Recognising and responding to English article usage errors: an ICALL based approach

Susan Sentance

Ph.D.

University of Edinburgh

1993

Abstract

Artificial Intelligence techniques are increasingly being used to enhance the area of Computer-Aided Instruction. This thesis is concerned with the area of Computer-Aided Language Learning, a subset of Computer-Aided Instruction, and demonstrates how various Artificial Intelligence techniques can be incorporated into a language learning system to produce an intelligent educational tool. In this thesis, the focus is on the use of English articles, which is a subtle area of the English language with which even advanced students of English have difficulty.

This thesis describes *ArtCheck*, an Intelligent Computer-Aided Language Learning (ICALL) system which detects, analyses and responds to English article usage errors. This system has three main features: it has **knowledge** of the article usage domain; it dynamically creates a **model** of the student; and it **adapts** to the individual student. The system's **knowledge** of the domain consists of a set of article usage rules which reflect standard teaching practice. The information necessary to apply the rules is extracted at the natural language processing stage, and includes structural and contextual information. The system **models** the state of the student's knowledge at all times, in order to give informative explanations to the student about any errors which are made. It is able to generate mal-rules which account for consistent errors made by the student, using **version spaces** and the **candidate elimination algorithm**. The student model can be described as **dynamic** because the generation of mal-rules can create new parts of the student model, in response to student behaviour, which are not predetermined by the system designer. The system **responds** to individual students by giving explanations of errors which are tailored to the student's level of ability and preferred learning style. The type of explanation given is also dependent on the system's assessment of the source of the error, and reflects any mal-rule which the system believes the student to be operating with.

Thus, *ArtCheck* demonstrates the role of a **dynamic** and **adaptive** student model in an ICALL system for English article usage. It is hoped that the use of Artificial Intelligence research in language tutoring systems will lead to sophisticated intelligent systems which adapt to individual students.

I declare that this thesis has been composed by myself and that the work described in it is my own:

Susan Sentance

This thesis is dedicated to my son, Lewis

Table of Contents

List of Figures

List of Tables

Acknowledgements

In acknowledging the help and support I have received whilst carrying out the research described in this thesis, I would firstly like to thank the Science and Engineering Research Council for providing the funding for this research, under award number 89313446.

I am also indebted to my supervisors Helen Pain and Elisabet Engdahl, for offering guidance and direction, support and encouragement, and for their comments on earlier versions of the thesis. I am also grateful to Paul Brna for helpful suggestions and illuminating conversations, and for commenting on an earlier draft of the thesis.

I would like to thank various other individuals who have willingly offered their help at different stages of the research. Firstly, to Pirkko Raudaskoski of the University of Oulu for providing information about the Finnish language and the teaching of English articles in Finland, for helping with the data collection exercises, and for general support and encouragement throughout. I would also like to thank Paul McIlvenny, also of the University of Oulu, for devoting a whole English lesson to data collection for my project. I would also like to thank Alan Black and Chris Mellish for help with the natural language tools incorporated into the system *ArtCheck*.

I am grateful to all the teachers and students who have been involved in data collection and evaluation exercises. I would like to thank Joan Norlund, Pearl Lönnfors and Nanette Lindeberg of the University of Helsinki, for introducing me to their exchange students who later participated in a data collection exercise, and for valuable insights into the teaching of English articles. I would also like to thank the many students who have participated in data collection and evaluation exercises, from the University of Oulu, the University of Helsinki, and the Institute of Applied Language Studies in Edinburgh. I also appreciate the assistance of Gregorio San-Roman-Leon and Siu Wai Leung with the external evaluation.

On a personal level, I am grateful for the support and encouragement and practical help I received from my husband Paul Evans, my colleague Siani Baker, and my childminder Tricia Smith. I hope they feel it has all been worthwhile!

Chapter 1

Introduction

Artificial Intelligence techniques are increasingly being used to enhance the area of Computer-Aided Instruction. This thesis is concerned with the area of Computer-Aided Language Learning, a subset of Computer-Aided Instruction, and demonstrates how various Artificial Intelligence techniques can be incorporated into a language learning system to produce an intelligent educational tool.

Computer-Aided Language Learning (CALL) has in the past consisted of rigid and inflexible programs which have tended to promote learning of a language through repetition (Pusack, 1983). Now, however, more powerful and adaptable programs can be developed using new research and technology, and Artificial Intelligence is able to contribute to these developments.

Educational computer programs can provide a one-to-one learning environment, where students can proceed through the teaching material at their own pace, and can provide the opportunity for students to practise in particular areas of the subject matter about which they are unsure. Such programs can act as a supplement to the classroom teacher, or in some cases, such as distance learning programmes, form the main body of the teaching material. The use of educational computer programs can aid teachers by saving preparation time and marking time, and can offer extra one-to-one tuition which the teacher may not have time to provide.

However, for an educational system to be of real value to students and teachers, it must be as knowledgeable and responsive to a student's needs as a human teacher would be. Systems which are built with these aims are known as **Intelligent Computer-Aided Instruction** (ICAI) systems.

ICAI systems differ from CAI programs in that they have **knowledge** of the domain which is being taught, and thus do not have to be primed with the answers to specific questions. In addition, they aim to **model** the student's knowledge, and can **adapt** to the individual level of the student. An intelligent instructional program should also give the student a certain amount of control over what is learned and how the learning takes place.

Much research is being carried out which aims to build prototypal intelligent teaching tools which will form the basis of the educational programs of the future. This thesis hopes to contribute to the research in this field by describing how research from Artificial Intelligence and second language education can be incorporated into an intelligent language learning system for English articles.

Section 1.1 will briefly overview current developments in Artificial Intelligence and Education. Section 1.2 will give the theoretical motivation for carrying out the particular piece of research which is described in this thesis. Section 1.3 will describe some of the factors which influenced the design of the system. Section 1.4 will outline the aims of the thesis. Finally, Section 1.5 will describe the structure of the thesis and the contents of the individual chapters.

1.1 Setting the scene: Artificial Intelligence and Education

There have been many developments in Artificial Intelligence (AI) and Education over the last few decades. This is illustrated by the many Intelligent Tutoring Systems (ITSs) or Intelligent Computer-Aided Instruction (ICAI) systems which have been built to illustrate advances in the field. ITSs have been built for various domains including mathematics, programming, and language learning.

Research in AI and Education has focused on a number of issues. These include improving the instructional abilities of the system, modelling the student's knowledge and misconceptions about the domain, interacting naturally with the student and giving clear explanations of errors, and collaborative learning.

Of particular interest here is how intelligent educational programs can deal with errors made by the student. An ICAI system needs to have an accurate representation of the subject matter in order to be able to detect errors in student input, and should also be able to understand, at some level, the reason for students' errors. In order to do this, the system needs to be able to have a student model which reflects a student's beliefs about the domain.

The next section will briefly outline a domain-independent structure for an ICAI system. The area of Intelligent Computer-Aided Language Learning (ICALL) will then be discussed. Finally, the terminology which will be used throughout the thesis will be explained.

1.1.1 Structure of an ICAI system

An ICAI system can take many forms, depending on the emphasis of the particular system, and the particular domain which is being taught. However, it is helpful to consider a typical architecture of an ICAI system. One such example is shown in Figure 1–1 (Brecht (Wasson) & Jones, 1988).

Figure 1–1 shows a modular system consisting of five modules, the expert, the student model, the psychologist, the instructional system and the interface. Each of these modules has a different role to play.

- **The expert**

  This component of the system contains all the knowledge about the domain. The system can then use reasoning together with this expert knowledge to solve problems or answer questions.

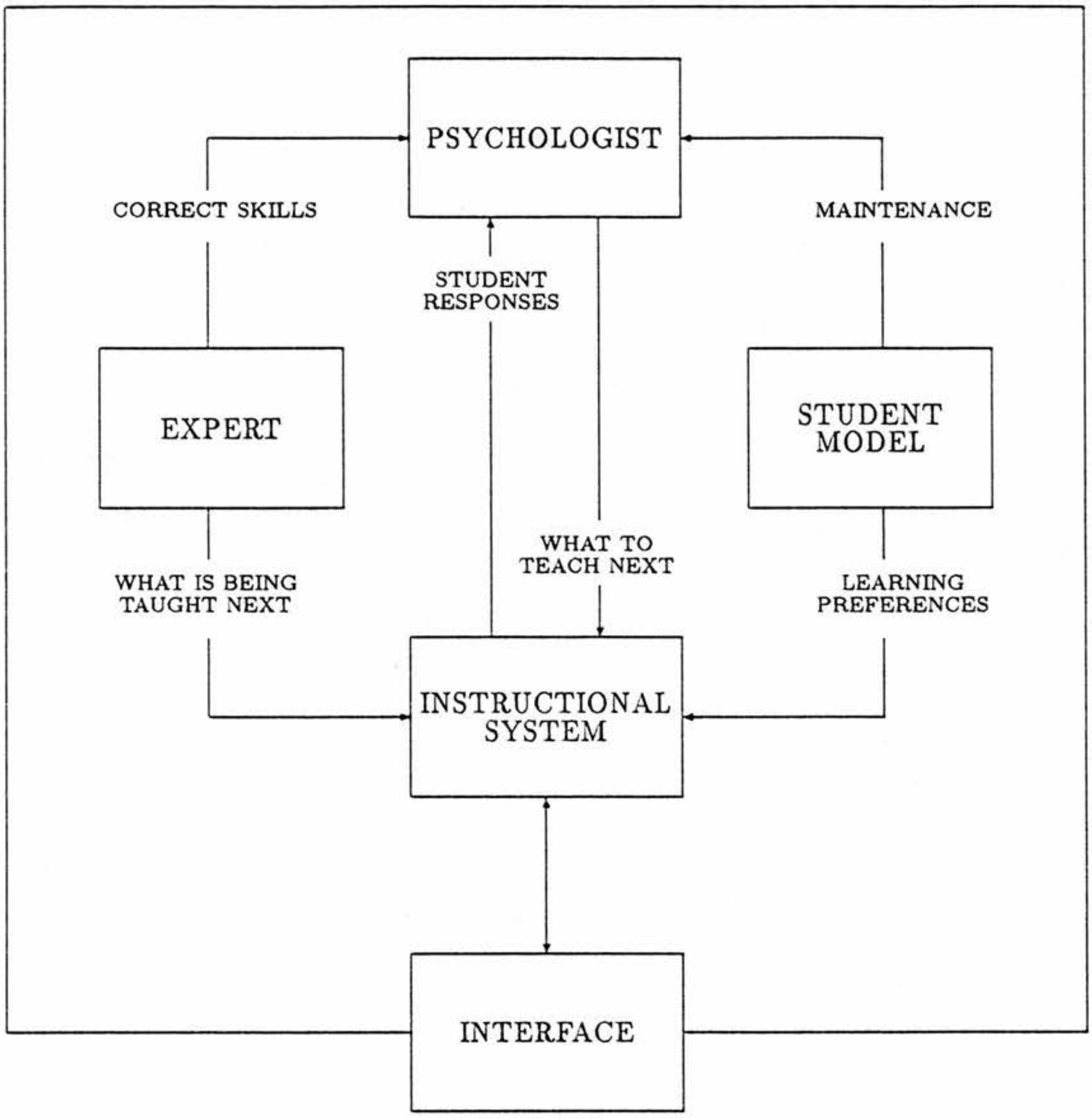


Figure 1–1: An ideal ICAI system (Brecht (Wasson) and Jones, 1988)

- **The student model**

  The student model holds the system's beliefs about the knowledge the student has of the domain, the learning preferences the student may have, and keeps a record of the student's progress while using the system.

- **The psychologist**

  The psychologist is responsible for updating the student model, diagnosing the student's errors, and deciding what the student should be taught next.

- **The instructional system**

  The instructional component of the system knows about the system's teaching strategies, contains appropriate teaching material, and structures the student's tuition.

- **The interface**

  The interface is the part of the system the student has contact with. The interface is responsible for translating the student's input into a form understandable by the rest of the system, and providing clear and coherent responses to the student input.

In individual systems, some of these components will be emphasised more than others. For example, a remediative system which only offers tuition when an error is detected might concentrate more on the psychologist and student model, and have a less comprehensive instructional system.

It was mentioned earlier that ICAI systems have been developed for a variety of different domains. The next section will discuss systems whose domain is language learning.

1.1.2 Intelligent Computer-Aided Language Learning

Intelligent Computer-Aided Language Learning (ICALL) systems are a subset of ICAI systems and are designed to teach students a language other than their own native language. This is a very important area in which to develop ICAI systems because of the types of learners which are involved. Other subject areas, such as mathematics, are primarily studied by school children or university students, for the purpose of passing examinations. Language learning differs in that, although it is studied by children for examination purposes, there are many more adult language learners who have a different motivation for learning a language. Adult language learners can fall into several different categories: those who are living in a different country and need to speak the language to improve their quality of life;

those in business who need to learn other languages to communicate with other people in the course of their work; and those who wish to learn a language as a hobby, or in some way for their own personal satisfaction. For such learners, the motivation to learn is much greater, and adult language learners are often **active** learners, seeking out opportunities to improve their ability to speak a language, and materials which will help them.

Having learners who are already motivated and looking for helpful teaching materials makes the task of the ICALL system developer much easier. It also suggests that more computer-based materials for language learners would be welcomed. In addition, many adult language learners teach themselves to learn a language, or for some other reason, spend a lot of time studying on their own, and computer-based learning materials are very useful in these cases. Many Computer-Aided Language Learning (CALL) systems are available, but most teach by repetition and do not have any knowledge of the subject they are teaching (Pusack, 1983). Therefore, the development of ICALL systems is an exciting and much-needed area of research.

There are, however, special problems associated with the development of ICALL systems which do not affect developers who choose other domains. The first is obviously that students wish to **speak** as well as write a language, and most computer systems do not offer speech understanding and generating facilities. Even with a speech processing facility, an added complication is that while a system may be able to understand an utterance in standard English (or whatever language is being taught), it may have problems understanding the ill-formed English of a language learner. More research is needed in this area, which will hopefully lead to the development of intelligent language learning systems with speech-processing facilities.

Thus, ICALL systems at present concentrate on the teaching of the written, rather than the spoken language. Here there are other problems associated with the provision of a comprehensive natural language interface, which is an area in which much research has been carried out. For an intelligent system to have the necessary knowledge of its domain, it should be able to understand the language student's input in the target language, although an alternative method of interacting with the student may be through menus or similar interfaces. As with speech understanding, the understanding of ill-formed input causes additional problems.

There is one further specific problem with regards to developing ICALL systems and that is the choice of language in which the student communicates with the system. Some systems use the target language with which to communicate, but there are problems associated with this, such as elementary students not being able to understand the instructions to use the system, and also not being able to understand the explanations which the system gives relating to errors made. The alternative is for the system to do most of the communication with the student in the student's native language. The problem then is that the system can only be used by students from a specific language background. Which of these developments is appropriate may depend on whether the target language for which the system is developed is the language of the country in which the system is used. In the case of *ArtCheck*, the system has been developed for English and communicates in English, because it is targeted mostly at students of a variety of language backgrounds who are studying in English-speaking countries.

1.1.3 Some terminology

This section will briefly clarify some of the terminology and standards to be used throughout this thesis.

- The terms **Intelligent Tutoring System(ITS)** and **Intelligent Computer-Aided Instruction(ICAI) system** will be regarded as synonymous for the purpose of this thesis, although systems will be most commonly described as ICAI systems.

- **Intelligent Computer-Aided Language Learning (ICALL)** is regarded as a subset of **Computer-Aided Language Learning (CALL)**. ICALL systems are those which have knowledge of the area of language which is being taught, or demonstrate intelligent behaviour in some other way.

- The term **user model** describes a structure which holds the system's beliefs of the state of the user's knowledge, goals, or plans. Expert systems usually have a user model. ICAI systems are thought of as having a **student model** or a **learner model**, as the user of the system is a student or learner. A student model contains the level of expertise of the student, and perhaps some information about how the student prefers to be taught. The term **student model** is preferred over learner model in this thesis, though these terms are synonymous. User models and student models are not synonymous terms, but when considering the contents and function of a student model, it is essential to consider previous related work in user modelling. For example, Chapter 2 discusses some of the work on user modelling which is related to this thesis. However, when the system *ArtCheck* is described, the term **student model** will always be used.

- For the sake of consistency, all human learners and teachers referred to throughout this thesis will be described as *she*.

- This thesis contains many examples of sentences in English, either produced by language students, or exemplifying techniques incorporated in *ArtCheck*. Such examples will be italicised, and incorrect sentences marked with an asterisk.

1.2 Motivation for the thesis

This section will discuss the motivation for carrying out the piece of research described in this thesis. It has already been described how language learning is a useful domain for an ICAI system. Some ICALL systems have been developed to concentrate on one particular grammatical area, while others have a wider scope. This research concentrates on one particular area of the English language, that is, English article usage. Section 1.2.1 will discuss the reasons for using this particular domain.

The system which has been implemented can be described as a **remediative tool**, in that it concentrates on the errors which students make when using English articles. Section 1.2.2 discusses the motivating factors behind this research.

1.2.1 The domain of English article usage

The importance of intelligent computer-aided systems for the language learning domain was discussed in Section 1.1.2. Article usage is a subtle area of the English language which learners often have difficulties with. It has been seen that learners who have an equivalent article category in their native language do not have as many difficulties as those whose native language does not include any equivalent to the article (Kellerman, 1984; Oller & Redding, 1971). Learners who have an equivalent category to the article in their language are therefore making use of **positive transfer**[1] in using the English article. This indicates that rather than negative transfer or interference, it is a **lack of positive transfer** which gives language learners from non-article bearing languages their problems.

Therefore, it was decided that the article usage of learners whose language background does not have an equivalent to the article would be an interesting area to use as a domain. Another reason for this choice of domain was that no attempt has been made to carry out a computational analysis of this domain, though there have been many attempts to determine a comprehensive set of article rules, as recently as (Kurup *et al*, 1992). Kurup *et al* describe a tutoring system for article usage, but in this system the article usage rules are not applied by the system itself. One of the aims of the system *ArtCheck*, as described in this thesis, was to implement the article usage rules by maximising the information available to the system as a by-product of the natural language interface. Even though a purely computational analysis may not be able to be an infallible predictor of article

---

[1]Transfer is the use of structures from one language in another language. Negative transfer is when this leads to an error, and positive transfer when it gives the correct result.

usage, it was thought that, using a combination of structural and contextual information, the system could have knowledge of many of the common usages of articles. In order to have this knowledge of the domain, the system would have to be able to process the natural language input of the student, and thus, this would be a further requirement of the system.

1.2.2 Recognising and responding to errors

Student modelling is an area of AI and Education which has provoked a considerable amount of research interest. The student's knowledge of the domain can be modelled by observing the student's behaviour, including correct and incorrect answers to questions. When the student gives an incorrect answer to a question posed by the system, it may be assumed that the student has some incorrect belief about the domain. In this case, the system has to firstly decide what the correct belief would be, and then what is the incorrect belief that the student has about the domain. Modelling the lack of a correct belief can be easily achieved using an overlay model. However, ascertaining an incorrect belief is more difficult to model, and much research has been done in this area. This will be discussed in more depth in Chapter 2. The bug library (Brown & Burton, 1978) has been suggested as a solution to the problem, whereby typical misconceptions about the domain are retained and the student's errors matched up against them, in order to ascertain the student's incorrect belief. However, with this method, the student model is **static** with fixed pre-determined limits.

In contrast, the **generation** of the analyses of errors on-line creates a **dynamic** model. It was decided to implement such a dynamic model for the article usage domain. The advantages of this would be that the system would cope with any unanticipated article usage errors, and a time-consuming empirical analysis is avoided. Language learners are all different and it is not necessarily an easy task to attempt to characterise all possible types of errors in advance. Generative student models have not been extensively implemented, so one of the aims of this project was to implement this type of model within the article usage domain. In order for the system to **learn** a rule which accounts for the student's errors, it is necessary to use machine learning techniques, and the implementation of these is discussed in Chapter 6.

Another motivating factor for this research was the need to provide informative and individualised feedback to students about their errors. The student model and the explanation facility can work together to this end, as the generation of a tailored response utilises the detailed information held about the student in the student model. There are many ways in which an explanation can be tailored to a student, the most common of which is in terms of the student's level of expertise regarding the domain knowledge. In developing educational materials it is necessary to consult the subject experts, not just for their knowledge of the domain, but also for their knowledge of suitable teaching and explanation strategies. In the case of second language learning, some recent research has been carried out as to how the student's awareness and good use of learning strategies can facilitate language learning. It was therefore decided that this would be an interesting area to develop within an ICALL system, even if only to a limited extent.

1.2.3 Motivation: a summary

In summary, there were several different motivating factors behind the research project described in this thesis. Firstly, the article usage domain is an area of difficulty for students, so an ICALL system for this domain would be a useful tool. Secondly, a computational analysis of this domain has not been attempted before. Thirdly, the generation of mal-rules can give analyses of unanticipated student errors, and this was to be attempted for the article usage domain. Finally, explanations can be tailored by exploiting the student model, and research into the role of learning strategies in second language learning suggested a particular way, in addition to the student's expertise about the domain, in which the feedback to the student could be further individualised.

1.3 Design criteria

The previous section described **why** the research project was carried out. Having decided what to do, the next stage was to decide **how** this was to be accomplished. The methods and techniques used during the various stages of the project will be described in detail in the appropriate chapters. In this section, the design decisions which were made prior to the development of the system will be discussed.

- **A remediative system**

  It was discussed in Section 1.1.1 that one of the components of an ICAI system is the instructional system. In *ArtCheck*, it was expected that the students who were to use the system would vary in the exposure and teaching they had had on article usage. Therefore, the system was designed as a remediative system, whereby the focus of the system is on the student's errors, and with less actual instruction. This means that the system teaches a rule when the student makes a mistake involving that particular rule, rather than having set lessons on article usage. The role of *ArtCheck* was to fill in the gaps in the student's knowledge, and highlight areas where the student had some difficulty.

- **Construction of the expert model**

  In an ICALL system for article usage, the expert model contains the rules necessary to predict correct article usage. In order to determine what these rules are, it is necessary to either consult an expert source, and collect and analyse some data. For this project, English text books and grammar books were taken as the expert source, and the article usage rule base was built up using these. In addition, some data collection was carried out to confirm the findings from the expert sources.

- **Content of the student model**

The student model can contain various pieces of information about the student. Only information which can be **used**, for example, by the explanation facility, should be held (Self, 1990). Therefore, in *ArtCheck*, the contents of the student model were determined by how the explanation was to be tailored. The explanation was to be tailored according to the level of expertise of the student, the student's learning strategy, and the source of the error. Thus, information which would enable this to happen was to be elicited from the student, or determined as the system was running, and then retained in the student model.

- **Eliciting errors from the student**

It was discussed above that *ArtCheck* was designed primarily as a remediative system. Therefore, one of the aspects of the design of the tutorial component was how errors were to be elicited from the student. The system had to be able to understand natural language, and to use the information gained as a result of natural language processing to determine correct article usage. One way of eliciting article usage errors was just to allow students to type sentences into the system. To help them in this task, it was decided that the system would suggest a number of simple topics which they could write about. This option involves students typing in whole sentences and having some keyboard skills, so, if this was to be problematic for students, an additional means of eliciting errors would be necessary. It was decided to also provide **fill-in-the-gap** type exercises where the student then only had to indicate the choice of article, and minimal typing skills were necessary. These fill-in-the-gap exercises were to be in the form of passages each including several article gaps, so that the context of each noun phrase would be apparent to the student, and this information could be taken into account. The fill-in-the-gap option in *ArtCheck* is known as the **GAP option**, and the free input option known as the **WRITE option**. These different modes of the system will be referred to throughout the text. More details of the two options are given in Appendix A.

1.4 Aims of this research

The overall aim of the **research project** was to build a computational tool which addressed a particular problem in the area of language learning, and which was intelligent to the extent that it had knowledge of the domain and used reasoning to analyse and explain the behaviour of the students who use it. The overall aim of the **thesis** is to describe how the research project was carried out and the contribution that this work has made to the field of ICALL and AI.

More specifically, this research had the following aims:

- To develop an ICALL system for English article usage which demonstrates the use of various Artificial Intelligence techniques.
- To implement a set of rules for the article usage domain.
- To demonstrate the use of candidate elimination and version spaces in the generation of mal-rules.
- To demonstrate the interaction of the explanation facility and the student model in the generation of explanations which are tailored to the student's level of ability and learning preferences.

Later sections of this thesis will return to these specific aims and attempt to establish how far they have been met.

1.5 Structure of the thesis

The remaining chapters of this thesis will be structured as follows:

- Chapter 2 will outline the related research carried out in the areas of user (and student) modelling and explanation, and pinpoint areas of research that have been used in the development of the system *ArtCheck*.

