theory-to-design. In **Journal of Educational Multimedia and Hypermedia**, Vol 3, No 2, pp 141-154.

Jang J -S R ,1997, Fuzzy Inference Systems. In **Neuro-Fuzzy and Soft Computing**, Jang J -S R, Sun C -T, Mizutani E eds), pp47-72, Prentice-Hall International; ISBN: 0132610663.

Jang J -S R, Sun C -T, Mizutani E (eds),1997, Neuro-Fuzzy and Soft Computing; Prentice-Hall International; ISBN: 0132610663.

Jennings A, Higuchi H, 1993, A User Neural Network Model for Personal News Service. In **Australian telecommunications Research**, Vol 27, No 1, 1993, pp 1-12.

Jonassen D H, Wang S, 1993, Acquiring Structural Knowledge from Semantically Structured Hypertext. In **Journal of Computer-Based Instruction**; Vol 20 No. 1 pp 1-8, 1993.

Jonassen D, 1989, Semantic networking approaches to structuring hypertext. In **Proceedings of Hypertext II, University of York,** pp 78-85, 1989.

Jonassen D, Grabinger R S, 1990, Problems and issues in designing hypertext/hypermedia for learning. In **Hypermedia for Learning**, pp 3-25, D.Jonassen (ed). Springer-Verlag.

Keiser G E, 1990, AI Techniques in software engineering. In **Knowledge Engineering: Applications**, ed A. Hojjat, pp 127-135. McGraw-Hill Publishing, 1990.

Kibby M, Mayes T, 1989, Towards Intelligent Hypertext. In **Hypertext theory into practice**. Pp 138-144. Blackwell Scientific.

Kibby M, Mayes T, 1990, Learning about Learning from Hypertext. In **Hypermedia for Learning**; D.Jonassen (ed), pp 135-143. Springer-Verlag.

Kinshuk, Patel A, 1997, A Conceptual Framework for Internet based Intelligent Tutoring Systems. In **Knowledge Transfer** (Volume II), pp 117-124, Behrooz A (ed); Pace, London.

Kobsa A, Müller D, Nill A, 1994, KN-AHS: An adaptive hypertext based on an information relevance model. In **4th International Conference on User Modelling,** UM-96, Hyannis, MA, pp 31-36**.**

Kohonen T, 1989, Self-Organisation and Associative Memory. Springer-Verlag; ISBN: 3540620176.[DJM6]

Kosko B, 1996, Fuzzy thinking: the new science of fuzzy logic, Hyperion; ISBN: 078688021X.

Kosko B, 1997, Fuzzy Engineering, Prentice Hall Press; ISBN: 0131249916

Kruse R, Gebhardt J, Klawonn F, 1994, Foundations of Fuzzy Systems. John Wiley and Sons; ISBN: 047194243X.

Linard M, Zeiliger R, 1995, Designing a Navigational Support for an Educational Software. In **Proceedings of EWHCI'95, Moscow, Russia,** Gornostan, Unger (eds), pp 63-78, Springer Berlin.

Littleford A, 1991, Artificial Intelligence and Hypermedia. In **Hypertext/Hypermedia Handbook**, Berk G, Deblin J (eds), McGraw-Hill, ASIN: 0070166226.[DJM7]

Lowe D, Hall E, 1999, Hypermedia and The Web: An Engineering Approach, Wiley; ISBN 0471983128.

Major N; 1995; How Generic can Authoring Shells Become? In **Integrating Hypermedia and Intelligent Tutoring Technologies: From Systems to Authoring Tools. Proceedings of AI-ED-95 Workshop on Authoring Shells for Intelligent Tutoring Systems**, Washington, DC 1995, pp 75-82.

Marchionini G, 1989, Information-seeking strategies of novices using a full-text electronic encyclopædia. In**Journal of American Society for Information Science,** Vol 40, No 1,Jan 1989, pp.54-66.

Maren A J, Harston C T, Pap R M, 1990 The Handbook of Neural Computing Applications, Academic Press; ISBN: 0124712606.

Masters T, 1993, Practical Neural Network Recipes in C++, Academic Press; ISBN: 0124790402.