- Chapter 3 will discuss work which has been done in the area of second language learning, and some ICALL systems which have been developed.

- Chapter 4 will discuss in detail the domain of English article usage. It will describe research which has been carried out with the aim of defining the domain, and discuss the difficulties which language learners have when using English articles.

- Chapter 5 will discuss the implementation of the article usage rules.

- Chapter 6 will discuss how the system dynamically models the student, including how mal-rules are generated in *ArtCheck*.

- Chapter 7 will describe how the system uses the information held in the student model to give an explanation which is tailored to the student's level of expertise and learning preferences.

- Chapter 8 will describe the process and results of evaluating the system.

- Chapter 9 will discuss how the developed system compares with other similar systems, the contribution made by this thesis, and how the work could be developed further.

- Chapter 10 gives a short conclusion to the thesis.

Chapter 2

Background: Student modelling and Explanation

The claim of this thesis is that a dynamic and adaptive student model can be developed as part of a tutoring system for English articles to enhance the learning experience of the student using the system. Before this can be examined in detail, some background issues in the areas of student modelling and explanation will be examined, with reference to relevant work in this area. This will help to set the scene for the discussion which will follow in later chapters, and will enable the contribution of this project to be considered in the light of work related to it.

Student modelling is an area in which research interest has recently been growing. The advantage of incorporating a student modelling component into a system is that it will enable it to adapt to individual students. It will be able to give explanations to the student at a level they can understand, find out where there are misunderstandings and diagnose errors, and decide what new topics should be introduced to the student, and how best to do this effectively. In order for a system to demonstrate this behaviour, it must have a mechanism for building and maintaining a model of that student.

Some of the work described in this chapter has been carried out in the area of user modelling as opposed to student modelling. A system which has an educational aim is said to be used by students and have a student model; other systems such as expert systems have a user model. Thus, in this thesis, the difference between a student and a user model is taken as the context in which the model is developed. Therefore, as laid out in Section 1.1.3, the term **user model** will be used when it is felt that **student model** is inappropriate.

User models and student models may be required in different kinds of systems which interact with users, including intelligent tutoring systems, dialogue systems, and advisory systems, and may have different functions in different systems. One important function of the user or student modelling component is to model the domain knowledge of the user or student, and many intelligent tutoring systems have been concerned with this (Clancey, 1987; Carr, 1977; Brown & Burton, 1978; Sleeman, 1982; Sleeman, 1987).

2.1 Modelling the student's domain knowledge

2.1.1 Overlay modelling

There are several ways of modelling the domain knowledge of the student. One such approach, introduced by Carr and Goldstein (Carr & Goldstein, 1977), is the **overlay model**. In a system such as an intelligent tutoring system or expert system, the system plays the role of the teacher or expert, that is, it has available to it the knowledge of the domain. In overlay modelling, the student's knowledge is represented as a subset of the expert knowledge; thus the term overlay comes from the idea of a template which "overlays" the expert knowledge. The model is maintained by **assignment of credit**, where the system marks what pieces of information the student does and does not know.

Clancey made use of overlay modelling in his system GUIDON (Clancey, 1987), an intelligent tutoring system built to assist medical students learning from the expert system MYCIN (Shortliffe, 1976). In a GUIDON tutorial session, the student plays the role of consultant. A "case" is described to him in general terms, following which he has to ask questions and form hypotheses of his own. GUIDON compares the student's questions and hypotheses to those asked by MYCIN, and can analyse student performance on this basis, knowing at every moment what the expert program would conclude based on the evidence available to the student. GUIDON's knowledge is represented in the form of domain rules and teaching rules. Whenever the expert program successfully applies a domain rule, GUIDON has to decide if the student also knows the domain rule, and consequently updates the student model.

The disadvantage of overlay modelling is that if the student is following a different problem solving approach to the expert, or believes something which is not true of the domain, this cannot be represented in the student model, as it does not constitute a subset of the expert knowledge. Other limitations of overlay modelling are that it cannot predict what a student might know based on partial information, and does not represent the order in which students typically learn new information in a domain (Chin, 1989). Nevertheless, this approach has been used widely, as it is easy to implement and can be effective. However, for a diagnostic model, what is required is a way of modelling which can cope with incorrect knowledge acquired by the student.

2.1.2 The genetic graph

A utilisation of overlay modelling which attempted to deal with the situation where the student's knowledge deviates from expert knowledge, was the **genetic graph** (Goldstein, 1982). In this approach, the expert knowledge is represented as a genetic graph, and the student knowledge is overlaid on top. The genetic graph representation differs from GUIDON and similar systems in that instead of the expert knowledge being represented as rules and facts comprising the knowledge-base, the facts and rules are represented as the nodes of a graph and the interrelationships between them as the edges. In representing the various relationships between the rules, the graph enables student knowledge to be modelled, even where it is not in exactly the same form as the expert knowledge. This technique has been incorporated into a tutoring system called WUSOR, a system designed to teach students to play the game WUMPUS or WUMPUS-hunting. The expert module contains all the logical and probabilistic rules which the student must learn, and the WUMPUS Advisor helps by pointing out rules at certain stages of the game, and commenting on moves which the student has elected to make. The links of the graph relate rules to other rules. The links used in WUSOR are **specialisation/generalisation**, **analogy**,**deviation/correction**, and **simplification/refinement**. It is the **deviation** link which enables misconceptions to be modelled. The definitions of these links are shown in Table 2–1.

Generalisation	R^1 is a generalisation of R if R^1 is obtained by quantifying over some constant.
Specialisation	Specialisation is the inverse of generalisation.
Analogy	R^1 is analogous to R if there exists a mapping from the constants of R^1 to the constants of R.
Refinement	R^1 is a refinement of R if R^1 manipulates a subset of the data manipulated by R on the basis of some specialised properties.
Simplification	Simplification is the inverse of generalisation.
Deviation	R^1 is a deviation of R if R^1 has the same purpose as R but fails to fulfill that purpose in some circumstances.

Table 2–1: The genetic links (Goldstein, 1982)

A portion of the genetic graph in WUSOR is shown in Figure 2–1. The contribution of the genetic graph to student modelling is that while the student's knowledge is not normally a strict subset of the expert's knowledge, it may fit into a framework which can include simplified, deviated or more general versions of the expert knowledge needed to acquire competence in the skills of the game. The genetic graph method therefore gives more scope of representation than the overlays used in GUIDON. Utilisation of deviation links to link correct rules with incorrect versions of them will enable incorrect student knowledge to be modelled, which is not possible with the straightforward overlay model used in GUIDON. However, the genetic graph in WUSOR is **static**, in that the graph exists for the game before it is played and does not change. Only deviations, simplifications etc. which have been noted previously and are specifically included occur in the graph.

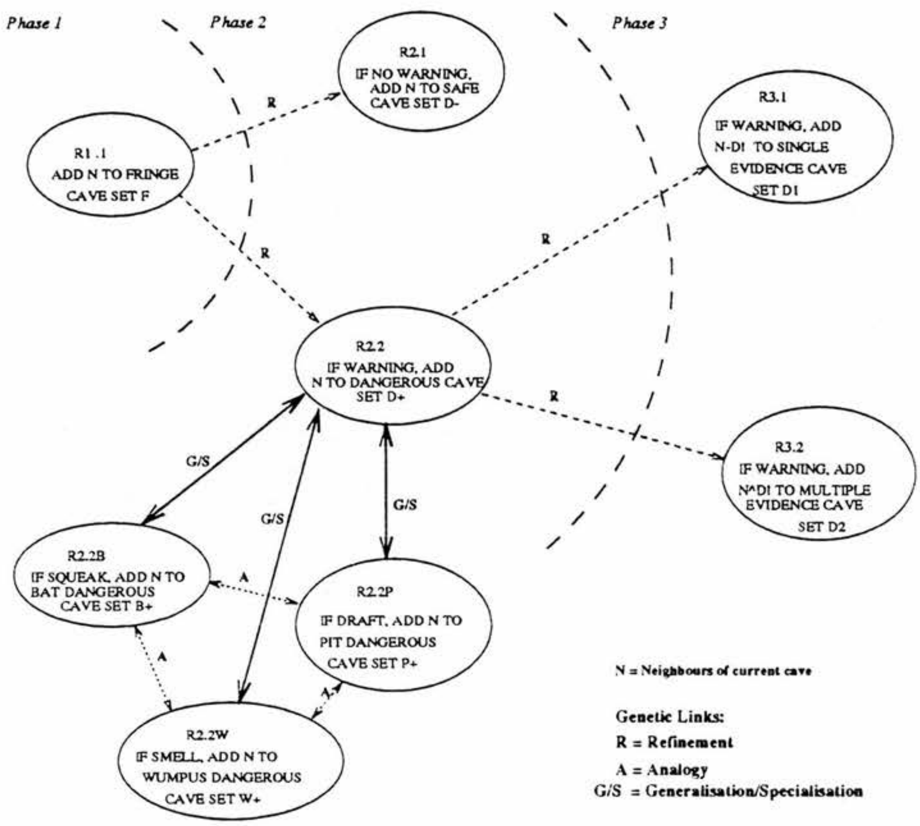


Figure 2–1: A region of the genetic graph (Goldstein, 1982)

Some more recent work on genetic graphs (Brecht (Wasson) & Jones, 1988) has shown that this technique can be transferred fairly easily to other domains, for example, the domains of subtraction and ballet, with the addition of new kinds of links to capture other relationships between rules. But, in order for any student model to be capable of fully representing student knowledge, it must be able to cope with deviations from expert knowledge which have not previously been encountered, that is, it must be a **dynamic** model. Brecht (Wasson) and Jones advocate the use of genetic graphs for dynamic modelling. They discuss an implementation which incorporates the generation of analogical links within a genetic graph (Escott, 1988) (cited in (Brecht (Wasson) & Jones, 1988)). Another way of using a genetic graph for dynamic student modelling would be to create new deviation links in response to errors made by the student. This has not previously been attempted with genetic graphs, and it is this use of genetic graphs which is described in this thesis.

2.1.3 Perturbation modelling

One of the chief functions of the student model is remediation (Self, 1987). Remediation involves a system detecting an error, analysing the error, uncovering the source of the error if possible, and giving a helpful explanation to the student. In order for this to be achieved, the system must have a means of recognising and representing the student's errors.

It is necessary at this point to clarify the distinction between terms such as **error**, **bug**, **mal-rule** and **misconception**. The use of these terms is often inconsistent in the literature, so it is necessary to define them clearly. Dillenbourg and Self distinguish between the behavioural and conceptual level of representation (Dillenbourg & Self, 1992). According to this framework, an **error** can be defined as a discrepancy between the learner's behaviour and the system's behaviour. A **bug** is a discrepancy between the system's representation of **behavioural knowledge** and the system's representation of the learner's (representation of) behavioural knowledge. Similarly, a **mal-rule** is a rule which describes this discrepancy. Finally, a **misconception** is a discrepancy between the system's representation of **conceptual knowledge** and the system's representation of the learner's conceptual knowledge. These definitions will be used in this thesis.

Attempts to model the student without using overlays have been termed **perturbation modelling**. This involves building up a model of the student which incorporates misconceptions held by the student. Several systems have been implemented which account, to some extent, for student knowledge which deviates from expert knowledge.

One way of doing this is to use a **bug library** or a list of **mal-rules** to help with the diagnosis of misconceptions. BUGGY is a system designed to help arithmetic teachers diagnose bugs (Brown & Burton, 1978). The aim of the system is to build a **diagnostic model** of the student, that is, a model of the internalised set of incorrect instructions or rules capable of duplicating a student's behaviour. BUGGY works off-line with a pre-defined arithmetic test and the student's answers. Each subtraction exercise is divided into a number of goals. For each correct method of satisfying a goal, there are a number of alternative "buggy" methods. The role of the system is to diagnose which of these buggy procedures is being used. BUGGY is successful in that it provides a detailed model of the subtraction process. A later system, DEBUGGY (Burton, 1982) generates combinations of primitive bugs and tests them against the student's answers until it comes up with a plausible diagnosis.

Leeds Modelling System (LMS) is a system which diagnoses bugs in basic algebraic skills (Sleeman, 1982). It produces a model of the user of the system, and from that can predict the student's behaviour. It considers itself to have succeeded when the student exhibits the same behaviour in respect to a set of algebra problems as the model predicts. The domain is represented in the form of production rules, as is the model of the student, and at each level of difficulty there exist rules and **mal-rules**, where a mal-rule is a deviant form of a rule. At each level there are various combinations of rules and mal-rules which the student could be using. Each of these combinations represents a possible student model, giving a large search space. The system deduces the student model both by inferring from student behaviour which rules are being used, and by the use of heuristics which eliminate functionally equivalent models. The algebra problems presented to the student are then selected to allow the system to discriminate between the remaining possible models, and eventually the system selects the correct model of the student.

There are several drawbacks to the bug library or mal-rule approach. Firstly, the system's knowledge of particular bugs has to be hand-coded into the system, which is obviously a time-consuming task and not necessarily exhaustive. Secondly, the predefined library of bugs or mal-rules means that the system cannot cope with unanticipated bugs. Finally, in the case of LMS, the search space is very large.

Payne and Squibb carried out a study of arithmetic errors of school children which challenged the mal-rules reported by Sleeman (Payne & Squibb, 1990). The students completed an algebra test, and the errors made were then analysed in terms of a large set of mal-rules, including those proposed by Sleeman. Few of Sleeman's mal-rules were observed, and many other mal-rules which had not been reported by Sleeman were noted. The three schools which took part also showed remarkably different results. In addition, the mal-rules did not seem to be used regularly by students. Generally, students who used a certain mal-rule used it on less than half of the occasions when it could be applied.

This research raises two main points. Firstly, if a library of mal-rules built up by looking at the behaviour of one set of students is not relevant for other sets of students, then the collection of mal-rules, at least for this domain, may not be a feasible exercise. Secondly, if the students do not use mal-rules consistently, then it may be that having mal-rules at all is not valid.

A further development was **repair theory** (Brown & VanLehn, 1980). Repair theory was described as a **generative theory of bugs**. The main reasoning behind this theory is that when a student is applying an incorrect procedure, she will eventually come across a point where she does not know the next step to take. This is known as an **impasse**. It is suggested that the student is then inventive and formulates a **repair** for the impasse. This theory requires the following design criteria: a representation of the correct procedural skill; a set of principles for the **deletion** of fragments (thus simulating an impasse); a set of repair heuristics; and finally, a set of **critics**. Critics are used to filter out those repairs which are psychologically invalid. The complete set of all possible bugs is the set of all valid repairs to all possible impasses, before the critics have been applied. A number of criticisms have been directed at this approach (Hennessy, 1990). These are mainly concerned with the lack of deep semantic knowledge of the domain, and the related issue of how the surface bugs can be related to the underlying misconceptions.

2.1.4 Dynamic student modelling

Student models can be categorised as **static** or **dynamic** models (see Section 2.1.2). A static student model is one whose limits are predetermined before the system is used. As a result, it cannot react to student input which is not anticipated, and a lot of time and effort has to be spent in building up the domain specific information about misconceptions, for example, in a bug catalogue. In contrast, a dynamic, or generative, student model can infer the student's misconceptions from analysing the student's behaviour, and hence can diagnose misconceptions underlying errors which have not been encountered before. Ideally, work in student modelling should be moving towards dynamic student models:

> *"The main hope for powerful ICAI systems is that from a smaller amount of initial knowledge they will be able to infer a student model from the student's response."* (Gilmore & Self, 1988, p181)

Several systems have made an attempt towards dynamic student modelling, by inferring misconceptions from observed student behaviour, and trying to generate an analysis of the student's misconception on-line. This thesis describes a system which claims to have such a dynamic student model, and is able to generate mal-rules.

A system which followed on from LMS, PIXIE (Sleeman, 1987), has a post-interactional analysis stage which processes unanticipated errors and incorporates them into the domain knowledge base. This is done by working backward from a student's incorrect answer towards the question to infer a mal-rule where there is a gap. All the mal-rules associated with a rule can be generated by systematically removing one or more of the rules' sub-steps (Sleeman, 1983). Another system which generates new parts of the student model is Automated Cognitive Modelling (ACM) (Langley *et al*, 1984). This system will be described in more detail in Section 2.1.5.

In order to generate parts of the student model, the system must be able to **learn** by observing student behaviour. This can be achieved by using **machine learning techniques**. The next section describes some of these techniques and how they have been used in the context of student modelling.

2.1.5 Machine learning techniques

Machine learning is concerned with developing and implementing computational theories of learning. In this section, three different techniques of machine learning will be described, with particular reference to how they can be or have been applied to student modelling. The techniques are: **decision trees**, **version spaces**, and **focusing**.

Decision trees

A decision tree is used in the classification of an object or concept. The nodes of the tree represent features of the object or concept, and the branches indicate the feature values. The complete classification is represented by the **path** from the root of the tree to the leaf. In machine learning, rules or concepts can be learned by using the data available to build a decision tree:

> *"A rule is expressed as a decision tree: each interior node consists of a test of an attribute with one subtree for every possible value of that attribute, and each leaf has an assigned class signaling the appropriate outcome of the classification rule."*(Quinlan, 1986, p151)

Decision trees have been used in the machine learning systems CLS (Hunt *et al*, 1966), ID3 (Quinlan, 1983), and various other related systems. The main advantage of decision trees in machine learning is that they can handle disjunctive concepts in the data (Bundy *et al*, 1985). However, they cannot be used for finding generalisations which account for data which is related.

A system which uses a form of decision, or discrimination, trees to generate production rules is the student modelling program Automated Cognitive Modelling (ACM) (Langley *et al*, 1984). ACM is based on the idea that problem solving involves a heuristic search through a problem space (Langley & Ohlsson, 1984). Inductive inference (Quinlan, 1986) is used to find a solution path through the problem space. This represents an incorrect rule used by the student. ACM has been developed for the domain of subtraction, using the already available empirical data about subtraction bugs. The aim of the system is to model the procedures involved in carrying out a subtraction task, both correct and incorrect.

The problem space in ACM consists of a set of primitive operators, and abstract condition types which are used to form the rules. In the subtraction domain, these are properties such as *greater than* and *above* which are instantiated during the modelling process. The input to the system is a set of problems and the student's answers. The conditions are used as tests to generate all possible answers to each problem, which are then compared with the student's answers. A discrimination tree is built up which represents all the possible steps in the problem. Each step which disagrees with the student's answer being modelled is marked as a negative instance, and each step which is in line with the student's answer is marked as a positive instance. The discrimination tree is then translated into a set of condition-action rules, which represent a model of the student's behaviour. Table 2–2 shows a bug which has been modelled by ACM.

---

find-difference
If you are processing *column 1*,
  and *number 1* is in *column1* and *row1*,
  and *number 2* is in *column2* and *row2*,
  and *number1* is greater than *number2*[1],
then find the difference between *number1* and *number2* ,
  and write this difference as the result for column1.

shift-column
If you are processing *column 1*,
  and you have a result for *column1*,
  and *column2* is left of *column1*,
then process *column2*.

Table 2–2: Model for the "smaller-from-larger" subtraction bug (Langley *et al*, 1984)

ACM has been implemented and run on common subtraction bugs. One of the problems, however, with this method, is how to discriminate between multiple solution paths. The use of psychologically plausible heuristics, if found, would aid in this.

Version spaces and candidate elimination

The version space technique was introduced by Mitchell in 1977 (Mitchell, 1977). A **version space** is **the set of current hypotheses** of the correct statement of a rule which predicts some fixed action. The version space has an upper and lower bound, from the maximal specific versions (MSV) of the rule, to the maximal general versions (MGV) of a rule. Within these two bounds are all the hypotheses which account for the positive and exclude the negative training instances of the rule or concept.

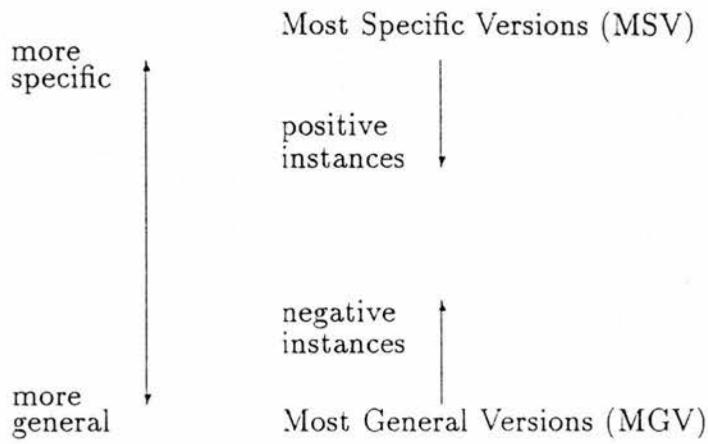


Figure 2–2: The effect of positive and negative training instances of version space boundaries (Mitchell, 1977)

Candidate elimination describes the way that the training instances narrow the version space (Mitchell, 1977). Both **generalisation**, that is, the use of positive instances to make the MSV more general, and **discrimination**, whereby negative instances are used to make the MGV more specific, are used. Therefore, as seen in Figure 2–2, the positive instances move the boundaries of the version space towards the more general rules, and the negative instances move the boundaries towards the more specific hypotheses. Candidate elimination has been implemented in LEX (Mitchell *et al*, 1983). The combination of discrimination and generalisation makes candidate elimination an efficient way of generating rules. It thus seemed a suitable technique to incorporate into the system *ArtCheck*.

Elsom-Cook developed a system IMPART which uses the idea of a lower and an upper bound on the student model (Elsom-Cook, 1988). His idea is that the state of the student's knowledge of the domain cannot be accurately defined, but only approximated by providing bounds within which the system believes it lies. In IMPART, a **set** of version spaces (rather than just one) is maintained, representing the fact that each new event may generate a completely new lower and upper bound. This differs from Mitchell's ideas in that the partial ordering of the bounds is lost, and the version space is not reduced with the addition of new data. However, in theory, Elsom-Cook's interpretation allows for a better treatment of inconsistent data.

Focusing

Focusing is similar to candidate elimination and version spaces in that positive and negative examples are used to focus on candidate hypotheses. The difference is in the way that the candidate concepts are represented (Gilmore & Self, 1988). The description space is represented in terms of features, the values of which have a structure which can be represented as a tree. The generalisation and discrimination of the features can be seen as a movement up or down the tree respectively. For example, the description feature *red* can be generalised to the value *coloured* with the inclusion of new positive instances, and similarly, the feature with the value *coloured* can be discriminated to hold the value *red*.

Gilmore and Self consider the possibility of using focusing as a learning algorithm in the context of student modelling (Gilmore & Self, 1988). The advantage of this and similar methods is that it learns incrementally. They note two particular disadvantages of this method, which obviously hold for candidate elimination as well:

(i) The algorithm as it stands can only be used for learning conjunctive concepts. It cannot be assumed that the rules in most domains can be expressed without some disjunctions.

(ii) The second disadvantage concerns the process by which the negative instances are used to discriminate, that is, make the most general versions of the candidate hypotheses more specific. The different ways of making a concept more specific mean that there may be many possible MGVs, causing a memory overload for an implemented system.

Gilmore and Self propose that focusing (or presumably, the similar candidate elimination algorithm) as a learning algorithm may have a role to play in collaborative learning systems. Earlier psychology literature suggests that it may be a psychologically credible technique (Bruner *et al*, 1956). A variant of the focusing algorithm, MULTI, (Gilmore, 1986) has been developed which overcomes the above disadvantages, by regarding negative instances as examples of alternative concepts.

In this section, we have described a selection of ways of modelling the student's knowledge of the domain, from the simple overlay model, to the application of machine learning techniques. The next section moves away from the student model's function as regards domain knowledge, and discusses some other aspects of the student which may be modelled.

2.2 Modelling other aspects of the student

Student models may be used for other functions besides modelling domain knowledge. For example, the system GRUNDY (Rich, 1979), which selects books for library users, attempts to model the **personality traits** of its users. Another system, HAM-ANS (Morik, 1985), detects and models the user's **preferences** in booking a hotel room. Other systems concentrate on the need of the user modelling component to model the **plans and goals** of the user. Examples of such systems are Carberry's TRACK system (Carberry, 1988), which infers the user's underlying task-related plan from the on-going dialogue, UC (Chin, 1989), a Unix advisory system, and PROUST(Johnson, 1986), which diagnoses errors in students' Pascal programs.