Mathé N, Chen J, 1994, A user-centred approach to adaptive hypertext based on an information relevance model. In **4th International Conference on User Modelling,** Hyannis MA, pp 107-114.

Maurer H, 1996, Hyper-G now Hyperwave: The Next Generation Web Solution. Addison-Wesley; ISBN 0201403463.

McAleese R, 1993 Navigation and Browsing in Hypertext. In **Hypertext Theory into Practice**; R.McAleese (ed). pp 6-44, Ablex Pub Corp; ISBN: 0893915750.

McAleese R, 1990 Concepts as hypertext nodes: The ability to learn while navigating. In **Hypermedia for Learning**; D.Jonassen (ed), pp 97-115, Springer-Verlag.

McAleese R, 1998 The Knowledge Arena as an Extension to the Concept Map: Reflection in Action; In **Interactive Learning Environments,** Vol. 6, No. x. (Available from www.icbi.hw.ac.uk/~ray.

Micarelli A,  Sciarrone F, 1996 A case-based system for adaptive hypermedia navigation. In **Advances in Case-Based Reasoning Lecture Notes in Artificial Intelligence**, I. Smith and B. Faltings (eds.): pp. 266-279Berlin: Springer-Verlag.

Micarelli A, Sciarrone F, 1996b, A Case-Based Toolbox for Guided Hypermedia Navigation. In **Proceedings of the 5th International Conference on User Modelling, Kailua-Kona, Hawaii.** pp 129-136.

Micarelli A, Sciarrone F, 1998, A Case-Based Approach to Usr Modelling. In **Proceedings  of EWCBR-98 in Dublin,** pp 77-93**.**

Midouser D, Nachmias R, Oren A, Lahav O, 1999, Web Based Learning Environments (WBLE): Current state and emerging trends. In **Proceedings of Ed-Media'99,** Seattle 1999, pp 753- 758.

Minsky M, Papert S, 1969, Perceptrons: An Introduction to Computational Geometry; Boston: MIT Press; ISBN: 0262631113.

Moody G, 1998, A New Dawn. In **The New Scientist**, No 2136, 30 May 1998, pp34-37, Reed Business Information.

Moore D J, Hobbs D J, Mullier D, Bell C, 1997, Interaction Paradigms with Educational Hypermedia. In **Proceedings of Euromicro, Budapest, Hungary 1997.** pp.65-71.

Mumford A, 1995, Putting Learning Styles to Work: An Integrated Approach. **In Industrial and Commercial Training**, Vol 27, No 8, pp 28-35.

Pereira D C, de Oliveira A, Vaz J C G, 1991, Hypermedia and ITS. **In Intelligent Systems in Education**. Micarelli, Belastra (eds), pp 207-223.

Preece J, Rogers Y, Sharp H, Benton D, 1993, Human Computer Interaction, Addison-Wesley; ISBN: 0201627698.

Rada R, 1990, Intelligent Hypertext. In **From Text to Hypertext**, McGraw-Hill; ISBN: 0077074017.

Reder L, 1985, Techniques Available to Author, Teacher, and Reader to Improve Retention of Main Ideas of a Chapter. **In Thinking and Learning Skills, Volume 2**, Chipman S F (ed), pp 37-64.

Rich E, Knight K, 1991 Artificial Intelligence, second edition. McGraw-Hill.; ISBN: 0070522634.

Ridgeway J, 1989 Of course ICAI is impossible...worse though, it might be seditious. In **Artificial Intelligence and Human Learning**; J.Self (ed), pp 28-48. Chapman and Hall.; ISBN: 041230130X.

Robotham D, 1995, Self-Directed Learning: The Ultimate Learning Style? In **Journal of European Industrial Training**, Vol. 19,No. 7, pp 3-7.

Rosenblatt F, 1958, The Perceptron: A probabilistic mode; for information storage and organisation in the brain, In **Psychological Review**, Vol. 65, pp. 386-408 .

Rosenblatt F, 1962, Principles of Neuro Dynamics: Perceptrons and the Theory of Brain Mechanisms. Washington, D.C. Spartan Books.

Rumelhart D E, Norman D H,  1978; Accretion, tuning and restructuring: three modes of learning. In **Semantic Factors in Cognition**; Cotton J W, Klatzky R (eds); Lawrence Erlbaum.

Rumelheart D E, McClelland J L, 1986, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1, Cambridge, MA: MIT Press ; ISBN: 026268053X

Schroeder M, 1992, Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise, W.H. Freeman; ISBN: 0716723573.

Schwarz E, Brusilovsky P, Weber G, 1996, World-wide intelligent textbooks. In **Proceedings of ED-MEDIA'96 - World Conference on Educational Multimedia and Hypermedia**, Boston MA.  AACE, pp. 302-307.

Sejnowski T J, Rosenberg C R, 1987, Parallel networks that learn to pronounce english text. In **Complex Systems,** Vol 1**.** no.1; Feb. 1987, p.145-68.

Skapura D, 1996, Building Neural Networks, Addison-Wesley; ISBN 0201539217.

Sleeman D H, Brown J S (eds) 1982 Intelligent Tutoring Systems. Academic Press; ISBN: 0126486816.

Spohrer J, Vronay D, Kleiman R, 1991 Authoring Intelligent Multimedia Applications. In **1991 IEEE International Conference on Systems, Man and Cybernetics**, Vol 3, p.1751-1756.

Tait K, 1998, Replacing lectures with multimedia CBL: Student attitudes and reactions. In **Instructional Science**, No. 26, pp409-438

Theng Y, Thimbleby H, 1998, Addressing Design and Usability Issues in Hypertext and on the World Wide Web by Re-Examining the “Lost in Hyperspace” Problem. In **Journal of Universal Computer Science,** vol. 4, no. 11, pp 839-855.

Trigg R, 1988, Guided Tours and Tabletops. In **Tools for Communicating in a Hypertext Environment**, ACM Trans. on Office Information Systems, Vol 6, No 4, Oct 1988, pp398-414.

Tsutsui S, 1991, Knowledge Based versus Neurocomputing. In **1991 IEEE International Conference on Systems, Man and Cybernetics**, Vol 3, 1821-1826.

Tudhope D, Taylor C, 1997, Navigation via Similarity: Automatic Linking Based on Semantic Closeness. In **Information Processing & Management,** Vol. 33, No. 2, pp 233-247.

Tunbridge N, 1999, The Human Touch**.** In **New Scientist No 2170,** pp 34-37, Reed Business International.
(examples cited www.inxight.com, www.cartia.com)

Von Altrock C,1996, Fuzzy Logic and NeuroFuzzy Appliances. In **Proceedings of the Fifth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE '96** (Cat. No.96CH35998). IEEE, New York, NY, USA; 1996; 3 vol. (xxxi+2214+ICNN 16) pp.2091-2093.

Widrow B, Hoff M E 1960, Adaptive switching circuits. In **1960 TRE WESCON Convention Record**, New York: IRE Part 4, pp. 96-104.

Woodhead G, 1991 Hypertext and Hypermedia. Sigma; ASIN: 0201544423.

Woods P, Warren J, 1995, Rapid Prototyping of an Intelligent Tutoring System, In Proceedings of the Australian Society for Computers in Learning in Tertiary Education Conference 1995 (ASCILITE'95).

Zeiliger R, Reggers T, Peeters R, 1996, Concept-Map based Navigation in Educational Hypermedia: a Case Study, In **Proceedings of ED-MEDIA'96,** pp 714-19, AACE, Boston MA.[DJM8]

Zhao Z, O'Shea T, Fung P, 1993, Visualisation of semantic relations in hypertext systems. In **Proceedings of ED-MEDIA'96 - World Conference on Educational Multimedia and Hypermedia,** 556-64, AACE, Boston MA.

* Web links verified as working 8/1999

Page: 384
[DJM1]Chapter 9
Page: 384
[DJM2]Chapter 9
Page: 385
[DJM3]Chapter 2
Page: 386
[DJM4]Chapter 3
Page: 386
[DJM5]Chapter 2
Page: 392
[DJM6]Chapter 6
Page: 393
[DJM7]Chapter 9
Page: 400
[DJM8]Chapter 3