Some systems infer a user model by referring to a pre-defined list of **stereotypes**. The first system to use stereotypes was Elaine Rich's GRUNDY (Rich, 1979), which recognises such stereotypes as *Feminist*, *Intellectual*, and *Sports-Person*. The user modelling component of UC, KNOME (Chin, 1989), has a double stereotyping system. Within this, Unix users are described as being either *novice*, *be-*

ginner, *intermediate* or *expert*, and from this classification, default inferences can be made. Facts about Unix are classified as **simple**, **mundane**, **complex**, or **esoteric**, depending on the level of difficulty they present to the user. Thus, given some facts which the user knows, the system can infer the user's level of expertise, and given the user's level of expertise, KNOME can predict what other facts the user is likely to know. GUMS, a domain-independent user modelling system using default reasoning (Finin, 1989), also uses stereotypical reasoning as one means of inferring information about the user. Stereotypes in GUMS are organised in hierarchies depending on the domain being used. For example, as well as the more general stereotype **programmer**, there may be more specific stereotypes within this classification such as **novice programmer** and **Unix hacker**, which inherit some of the default properties of the more general class.

There are many other aspects of the user which can be modelled, for example the motivation of the learner (del Soldato, 1992). The reason for doing this is so that the system can have access to extra information which can be used to give more individualised explanations. As will be described in Section 3.2, language learners can have different strategies for learning a new language, and also, awareness of their learning preferences. Therefore, this would be an interesting facet of the learner to be modelled in an ICALL system.

2.3 Explanation

Explanation is an essential part of any expert system or intelligent tutoring system. It is by its ability to communicate with and explain its reasoning to a user that a working system will often be judged. Thus, it is an area of research which has generated a lot of interest in recent years. This section will describe briefly some of the major developments in explanation, and the way the user model and the explanation facility can interact.

2.3.1 Early attempts at explanation

The most basic approach to explanation is to use canned text, that is, responses previously prepared by the system designer, in response to user input. However, this is obviously not adaptive and flexible enough for the purposes of an intelligent system.

Another early approach is to somehow translate the steps taken by the expert model in reaching its conclusions and thus form an explanation giving the system's reasoning. This is the approach used in MYCIN (Shortliffe, 1976).

In their survey of explanation, Moore and Swartout (Moore & Swartout, 1988) gave these five criticisms of early work done on explanation.

- Explanations were **narrow**, that is, they could only cope with a few types of questions.
- Explanations were **inflexible**, that is, they could only be presented in one way.
- Explanations were **insensitive**, that is, they were not tailored to individual users.
- Explanations were **unresponsive**, that is, they could not answer follow-up questions or offer alternative explanations.

- Explanations were **inextensible**, that is, new strategies for explanation could not be added easily.

Taking into account the shortfalls of many early explanation facilities as outlined above, much research recently has focussed on improving the explanation capabilities of expert systems and intelligent tutoring systems. Work in explanation can be divided into two areas (Cohen *et al*, 1989): that which emphasises enhancing the form and content of the system itself, for example, (Swartout, 1981; Clancey & Letsinger, 1981; Wick & Thompson, 1992); and that which concentrates on modelling aspects of the user which can then be used to improve the explanation, for example, (McKeown, 1985; McKeown *et al*, 1985; Paris, 1988).

2.3.2 Developments in explanation: enhancing the system structure

Early attempts at explanation in expert systems like MYCIN did not involve any general model of explanation. In a subsequent system, NEOMYCIN (Clancey & Letsinger, 1981), which was developed to teach the expert knowledge contained in MYCIN, the diagnostic strategy is separated from the domain knowledge. NEOMYCIN is thus able to produce abstract and concrete explanations of its reasoning strategies, and answer *why* and *how* queries. These two queries form the foundation of nearly all explanation facilities to date (Wick & Slagle, 1989).

Another system, XPlain (Swartout, 1981), also represented the problem-solving knowledge explicitly and separately from domain knowledge. It used a domain principle and domain rationale to record the designer's rule justification by using an automatic programmer to build the expert system. Thus, it was also able to give justifications for its behaviour.

More recent work (Wick & Thompson, 1992) proposes the **decoupling** of the line of explanation from the line of reasoning. Wick and Thomson argue that the process of explanation should include the ability to reinterpret data and even find additional information supporting the new line of explanation.

An important feature of explanation generation systems is their ability to respond to the user's feedback. The Explainable Expert System (Moore & Swartout, 1988) has an explanation facility which is **reactive**, that is, it reacts to the user's follow-up questions. In this system, the intentional structure of responses is explicit. Follow-up questions are related to the current context, and this feedback and the intentional structure is used to plan responses. In addition, there are different explanation strategies for realising a given discourse goal. A **discourse goal** is a goal for a particular part of the discourse, and is stated in terms of the effect that that particular piece of discourse is intended to have on the hearer or reader.

Many explanation facilities either use canned text or some kind of template which can be filled with translated code, to create responses for their users. Such responses are often stilted and inflexible. More recently, researchers have been making use of the work done in the area of language generation. Language generation systems, for example (Mann, 1983; McDonald & Pustejovsky, 1985), can be incorporated into an explanation system to make explanations more natural and readable.

2.3.3 Developments in explanation: exploiting the user model

Another way of improving the explanation facility of a system is by making use of the information in the user or student model. The explanation can then be tailored to individual users. The manner in which this is implemented in a particular system depends on the information held about the user.

The most obvious aspect of the user to be modelled is the user's level of expertise with respect to the domain knowledge as described in Section 2.1. This can be used in the generation of an explanation in various ways. In TAILOR (Paris, 1988), descriptions are given of devices as found in texts such as encyclopediae, which are tailored to the user's level of expertise by using one of two discourse strategies. These strategies are the **constituency schema** (McKeown, 1985), for experts, and the **process description**, for novices. A mixture of these strategies can be used for users who fall between the extremes of novice and expert, and thus an *a priori* list of stereotypes is not required. In another system, developed by Wallis and Shortliffe (Wallis & Shortliffe, 1982), the user declares her level of expertise and desired level of detail as integers between 1 and 10. These figures act as upper and lower bounds on the complexity of concepts that will be included in the explanation.

Other systems tailor the explanation by modelling the user's **goals** or **plans**. One example is Advisor (McKeown *et al*, 1985), where the explanation given to the user reflects what the user actually wants to know, as represented by the inferred goals. AQUA, a Unix advisory system (Quillici *et al*, 1988) generates explanations of the user's misconceptions by referring to the user's inferred goals or plans. A misconception is diagnosed where the user's beliefs about the causes and effects of Unix commands are inconsistent with that of the system.

Explanations of the user's misconceptions obviously depend on the information held in the user model which leads the system to believe that there is a misconception. In McCoy's system, ROMPER (McCoy, 1988), the misconception corresponds to a misclassification or misattribution of an object in the domain. The explanation given depends on the type of misconception and on the highlighting of the user model. The user model is **highlighted** in the sense that several factors, including the focus of the dialogue, and what has been mentioned in the discourse already, are taken into account in determining the **active perspective** of the user model at any given time. The active perspective means that certain objects and attributes appearing in the user model are given more prominence. This method is dependent on the domain knowledge being in the form of a taxonomy, and is thus not suitable for all domains.

It has thus been seen that the interaction between the user model and the explanation component plays a important role in the generation of individualised explanations (Kass & Finin, 1988). In *ArtCheck*, the implemented system which is described in this thesis, the explanation facility makes use of all the information retained in the user model when generating the appropriate explanation.

2.4 Conclusion

This chapter has described some of the contributions to the area of student modelling and explanation, in so far as they relate to the research under discussion in this thesis. Section 2.1 described how the field of student modelling has mainly concentrated on better modelling of the student's knowledge of the domain. Of particular importance in this area is the modelling of the student's **misconceptions** about the domain. One method used was a bug library or a list of mal-rules, which included all the bugs or misconceptions about the domain knowledge which the system might expect the student to have. This approach has two main drawbacks. Firstly, the list of mal-rules or bugs can never be exhaustive, and there is always the possibility that the student may have errors that are not included. Secondly, generating the list of mal-rules or bugs is a very time-consuming process which requires a lot of data collection and analysis if it is to be done properly. It is therefore desirable to find another approach which overcomes these drawbacks. One alternative approach is to **generate** the mal-rules, using machine learning techniques. One of the aims of this thesis was to do this for the article usage domain. The machine learning technique which was chosen for this purpose was candidate elimination using version spaces.

Another point of interest in this chapter is the work by Brecht (Wasson) and Jones (Brecht (Wasson) & Jones, 1988). They examined the original genetic graph representation (Goldstein, 1982) to see if it could be adapted for other domains. They implemented it for the subtraction and ballet domains. Most importantly, they claimed that it could be used for representing **dynamic** student models. Their point was that if a means could be found to generate new rules, whether mal-rules or other rules, these could then be attached to the genetic graph while the system was running, thus avoiding the limitations of **static** student models. The system's knowledge of the article usage domain is represented in this thesis as a genetic graph, in order that newly generated deviant rules can be added dynamically to the genetic graph, as proposed by Brecht (Wasson) and Jones. The development of these ideas into the system under discussion, *ArtCheck*, will be described in Chapter 6.

Section 2.2 described other aspects of the user which could be modelled, in addition to domain knowledge. The role of learning strategies in language learning will be discussed in Chapter 3. It will be seen that this is one aspect of the student which can be modelled in an ICALL system.

Section 2.3 gave a brief overview of relevant research in the broad field of explanation. It was seen that the information in the student model can be used in tailoring the explanations of misconceptions to individual students. It will be described in Chapter 7 how various aspects of the system's beliefs about the student can be used to individualise the interaction in an English article checking system.

The next chapter considers some of the issues in the area of second language learning, and discusses relevant research which has been carried out in this area.

Chapter 3

Background: Language learning

In this chapter, some of the work related to the area of language learning will be discussed. Firstly, a brief history will be given of the **theories of second language acquisition**, mainly concentrating on the role of **language transfer** in second language acquisition. Secondly, the **strategies** of language learners will be discussed. Finally, some of the **Intelligent Computer-Aided Language Learning (ICALL) systems** which have been developed will be described.

3.1 Theories of second language acquisition

From the 1950s through to the 1970s, two distinct approaches to the area of second language acquisition were developed: **contrastive analysis** and **error analysis**.

3.1.1 Contrastive analysis

The original **contrastive analysis** hypothesis was put forward by Lado (Lado, 1957). Contrastive analysis was basically interested in **predicting** errors by comparing the target language (TL) with the mother tongue or first language (L1) (Singleton, 1981). The errors could be predicted because they resulted from strategies of language use being **transferred** from the L1 to the TL, known as **language transfer**. Language transfer can be defined as being **positive** or **negative**. **Positive transfer** is the term used to describe transfer from the first language to the target language which results in correct use of the target language, and **negative transfer** to describe transfer which results in errors. Negative transfer is also known as interference. The version of contrastive analysis which emphasised its **predictive** ability was known as the **strong form**, while the **weak form** of contrastive analysis concentrated on its ability to **diagnose** errors (Ellis, 1985).

3.1.2 Error analysis

In the 1970s, the contrastive analysis approach was much criticised, and is now considered to be invalid, particularly the strong predictive form. It was seen that the predictions as to what would and would not cause problems for learners were unreliable (Kellerman, 1984). Other objections to contrastive analysis concerned its theoretical validity, and its relevance to language teaching (Ellis, 1985).

Evidence was given for **universal** orders of development of learners of English with markedly different mother tongues (Dulay & Burt, 1974). It was claimed that learners from different backgrounds followed the same stages of development of a target language, and also that even the equivalence of an utterance in the target language with one from the learner's mother tongue did not necessarily justify the assumption that the psycholinguistic process of L1 transfer had taken place.

Thus, the emphasis moved from the prediction of errors, to the **attribution** of the cause of the error, known as **error analysis**. Error analysis was described as an attempt to account for errors which could not be accounted for by contrastive analysis, and to bring second language acquisition in line with the current theoretical work in linguistics (Dulay *et al*, 1982). Many errors were found which could not be accounted for with language transfer. In one study, it was claimed that only a third of all errors could be attributed to interference or negative transfer (George, 1972), though other studies produced different statistics for this (Ellis, 1985).

One development emerging from this approach was the idea of a **transitional competence** (Corder, 1967) or **interlanguage** (Selinker, 1974). What was meant by this was that there were definite stages through which the learner moved when progressing from the L1 to the TL, (or from **no** knowledge of the TL to **complete** knowledge of the TL). It was claimed that each intermediate stage was a **definite system** in its own right with its own rules. As the learner moved closer to the TL, more of the rules of the TL appeared in her interlanguage (Selinker, 1974). In this approach, transfer was just one of several factors which influenced the formation of the interlanguage. According to Selinker, other factors included generalisation, transfer of training, strategies of second language learning, and strategies of second language communication. Corder described the learner's speech as an **idiosyncratic dialect**, similar to the language of infants, poets and aphasics (Corder, 1974).

Central to error analysis was an interest in the similarity between first language and second language acquisition. It was claimed that the strategies used to learn a second or subsequent language were the same as those used to acquire a first language (Corder, 1974). Dulay and Burt carried out various studies of the second language acquisition of children, for example (Dulay & Burt, 1973). They rejected the role of transfer altogether, and suggested that L1-like errors could be analysed as overgeneralisations of TL material, or as parallels to L1 acquisitional forms (Dulay & Burt, 1974). The claim was that most people, regardless of their language background, learn structures in a fairly set order (Dulay *et al*, 1982). There then followed a surge of interest in *morpheme studies* for different languages, which seemed to reveal that there was a universal order of acquisition irrespective of L1 background or age, and even some similarities with L1 acquisition orders (Kellerman, 1984).

3.1.3 Current thinking on language transfer

In this section, the current status of language transfer in second language acquisition will be outlined.

One of the problems with error analysis was that there were often several different analyses of the same error. For example, consider the ill-formed French sentence (Singleton, 1981):

**A-t-il revenu?*

which should read,

Est-il revenu?

One analysis of this error is that negative transfer has taken place from the equivalent English sentence, *Has he returned?* where the verb *have* is used. An alternative analysis, disputing the role of language transfer, is that **overgeneralisation** has taken place from sentences like *A-t-il répondu?*. It was suggested (Ellis, 1985) that the different interpretations depend on the individual researcher's bias. An obvious solution would be to accept both factors as contributors to the error (Chesterman, 1977).

Another problem with error analysis was that the studies which were intended to show the invalidity of transfer as a source of error were generally restricted to morphological data (Singleton, 1981). Even then, Kellerman quotes examples of morpheme studies which did show evidence of L1 interference (Kellerman, 1984).

Some data which has been gathered regarding the time taken to learn a new language shows that the similarity of the target language and the native language makes the acquisition of the target language quicker (Odlin, 1989), which indicates that the subject of transfer should not be discounted altogether.

More recent work in second language acquisition has re-emphasised the importance of language transfer, now sometimes known as **Cross-Linguistic Influence**. Transfer is seen to occur in all areas of language development, not just in morphology and syntax (Odlin, 1989). In the area of syntax, Odlin gives examples of errors relating to word order, relative clauses, verb phrases and articles, which can all be accounted for by language transfer. Most transfer errors are caused by **negative transfer**. However, in the case of article usage errors, a **lack of positive transfer** can be seen as the source of the error (Odlin, 1989).

In summary, there have been two extreme views on the subject of language transfer. Firstly, there was the view that language transfer was the only source of error (Lado, 1957) and secondly, there was the view that language transfer had no relevance in the attribution of causes of error (Dulay & Burt, 1974). Neither of these extreme views are adhered to in the current literature, and the consensus now seems to be that while Cross-Linguistic Influence plays a very important role in second language acquisition, other factors are also involved (Selinker, 1992, p172).

3.2 Learner Strategies

This section examines the role of the **learner's strategies** in language learning. Learner strategies, sometimes known as **learning styles** (Ellis, 1992), can be defined as:

> *"...strategies which contribute to the development of the language system which the learner constructs and affect learning directly."* (Rubin, 1987, p23)

Language learners can be seen to use a variety of different strategies in learning a language. Learners are usually aware of the strategies they use, and have opinions on successful and unsuccessful ways of learning (Horwitz, 1987). It is claimed that students themselves are the most accurate source of information regarding their use of learning strategies (O'Malley & Chamot, 1990). What students think about the process of language learning affects how they tackle it, and thus a particular instructional approach may not work as well for one student as for another (Ellis, 1992). As a result of this, students often complain about how they are being taught when they believe that there are better ways of learning a language (Horwitz, 1987). In a study of student beliefs about language learning described by Horwitz, more than 50% viewed learning grammar and vocabulary as the most important part of learning a language.

Different researchers in this area have come up with different sets of strategies, for example (Naiman *et al*, 1978; Bialystok, 1983; Færch & Kasper, 1983; Rubin, 1987; O'Malley & Chamot, 1990). One set is that proposed by Rubin (Rubin, 1987). Rubin, like O'Malley and Chamot, categorises strategies as **metacognitive**, **cognitive** and **social**. She describes metacognitive strategies as being concerned with the **regulation** and **monitoring** of the learning process. Cognitive strategies are said to cover areas such as **practice, memorisation, clarification, guessing, deducing** and **monitoring**. Social strategies are strategies which learners use to give themselves opportunities to be exposed to the language and practise their skills. These strategies can be used either consciously or unconsciously by students in acquiring a new language.

A distinction can be made between strategies to do with **remembering** and those to do with **communicating** (Wenden, 1987). In Wenden's survey, some students said they preferred to sit down and memorise the learning material, whereas other students made a point of not doing that, and learned by practising and listening to the language. This is described by Ellis (Ellis, 1992) as the distinction between **norm-oriented** learners and **communicative-oriented** learners:

> *"Norm-oriented learners are those who are concerned with developing knowledge of the linguistic rules of the second language, while communicative-oriented learners are those who seek to develop their capacity to communicate effectively in the L2 irrespective of formal accuracy."* (Ellis, 1992, p163)

Ellis has carried out some studies of adult learners of German which suggest that this distinction is both a valid one and relevant to future research in this area.

O'Malley and Chamot have proposed that students should be **trained** to use learning strategies to their benefit (O'Malley & Chamot, 1990). There are two possible interpretations relating to how this "training" can take place. On the one hand, teachers can promote the use of learner strategies by encouraging students to identify strategies that work well for them (Wenden, 1987). Alternatively, there may be **good strategies** that are associated with the successful language learner, and which less successful learners should try to use (Ellis, 1992).

To summarise, learner strategies are important, because less successful language learners may be able to use the strategies of more successful learners to good effect, and awareness of different strategies can enhance awareness of the language learning process.

"Learner strategies parallel theoretically derived cognitive processes and have the potential to influence learning outcomes in a positive manner." (O'Malley & Chamot, 1990, p217)

In an ICALL system, the learner model can be used to individualise the instruction for particular users. Therefore, if learners prefer and benefit from different forms of instruction, the incorporation of information about the learner's strategies would be a desirable extension of the learner model. This is implemented in *ArtCheck* to illustrate the usefulness of learning strategies in tailoring explanations to the individual student.

3.3 Intelligent Computer-Aided Language Learning (ICALL) Systems

This section will consist of a review of some of the work done in the area of Intelligent Computer-Aided Language Learning (ICALL) systems. These systems tend to be based on research either from the area of second language acquisition or from Artificial Intelligence. Existing language tutoring systems can be categorised in a number of ways. Some systems are specific to certain languages, and some are designed to be more general. Some systems concentrate on a few particular constructions or grammatical features, whereas others aim to teach the whole language. Here, three types of systems will be described:

- Systems which take into account the mother tongue of the learner
- Systems which concentrate on one or a few particular constructions
- General language learning systems.

3.3.1 ICALL systems and language transfer

One particular piece of work in Artificial Intelligence concerned with the effects of Cross-Linguistic Influence is a system which has been developed by Ethel Schuster, to aid in the teaching of English to Spanish students (Schuster, 1986; Schuster & Finin, 1986). The system, VP^2, is intended to introduce students to verb particles and verbs with prepositions in English, which is often a difficult area for speakers of other languages to grasp, due to the number of idioms in the English language. Schuster adheres to the view that transfer from the learner's native language affects learning of the second language. The student model consists of the speaker's own grammar, Spanish, while the expert model is the grammar of the TL, in this case, English. The system provides sentences for the learner to translate, where a verb particle or verb+preposition is required in the response. An example from the system VP^2 is given below:

TUTOR: TRANSLATE THE FOLLOWING SENTENCE
Moris penso en comprar un carro
STUDENT: *Moris thought in buying a car*
TUTOR: You used the incorrect preposition < **in** >.
In English you can use <think of> or
< think about> in this sentence.
Note that the direct translation of <think of>
- <pensar de> does *not* exist in Spanish.
In English, you can also use <think up>
(an excuse, invent); <think over> (review);
<think out> (consider, examine).

Errors are analysed by consulting first the expert model, then the student model, to see how the interference of the L1 grammar has affected the student's response. An explanation is given where there is negative transfer, which is backed up by other information about the construction being taught.

Another system which is based on the belief that negative transfer causes many errors in language has been developed by Wang and Garigliano (Wang & Garigliano, 1992).

This system is designed for English speakers learning Chinese. The student's native language, English, is taken into account when diagnosing errors. The motivation Wang and Garigliano give for considering negative transfer as a major contributor to the errors made is firstly, that people rely on previous knowledge when learning a new skill, and secondly, that this previous knowledge can sometimes hinder their learning (Halasz & Moran, 1982). A mixed grammar is used which consists mainly of Chinese rules and some partial English rules. The rules used reflect the findings of an empirical study (Wang & Garigliano, 1992). When the sentence is parsed, initially only the Chinese grammar rules are used in the parsing. If the parsing is unsuccessful, the system then attempts to parse part of the sentence using the partial English grammar rules.

The student model is used to detect errors and classify the errors as being due to transfer or not. There are three types of transfer error recognised: lexical transfer, that is, where the direct translation or words is inappropriate; idiomatic transfer, where idioms in one language do not transfer correctly to another; and syntactic transfer, where the structure or ordering of a phrase is incorrectly transferred from one language to another. The student's performance is evaluated depending on the generality and frequency of the errors made.

Scripsi (Catt & Hirst, 1990) is a system which has been developed for French or Chinese students of English. It is based on research in second language acquisition which has identified **transfer** and **overgeneralisation** as two of the most important sources of errors made by language learners, and includes a representation of both the native and the target languages.

Scripsi takes whole English sentences as input. The system then parses the input, diagnoses any transfer or overgeneralisation errors, and reports the results to the student. A feature-based context free grammar is used.

Transfer errors are diagnosed by applying L1 grammar rules when applying the L2 rules fails to give a parse for the input. Examples of this for French and Chinese learners are given below.

(i) French learner:

He has hunger.

(1) Avoir faim is expressed in English as TO BE HUNGRY

(ii) Chinese learner:

He very happy.

(1) Chinese usage: sentence lacks a copular verb.

The other type of error which is diagnosed is that caused by **overgeneralisation.** In Scripsi, rule overgeneralisation is equated with **constraint violation**. Examples of the sorts of constraints used are subject-verb agreement and word order restrictions.

Thus, if a parse cannot be found for the sentence, the system checks to see if any constraints have been violated. If so, the system allows **constraint relaxation** to enable a parse to be found, and reports the violation of the constraint as the error. For example,

My friend wroted a book.

(1) Verb WRITE has irregular past tense WROTE.

As can be seen from these examples, the explanations of errors given to the student are not particularly user-friendly. The student model is used purely for diagnosis and not for tailoring responses to students, or modelling the student's ability.

3.3.2 ICALL systems concentrating on particular areas of grammar

Some ICALL systems concentrate on just one, or a few areas of grammar. This section describes three examples of such systems. ET (English Tutor) (Fum *et al*, 1988) is a system which aims to instruct students in the use of English tenses. XTRA-TE (Kurtz *et al*, 1990) is a system which concentrates on agreement. The Fawlty Article Tutor (Kurup *et al*, 1992) is a system which deals specifically with English articles. These three systems will be discussed in turn. The system described in this thesis, *ArtCheck*, also concentrates on one area of grammar, in this case, the use of English articles.

ET (Fum *et al*, 1988) consists of a tutor, a domain expert and a student modeller. The tutor decides which exercises to use, and decides how to correct the student. The domain expert applies the domain knowledge to solve the assigned exercises. The student modeller analyses the responses and maintains a student model. The system's domain knowledge includes a set of production rules which are used to generate the tense of the verb.

The system operates as follows. The system gives the student an exercise consisting of a sentence in which the student has to conjugate the verb. Similar exercises are then presented to the student until the system has completed a topic or sub-topic and has formed a diagnosis of the student's performance. If the student makes an error, the system forms a hypothesis for that error and then chooses subsequent exercises which are designed to confirm the hypothesis. The system has access to a **bug catalogue** which consists of some stereotyped mal-rules used in the conjugation of English verbs. If none of these are applicable, the system then tries to **generate new mal-rules** which could explain the student's reasoning.

XTRA-TE is a language learning system for Chinese students learning English (Kurtz *et al*, 1990). It accepts free form student input which can be analysed both syntactically and semantically. It concentrates on syntactic and semantic agreement of nouns, verbs and adjectives, and pronominal gender agreement.

The student model consists of a vector of values which reflect the student's understanding of a particular concept. Fuzzy set theory is used to set the values of each concept.

There are four levels of familiarity, corresponding to how well the student knows each concept. The familiarity level is then used to adapt the correction of the student's errors. For example, if the student is familiar with the concept, no hint is given as to the correct answer after an error, but if the student is less familiar, an indirect or direct correction may be given.

Another feature of the system is the multilevel relaxation of agreement restrictions to determine the source of the error. The system first specifies that syntactic and semantic agreement must be enforced when parsing the input. If no parse is found, then the system first relaxes the syntactic agreement restrictions, then the semantic agreement restrictions, and then both, until a parse is found.

XTRA-TE is built on an existing machine translation system called XTRA with extensive English and Chinese dictionaries.

The Fawlty Article Tutor is a recently developed system which deals specifically with English article errors (Kurup *et al*, 1992). The domain knowledge consists of a set of production rules, based on six dimensions of the noun. The dimensions are **singular/plural**, **mass/count**, **common noun/proper noun**, **definite/indefinite**, **first introduction/subsequent introduction**, and **specific/general**.

In this system, the student is offered a scenario, in the form of a short passage, in which one noun phrase occurs with the article missing. The student has to decide which of the rules should be applied, depending on the combination of dimensions present, and thus which article to choose. The tutoring component controls the sequence in which scenarios are presented to the student. When the student makes an error, the system determines which of the dimensions are wrong and explains it to the student.

One of the problems with this system is that the domain of article usage has been reduced to just 11 rules, which are not presented in a way which is easy to remember. In addition, the rules do not appear to cover all usages of the articles, though this is a difficult goal for any system to achieve. As this system concentrates on the same domain as the system described in this thesis, the two systems will be compared in detail in Chapter 9.

3.3.3 General ICALL systems

Some ICALL systems are designed to be more general language teaching tools, in that they teach a wider variety of constructions, or can be applied to more than one language.

Schwind developed a system for French students using German (Schwind, 1990). The system deals with agreement errors, syntactic errors and semantic errors.

The system is based on a very complete German language knowledge base, which includes structural and semantic knowledge of German. The semantic knowledge is represented using features which indicate the properties of nouns and the selectional restrictions of verbs.

The system communicates with the student in French, and is able to generate simple French sentences. When the system is running, the student is given a list of verbs, nouns, and adjectives and asked to construct a sentence using those elements. The student can also ask questions of the system, for example, the meaning of a word or the conjugation of a verb. When the student has constructed a sentence, the system checks for several types of errors: agreements errors; omission or addition of words; misordering of words or constituents; and selectional restriction errors. If the system detects an error, it does not immediately correct it, but encourages the student to correct the error, by first asking a leading question, then stating the grammatical rule which has been broken, and then offering some examples. If all these options have been tried unsuccessfully, the system will give the student the correct answer.

A still more general approach to language learning has been developed by Ghemri (Ghemri, 1992). Here the principles of Government and Binding (GB) theory are used to develop a **language independent** approach to the diagnosis of misconceptions in the area of language learning.

The idea behind GB theory is that all human languages can be described by a common set of **principles** which they all obey, and a set of **parameters** which explain the divergences among them.

In Ghemri's system, the errors in learning a second language can be linked to setting a parameter incorrectly. The learning strategy adopted by the learner corresponds to the setting of the parameters. The different learning strategies defined are known as L1, L2, universal grammar (UG) or unmarked. The most frequently used learning strategy is then associated with each parameter.

The system can be used for a variety of languages, providing the parameter settings for that language are known.

3.4 Conclusion

This chapter has covered a wide selection of topics concerned with language learning. Firstly, a brief history of the research development in second language acquisition was given. It emerged from this that **language transfer** is considered an important source of error in language learning. Many errors, including syntactic errors, can be caused by either **negative transfer** or **lack of positive transfer**. The next section discussed the **learner strategies** of language learners. It was seen that awareness of the different strategies which can be used can improve the learner's acquisition of a new language. The incorporation of information about learner strategies in the student model of an ICALL system would seem to be a desirable development.

The final section gave brief details of some ICALL systems which have been developed. The purpose of this was to show the type of work which has done in this area before moving on to describe a new ICALL system. The ensuing chapters will describe the design and implementation of the system *ArtCheck* in some detail. The next chapter will describe the domain which has been chosen for this system.

Chapter 4

The domain: English articles

Articles in English are those small words *a, an* and *the* which precede nouns, whose main function is to indicate the definite or indefinite nature of a noun phrase. The indefinite article, *a/an,* is mostly used to introduce **new** information into the discourse. Such information then becomes **given** (Halliday, 1967) information which can then be referred to using a definite article, *the*.

For example, the sentence

I saw **a girl** *crossing the street.*

demonstrates this use of the indefinite article, where the article is being used to introduce a new noun phrase into the discourse. This sentence could then be followed by,

The girl *looked just like my sister.*

where the definite article in the noun phrase *the girl* can now be used, because the particular girl in question has been clearly identified in the preceding discourse.

The definite article can also be used to specify a particular referent where it can be assumed from the **context** that only one such referent exists and no ambiguity will arise. An example of this occurs in *the street* in the sentence above. This may be new information but can be used with the definite article because in the usual context only one *street* is available as a referent. Another example of this usage of the definite article can be seen in,

Sit down in front of **the gas fire**.

In contrast, the use of the **indefinite** article in the sentence

I am going to buy **a gas fire**.

is appropriate, as the noun phrase *a gas fire* is intended to be non-specific and not to refer to any particular gas fire.

The lack of an article before a noun phrase is commonly referred to as the **zero article**, as exemplified by the sentence,

Do you take **milk** *in your coffee?*

For the purposes of this thesis, the subject of English articles is being used as the **domain**, and as such provides the material or knowledge base which is to be used. It is not the intention of this thesis to produce a new analysis of English articles which supercedes the work described below, or a new set of rules of English article usage for the language learner. The purpose of this chapter is to **describe** the domain, and the problems it creates for the researcher and language learner.

Any discussion of this domain involves answering two questions:

- What are English articles and how can they be defined?
- What are the problems facing the language learner with respect to English articles?

The first question will be addressed in Section 4.1, which will give a brief history of the research into English articles. The second question will be discussed in Section 4.2 and Section 4.3. In Section 4.2, the use of English articles will be described as presented in traditional English grammars and English language teaching manuals. Section 4.3 will examine the specific difficulties that certain language learners have with the English articles.

4.1 English articles: the research issues

The study of the English articles is a complex area which has interested many researchers in linguistics and philosophy in the past. A brief account of the theoretical issues and main contributors involved will be given in this section. For a more detailed survey, see the recent book by Andrew Chesterman (Chesterman, 1991).

Firstly, the questions any analysis of article usage has to answer are (Chesterman, 1991):

- Which kinds of nouns may in principle take which article?
- Under what circumstances may - or must - a given noun or noun type take a given article?

The following discussion highlights the theoretical issues surrounding the answers to these questions.

4.1.1 Familiarity theory

The most important works on article usage in the first half of this century were those written by Christophersen (Christophersen, 1939) and Jespersen (Jespersen, 1949).

Christophersen introduced the **familiarity-unity** or **familiarity** theory. The notion of familiarity was defined as follows:

"The article **the** *brings about it that to the potential meaning (the idea) of the word is attached a certain association with previously acquired knowledge, by which it can be inferred that only one definite individual is meant"* (Christophersen, 1939, p72)

Thus, using *the* meant that the associated word was **familiar** and, correspondingly, words preceded by the zero article or *a* were **unfamiliar**. All nouns were categorised as **unit-words** or **continuate-words**, although some nouns could belong to both categories. A unit-word was defined as *"an individual or unit belonging to a class of similar objects"*, for example *dog*, and a continuate word as *"something ... continuous and extending indefinitely in space and time"*, for example, *water* or *happiness*. In accordance with this definition, the function of the indefinite article *a* was to stress the **element of unity** of the word. Where a word could be classed as both a unit-word and a continuate-word, the presence of the article *a* **added** the element of unity to the word. Thus, *a* represented unity, *the* represented familiarity, and the use of the zero article indicated the absence of both familiarity and unity.

The distribution of the articles within these concepts is shown in Figure 4–1.

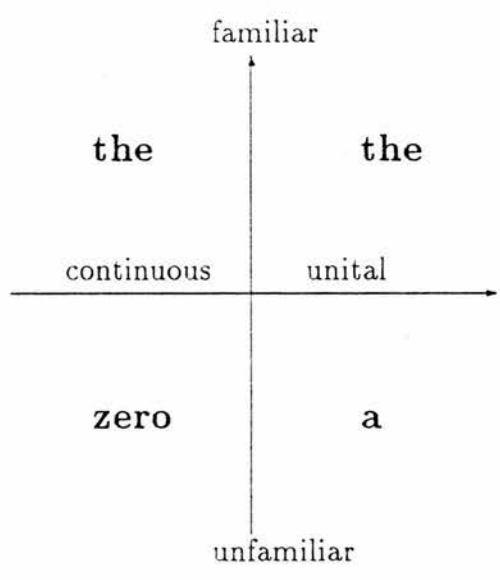


Figure 4–1: Christophersen's familiarity-unity theory

Jespersen's description of the articles was based on Christophersen's work. He proposed three **stages of familiarity**:

- The first stage was **complete unfamiliarity**, as in the examples *an apple* and *he drinks milk every day.* In this case the articles *a* or *zero* were used.

- The second stage was **nearly complete familiarity** where the word can be identified by the context or by the whole situation. In this case the article *the* was used, as in the example: *"Once there lived an old tailor in the village. The tailor was generally known in the village as the crook."*

- The third stage was "**familiarity so complete that no article (determinative) was needed**". Examples of this use were proper names, God, direct address, and before regular meals and places such as *church*, *prison*, *town* etc.

4.1.2 Extensitivity

Guillaume, whose work was on French articles, took a rather different approach (Guillaume, 1919)(cited in (Hewson, 1972; Chesterman, 1991)). He introduced the notion of **extensitivity**, which is an abstract notion which can be seen as being the difference between not using an article and using one. One of the aspects of Guillaume's theory is that any noun can take any article in principle, and the nouns that regularly reject a certain article are the exceptional ones, rather than the other way round (Chesterman, 1991). Guillaume's theory of language is **psychomechanical**, embedding the idea of a movement from the potential to the actual. Thus, each article has a **kinetic** aspect as well as a **static** aspect. He defines the kinetic aspect of the **indefinite** article as a movement towards the **singular** and **particular**, and the kinetic aspect of the **definite** article as a movement towards the **general** and **universal**. Hewson later applied the notion of extensitivity to the articles in English (Hewson, 1972), giving an explanation of their use from a psychomechanical point of view. He also elaborated on some of the later, unpublished work of Guillaume.

4.1.3 A structural approach

Yotsukura carried out a **structural analysis** of a large corpus (Yotsukura, 1970). Rather than the traditionally accepted three articles, *a*, *the*, and zero, she considered there to be **five** articles, *a*, *the*, *some*, the zero article and the null article. The essential difference between the **zero** article and the **null** article is that the zero article is indefinite and occurs before mass and plural nouns, and the null article is definite and occurs with proper nouns and, in certain cases, with singular count nouns (Chesterman, 1991).

For her study, Yotsukura considered 9000 occurrences of the 100 most commonly occurring words from nine standard high school text books. She arrived at 38 formulae describing the uses of articles in structural terms. These formulae took into account the classification of the noun as count or mass, singular or plural, and whether and how the noun was modified. She also listed groups of nouns which behaved exceptionally, such as a group of singular count nouns which can occur in subject position without an article, (*part, man, woman, age, life, ...* etc.).

One of the limitations of a purely structural analyses is that it cannot be specific enough to always indicate which article to use where. Of the 38 formulae given, only 17 give one article as the only possibility. Presumably contextual information would need to be applied to the other cases to restrict the choice of articles.

Although Yotsukura recognised the limitations of a study which restricts itself to structural information, particularly in the lack of information about the context of the noun phrase, she indicated that concentrating on the structural aspect of article usage would be useful for any future computational analysis.

4.1.4 Locatability and Inclusiveness

In a later study, Hawkins, on the other hand, claimed that a syntactic description of the articles would never be adequate and that only a **pragmatic** and **semantic** account would capture the distinction between the different usages (Hawkins, 1978). He proposed an account of the articles based on **speech-act theory**:

"The difference in truth conditions between definite and indefinite articles are often a natural consequence of pragmatic (speech act) differences between them and ... a large number of article ungrammaticalities are in turn directly explainable in terms of the logical and pragmatic meanings of the articles." (Hawkins, 1978, p15)

In Hawkins' analysis, the use of the definite article instructs a hearer to **locate** the referent within the context of the shared speaker-hearer knowledge. This is known as **the location theory**.

Another distinction Hawkins made was between **inclusive** and **exclusive** reference. When a noun phrase has inclusive reference, **all** the objects in the shared set, (*"the pragmatically limited domain of quantification"*), are being referred to.

In some more recent work, Chesterman used the two concepts of **locatability** and **inclusiveness** developed by Hawkins, in his own analysis of the English articles (Chesterman, 1991). Chesterman's book is an analysis of definiteness both in English and Finnish (a language with no grammatical category equivalent to the article). He claimed that definiteness and indefiniteness are not just opposites but *"qualitatively different concepts"*.

Chesterman, like Yotsukura (Yotsukura, 1970), used the five articles *a*, *the*, *some*, the indefinite zero article, and the definite null article. His analysis of definiteness involved three oppositions:

- The first opposition was between **locatable** and **non-locatable** referents (Hawkins, 1978). His definition of locatability modified Hawkins' definition to include non-referential items such as **properties**. He also argued that while locatability is a necessary condition for the definite article *the*, the absence of it is not a **necessary** condition for *a* or *some*. This is shown by the example, *Fred lost a leg* (Chesterman, 1991), where *a leg* is clearly locatable. On the other hand, the zero article must always be non-locatable, as in the question, *Do you take sugar?*. The null article, which is used for proper nouns, such as *John*, which are uniquely identifiable, and other definite singular count nouns, does have the property of locatability.

- The second opposition was between **inclusive** and **exclusive** referents. Chesterman defined **inclusiveness** as referring to *more or less all*, or *pragmatically all* the members of a set, claiming that Hawkins' original all-or-nothing definition was too strong. Thus, the use of the null article was inclusive as only one member of a set was being referred to; the use of the definite article *the* was inclusive because all the members of the shared set were being referred to; the use of the zero article did not say anything about inclusiveness; and *a* and *some* necessarily involved exclusive reference.

- The third opposition which Chesterman used was between **limited extensitivity** and **unlimited extensitivity**. The notion of extensitivity was taken from the work of Guillaume (Guillaume, 1919), mentioned above. Chesterman defined **limited extensitivity** as being the use of a surface article, that is, *a*, *some*, or *the*, and **unlimited extensitivity** as being where there the zero article or null article were used.

These three oppositions are summarised in Figure 4–2.

	Locatable	**Inclusive**	**Limited extensitivity**
zero	-	+/-	-
some	+/-	-	+
a	+/-	-	+
the	+	+	+
null	+	+	-

Figure 4–2: A feature matrix showing the distribution of the articles

Chesterman then redefined these oppositions in terms of sets. He used two types of sets: referent sets and entity sets. A **referent set** is the set of all known referents, and **inclusiveness** is defined in these terms. An **entity set** is the set of all the referents and the entities associated with those referents, and this is the type of set which is used to define the notion of **locatability**. For example, if there is a referent set consisting of *car*, the corresponding entity set will include *steering-wheel,engine*, etc., and all the members of this set will be regarded as locatable. The functions of each of the articles can be summarised in terms of sets as follows:

a(n) NP : one member of a referent set.

some NP : not-all (members) of a referent set.

the NP : (pragmatically) all (the members) of a locatable referent set (where 'locatable' means 'locatable in a shared entity set').

null NP : a locatable, one-member referent set itself.

zero NP : a referent set itself (which must not be a one-member set).

4.2 English articles and traditional grammar

The previous section demonstrates how researchers in this area have struggled to find an adequate analysis which accounts for all the different occurrences of the articles in the English language. However, from the point of view of the grammarian and the language learner, a non-theoretical description of the use of the articles is required, which can be used to decide which article is appropriate in a given context, and includes the idiomatic exceptions to the general rules which language learners need to know.

Thus, in this section, the description of English articles given by traditional English grammars (Quirk *et al*, 1972; Leech, 1989; Close, 1972) and English language teaching books, for example (Arnold *et al*, 1988a; Arnold *et al*, 1988b; Westlake *et al*, 1988), will be outlined. The standard way of teaching English articles to language learners has been to present a set of rules giving environments in which the articles can be used. A typical set of rules is given in the next section.

4.2.1 A set of rules describing article usage

- **Using the indefinite article** (*a/an*)

  - The indefinite article is used when introducing new information, for example, *"Once upon a time, there was a beautiful princess"*.
  - Of the two forms of the indefinite article *a* and *an*, *a* is used preceding words beginning with a consonant, and *an* is used preceding words beginning with a vowel.
  - The indefinite article is used only for singular count nouns[1]. There is no plural form of *a/an*.
  - The indefinite article is used after the verbs *be*, *become* and *remain* and before a singular count noun which is an indefinite complement, for example, *He became a teacher.*
  - The indefinite article is used in idiomatic phrases to do with rate and frequency, eg. *five miles an hour*, *twice a day.*
  - The indefinite article is used in the expressions *a hundred...*, *a half*, *a dozen*, *a million* etc, for example, *she bought a dozen eggs.*
  - The indefinite article is usually used within the structure *There is a...*, for example, *There is a book on the table.*

- **Using the definite article** (*the*)

  - The definite article *the* can be used in the singular or plural and for mass or count nouns.
  - The definite article is used when the noun it precedes has been mentioned before.

[1] A noun is known as a **count noun** if it has a plural form, and refers to an object which can be counted, eg, *book*. A noun is known as a **mass noun** if it is uncountable, eg *bread*.

- The definite article is used when the context makes it clear what is being referred to, for example, in a passage about rabbits, ".. *After two or three weeks of spoiled lettuces and nibbled cabbage-plants, the cottager had lain in wait and shot him (the rabbit) as he came through* **the potato-patch** *at dawn*" (Adams, 1972, p31).
- The definite article is generally used when the noun is modified by a prepositional phrase using the preposition *of*, for example, *the capital of Finland.*
- The definite article is used where the noun is normally classified as an adjective but is being used as a noun, for example, *the poor*, *the blind.*
- The definite article is used for the following cases of proper nouns. The default for proper nouns is the zero article.
    * Proper nouns which also exist as common nouns, eg *the Grand Hotel*
    * Plural names, eg *the Netherlands,the Smiths*
    * Geographical names, eg *the Mediterranean*, *the Atlantic*
    * Newspapers, eg *the Guardian*
    * Hotels and hospitals, eg *The Royal Infirmary*
- The definite article is often used when the noun is modified by a relative clause.
- The definite article is used for certain common nouns which are thought to be salient in all contexts, for example, *the world*, *the sky*, *the moon*, *the sun.* In this case the definite article is known as the "indexical the" (Quirk *et al*, 1972).
- The definite article is used after the words *all* and *half* and before the appropriate noun, for example, *half the cheese*, *all the students.*
- The definite article is used with the superlative, for example, *the happiest baby*, *the best film.*
- The definite article is used before ordinal numbers, for example, *the first chapter.*

- The definite article is used before the words *same* and *only*, for example, *the same college*, *the only student.*
- The definite article is used in comparative expressions of the form *article+comparative+article+comparative*, for example, *the more the merrier.*

- **Using the zero article**

  Where no article is used, this is commonly known as the **the zero article**.

  - The zero article is usually used with mass nouns, both concrete and abstract for example, *bread*, *milk*, *money*, *poetry.*
  - The zero article is usually used with plural count nouns, for example, *Dogs are very lovable animals.*
  - The zero article is usually used with proper nouns, for example, *Paul*, *Australia.* For exceptions to this, see the rules on the usage of the definite article.
  - The zero article is usually used when a number precedes the noun, eg *eight dogs*, *five books.*
  - The zero article is used before singular count nouns in certain cases involving prepositional phrases and some semantic categories of nouns, as follows:
    * time and season, eg *in spring,at night*
    * meals, eg *at breakfast*
    * place, eg *at school,in hospital*
    * transport, eg *on foot*, *by bus*

4.2.2 Specific vs generic vs unique reference

The rules given above do not take into account one very important variable, that is, the *function* of the article (Quirk *et al*, 1972). The function of the article relates to the type of reference which is intended. It can be described as **specific**, **generic** or **unique**. Most of the rules given in Section 4.2.1 relate to noun phrases used with specific reference. The difference between these three types of reference is described below.

- **Specific reference**

  Specific reference is where the speaker/writer intends to refer to a specific member or specific members of a general class. For example,

  > *I have* **a very old cat.**

  The rules for using articles where specific reference is intended are as given in Section 4.2.1.

- **Generic reference**

  Generic reference is where the speaker/writer intends to refer to a class in general, without reference to any particular instance of that class. For example,

  > **The cat** *has been a domestic pet for thousands of years.*

  The rules for using articles where generic reference is intended are slightly different to those given in Section 4.2.1:

  - When a singular count noun is used with generic reference in mind, the definite article is used, for example, **The aeroplane** *has revolutionised travel.*
  - When a plural count noun or mass noun is used with generic reference in mind, the zero article is used, for example, **Cars** *cause great damage to the environment*, **Money** *is the root of all evil.*

- **Unique reference**

  Unique reference is used to mean proper nouns, where the particular instance being referred to is stated explicitly. The rules for article usage with proper nouns are as given in Section 4.2.1, the basic principle being that the zero article is the default, with the definite article being used before some exceptional cases.

4.3 English articles and the language learner

4.3.1 Article-less languages

As discussed in Section 3.1, the acquisition of a second language, and the problems learners have with this, can be greatly affected by the learner's native language. In learning the grammatical structures and the vocabulary of another language, learners automatically make reference to the knowledge they already have about language, that is, knowledge of their own native language (Singleton, 1981). This is known as **language transfer**, and can be **positive** or **negative**, as defined on page 37. It has been shown that the less extreme these structural differences are the easier it is for learners to acquire the language (Odlin, 1989).

The use of English articles is not normally a problem for non-native speakers as many other languages have an equivalent category, and the strategies which govern their use can be transferred successfully (Oller & Redding, 1971). This is a case of positive transfer. However, there are languages which do not, and for native speakers of these languages, using English articles appropriately can cause difficulty. Examples of languages where there are no articles are Finnish, Russian, Japanese, Basque, Turkish, Slovak, and Arabic. The problem that learners of these language backgrounds have with articles can be seen as a case of a **lack of positive transfer**, rather than negative transfer, as discussed in Section 3.1.3.

Finnish: a language with no articles

In Finnish, words are formed by adding a variety of affixes to basic word stems; these affixes include markers for case (there are 15 cases in Finnish), number, tense, and emphasis. There are no words or affixes for gender, and no articles. The vocabulary bears no similarity to English vocabulary, though there are many loan words from Swedish.

One of the differences between the two languages is the way they express the definiteness or indefiniteness of objects referred to. In English we use articles. In Finnish, however, there is no explicit syntactic category with this function. The way definiteness and indefiniteness is expressed in Finnish with nouns with specific reference is known as **species**, of which the strategies used include word order and case-endings. In terms of word order, sentence-final position is used for new information, and noun phrases occurring in non-sentence-final positions can thus often be read as definite, as shown in Figure 4–3. Case also plays an important role in determining definiteness. The Finnish partitive case is used to indicate indefinite quantities of things, and as such is thus sometimes used in opposition to the accusative case, which is also shown in Figure 4–3. In the area of generic reference, however, Finnish does not have any clear equivalents. Generally, in Finnish, the singular refers to individual members of a class (specific function), and the plural is used to refer to a whole class (generic function). This means that there is no concept of a generic singular noun in Finnish.

Therefore, for the Finnish student learning English, the **article** represents a completely new category to learn for which Finnish has no equivalent, and the student also has to try to "forget" ways of expressing definiteness and indefiniteness which come most naturally to them. Teaching articles to Finnish students can be a considerable problem for English teachers.

Other languages without articles

The definiteness strategies of some other languages which do not have articles will now be briefly described.

Case endings

Otan	kirjat	Otan	kirjoja
take-1sg	*book-accusative-pl*	*take-1sg*	*book-partitive-pl*
I take the books		*I take some books*	

Word Order

Huonessa	on	tuoli	Tuoli	on	huonessa
room-inessive	*be-3sg*	*chair-nom*	*chair-nom*	*be-3sg*	*room-inessive*
There is a chair in the room			*The chair is in a/the room*		

Figure 4–3: Definiteness strategies in Finnish

- **Russian**

Russian, like Finnish, uses case endings and word order to show definiteness. Definiteness is marked on some **object** noun phrases by using the accusative marker for a definite noun phrase and the genitive marker for indefinite. On **subject** noun phrases the nominative case is used for definite noun phrases and the genitive for indefinite. In this way, the Russian genitive case acts rather like the Finnish partitive.

- **Basque**

Basque is an agglutinative language, like Finnish. It does not have articles, but does instead have a morpheme which represents definiteness in the case of definite noun phrases. The surface morpheme varies according to various other qualities of the noun, for example, whether it is singular or plural, whether it occurs in subject or object position in the sentence, and whether it occurs with an intransitive or transitive verb. To indicate that a word is indefinite, the word *bat* meaning *one* sometimes comes after it.

- **Polish**

  Polish uses word order, prenominal pronouns and stress and intonation to indicate definiteness (Szwedek, 1976). As with Finnish, new information is found in sentence-final position and given information in non-sentence-final position. Sentence stress is normally put on new information and not on given information. In some cases, dependent on the word order and sentence stress, the demonstrative pronoun *ten* and the indefinite pronoun *jakiś* are obligatory, and function similarly to the English definite and indefinite article respectively.

- **Turkish**

  Turkish is another agglutinative language, rich in vowel harmony and consonant assimilation. It does not have articles, but the word *bir* for one is sometimes used for the indefinite *a/an.* If the noun is used as the direct object of a verb then the definite objective case is used to indicate definiteness. In other situations the definiteness of a noun phrase is indicated by the context.

4.3.2 Common difficulties experienced with articles

The effect of inter-lingual interference

A detailed study was conducted in Jyväskylä, Finland, in order to determine what sorts of errors Finnish university students made with English articles (Herranen, 1978) and to what extent these errors were attributable to inter-lingual interference (or negative transfer).

Order	Name of Category	Nr.of Errors	Rel. Freq.(%)
1.	The generic definite article before singular count nouns, eg. "Tiger and cat *belong to the same family of animals.*"	77	57
2.	The definite article before plural count nouns with bounded generic reference,eg. "People in the corner seats *pull up the windows and hold them shut...*"	32	36
3.	The generic zero article before non-count nouns, eg. *"From this time comes also his love towards* the water *as a symbol of happiness."*	110	35
4.	The "indicative" definite article before common noun turned name, eg. *"Dinner will be held at* Grand Hotel."	56	31
5.	The specific zero article before plural count nouns.	14	31
6.	The specific zero article in intensive relation, eg. *"He turned* a communist *in order to help the poor."*	25	28
7.	The specific zero article in prepositional phrases, eg. *"The darkness outside the windows is touched by* the puffs of cloudy whiteness."	31	23
8.	The specific definite article before nouns make particular by the context.	117	19
9.	The specific definite article before nouns made particular by a modifier, eg. *"The railway carriage is dark except for* a feeble glimmer *of the small lamps in the ceiling."*	125	17
10.	The zero article with unique reference, eg. *"Henderson lived in* the dry Africa."	44	16
11.	The specific indefinite article before nouns with modification, eg. *"It was* a very skillfully made plan."	22	12
12.	The definite article with unique reference having partitive meaning.	5	11

Table 4–1: Herranen's hierarchy of difficulty based on relative frequencies of errors

Herranen carried out two studies: the first was a traditional error analysis of the essays written by 90 students at Jyväskylä University, to see what types of errors were made by students; and the second was a more sophisticated multiple-choice exercise given to 45 students designed to test for the types of errors which were more difficult to elicit in normal written text, such as nouns occurring with generic and unique reference (see Section 4.2.2). The students tested were at different levels of English. She recorded 325 errors from the first study, and 724 from the second. Table 4–1 lists the most relatively frequently occurring types of errors, where the number of occurrences of incorrect article usages are compared with the number of correct article usages of the same type. Where an example of an error corresponding to a category is given in the table, it is taken from Herranen's own data. The absence of an example in the table indicates that Herranen did not give one.

Herranen's particular interest was in deciding how many of the errors made could be attributed to inter-lingual interference. From the errors she found, she analysed the following as having an interference component:

(i) **The omission of the indefinite article** *a/an*, in cases where a noun being used with specific reference occurred in sentence-final position. Herranen analysed this as being due to the fact that, in Finnish, such a noun would be identified as being indefinite by virtue of its sentence-final position. The use of this strategy in English would lead to the article being omitted.

(ii) **The omission of the definite article** *the*, in cases where a noun being used with specific reference occurred in a non-sentence-final position. The analysis of this error mirrors that above, in that the strategy of indicating definiteness with word order may cause interference, leading to an omission of the definite article.

(iii) **Using the indefinite** *a/an* **instead of the definite article** *the*, in cases where a modified noun with specific reference occurs in sentence-final position. In English the definite article is used because the noun is modified.

Herranen analyses the errors of this type as being caused by interference of the word order strategy once more, whereby the occurrence of the word in sentence-final position leads to an indefinite article being used, (a correction of error type (i)), when in fact the modification of the noun means that it can no longer be regarded as new information. (This analysis may not be correct, as in fact a modified noun sometimes occurs with the indefinite *a/an*, and sometimes with the definite article *the*, and which article it takes depends on whether the modification is restrictive enough to render the noun uniquely identifiable to the hearer).

(iv) **Addition of the indefinite article** *a/an* where a mass noun is used with generic reference. Herranen analyses this error as possibly being due to the Finnish student incorrectly classifying a mass noun as a count noun, in cases where the equivalent noun in Finnish would be categorised as a mass noun, or vice versa.

(v) **The use of the zero article or the indefinite article** *a/an* **instead of the definite article** *the* where a singular count noun is being used with generic reference. This was the type of article usage which Herranen claimed was the most difficult (see Table 4–1). There is no equivalent of the generic definite article in Finnish at all, and in Finnish this concept would be interpreted via the semantics of the sentence. This can only be regarded as inter-lingual interference because it is the complete absence of an equivalent concept in Finnish which causes the Finnish learner such difficulty. Herranen adds that this difficulty could have been *"reinforced by faulty teaching and faulty materials"*, because the distinction between specific and generic usages of nouns is not made in English teaching books available in Finland.

Students tested in Oulu and Edinburgh

As part of the project under discussion in this thesis, two smaller data collection studies were carried out, with the aim of corroborating Herranen's findings concerning the parts of the English article usage system students found particularly difficult. The first study took place in Oulu, in Finland, and involved 38 advanced level students from the Department of English in the University of Oulu. The second study involved 15 students from Helsinki University, who were spending a year on an exchange programme at Edinburgh University. The second group contained students from all disciplines, and these students had already spent 8 months living in Edinburgh when they took part in the study.

The Oulu group was given a multiple-choice exercise and a comprehension exercise. The Edinburgh group was given a (longer) multiple-choice exercise. Both groups experienced most problems with the generic use of *the* with a singular count noun, thus agreeing with the research carried out by Herranen. For example, the following two multiple choice questions are those which both groups found the most difficult:

(For each question, students were instructed to select the correct answer from the choice of **A**, **B**, and **C** given).

(i) A An aeroplane has revolutionised travel.
B The aeroplane has revolutionised travel.
C Aeroplane has revolutionised travel.

(ii) A The olive grows only in warm climates.
B Olive grows only in warm climates.
C An olive grows only in warm climates.

In the first question, 26 out of a total of 53 students selected the incorrect answer C, and 3 selected the incorrect answer A. In the second question, 25 students selected the incorrect answer B, and 13 selected the incorrect answer C. Thus, 55% of students answered incorrectly in the first case and 72% in the second. According to Herranen, the distinction between specific and generic reference is not only a difficult one for the student, but possibly a distinction that has not been taught correctly to students. She suggests that improved teaching methods might include making this distinction quite clear.

The Oulu students also completed a comprehension exercise to uncover the sorts of article usage errors which occurred in free writing. There were not as many article errors in this part of the exercise, although one of the problems in analysing the data was determining whether an error was to do with the articles or something else, as in the example:

> " **People should act now in order to save the world from* **the nature destruction.**"

In this sentence *the nature destruction* is incorrect, but it is unclear whether there is an article error. The student could have said *nature's destruction* in which case there should be no article. *The nature* is a fairly common error for students to make. Or, alternatively, the student could have said *the destruction of nature* in which case the definite article is required.

The comprehension exercise given to the Oulu students was designed to elicit many occurrences of generic usages of noun phrases. However, this was not a major source of difficulty for the students as they were able to express the generic function with the zero article and plural noun instead of with the definite article with singular noun, as shown in the following example:

> *"Rainforests, for example, are (of) vital importance to the climate of the whole world."*

The generic sense could also have been expressed in the following way:

> *"The rainforest, for example, is (of) vital importance to the climate of the whole world."*

Thus, the students could avoid a construction with which they lacked confidence because they could express the same idea in another way. This behaviour is known as **avoidance** (Kleinmann, 1977). Therefore, if the use of the generic singular form is only a problem in multiple choice exercises, it suggests that it may not be a real problem for Finnish students at all.

Also in the comprehension exercise, there were many errors made with words like *climate*, for example,

> " **Burning of* CO_2 *will stop and* **climate** *will change.*"

> " ***Climate** *will change everywhere, we will have to deal with a lot of refuges, many animals will die.*"

The noun *climate* occurs with *the* because it is an **indexical noun** and can be definite in any context. The fact that the students consistently used the zero article with this noun indicates they may have thought it was a mass noun.

Other common errors were with modified noun phrases, such as:

> " **Because* **destruction of rainforests** *in one country can kill people in another.*"

and mass nouns, such as:

> " **If the rainforests are cut down, the rains wash away the fertile part of the soil, no plants can live there any more,* **the erosion** *starts, and* **the desertification** *and the soil dries, and people run out of water.*"

More details of this exercise can be found in Appendix C.1.

Another study in Edinburgh[2] involved 15 students of mixed nationalities and mixed levels of English. All the students had as their first language one which did not include articles. They were given a short test of 10-12 multiple choice questions which was tailored to their level of ability. The students made a variety

---

[2]This study was actually part of an evaluation exercise. The findings discussed here were taken from an analysis of the **pre-test** part of the exercise only, as the other data from the exercise involves some teaching on articles and will be examined in full in Chapter 8 on the evaluation of the system.

of errors, and all of the students made at least one error. The number of students involved is obviously not high enough to give any sort of statistical analysis, but it is possible to indicate some particular sources of difficulty for these students. (The generic use of the article was not tested in this exercise, as it was thought to be too difficult). The most common errors found were:

- Using the zero article instead of *the* in the following situations:
  - Plural proper names, (for example, *the United States of America*).
  - Proper nouns which also occur as common nouns, which should also be preceded by *the*, (for example *The Royal Infirmary*).
  - After *all*, (for example, *all the students were there*).
- Using the zero article instead of *a* in the following situation:
  - Where the noun is a complement of the verb *to be*, (for example *Sally is doctor*).

The learner's and the teacher's view

As part of the studies described above, the participating students and some of the teachers were asked about their experiences with learning and teaching English articles. More details of the types of questions asked are given in Appendix C. In addition, a questionnaire was sent out to several teachers of English in Finland asking for more detailed information on the teaching of the English articles. This questionnaire can be found in Appendix C.3. The following paragraphs summarise the opinions gathered from these sources.

Article usage is a subtle area of the English language. Correct article usage is not regarded as important as, for example, using the correct verb form or number, as even with the wrong article, the learner can (usually) still be understood. However, students whose first language does not contain articles readily acknowledge that English articles are a great problem for them. Teachers of English also recognise this as an area of difficulty for their students. Some students and teachers have the view that it is impossible to be completely free of article errors and that the use (or misuse) of articles will always distinguish the native English speaker from the non-native speaker.

Many students complain about the number of article usage rules which they need to learn, and in addition, about the many exceptions to the general rules which they keep encountering. Some students admit to guessing which article to use, and to generally lacking confidence in article usage, even when otherwise their standard of English is very high. Some Finnish students questioned made particular comments about finding the choice of article to accompany a generic usage of a noun very difficult, and also remembering which types of proper nouns took an article and which did not. Teachers questioned also commented on the large number of rules which the students were expected to learn. However, one teacher commented that, in her opinion, where the rote learning of rules had been abandoned as a teaching strategy, the performance with articles was worse, because the students did not have a rule to refer to when they needed it. Another teacher felt that the context-based rules were difficult for the student to grasp (that if the noun has been mentioned before, or brought into the context in some way, it must be used with the definite article), especially for students who were weak in the area of reading comprehension and thus could not trace back the antecedents of a given noun phrase. It was also mentioned that a particular area of difficulty for students was to know when one rule would overrule another, which is an area not covered in typical grammar and teaching books. Another criticism of teaching manuals given was that they only give examples of article usage at the sentence level, when the decision about article usage mostly depends on the context in which the noun phrase occurs. Thus the opinion of both the teacher and the learner is that correct article usage is difficult to achieve.

4.4 Conclusion

From the above discussion, it can be seen that the area of article usage poses considerable problems for the language learner whose first language does not contain articles, whatever their level of English. The research which has been done in this area and was described earlier in the chapter shows that it is also a difficult area to define at a theoretical level. With this in mind, it seems that the domain of English article usage is a challenging one to take as a starting point for an intelligent computer aided language learning (ICALL) system. The next chapter will show how a formalised account of article usage has been developed and implemented, and then used as a knowledge base within the ICALL system *ArtCheck*.

Chapter 5

A computational account of English article usage

The previous chapter described the domain of English articles, and the problems that it creates for certain language learners. In this chapter, a computational account of article usage will be described, as developed in the system *ArtCheck*. The following sections will show how a plausible set of rules describing article usage can be implemented, using a variety of information sources. In Section 5.1 a set of rules is chosen which can be said to describe the majority of usages of the articles. Section 5.2 and Section 5.3 will discuss how the article usage rules can be applied. Section 5.4 will describe how the rules of article usage can be used as a knowledge base for an ICALL system.

5.1 Primary considerations

5.1.1 Defining a rule set

It was stated in Section 1.2 that for a computer system to say it knows about article usage, it must be able to accept whole sentences as input and predict the correct article usage in each noun phrase. Therefore, a description of article usage is required which handles most, if not all, occurrences of articles in English. Several alternative descriptions of articles were discussed in Section 4.1. When considering the sort of description which a computer system could use to predict correct article usage, it is necessary to consider the type of information which is implied, and the type of information a computer system could plausibly have access to. For example, applying the concept of **locatability** (Hawkins, 1978; Chesterman, 1991) requires extensive pragmatic knowledge which the computer does not have access to. If the necessary pragmatic knowledge is fed into the computer the onus is then essentially on the system designer and not the system to predict correct article usage. This objection applies to other descriptions of article usage in semantic and pragmatic terms. Thus, a more structural approach is preferred from a computational point of view, for example Yotsukura's 38 formulae, or a traditional rule set as exemplified in Section 4.2.1.

The majority of language students have been taught using rules for article usage as given in Section 4.2.1. It was thus felt that this should be taken as the starting point for this exercise. One of the problems with some of these lists of rules is that there are many exceptions. Thus, some of Yotsukura's rules, excluding very uncommon usages, were then added for completeness. These rules cover the specific and unique functions of the noun phrase as defined in Section 4.2.2. The application of the rules for generic usage will be described separately in Section 5.3.5. This set of article usage rules is shown in Table 5–1.

Article	When used	Information needed
a/an	For new information	Record of previous NPs used.
	For singular count nouns	Number and count/mass noun.
	With rates and frequencies	Types of words affected.
	After *be*,*become*,*remain* with singular count complement	Number and count/mass noun *and* What precedes NP.
	Before *hundred*,*half*, etc	Types of words affected.
	Before *lot of*,*kind of*,*type of* etc.	If prepositional phrase *and* List of affected words.
	After *such* and before singular count nouns	Number and count/mass of noun *and* What precedes NP.
	With the adjective *certain*	If premodifying adjective.
	After *half* and before singular count nouns	Number and count/mass of noun *and* What precedes NP.
the	For given information	Record of previous NPs used *and* Contextual information *and* General/world knowledge.
	Before *of* prepositional phrases	Details of modifier.
	Where a noun is also classed as an adjective.	Category of word.
	With relative clauses	Details of modifier.
	After *all*,*half* and *both*	What precedes NP.
	Before *same* and *only*	Details of modifier.
	Before the superlative	Morphological information *and* Details of modifier.
	Before ordinal numbers	Morphological information *and* Details of modifier.
	Before plural proper nouns	Category of word *and* Number of noun.
	Before proper nouns which are also common nouns	Category of word.
	In comparative expressions such as *the more the merrier*	Recognition of structure
zero	With mass nouns	Count/mass noun.
	With plural count nouns	Number and count/mass of noun.
	With proper nouns	Category of word.
	When a number precedes noun	Details of modifier *and* Which adjectives are numbers.
	With certain singular count nouns to do with seasons, meals, time, transport, and institutions, and with certain prepositions.	List of types of nouns *and* Prepositional phrase modifier

Table 5–1: The information needed to apply the article usage rules

5.1.2 Information required

Having started with a relatively plausible set of article usage rules, the next stage was to decide what sort of information the computer must have access to in order to apply each rule. In Table 5–1, the set of article usage rules is given and in addition, the information needed to apply each rule.

For the purposes of this thesis, three articles were considered, *the*, *a/an* and zero. *A* and *an* are dealt with together, as they exist as two surface variants of the same article. A rule has been implemented which decides between *a* and *an*, but this is applied at a different level of the system.

Table 5–1 is divided into three sections, one section for when to use each article. Each rule takes the form:

When conditions X apply, use article Y.

The table shows the information the computer would need to have in order to apply the conditions *X*. For example, to apply the rule:

Rule: *Use* **the** *where a noun is modified by a superlative adjective.*

the system would have to know if the noun was **modified** and if so, by what (syntactic information), and whether an adjective was a **superlative** form (morphological information). It is necessary then to consider from where this information will come.

If the computer system, in addition to checking the article usage, is able to process the natural language input, one of the possible information sources available is the natural language processor. This will be discussed in the next section.

5.1.3 Natural language processing

As the system processes the sentence, it also gains a lot of information about its components and this can be used in the prediction of correct article usage. Some of this information can be extracted from the individual parts of the system and can be used in the application of the article usage rules.

The **natural language processor** is responsible for transforming the student's input into a form which the system can use. It can be seen as consisting of four components: the **parser**, the **lexicon**, the **grammar** and the **morphological analyser**.

The parser

The parser used in the natural language component is an indexed **chart parser** which originated from (Gazdar & Mellish, 1989). The parser makes reference to a grammar, a lexicon and a morphological analyser while determining the **syntactic structure** of the student's input. From the parser, a considerable amount of information about the noun phrase can be extracted, including the nature of any pre- or post-modifying elements such as adjectives and relative clauses, and the context in which the noun phrase occurs. Because the system was intended for use by non-native speakers of English, it was desirable that the parser could handle ungrammatical input, although the only ungrammatical input which was to be actually analysed was the incorrect usage of articles. The parsing of ill-formed input will be discussed further in Section 5.5.3.

The lexicon

The lexicon is a long list of the words the system knows. With each word is given its syntactic category and various other pieces of information, known as **features**, which tell the parser in what kind of structures it may be used. The most interesting information held in the lexicon in terms of article usage is whether a noun is a **count** or **mass** noun, and whether a noun is a **proper noun** or a **common noun.**

The lexicon which was used in *ArtCheck* originated from a Lisp lexicon implemented as part of the Alvey Morphological Tools project and described in (Ritchie *et al*, 1992). All the lexical entries, together with the features which were required by the *ArtCheck* grammar, were extracted from the Alvey lexicon and translated into Prolog.

The grammar

A context free feature grammar is used. For reasons which will be explained later in this chapter, the grammar has not been designed to cater for all aspects of English. Nevertheless, a reasonable subset of natural language is included in this grammar. The features used include number, person, tense, count/mass etc. The choice of features used was mainly determined by the article usage checking requirements rather than by a desire to encapsulate the whole of the English language.

The morphological analyser

The morphological component of a natural language understander works in conjunction with the lexicon and parser, and validates words in student input by using the appropriate morphological rules to expand the relevant entry in the lexicon. It then knows how each word in the input to the system is constructed, that is, the tense and number of a verb, and whether a noun is **singular** or **plural**. The morphological analyser can also give information about whether the **superlative** or **comparative** forms of adjectives are being used.

The morphological analyser in *ArtCheck* contains a set of **spelling rules** which enable the system to recognise many more words than the roots given in the lexicon. The spelling rules used in *ArtCheck* are given in (Ritchie *et al*, 1992). Again, these were translated from Lisp into Prolog for the purposes of this project.

All the above components of the natural language processor have a part to play in the prediction of correct article usage. To return to the example rule given above:

Rule: *Use* **the** *where a noun is modified by a superlative adjective.*

it was stated that to apply the rule, the computer would have to know whether the noun was modified, and the morphological form of any modifying adjective. The syntactic information required would be available from the parser, and the morphological information from the morphological analyser. Therefore, it can be seen that several of the article usage rules can be applied with access to this information.

5.2 Applying the article usage rules

5.2.1 A three stage process

The application of these article usage rules to natural language input not only predicts what the correct article usage will be, but detects if there is an error in article usage in the input. The next decision which had to be made was at which stage this application of the article usage rules was to take place. The first impression might be that the appropriate stage to apply the article usage rules is during the parsing process. However, this was decided against for two reasons:

- During the (chart) parsing process there are many attempts to parse a noun phrase. However, as the rest of the sentence unfolds, not all the elements of the sentence which **could** be parsed as noun phrases can be in the given context. For example, in the sentence,

  The cats play in the garden.

the word *play* could be categorised as a **noun** or as a **verb**. Once the whole sentence is parsed, it is clear that *play* cannot be parsed as a noun in this context.

- A related reason to that given above concerns post-modification. From the article usage rules given in Table 5–1, it has been seen that any post-modification of a noun, by, for example, a relative clause or prepositional phrase, can affect the choice of article for that noun. However, before the post-modifying part of the noun phrase is reached, the parser will already have decided that there is a non-post-modified noun phrase. Applying the article usage rules at this stage would be a wasted effort, as the rules would have to be applied again when the complete noun phrase was parsed. A similar argument applies to compound noun phrases.

Thus it is preferable to check the article usage of a sentence after the parsing process has been completed. The prediction of correct article usage can then be divided into three stages. Firstly, the parsing takes place. Then the parsed input is examined and the noun phrases taken out with all the relevant information which has been acquired. This is known as **extracting the noun phrases**. Finally, the article usage rules are applied to each noun phrase. This is known as **article checking**. This process is shown in Figure 5–1.

5.2.2 Five structural dimensions

It can be seen that many of the article usage rules can be applied by making reference to existing information available from the parser, lexicon and the morphological analyser. To describe this in another way, there are five structural dimensions which affect the decision about which article to choose. These are:

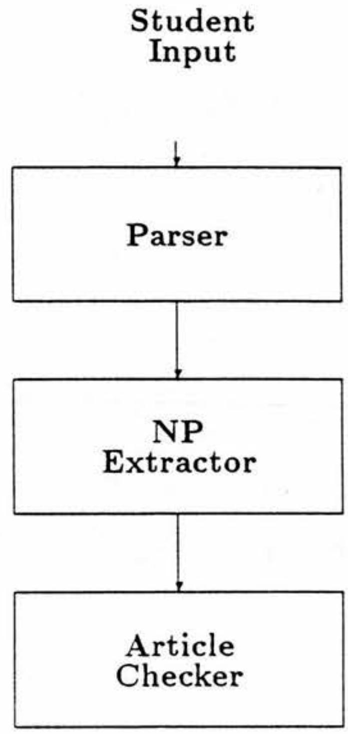


Figure 5–1: The three stages of article checking

1. Number
2. Count/mass
3. Proper/common noun
4. Linguistic environment
5. Modification

These will be exemplified in turn.

Number

Whether a noun phrase is **singular** or **plural** is very important in the decision about which article is correct. The indefinite article *a/an* can only occur with singular nouns. The zero article can usually only occur with plural nouns, but there are some notable exceptions to this. The article *the* is freely used with both singular and plural nouns.

Countability

This distinction concerns whether a noun is **count** or **mass.** A mass noun cannot occur with the article *a/an*. A count noun does not usually occur with the zero article, unless it is in the plural.

Proper or common noun

Proper nouns and common nouns obey different sets of article rules. A proper noun usually occurs with the zero article. However, if the proper noun is in the plural, or if it also occurs as a common noun, then it can occur with the definite article *the*.

Linguistic environment

Another factor affecting the choice of the article is its immediately preceding linguistic environment in the sentence. For example, where a noun is a complement of the verbs *be*, *become* and *remain*, it usually occurs with the indefinite article *a/an*. Alternatively, if a noun phrase is preceded by a **quantitative determiner** such as *all* or *both*, it usually occurs with the definite article *the*.

Modification

The fifth dimension which affects the choice of article of the noun is the presence of any modifier before or after the noun. Generally speaking, if a noun is modified, it increases the possibility that that noun can occur with the definite article *the*, even if the noun has not been mentioned before. More specific cases are given in Table 5–1, for example, if the noun is modified by a prepositional phrase including the preposition *of*, the definite article is almost always used, as in the phrase *the middle of the road*.

5.2.3 Progress so far

It can be seen from comparing the list of rules in Section 5.1.1 with the information available, as seen in Section 5.1.3, that the computer system can apply many of the article usage rules by utilising the information already in the system. These rules are given in Table 5–2.

Article	When used	Information needed
a/an	For singular count nouns	Number and count/mass noun.
	After *be,become,remain* with singular count complement	Number and count/mass noun *and* What precedes NP.
	After *such* and before singular count nouns	Number and count/mass of noun *and* What precedes NP.
	With the adjective *certain*	If premodifying adjective.
	After *half* and before singular count nouns	Number and count/mass of noun *and* What precedes NP.
the	Before *of* prep.phrases	Details of modifier.
	Where a noun is also classed as an adjective.	Category of word.
	With relative clauses	Details of modifier.
	After *all,half* and *both*	What precedes NP.
	Before *same* and *only*	Details of modifier.
	Before the superlative	Morphological information *and* Details of modifier.
	Before ordinal numbers	Morphological information *and* Details of modifier.
	Before plural proper nouns	Category of word *and* Number of noun.
	Before proper nouns which are also common nouns	Category of word.
zero	With mass nouns	Count/mass noun.
	With plural count nouns	Number and count/mass of noun.
	With proper nouns	Category of word.

Table 5–2: Article usage rules (1) - Easily implemented

The remaining cases, which will be discussed in the next section, can be seen in Table 5–3. The table includes a reference to the section in which the application of particular rules will be discussed. Those rules for which a section number is not given will be mentioned in Section 5.3.6.

Article & When used	Information needed	Section
Use **a/an:**		
For new information	Record of previous NPs used.	5.3.3
With rates and frequencies	Types of words affected.	5.3.2
Before *hundred,half*, etc	Types of words affected.	5.3.2
Before *lot of,kind of,type of* etc.	If prepositional phrase *and* List of affected words.	
Use **the:**		
For given information	Record of previous NPs used *and* Contextual information *and*	5.3.3
	General/world knowledge.	5.3.4
In comparative expressions such as *the more the merrier*	Recognition of structure	
Use **the zero article:**		
When a number precedes noun	Details of modifier *and* Which adjectives are numbers.	5.3.2
With certain singular count nouns to do with seasons, meals, time, transport, and institutions, and with certain prepositions.	List of types of nouns *and* Prepositional phrase modifier	5.3.1

Table 5–3: Article usage rules (2) - The remaining cases

5.3 Outstanding cases

The five structural dimensions discussed in Section 5.2.2 give a certain amount of information about the choice of article. It is obvious however, that more information is required. For example, nothing has been said as yet about whether a noun is **given** or **new**[1] information (Halliday, 1967). There are also a number of idiomatic uses of the articles which require more explicit information. Some of the cases of article usage which are outwith the scope of the five structural dimensions are as follows:

(i) The use of the zero article in certain prepositional phrases when the noun is of a certain **semantic category**, for example, a meal, a season, a form of transport or an institution.

[1]A noun is said to be **given** if it is clear to the hearer/reader to what it refers, and **new** if it is the first time it has been mentioned.

(ii) **Idiomatic usages** of the indefinite article in phrases to do with rates and frequencies, such as: *a dozen eggs*, *five miles a day*, etc.

(iii) The use of the definite article where the noun is known by the context. In other words, either the noun has been mentioned before, or it is implied by what has been mentioned before, or it has been paraphrased earlier in the discourse.

(iv) The use of the definite article in cases where **general** or **world knowledge** indicates that the article is known, as in the example, *The world is round.*

(v) The use of the articles where the sense of the noun phrase is **generic**, as in the example, *Dogs are lovable animals.*

These cases will be examined individually.

5.3.1 Semantic categories

The zero article is used before singular count nouns in certain cases involving prepositional phrases and some semantic categories of nouns, as follows:

- time and seasons, eg *in spring,at night*
- meals, eg *at breakfast*
- place, eg *at school,in hospital*
- transport, eg *on foot, by bus*

No semantic interpretation of the sentence is available in this system. However, in order to apply this rule, the system must have information about the **type** of noun phrase it is dealing with, or the **semantic category**. For example, it needs to know that **dinner** is a type of meal, **boat**, a type of transport, and **prison**, a place or institution. The next issue is where this information was to be included so that the system could have access to it. Three alternatives were considered:

(i) The article usage rule itself could be expanded, so that the system checked for each and every one of the possible nouns or proper nouns each time it applied the rule. This of course would be a time-consuming way of applying the rules, and also writing a rule which tested for an infinite number of proper nouns would be impossible.

(ii) The lexicon could be expanded, so that it marked exceptions with the article they occurred with and in which context. The rule would just check if the lexical entry was marked as an exception and then check the conditions for the exception applying.

(iii) The lexicon could be expanded, so that it gave the semantic category of each noun, or just of nouns whose semantic category is relevant to any of the rules. The rule would then check for the semantic category of the noun, and in the second case above, also for the presence of the relevant prepositions in in the noun phrase.

Of these three options, the final option would appear to be preferable, as it minimises the extra coding that would have to appear in the lexicon, and also the amount of testing that would have to be done in each rule. An example of a lexical entry and a rule is given below:

Lexical entry:

```
n(pn:neg,number:singular,type:count,semantic:meal)---- >[dinner].
```

Rule: *If the noun is a singular count noun of semantic category meal and the modifying preposition is one of the list [at,over,during], then use the zero article.*

The semantic feature illustrated above does not need to be restricted to the semantic categories of nouns. For example, there are some cases where the type of adjective which modifies a noun is important to the decision about article usage. For example, in the following rule:

Rule: *An article is sometimes not required when the adjective is of type number.*

An example of a lexical entry for an adjective with a semantic feature attached would be:

Lexical entry:

```
a(sem:number) ---> [eight].
```

With the semantic category feature attached to adjectives, the system will also be able to distinguish **superlative adjectives** from other adjectives, as required by the article usage rules.

5.3.2 Idiomatic article usages

In this section, the article usage in expressions of rates, frequencies and quantities will be discussed. The following sentences give examples of this:

Frequency:	*I go swimming twice a week.*
Rate:	*This car goes ninety miles an hour.*
Quantity:	*I won a hundred pounds.*

The article usage in these sentences can be expressed by the following rules:

- Use *a/an* after adverbs to do with frequency and before nouns to do with time.
- Use *a/an* after expressions of quantity and before nouns to do with time.
- Use *a/an* before quantity nouns.

The sort of information which the computer needs in order to apply these rules is which nouns are **time nouns**, or which adverbs are **frequency adverbs**. In other words, the system needs to know what sort of **semantic categories** certain words fall into. Therefore, the solution is the same as that described in Section 5.3.1. The lexical entry for each of the affected nouns will have the semantic category of the noun included. There will also be a semantic feature attached to adjectives and adverbs. For example:

Lexical entries:

```
n(pn:neg,number:singular,type:count,sem:time)--->[day].

adv(sem:frequency)--->[twice].
```

The rule for frequencies is then written as follows:

> Rule: *Use* **a** *before nouns to do with time occurring after frequency adverbs.*

The implemented versions of all the article usage rules is given in Appendix D.

5.3.3 Contextual knowledge

The definite article is used where the noun phrase can be said to be **given** information, that is, in conversational terms, it is known by both speaker and hearer. This is the most important use of the definite article. The three cases which will be discussed under this heading are where the input to the system includes the following:

- A noun phrase which has been mentioned before.
- A noun phrase which is known by virtue of the context in which it occurs.
- A noun phrase which is a paraphrase of one mentioned earlier.

Finding out which noun phrases fall into these three categories is the *"definite reference problem"* (Hirst, 1981). The system does not have any semantic or pragmatic knowledge available to it, so some other means of detecting what is **given** information must be found.

Identical noun phrases

The first case is where an identical noun phrase has been mentioned before, or a noun phrase with the same head noun[2]. The solution is to maintain a **discourse model** which keeps a record of all noun phrases which have occurred in the previous discourse. This can be used to check if a noun phrase has been used before. If it has, then it can be taken that it is **given** information and can be referred to with the definite article.

Associated noun phrases

The second case is where noun phrases are used which the hearer/reader is expected to know because of the context in which they appear, that is, they have not been mentioned before, but another noun phrase has been used which causes them to be part of the current context, for example,

> *"He got quickly into his car and tried to start* **the engine.**"

In the above example, the noun *engine* can be referred to with the definite article because the use of the word *car* means that *engine* is given information. There are many uses of nouns in ordinary English which fall into this category, and it is essential that a system which claims to be able to predict correct article usage makes some attempt to detect this type of given information. One possibility was of course to restrict the domain to essentially one context, in which all that the hearer/reader will be expected to know, including a selection of paraphrases, would be encoded as part of the domain. This would not have been satisfactory, because such a restricted domain would mean that the system would have extremely limited use. Thus, an alternative solution was proposed.

Sanford and Garrod have shown through a series of reading time experiments, that key words in a text evoked **scenarios**, that is, they gave the reader the *"knowledge*

---

[2]The **head** noun is the primary noun in the noun phrase.

of settings and situations as constituting the interpretative scenario behind a text" (Sanford & Garrod, 1981). In essence, they *"extended the domain of reference"* for the reader. The experiments carried out by Sanford and Garrod showed that readers were able to make quicker references to words which were **semantically related** to what had gone before, and which could be assumed to exist in the **extended domain of reference**, than to non-related words in the text. They suggested that once a key word had been used, words which were semantically related to it were already available to the reader, and could be referred to without any extra inferences being made. Sanford and Garrod discussed special links which showed what sort of semantic relationship there was between the noun and each of the associated entities. Examples of such links were KIND-OF, ASPECT-OF and SUPERSET-OF.

One of the problems in implementing the article usage rules was how to detect that certain noun phrases were accessible to the reader and could thus be used with the definite article. This problem was solved, at least partially, by implementing the ideas put forward by Sanford and Garrod. Within the system, the lexicon already contained certain pieces of syntactic information about each word it contains. The lexicon was augmented by the addition of certain other information about semantically-related entities and concepts. It was found that it was not necessary for these purposes to know what the relationship was, but merely to know that there was one. The semantically-related nouns were thus attached to the noun's entry in the lexicon in the form of a list.

As a result of this, the augmented lexicon includes, in addition to information about a semantic category of a noun, a list of all the associated entities that can also be referred to[3]. The semantically-related items evoked as a result of any lexicon look-up are also recorded in the discourse model as "previously mentioned information" which can be referred to again as given information. No extra infer-

---

[3]This has been implemented for some, but not all, of the nouns in the lexicon in *ArtCheck*.

ences are required to access the related information. This can be illustrated with the following example:

The following text is taken from Watership Down by Richard Adams (Adams, 1972):

> *"General Woundwort was* **a singular rabbit**[4]. *Some three years before, he had been born - the strongest of a litter of five, in a burrow outside a cottage garden near Cole Henley. His father, a happy-go-lucky and reckless buck, had thought nothing of living close to human beings, except that he would be able to forage in their garden in the early morning. He had paid dearly for his rashness. After two or three weeks of spoiled lettuces and nibbled cabbage-plants, the cottager had lain in wait and shot him as he came through the potato-patch at dawn. The same morning, the man set to work to dig out* **the doe** *and her growing litter."*

In the first sentence, the noun *rabbit* is introduced. In the original lexicon, this would have the following lexical entry:

```
n(pn:neg,number:singular,type:count,semantic:animal)---> [rabbit].
```

In the augmented lexicon the entry for a rabbit would be:

```
n(pn:neg,number:singular,type:count,semantic:(animal,
    [animal,bunny, doe,buck, hutch] ---> [rabbit].
```

Thus, the nouns *animal, bunny, doe and buck* are all added to the student model as available referents. Then, when in the last sentence, the noun *doe* is introduced, the system can immediately know that the usage of the definite article with this noun is acceptable.

---

[4]the emphasis here has been added

Chesterman presents the same idea, in that he distinguishes **referent sets** and **entity sets** (Chesterman, 1991), (discussed in Section 4.1). A referent set contains all the referents for a given noun phrase, whereas an entity set contains the referents and all the associated referents that could also be *"located"*, or known about. In this case, the appropriate entity set for each noun was included in the lexicon.

The result of these changes is that the context which is evoked by particular words is available to the system as **known information**.

Paraphrases

The third case described above involves a noun phrase being used in a paraphrased form. The noun has been mentioned before, but in a different form, and semantic and pragmatic knowledge is required to make the connection between the noun phrase and its antecedent. An example might be:

> "**British Airways** *announced that due to the recession they would be making redundancies in the spring.* **The airline** *regretted the action but said that the matter was beyond their control."*

In this example, the proper noun phrase *"British Airways"* is later referred to as *"the airline"*. Because the second noun phrase is a paraphrase of the first, it is given information which can be referred to with the definite article. To know which noun phrases are paraphrases of others requires a level of general and pragmatic knowledge which a computer does not have available to it, though in a restricted context, some of this information could possibly be included. Another example is the noun phrase *the action* in the same passage. In this context *"the action"* is a paraphrase of *"British Airways are making redundancies in the spring"*. It would be impractical for a computer to be able to acquire the sort of pragmatic knowledge required to trace back the antecedents of *action*. Other nouns which are similar to *action* in this way include *problem*, *situation*, *result*, *effect*, *matter* etc.

Therefore, it will not be possible for the computer to detect that the paraphrased noun phrase is given information, which is obviously a limitation of the system. A similar problem is discussed in the next section.

5.3.4 General/world knowledge

While reading a text, human readers make use of the vast amount of general or world knowledge they have access to, in order to assimilate the information before them. The problems of so-called intelligent computer systems, who do not have access to this mass of knowledge, are well-known and far-reaching. In normal discourse or text, the speaker or writer makes certain assumptions about what a hearer or reader would be expected to know, and can thus treat such information as **given**, even though it has not been previously mentioned in the discourse or text. In the case under discussion here, the problem is that the system does not have access to these assumptions, yet still has to make a decision about whether information can be said to be **given** or not.

To some extent, this system is not able to make decisions in these cases. However, some attempt has been made to make this less of a problem. The lexicon will contain markings for certain words which can be said to be salient in (nearly) *all* domains. These are nouns which can always legitimately occur with the definite article, for example, *world*, *moon*, *sun*, *press*, *media*, etc. These are known as **indexical nouns**. The definite article used in this case is known as the *"indexical the"* (Quirk *et al*, 1972).

5.3.5 Generic reference

As discussed in Section 4.2.2, the distinction between **generic** and **specific** reference is that generic reference is reference to a class of entities, and specific reference to a particular member or members of a class. For example, a **particular** *cat* is being referred to in the sentence,

My cat was run over yesterday.

whereas it is the **class** of cats which is referred to in the sentence,

Cats are run over every day in this city.

In general, the zero article is used for mass and plural count nouns where either specific or generic reference is intended, unless, in the case of specific reference, other rules override that generalisation and a definite article is used. However, where a singular count noun is referred to in the general sense, the definite article *the* is used, as in the example,

The cat has been a domestic pet for thousands of years.

Thus the article usage rules for generic reference are slightly different to those previously defined for specific reference. The system does not have the necessary information to know if specific or generic reference is intended, because the function of a noun phrase depends on the context and the sense in which it is used.

One suggestion which was considered was to consider the tense and aspect of the verb occurring where there was generic reference, in order to identify any generalisations which could be made as to particular contexts where such reference is intended. However, the result of this was not successful as there were no key constructions found which could trigger off the recognition of a generic usage of a noun phrase.

Thus, the generic usage of the noun phrase cannot be detected by this system and special rules for this case cannot be implemented. The choice remains then, as to whether the system is to forget about generic reference altogether and just deal with specific and unique reference, or whether more general rules will be used which accept both specific and generic alternative usages of a noun phrase. Both these alternatives have been tried in different modes of the system. This will be explained in more detail in Section 5.5.4.

5.3.6 Some exclusions

It has been seen that some uses of the definite article cannot be detected because of the problems with contextual information discussed above. In addition, several other article usage rules have been excluded, and are listed below.

- Some idiomatic expressions, for example, *a lot of*, *a kind of*, *a sort of*, *the more the merrier.*
- Some particular time expressions, for example, *next year*, *every day*, *last year*, etc.
- Conjunctions where two noun phrases were treated as one, for example (from the Oulu data),

  > "**Developed and developing countries** *should start co-operating in order to save rainforests which are left."*

- Answers to questions, for example (from the Oulu data):

  > *"It is the main carbon sink."*

These rules were not excluded because their implementation would have caused difficulties. Instead, it was felt that some of the more minor and exceptional cases should be left out, to save having a very large set of less common rules. The aim of this part of the system was not to find a complete watertight set of article usage rules, but to demonstrate how a computer was able to apply such rules.

5.3.7 Summary

To summarise, in addition to the structural dimensions discussed in Section 5.2.2, three changes were introduced to the lexical entries to enable the application of more article usage rules:

- The semantic category feature was added to nouns in the lexicon. This was used in the application of more idiomatic rules.

- Entries for nouns in the lexicon were augmented by a list of semantically related nouns. This enabled the system to detect that a noun was **given** in the discourse because a semantically related noun had been used before.

- Some nouns were marked as **indexical** in the lexicon. This was an attempt to partially overcome the problems presented by the students' general/world knowledge to which the system does not have access.

5.4 The article usage knowledge base

5.4.1 Production rules

The expert model is the information which the system has about the domain. The expert model for this domain is a set of article usage rules. It has been seen that the form of each article usage rule is:

When conditions X apply, use article Y.

Such rules can easily be implemented as **production rules**. Table 5–4 shows the final rule set which was implemented in the system. Here, the information which was used to determine the correct article is detailed in each case. This includes: whether the noun is a common or proper noun (CN/PN); the number of the head noun (Num); whether the head noun is mass or count (Count); and the linguistic environment of the noun (Environment).

Rule No.	Head Noun PN	Num	Count	Environment	Article used
1	CN	sing	count	New information	a/an (or the)
2	CN	plural	count		zero
3	CN		mass		zero
4	CN			Given information	the
5				Modifier: preposition *of*	the
6	CN	sing	count	Preceded by: preposition Sem.cat: season, meal,transport or place	zero
7	CN	plural	count	Modifier:none Semantic cat.: quantity	a/an
8	CN	sing	count	Preceded by: *how often* adjective Modifier: none Semantic cat: time	a/an
9	CN	sing	count	Preceded by: number + plural noun Modifier: none Semantic cat: rate	a/an
10	CN	plural		Lexical: HN categorised as adjective	the
11	CN	sing	count	Preceded by: *be* verb	a/an
12	CN			Modifier: superlative adjective	the
13	CN		count	Modifier: number	zero
14				Modifier: relative clause	the
15				Preceded by: *all*	the
16				Modifier: *same*	the
17				Modifier: *only*	the
18				Modifier: ordinal number	the
19	PN				zero
20	PN			Word also classified as CN	the
21	PN	plural			the
22	CN	singular		Indexical noun	the
23	CN			Modifier: adjective	the
24	CN	singular	count	Preceded by: none (subject) One of [*man,woman,age,family, cost,part,life,word*]	zero
25	CN	singular	count	Preceded by: *half*	a/an *or* the
25a	CN			Preceded by: *half*	the
26	CN	singular	count	Preceded by: *such*	a/an
26a	CN			Preceded by: *such*	zero
27		singular	count	Modifier: *certain*	a/an
27a				Preceded by: *certain*	zero
28				Noun = *one*	zero *or* the
99				First letter of next word is vowel or next word is *hour*, and word does not begin with *uni...*	*an*

Table 5–4: Final article usage rule set

Table 5–5 gives the English descriptions of all the implemented rules. In rule 1, the brackets indicate the wider selection of articles allowed by the weaker model for article usage. This will be discussed in Section 5.5.4.

The student's knowledge of the rules can be represented in the same way. However, it was necessary to find a means of representation which shows how the individual production rules are related to each other. This is particularly useful when it comes to modelling the student, as will be discussed in detail in Chapter 6. For this reason, the **genetic graph** was chosen as a means of representation.

Rule	Description
1	Use *a* (or *the*) before a singular count noun.
2	Use *zero* before a plural count noun.
3	Use *zero* before a mass noun.
4	Use *the* where a noun or semantically related noun has been mentioned before.
5	Use *the* where a noun is modified by a prepositional phrase beginning with *of*.
6	Use *no article* in certain idiomatic prepositional phrases involving the use of nouns to do with transport, seasons, meals or institutions.
7	Use *a* before nouns modified by quantity words such as *hundred* or *dozen*.
8	Use *a* before nouns to do with time occurring after frequency adverbs.
9	Use *a* before nouns to do with time occurring after frequency noun phrases.
10	Use *the* for common nouns which also occur in the lexicon as adjectives.
11	Use *a* before singular count nouns which occur as a complement of the verb *to be*.
12	Use *the* before nouns which are modified by superlative adjectives.
13	Use *no article* before nouns which are modified by cardinal numbers.
14	Use *the* before nouns which are modified by a relative clause.
15	Use *the* before nouns which are preceded by the word *all*.
16	Use *the* before nouns which are modified by the word *same*.
17	Use *the* before nouns which are modified by the word *only*.
18	Use *the* before nouns which are modified by ordinal numbers.
19	Use *no article* before proper nouns.
20	Use *the* before proper nouns which are also categorised as common nouns.
21	Use *the* before plural proper nouns.
22	Use *the* before indexical, or always definite, words like *world* and *sun*.
23	Use *the* before nouns which are modified by an adjective.
24	Use *no article* before the nouns *man,woman,life, part,age,family,word,cost*.
25	Use *a* after the word *half* and before a singular count noun, and *the* after the word *half* and before a plural or mass noun.
26	Use *a* after the word *such* and before a singular count noun, and *no article* after the word *such* and before a plural or mass noun.
27	Use *a* where a singular count noun is modified by the adjective *certain*, and *no article* where a plural or mass noun is modified by the adjective *certain*.
28	Use *no article* or *the* before the noun *one*.
99	Use *an* in preference to *a* when *a/an* is indicated, and where the following word begins with a vowel, or where the following word is *hour*, with the exception of words beginning with *uni*....

Table 5–5: Article usage rule descriptions

5.4.2 The genetic graph

In a genetic graph, the nodes of the graph represent the rules of the knowledge base and the edges or links between the nodes represent the evolutionary learning paths between the rules. The original genetic graph (Goldstein, 1982) used four different types of links: a generalisation/specialisation link; a refinement/simplification link; an analogy link; and a deviation/correction link. Additional links may be added as required (Brecht (Wasson) & Jones, 1988).

The genetic graph offers several advantages over a straightforward rule base. It gives a hierarchical representation of the domain knowledge, showing how individual rules are related. It allows the representation of all student knowledge which is a subset of expert knowledge, plus representation of known misconceptions. It can also be used to represent new links, for example, deviation links to dynamically generated malrules, which are added as the system is running. In addition, the genetic graph makes it easier to select appropriate teaching material. The links represent a possible learning path from one rule to the next, so it is clear from what the student knows already what can be taught next. Fuller explanations of misconceptions can also be given because of the extra information encoded in the genetic graph. Finally, it is easy to understand and a clear form of representation.

To represent the article usage knowledge base there are three kinds of link:

- **A specialisation/generalisation link**

  This shows that one rule is a more specialised or more general version of another rule. It is a bi-directional link. It is used in the explanation component of the system (see Chapter 7) to decide which rule should be **taught** if there are two rules which both select the same article.

- **A priority link**

  This shows that one rule has **priority** over another. It is used in the article checking component of the system to show which rule to **select** if two rules apply which indicate that two different articles should be used. It is the main tool for conflict resolution and will be discussed further in the next section.

- **A deviation link**

  A deviation link links an article usage rule with an incorrect rule used by the student. This is discussed in Chapter 6 on the student modelling part of the system.

Figure 5–2 shows a portion of the genetic graph for article usage. The rules shown in this portion mainly refer to singular count nouns. Where no link is shown between the rules, a choice of articles is available. The priority link points to the rule which has the most priority. The complete set of edges for the article usage domain is shown in Appendix D.

5.4.3 Conflict resolution

A major drawback of the article usage rules described in traditional grammars is that they do not indicate which rules overrule other ones, when more than one rule applies. The rules are not mutually exclusive and the conditions of the rules applying are very different and concern different aspects of the noun phrase, for example, the modification of the noun phrase, the context in which it occurs, or the number of the head noun. It is therefore an extremely common occurrence for more than one rule to apply at once, and for the co-applying rules to indicate that different articles should be used. For example, one of the errors found in the Oulu exercise (see Section 4.3.2) was as follows,

> *" *Destruction of rainforests is a global problem, not only a national one."*

The word *destruction* is usually regarded as a mass and abstract noun. The student is then probably applying the rule about mass nouns occurring with the zero article.

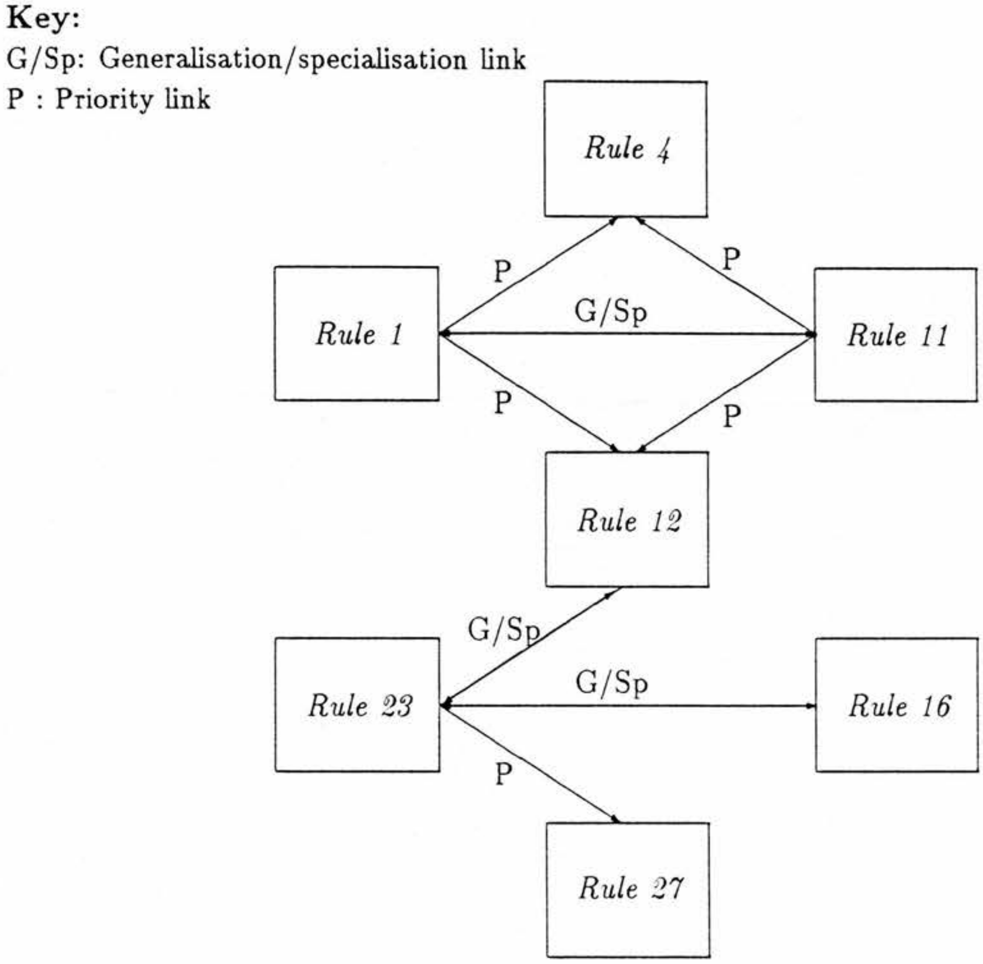


Figure 5–2: Portion of the genetic graph for article usage

However, another rule applies, which the student may or may not be aware of. If a noun is modified by a prepositional phrase with the preposition *of*, the definite article *the* is used. This rule takes priority over the previous rule, so in the sentence above should read:

> *"The destruction of rainforests is a global problem, not only a national one."*

Books which teach students about English articles (for example, (Westlake *et al*, 1988; Arnold *et al*, 1988a)) do not seem to include any indication of which rules take priority over others. This is probably because it is a very difficult issue. However for a computational analysis, it is essential that this is tackled. The following paragraphs describe how conflict resolution has been implemented in *ArtCheck*.

A number of alternative ways of describing the priorities different article usage rules have over each other were examined.

- **Weighting**

  The first strategy used to capture the relationships between different rules was **weighting**. Each rule was assigned a number between 1 to 9 depending on how much priority it had over other rules. Thus, where a rule with weighting 9 fired, it had priority over all other rules which applied. The assignment of weightings was done by finding examples of each of the combinations of the 29 rules applying. However, this strategy was unsuccessful in several cases. As an example, consider the following:

Example:	Rule 19 *(Use zero for a proper noun)* has priority over Rule 11 *(Use a/an after the verb <to be>)* so had a higher weighting, (for example, *I am Tom*). Rule 11 had priority over Rule 23 *(Use <the> before adjectives)*, so had a higher weighting, (for example, *I am a very good student*). So Rule 19 should have a higher weighting than Rule 23. However, Rule 23 has priority over Rule 19 in some cases, for example *I am the good-looking Tom, not the ugly one*, so the weighting system failed.

- **Semantic/lexical information**

  The next strategy which was tried was to do with the type of information occurring in the pre-conditions of the rules. It seemed that the more exceptional cases, that is, rules which required explicit information about the semantic category of the head noun, and those that insisted on a specific lexical item, took priority over other rules. So where there was a combination of rules applying, the system indicated that if one of the rules included specific lexical information or information about semantic categories, that rule took priority over the others. If none of the rules were of this type, no rule took priority, and the set of articles chosen was the conjunction of the articles selected by all the rules.

- **The genetic graph**

  The genetic graph was discussed above as the means of representing and relating the article usage rules. It then seemed that it would be a useful way of showing which rule had priority over another. Therefore a priority link was added to the genetic graph. The original priority links implemented linked the particular rules discussed above, where the pre-conditions of the rule included semantic and pragmatic information, to other rules which specified that a different article should be used. Other priority links were added as the system was developed. Where no priority link exists and two rules fire, there is a choice of articles which can be used in that case.

The strategy of implementing conflict resolution via the genetic graph has been reasonably successful. The major dilemma is that the more priority links there are, the more specific and accurate the system can be about article usage, but more prone to possibly making misjudgements. If there are less priority links, the system is less likely to make 100% successful predictions about article usage, but in more cases there will be a choice of articles which can be used, which makes the system weaker. If the predictive power of the system becomes too weak the system will lose some of its ability to detect article usage errors.

5.5 Implementation of article checking in *ArtCheck*

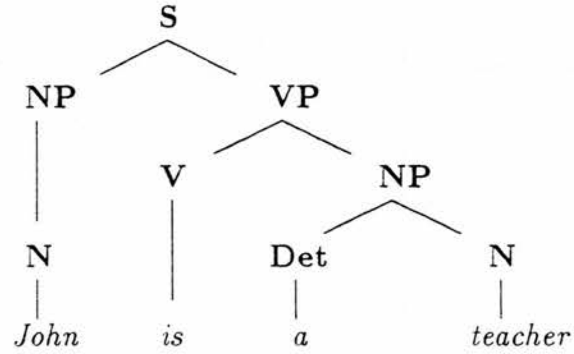
5.5.1 The article checking process

This section will describe how the implemented system takes a sentence of student input, processes it, and checks whether the article usage is correct. The system processes the complete sentence, regardless of whether the student is in GAP mode (where the student enters the appropriate article into a sentence presented by the system) or in WRITE mode. More details of the two modes are given in Appendix A.

The first stage is to parse the sentence. For example, the sentence,

John is a teacher.

results in the following parse:



All parses, if there is more than one, are considered at this stage. The resolution of multiple parses is done at a later stage.

The next stage is to extract the noun phrases from the parse. This is done by working through the parse and extracting the nouns and any structural or lexical information which may be relevant. The output from this procedure contains the following elements: determiner (Det); head noun (HN); proper/common noun (PN?); number; count/mass (Count); linguistic environment (Env); modifier; and semantic category.

The output from *John is a teacher* at this stage is:

Det	HN	PN?	Number	Count	Env	Modifier	Semantic cat.
zero	John	yes	singular	count	subject	nil	nil
a	teacher	no	singular	count	*is*	nil	nil

The output is then passed to the **article checker**. It applies the article usage rules to each noun phrase in turn. Each noun phrase is tested against each of the rules; when a rule fires it is added to the set which have fired for that noun phrase.

There are two noun phrases in the sentence *John is a teacher.* With the first noun phrase, *John*, the rule **19** (see Table 5–4) fires. This rule says that the zero article should be used with a proper noun. With the second noun phrase, *a teacher*, the rules **1** and **11** fire. Rule 1 applies to a singular count noun, and Rule 11 applies to a singular count noun which is a complement of the verb *to be*. Both rules select *a/an* as the correct article.

The next stage is to see if any of the rules which have fired have priority over the others. In the case of the first noun phrase, only one rule has fired so there is no question of priority. In the second case, two rules have fired, but both indicate the use of the same article. Therefore there is no priority link in this case either. The latter rules are linked by a **generalisation/specialisation** link in the genetic graph. This will mean that in the explanation component of the system, the more specialised of the two rules, rule 11, would be explained to the student if an error had been made.

The information which is then passed from the article checker to the student modelling component of the system is as follows:

- Correct or incorrect
- Details of noun (as above)
- Determiner used
- Rules which fired
- Determiner chosen

For this example, the output from the article checking component is shown below:

Correct/Error	Noun details	Det.used	Rules firing	Det. chosen
Correct	*John*, proper noun, etc.	zero	19	zero
Correct	*teacher*, common noun, etc.	a	1 & 11	a

5.5.2 Limitations of the grammar

The system was originally implemented with a restricted grammar, a small number of features, and a 6,000 word lexicon. As the system was developed, the grammar was extended to incorporate more complex sentence structures and longer sentences. However, the original set of features, which was quite small, was retained. For example, there are no features in the system to do with selectional restrictions of the verb. For example, the sentence

**I think a book.*

is parsed by the system, whereas having more features would be able to rule out this sentence.

One of the reasons for not having many features was to do with ill-formed input, as will be discussed in Section 5.5.3. Therefore, the extension of the grammar created two problems for the system:

- Too many parses were found which slowed down the parsing process.
- Deciding which was the correct parse to choose was difficult without semantic information, and put too great a burden on the strategies for resolving ambiguity.

Therefore, the system is able to cope with a subset of natural language, but falls short of being able to accept completely free student input. This was acceptable because the aim of the thesis was to show how the system could analyse article usage errors and not to be able to parse any English sentence. One of the ways the system could be improved is by extending the feature system, but this was not carried out, firstly, because of time restrictions, and secondly, because it was not central to the contribution of the thesis. However, with a complete and extended natural language interface the system would be able to parse more sentences and thus detect article usage in any natural language sentence.

5.5.3 Parsing ill-formed input

It was mentioned in Section 5.1.3 that in an ICALL system some attempt must be made to take account of the ill-formed input of the language learner. There are different types of ill-formed input: unrecognised words; incorrect agreement; and omitted words being three examples. Various proposals have been made for understanding incorrect agreement (Kwansy & Sondheimer, 1981; Kurtz *et al*, 1990) and spelling/typographical errors or omitted words (Weischedel, 1983; Mellish, 1989). In the system described in this thesis, the only ill-formed input of interest to the system is that concerned with article usage errors. However, it is desirable that the parser should be able to understand ill-formed input to the extent that it can extract from it the necessary information to check the student's article usage, although this is not an easy area to implement. It was originally decided to have a grammar with few features and relaxed agreement of nouns and verbs, in order that ill-formed input would be acceptable to the parser. It was not necessary for the parser to know whether the input (apart from article usage) was ill-formed or not. However, this presented difficulties for the system in that too many parses were found and it was difficult for the system to decide which was the correct one. This was more of a problem for this system because there were already too many parses being found due to the lack of features in the grammar, as discussed in Section 5.5.2. Therefore, the relaxation of agreement of nouns and verbs was abandoned by the system.

ArtCheck is able to deal with unrecognised words, by firstly asking the student to retype the word, and then, if the word is still unrecognisable, asking the student to enter the word in the lexicon. It does not at present deal with other kinds of ill-formed input. Suggestions as to how *ArtCheck* could be extended in this respect are described in Section 9.3.1.

5.5.4 A version of expert knowledge for specific and generic noun phrases

The final rule set for article usage was given earlier in Table 5–5. This is a strongly predictive model which, together with the priority links often predicts one article. However, due to several factors, including exceptional usages of articles, contextual information which the system does not select, and the inability of the system to distinguish between specific and generic noun phrases, the predictions will sometimes be incorrect. On the basis of this, it was decided that a slightly weaker model would be implemented alongside the strongly predictive one, which would make allowances for these factors. The main motivation for this was for the system to be able to predict article usage correctly when a noun phrase was being used with the generic function. It was discussed in Section 5.3.5 that the system is not able to detect the usage of a noun phrase with this function.

It was decided to implement the weaker model in the WRITE option and the stronger model in the GAP option. This decision was made because in the GAP option, the sentences which the system would have to parse are within the system's control and therefore the reliability of the system's responses can be ensured, whereas in the WRITE option, a less predictive but more generally correct model is to be preferred.

The only variation between the weak and strong models, as implemented in the WRITE and GAP modes of the system respectively, is that, in Rule 1, the strong model only allows the indefinite article *a/an*, whereas the weak model allows both the indefinite article *a/an* and the definite article *the*. Allowing *a/an* or *the* before singular count nouns takes account of both specific and generic usages of the articles with singular count nouns, for example:

I looked out the window and saw a giraffe. (specific function)

The giraffe has a very long neck. (generic function)

Where a plural or mass noun is used, the zero article is usually used regardless of whether the function of the noun phrase is generic or specific.

5.5.5 Resolving ambiguous parses

As was described in Section 5.5.2, one of the expected problems of natural language processing which this system has had to cope with is the problem of ambiguous parses. There are often two or more perfectly correct interpretations of the same sentence. In natural language processing, there are often also several implausible readings of the same sentence, which the system is unable to discount without semantic and pragmatic information. For example, consider the sentence,

The girl is singing.

There should be only one reading for this sentence, which is:

The girl is singing
NP VP

However, the word *singing* can be interpreted as a noun as well as a verb and another parse may be produced:

The girl is singing
NP V NP

It was mentioned above that one of the limitations of the natural language component of this system is that it does not have a rich feature system. The result of this is that many implausible parses of sentences are often found. The philosophy of this system has been that only noun phrases and article errors are of interest to the system and ungrammatical input in any other area will be ignored. The same principle applies to multiple parses. If the parsing ambiguity does not affect the system's prediction of correct article usage it is treated as irrelevant by the system.

Therefore, where ambiguous parses are given for a sentence of student input, the following procedure is followed: Firstly, the noun phrases are extracted from each parse. Secondly, the article usage for each parse is checked. Finally, if there are multiple sets of results which came from different parses, one result only is selected.

Strategy for resolving multiple results

At this stage the system is dealing with a list of results from the article checking. The system then has to decide which set of answers it is going to consider as the correct answer. As mentioned above, there is no semantic or pragmatic knowledge involved, so the decision made by the system has to be based on other factors. The proposed formula for selecting the correct result from a set of possible results originally included the following strategies:

Action	**Why?**
Remove duplicates :	If the results of article checking from different parses are the same, then this indicates that the difference between the parses was not relevant to the noun phrase or the choice of article.
Choose the mostly correct answer :	This was suggested to give the student the benefit of the doubt.
Take the minimum number of NPs :	If the parser parses a sentence incorrectly, it is more likely to decide non-NPs are NPs than the other way round. It is also preferable to omit an NP altogether than to confuse the student by calling words noun phrases when they should not be.
Take the solution with the most rules applying :	To maximise the amount of information available.

These strategies were tried in various combinations. It was found that the strategy of *"choosing the mostly correct answer"* did not work when the student made an error. So this strategy was omitted.

The formula which has been implemented is as follows:

IF more than one answer after article checking due to multiple parses.
THEN eliminate duplicates
  IF all answers are the same except for the rules applied

(ie same result, different reason)

THEN:

Choose that with most reasons (most rules applying)

ELSE

Find the answer with the least NPs

IF there is still a tie

THEN SELECT answer with the most reasons

This formula is generally successful in selecting the result which corresponds to the correct interpretation of the sentence. More details on the sorts of sentences the parser successfully deals with are given in Chapter 8 on the evaluation of the system.

5.6 Conclusion

In this chapter, it has been shown how a set of article usage rules can be implemented as the expert model in an ICALL system.

A traditional rule set mainly taken from English grammar books was chosen as the starting point. The rules were considered from the position of what sort of information a computer would need to apply them. It was seen that the natural language processor provides a lot of the structural information necessary to apply the rules. It is obviously advantageous to have a computer system which can check article usage from natural language input, rather than checking the article usage from pre-processed information about a noun phrase. Thus it can be seen that it is essential to have the natural language processor because it can be used as an information source when applying the article usage rules.

Section 5.3 showed how other rules using non-structural information could be applied. The semantic categories of noun phrases have been included in the lexicon, giving information which can be used to apply some of the more idiomatic article usage rules. In addition, a list of associated entities is included with a noun's entry in the lexicon, thus allowing nouns which have been brought into the context by the mention of an associated noun phrase to be referred to with the definite article.

Section 5.4 described the actual process of applying the article usage rules. The final set of rules implemented was given, and the procedure for applying them. The method of conflict resolution used in the system was described. From a computational point of view, conflict resolution is the key to accurately predicting article usage. For students learning to use articles, it is also a problem, as the grammar books and teaching material do not seem to mention which rules should take priority when more than one applies, which happens in many cases. Finally, the problem of resolving ambiguous parses was discussed.

Having predicted what the correct article usage within a noun phrase should be, and detected any article usage errors in the student input, the next stage in processing is the student modelling component, which keeps a record of the student's successes and failures with articles and attempts to analyse any errors which occur. This will be discussed in the next chapter.

Chapter 6

Dynamic student modelling

The student model is a key feature in an Intelligent Computer-Aided Instruction (ICAI) system, and represents the current state of the system's knowledge of the student. Recently, as intelligent techniques have been used more in Computer-Aided Language Learning (CALL) systems, student modelling has become a relevant area in the domain of language learning.

This chapter will describe the way in which student modelling is carried out in *ArtCheck*. Section 6.1 describes the structure of the student model. The remaining sections describe each of the processes in the modelling of the student's knowledge. Section 6.2 describes how the student model is initialised when the student first starts using the system, and how the information passed to the student modelling process from the article checking process is used to maintain the student model. Section 6.3 describes the process of analysing the student's errors and constructing a mal-rule to account for them, and Section 6.4 describes how this new information is then incorporated into the student model.

6.1 The structure of the student model

6.1.1 The genetic graph as a student modelling tool

As discussed in Section 5.4.2, expert knowledge is represented as a genetic graph. This is a useful form of representation from the point of view of domain knowledge, as it demonstrates the hierarchy of the rules and the general structure of the information contained within the domain knowledge. The genetic graph is used primarily as a user modelling tool, allowing the student knowledge to be overlaid on top of the expert knowledge. Where the user's domain knowledge is a **subset** of the domain knowledge, this can be shown clearly and simply using a genetic graph. Where the user's domain knowledge deviates from the domain knowledge, this can also be represented on the genetic graph using **deviation links**. Deviation links attach a mal-rule to the domain knowledge represented in the genetic graph. Section 6.3.5 will illustrate how deviation links and mal-rules can be generated **dynamically**. In this way, the genetic graph is being used as a **dynamic user model** which is less constrained by limits set down by the system designer than the bug library.

As mentioned in Section 1.1.3, the user model in *ArtCheck* will be known as a **student model** because, in the case of instructional systems, the user of the system can reliably be described as a **student**. The domain rules which appear in the student model in *ArtCheck* have been described in Section 5.4.2. As the student uses the system, the information in the student model is continually updated. This information refers to student rules and student edges in the genetic graph which the system believes that the student has acquired.

The next section will give details of the kind of information held in the student model in *ArtCheck*.

6.1.2 The contents of the student model

The student model can contain various pieces of information about the student, including the student's knowledge of the domain and various other aspects of the student, such as those described in Section 2.2. It is important not to fall into the trap of holding information in the student model simply because it is available. Information should only be held about the student if the system has a use for it (Self, 1990). In *ArtCheck* there are two levels of information held in the student model. At one level, there is **short term information** which can be discarded at the end of one session. At another level, there is **long term information** about the student which is retained from one session to another.

Short term information held about the student consists of the following:

- **Noun phrases used**

  A record is kept of all the correct and incorrect noun phrases which the student has used in the current session with the system. Obviously, the system needs to know the most recent noun phrase so that it can include it in the explanation given to the student. The other noun phrases are retained to enable the system to analyse the student's errors. The incorrect and correct noun phrases observed provide evidence to support the system's hypotheses about any mal-rule with which the student may be operating.

- **Complete sentences used**

  The system also keeps a record of complete and unparsed sentences. These are used to improve the explanation facility, so the system can repeat the sentence back to the student when giving an explanation. They are also used when explaining to the student about a mal-rule, when there is evidence for the mal-rule in errors originating from a number of complete sentences.

- **Diagnosed mal-rules**

  The system keeps a record of mal-rules which it has diagnosed and explained to the student, to avoid repetition in the explanations. The diagnosis of a mal-rule will be described in Section 6.3.

Long term information held about the student consists of the following:

- **Student outline**

  After the student has finished using the system, the system retains a snapshot of the student in the form of the student outline. This includes the student's level of ability in article usage, and the student's preferred learning strategy. This will be described in more detail in Section 6.2.5. During the running of the system, this enables the system to tailor the explanations to the student, and avoids the need to re-initialise the student model when the student uses the system again.

- **Rules acquired by the student**

  The most obvious piece of information which will be held about the student concerns the article usage rules which the system believes the student has acquired. The number of rules the student has demonstrated she knows gives an indication of her level of ability in article usage and is incorporated into the student outline. When the student makes an error, the explanations given differ depending on whether the student is thought to know the particular rule which has been violated or not. Therefore, in between the student's sessions with the system, the system retains the information about which rules it believes the student has used, and to a certain extent, the number of times each rule has been used. This will be discussed in Section 6.2.3.

- **Links between rules acquired by the student**

  The genetic graph is the chosen form of representation for the student model. With this form of representation, the relationship between article usage rules is represented as the edges of the graph. In terms of the student's knowledge, the student can be said to have acquired a link when she has acquired the rules at either end of that link. This information gives the system an accurate picture of the student and the amount of domain knowledge acquired. Some of the links on the graph, which may or may not be acquired by the student, show the **priority** that some rules have over other rules. The explanation facility can use this information about the student to give more individualised explanations. This will be explained in Section 7.3.3.

6.2 Building and maintaining the student model

6.2.1 Initialisation

As will be described in Chapter 7, the system **adapts** its explanations depending on the level of ability and learning preferences of the student. Adapted explanations are given by the system after each sentence entered. It is essential then that the system starts with some kind of student model, even if it is a very general one. One alternative is for all students to start with the same student model, representing the average student. Another alternative is to attempt to acquire some information in order to **initialise** the student model, and this is preferable because the information will be more reliable if it is obtained directly from the student (Self, 1990). This gives the system some basic information which it can use when generating the first explanations. As the student progresses with the system, the student model is refined and the explanations adapt to the student accordingly.

Thus, if the student is unknown to the system, the system makes a few enquiries to establish an initial student model. It first asks the student to indicate his/her general level of ability in English. The three levels available are **Beginner**, **Intermediate** and **Advanced**. These levels are used because they correspond to the terms used in many language schools and text books.

The second question the student is asked concerns learning strategies. The student is asked whether she prefers to learn by concentrating on the **rules** of English article usage, or by seeing **examples** of articles in use. Students can optionally indicate that they have no preference. Information about different learning strategies is sought because it has been shown that students differ in the strategies that they use when learning a foreign language (Rubin, 1987). This information will be used by the explanation component of the system in adapting the explanation to the student. The use of learning strategies in *ArtCheck* is discussed in Section 7.2.2.

If the student is already known to the system, it is not necessary for the system to make these enquiries. The information about the student's general level of English and learning preference is retained as part of the student model, which the system is able to retrieve.

The following sections will describe how the student model is maintained.

6.2.2 The input to the student model

It was described in Section 5.5.1 that when the article checking process is complete, certain pieces of information are passed to the student modelling component of the system. These are:

(i) A marker indicating that the noun phrase is **correct** or **incorrect**.
(ii) Information about the noun phrase, and the environment in which it occurs.
(iii) The determiner used by the student, or zero, if none used.
(iv) The article usage rules which fired.
(v) The correct article selected by the system.

After the article checking has taken place, the system uses the information it has about the student's performance to update the student model. Each noun phrase is considered individually. For each noun phrase in the student's input, the system checks if the determiner is **correct**, **incorrect** or **non relevant**. A non-relevant determiner is one that does not belong to the set of articles *a/an*, *the* and *zero*.

This includes possessive pronouns such as *your* and *my*, demonstrative pronouns such as *this* and *that*, and the word *some*. As seen in Section 4.1, *some* is sometimes considered to be a member of the set of articles, but for the purposes of this thesis it will be regarded as a non-relevant determiner.

6.2.3 Processing correct noun phrases

When a noun phrase with correct article usage is found, it is added to the list of correct noun phrases that the system keeps for the duration of one session. The system then checks which article usage rule fired when this sentence was processed and records this information in the student model. The rules which it believes the student has used are marked in the student model together with a degree of certainty, representing the certainty with which the system believes that the student knows the rule. On the student's first apparent use of the rule, it is marked in the student model as having certainty 1. On the second use of the rule, it is marked as having certainty 2. On the third and all subsequent uses of the rule, it is marked as having certainty 3. When a rule is marked in the student model with certainty 3, it is said to be **known** by the student.

If the student has entered a correct noun phrase, the system also updates the edges which it believes the student model contains. As described in Section 2.1.2, the genetic graph can be used as an overlay model, and in *ArtCheck*, the edges in the student model correspond to the edges in the expert model described in Section 5.4.2. The student is said to have acquired an edge when the system believes that she knows both rules at either end of the edge or link.

6.2.4 Processing other noun phrases

When the student has entered an incorrect noun phrase, it is added to the list of incorrect noun phrases. These will be examined later by the system when attempting to find an explanation for the student's errors. If a noun phrase occurs with a non-relevant determiner, that noun phrase is ignored by the student modelling component. However, as described in Section 5.3.3, all the noun phrases, no matter which determiner is used, are added to the **discourse model** for a complete record of the noun phrases used in the preceding discourse.

6.2.5 The student outline

When the system has dealt with all the noun phrases in the student input, it then updates the student model if necessary by checking the **student outline**. The student outline is used by the explanation component when generating adapted explanations for the student. It consists of three components: the student's general level of English; the student's combined level of English and article usage; and the student's learning preference. This is because the system communicates with the student in the target language, and both the student's general level of ability in English and her specific knowledge of article usage should be taken into account.

Stage 1:	0 - 6 rules
Stage 2:	7 - 12 rules
Stage 3:	13 - 18 rules
Stage 4:	19 + rules

Table 6–1: Knowledge of article usage rules

The student's knowledge of English article usage is determined by the system and is generated by considering how many article usage rules the student has acquired. There are 28 rules in total[1]. As mentioned before, a rule is said to be **known** when a student has used it three times. In total, there are four levels of ability, corresponding to **beginner**, **intermediate**, **advanced** and **very advanced**. The level of ability of the student is determined by combining the general level of ability in English with the student's knowledge of article usage rules. The system determines the student's knowledge of article usage by considering how many rules the student has acquired. The divisions of number of rules acquired between stages

[1]There is another rule, rule 99, which decides whether *a* or *an* should be used, but this is applied at a different level, after the other rules have applied.

of article usage knowledge are arbitrary and are given in Table 6–1. The system is concerned with the number of rules acquired, and not with specific rules acquired, because this gives the system a general idea of how competent the student is. In the WRITE option, the system cannot control which particular article usage rules the student attempts to use, so it is unfair to specify which rules should be acquired at each stage.

Level	Number of rules			
	0-6	7-12	13-18	19 +
Beginner	Level 1	Level 2	Level 2	Level 3
Intermediate	Level 2	Level 2	Level 3	Level 4
Advanced	Level 2	Level 3	Level 4	Level 4

Table 6–2: Four levels of expertise

These two factors, knowledge of article rules, and general competence in English, are combined into four levels of expertise as shown in Table 6–2. Again, the combination of these two factors into a general level for the student was determined arbitrarily. As discussed in Section 6.2.1, the student herself declares her general level of expertise in English, as being as a **beginner**, **intermediate** or **advanced**. The system initialises each student as either a Level 1, Level 2, or Level 3 student. Level 4 is reached when the student either makes very few errors, or has acquired the appropriate number of article usage rules.

After each complete sentence, the system checks to see if the student has acquired any more rules. If the student has shown that she knows enough rules to move up a stage, according to Table 6–1, the system puts the student up to a higher level if possible, according to Table 6–2. The student can also go up to a higher level, if, after entering more than 10 noun phrases, she has made less than 10% errors. The student is put **down** a level, if at this point she has made more than 70% errors[2]. It is important to correctly assess at what level of ability the student is,

[2]These figures were determined by experimentation with the system

as this is used in adapting the explanation in various ways. The next section will describe how the student's errors are analysed.

6.3 Analysis of errors

This section will describe the method for analysing the article usage errors made by the student. The motivation for doing this is to find a rule which represents the behaviour of the student when making errors, that is, a **mal-rule**. It is suggested that the mal-rule which is generated represents the student's underlying beliefs about the usage of articles, that is, it represents a misconception held by the student. If this is the case, then it is hoped that the system's explanation to the student regarding the errors made will help to remove that misconception.

Student modelling can be said to be **dynamic** when parts of the model can be generated in response to students' behaviour and thus specific types of students' errors do not have to be anticipated in advance. For the student model in this system to be dynamic, the mal-rules which represent the student's behaviour when making errors must be generated by the system.

In order to achieve this, machine learning techniques can be used (Gilmore & Self, 1988). One of the aims of machine learning is to learn concepts and rules from the given data. Similarly, in dynamic student modelling, the aim is to learn something about the student's behaviour from the errors made by the student. Thus, in *ArtCheck*, machine learning techniques concerned with rule induction are used to maintain a dynamic student model.

In *ArtCheck*, the student is able to type in sentences and receive feedback from the system about her errors. (See Appendix B for a sample session with the system.) The system keeps a record of all correct and incorrect noun phrases. If an incorrect noun phrase is encountered, an explanation of the error is given to the student. This will be described in detail in Chapter 7. The general idea is that if the system can determine a mal-rule which represents the student's error, then this will be explained to the student. In other cases, a more straightforward explanation is given.

Section 6.3.1 will explain how **version spaces** and **candidate elimination** operate in machine learning. The next sections, Sections 6.3.2 through to 6.3.5 will then turn to the article usage domain and demonstrate how these techniques can be utilised and implemented in *ArtCheck*. Sections 6.3.6 and 6.3.7 will discuss how the problematic areas associated with this method are addressed in *ArtCheck*.

6.3.1 Version spaces and candidate elimination

This section will describe the concept of **version spaces** and the **candidate elimination** approach to rule learning (Mitchell, 1977). This was described briefly in Section 2.1.5.

The learning of rules from training instances is also known as **rule induction**. This is more formally defined as follows:

> *"It is given that some fixed action, A, is advisable in some class of (positive) training instances, I+, but is inadvisable in some disjoint class of (negative) training instances, I-. The task is to determine a rule of the form P−> A, where P is a set of conditions or constraints from some predefined language. These conditions must be satisfied for action A to be invoked. The learned rule must apply to all instances from I+, but to no instances from I-."* (Mitchell, 1977, p305)

Given a positive training instance, there are a number of hypotheses which could explain the instance, some very specific to the particular training instance, and some more general. A **version space** is the **set of current hypotheses** of the correct statement of a rule which predicts some fixed action. The set of maximal specific versions (MSV) is the set of the most specific hypotheses in this space, and the set of maximal general versions (MGV) is the set of the most general. The hypotheses within the two boundaries can be described as being *more specific than* the MGV or *more general than* the MSV. *More specific than* can be defined as follows:

"R1 is said to be more specific than R2 if and only if it will apply to a proper subset of the instances in which R2 will apply." (Mitchell, 1977, p306)

The concept of *more general than* is obviously the inverse of this.

Candidate elimination describes the way that the training instances narrow the version space (Mitchell, 1977). Positive instances serve to make the MSV more general, and negative instances to make the MGV more specific. The algorithm for candidate elimination is given below.

1. Firstly, the MGV is initialised to have no constraints, or to a rule with no conditions. The MSV is initialised to a rule which represents the first positive instance. Each of the positive and negative instances are then considered in turn.

2. When the next positive instance of the rule is encountered, the hypotheses with which it is in conflict are eliminated. That is, the most specific hypotheses which hold for the first instance but do not hold for a subsequent instance are eliminated from the version space. This is known as **generalisation** (Bundy *et al*, 1985). The MSV is then the set of hypotheses which hold for all the positive instances of the rule.

3. When a negative instance is encountered, then the more general hypotheses in the version space which hold for both the negative and the positive instances are eliminated, in order to discriminate between the negative and positive instances. This is known as **discrimination** (Bundy *et al*, 1985). The lower bound becomes the set of the most general hypotheses which exclude the negative and include the positive instances.

4. Steps 2 and 3 are repeated until the MGV and the MSV are equivalent, that is, there is only one candidate hypothesis remaining.

Mitchell describes two main advantages that this method has over a traditional depth-first search, where the best-so-far rule is refined by comparison with each of the subsequent positive and negative instances (Mitchell, 1982). Firstly, the candidate elimination approach separates the deductive stage of seeing which hypotheses are **plausible** from the inductive stage of seeing which rule is the **best**. Secondly, when an incorrect decision is made in the depth-first approach, backtracking is required. This can be avoided in the candidate elimination algorithm.

Candidate elimination has been implemented in LEX (Mitchell *et al*, 1983). The following sections will discuss how it has been implemented in the article usage domain.

6.3.2 The pre-defined library of conditions

In order to describe the use of version spaces in this domain, the **language** in which the errors are described must first be defined. The errors can be expressed as a set of conditions which can be seen to apply when the error is made. A mal-rule is expressed as a production rule consisting of a conjunction of conditions which imply a certain action. The types of conditions which can occur in the left hand side of the rule are pre-defined according to the domain. In this section, the library of conditions which will be used in constructing mal-rules will be introduced. The next two sections will discuss how a set of positive and negative instances is acquired in this domain. Section 6.3.5 will describe, using an example, how the rule is constructed, and the final sections will discuss some of the limitations of this method, and how they have been addressed.

The rules of the article usage domain were given in Section 5–4. An error in this domain is said to have occurred when an article occurring in a noun phrase violates the article usage rule which the system believed was applicable. The analysis of errors in this domain consists of accumulating errors of this nature and attempting to find an explanation for them. Therefore, in this domain, the **error** is a positive training instance.

It has been seen in Section 5.1.2 that the article usage rules are production rules of the form:

When conditions X apply, use article Y.

The conditions which are used in the article usage rules given in Table 5–4 depend on the following criteria:

(i) Whether the head noun is a proper noun or a common noun.

(ii) Whether the head noun is singular or plural.

(iii) Whether the head noun is count or mass.

(iv) Whether the head noun is modified, and if so, by what.

(v) What immediately precedes the noun phrase.

(vi) What semantic category has been assigned to the head noun.

These criteria will be used in the generation of the mal-rules. There are many other permutations of these criteria than those included in the article usage rules. Therefore, there are many other possible rules which the student could be using, which have different combinations of these criteria.

The full list of types of conditions which are feasible in this domain is given in Table 6–3. This library of conditions can also be described as a **rule part library** (Sentance, 1992) in that they are all derived from existing article usage rules. In this way, the work required by the system designer is minimised, as the information which must be included in the pre-defined library of the conditions is already available.

The next two sections will describe how positive and negative instances are defined in the domain of article usage.

Condition	Potential instantiations of conditions
Head noun	A specific lexical item
Proper noun	Proper noun Common noun
Number	Singular head noun Plural head noun
Count	Count head noun Mass head noun
Modified by (i) Modified by (ii) Modified by (iii)	An adjective of a certain lexical form An adjectives of a certain semantic category An adjective A preposition of a certain lexical form A preposition A relative clause
Preceded by	Identical verb form/identical preposition/nothing A verb with certain person, number, and verb form A verb with certain person and number A verb with certain number A verb A *be* verb with certain person, number, and verb form A *be* verb with certain number A *be* verb A *have* verb with certain number A *have* verb A *do* verb with certain number A *do* verb A modal verb with certain person and number A modal verb with certain number A modal verb A certain preposition Any preposition
Semantic category	A certain semantic category A non-nil semantic category

Table 6–3: Types of conditions in the mal-rule

6.3.3 Positive instances

Mal-rules can be generated by observing the student's errors and deciding which rules appear to govern the student's behaviour. This process involves considering a number of errors which the student makes. A mal-rule cannot be generated from the observation of just one error, as the system does not have enough information about the consistency of the error to be able to decide that a student is applying an incorrect rule. A number of errors must be accumulated, before the system can **generalise** about the errors and find a common rule to explain them. In the terminology of machine learning, the errors in *ArtCheck* are the **positive training instances** from which the system can learn.

There is an assumption in many machine learning systems that the training instances are all instances of a single concept or rule (Dietterich *et al*, 1982). However, this is not the case in this domain. The student may make a number of article usage errors, which may or may not all be related to the same underlying misconception. These errors must be **grouped** into sets of related positive and negative instances. Domain knowledge is used to establish the set of training instances from the available data. Certain criteria are specified which can be used to establish whether the errors are related or not. After the set of training instances is put together, the system can analyse the data and decide if the student seems to be operating a mal-rule.

For the purposes of analysing article usage errors, there are two criteria used to ascertain the similarity between errors: firstly, the **article used** in making the error; and secondly, the **article usage rule** which applies in each case. These criteria are discussed below.

- **The article used**

  The production rules in the article usage knowledge base all state that given a certain set of conditions, a certain article is used. The set of articles consists of *the*, *a/an* and the zero article. Thus, there are three possible actions which can be taken, one corresponding to each of the articles. The analysis of errors made by a learner involves constructing a rule which defines a set of conditions in which one of these articles is used. It therefore follows that only the errors which use the same article can be represented by the same rule.

- **The article usage rule**
  One of the problems of learning by examples occurs when the system attempts to find a common explanation for two examples which are not related. The problem then becomes how to eliminate implausible hypotheses using a set of heuristics. It is obviously much easier if this step can be largely avoided. Therefore, the system attempts to ascertain the relatedness of errors before it attempts to analyse them as being evidence of a common mal-rule. One obvious way of determining whether errors are related is if they have an applicable article usage rule in common. This guarantees that the errors have one or more of the conditions in common. Being able to see which rule would have been the correct one to apply also helps the system when it comes to attaching the deviant rule on to the genetic graph, as will be discussed in Section 6.4.2.

The number of errors which must be observed before this procedure begins is obviously arbitrary. For the purposes of this system, considering the frequency of erroneous noun phrases in students' input, it was decided to wait until **three related errors** had been made before attempting to diagnose a mal-rule. The system thus does not consider the possibility of finding a mal-rule until at least three errors have been made. When this situation arises, the system divides the errors up according to which article has been used in each of the errors. Errors involving the use of the same article are grouped together. When more than three errors have been made with a certain article, the errors are examined to establish whether they have sufficient conditions in common for the same article usage rule to have occurred in at least three cases. Only when at least three errors have been observed using the same article and with an article usage rule in common will the process of attempting to generate a mal-rule to account for the errors be initiated.

6.3.4 Negative instances

It has been described that in the domain of *ArtCheck*, a positive instance is an incorrect noun phrase which has been used by the student. It would seem to follow then, that a negative instance of the error is a correct noun phrase. The problem with this is the difficulty in determining which correct noun phrases are negative instances of which errors. A correct noun phrase could possibly be seen to be a negative instance of many different errors, which would not be intuitively valid. Therefore, it is important that a correct noun phrase must show some similarity to the positive instance before it can be said to be a negative instance of an error.

The positive instances of the same error are related by the article usage rules which should have applied in each case, and the article which the student chose to use. The negative instances can obviously not be related in this way or they would be defined as errors themselves. Therefore, the correct noun phrases are related using **one** of these criteria.

For a negative instance to be seen as a correct form of an error, it must occur in the same context as the error. It is this context or set of conditions which dictates which article usage rule should apply. Therefore, the criteria which are chosen for defining a correct noun phrase as a negative instance of an error are as follows:

(i) The noun phrase must be correct.

(ii) It must involve a different article to that used in the incorrect noun phrase.

(iii) It must have at least one article usage rule in common with the positive instances of the error. This ensures that the rule which was supposed to have been violated in the case of the positive instances has been used correctly in the negative instance.

To exemplify the criteria for grouping positive and negative instances, consider some sentences involving incorrect and correct noun phrases:

Positive instances of the error (incorrect noun phrases) :

(a) **John is teacher.*

(b) **John is happy man.*

(c) **Jean is student.*

Negative instance of the error (correct noun phrases) :

(d) *I am a student.*

Sentences (a) - (c) include incorrect noun phrases which all involve the use of **the zero article** in the case where the following rule should have applied:

> Rule 11: *Use the article* a/an *where a singular count noun is used as the complement of the verb* to be.

Sentence (d) is an example of the correct article being used, and this rule being applied successfully. Therefore, sentence (d) can be seen to be a negative instance of the common error exhibited in sentences (a) - (c).

The next section will describe how the positive and negative instances of errors are used to reduce the version space, and hence construct the mal-rule.

6.3.5 Construction of a mal-rule

The above sections have defined the language in which the positive and negative instances are represented in *ArtCheck*, and discussed how to group correct and incorrect noun phrases used by the learner so that they become positive and negative instances of a rule. The next stage is to apply the candidate elimination learning algorithm to the positive and negative instances. This will be demonstrated by working through a simple example.

The set of correct and incorrect noun phrases which will be used in this example is as follows:

Incorrect noun phrases :

P1. **John is teacher.*

P2. **Sandy is pig.*

P3. **I am doctor.*

Correct noun phrases :

N1. *John is a good man.*

In this example, the positive instances of the mal-rule will be considered first, and then the negative instances.

The first positive instance is **John is teacher.* The noun phrase *teacher* should occur with the article *a/an*, and occurs instead with the zero article. The error in *teacher* can be described as follows:

Det	HN	PN?	Number	Count	Env	Modifier	SC
zero	teacher	no	singular	count	*is*	nil	nil

(PN = Proper noun; Env = Linguistic environment; SC = Semantic category)

The MSV of the version space is this description of the first positive instance. The MGV, in the absence of any negative instances so far, can be written as:

MGV1 : *In any noun phrase, use the zero article.*

The difference between the MSV and the MGV can be seen in the following table:

	Det	HN	PN?	Number	Count	Env	Modifier	SC
MSV1	zero	teacher	no	singular	count	*is*	nil	nil
MGV1	zero	??	??	??	??	??	??	??

The next positive instance, the incorrect noun phrase *pig* in *Sandy is pig* is a similar error. The only difference is that *pig* is the head noun and not *teacher*, so the MSV becomes more general with regard to this condition, as seen in the following table:

	Det	HN	PN?	Number	Count	Env	Modifier	SC
MSV2	zero	**??**	no	singular	count	*is*	nil	nil
MGV1	zero	??	??	??	??	??	??	??

In the third positive instance, P3, the incorrect noun phrase *doctor* in *I am doctor*, is preceded by *am* and not *is*, so the MSV becomes more general to reflect this:

	Det	HN	PN?	Number	Count	Env	Modifier	SC
MSV3	zero	**??**	no	singular	count	Sing.form of *be*	nil	nil
MGV1	zero	??	??	??	??	??	??	??

The next stage is to consider the negative instance, N1. This is the correct noun phrase *a good man* in the sentence *John is a good man.* When a negative instance is encountered, the following takes place:

> *"...each element of the MGV which matches the instance must be relaxed by a set of minimally more specific versions which do not match the instance. These new constraints are obtained by adding constraints taken from elements in MSV in order to ensure that they remain more general than some MSV, and thus remain consistent with previous instances."* (Mitchell, 1977, p308)

In order to keep the MGV as general as possible, initially only one constraint from the MSV is considered at any one time. However, if no maximally general versions could be found by adding only one constraint, then more than one would be added. The potential members of the MGV in response to the negative instance N1 are as follows:

	Det	HN	PN?	Number	Count	Env	Modifier	SC
MSV3	zero	**??**	no	singular	count	Sing.form of *be*	nil	nil
MGV2	zero	??	no	??	??	??	??	??
MGV3	zero	??	??	singular	??	??	??	??
MGV4	zero	??	??	??	count	??	??	??
MGV5	zero	??	??	??	??	Sing.form of *be*	??	??
MGV6	zero	??	??	??	??	??	nil	??
MGV7	zero	??	??	??	??	??	??	nil
N1	a	man	no	singular	count	Sing.form of *be*	adjective: *good*	nil

However, of these potential maximal general versions, only MGV6 is valid in this case, as it is the only one which excludes the negative instance N1. This hypothesis could be written as:

MGV6: *With any* **non-modified** *noun phrase, use the zero article.*

Thus, if no other data is provided, the final versions are as follows:

	Det	HN	PN?	Number	Count	Env	Modifier	SC
MSV3	zero	**??**	no	singular	count	Sing.form of *be*	nil	nil
MGV6	zero	??	??	??	??	??	nil	??

All the possible hypotheses which account for the data observed fall between these two boundaries. The complete description space is shown in Figure 6–1.

The ideal situation when working with version spaces is to have sufficient data to leave only one hypothesis between the lower bound and upper bound. However, this is unlikely to be the case in this domain. Therefore, the question arises as to which of the hypotheses is selected as the **best** once the candidate elimination process has been completed. As the topic under consideration is the analysis of errors, the best hypothesis in this case is the most specific one within the bounds of the version space. This is equivalent to the current MSV, or the **least upper bound** of the version space. This decision is made because when the mal-rule is

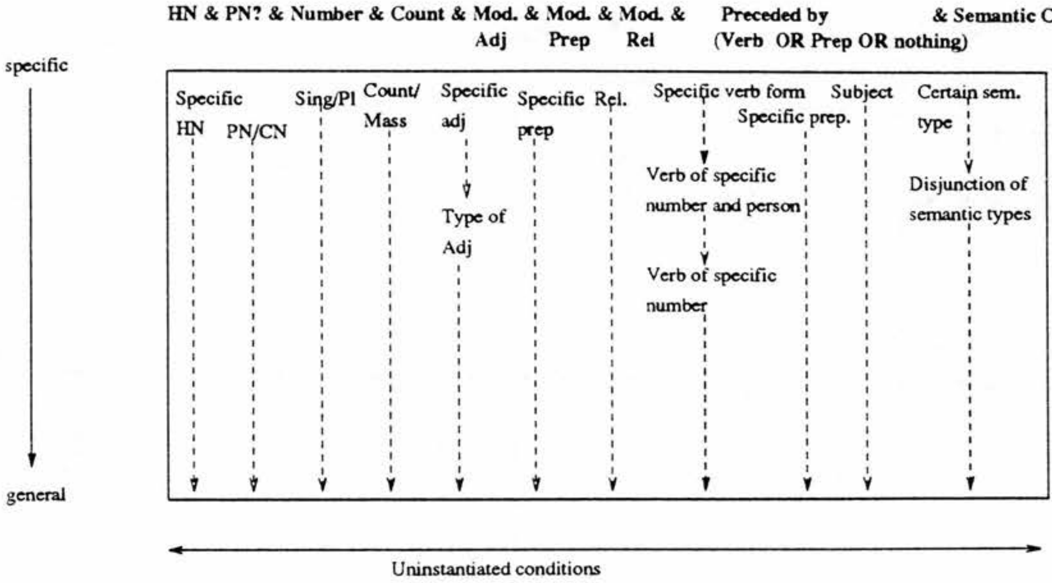


Figure 6–1: The structure of the description space

used by the system in giving an explanation of the student's error, the student will be helped by the most specific explanation available. The mal-rule which is proposed which accounts for this data is as follows:

> MSV3: *Where there is a singular, unmodified, common, count noun preceded by a singular form of the verb* be, **use the zero article**.

Mitchell describes the MSV as a **set** of maximally specific elements (Mitchell, 1977). If there were more than one element in the set MSV it would not be clear which would be the best hypothesis to select. However, as has been seen from this data, and also pointed out elsewhere (Bundy *et al*, 1985), the MSV will only ever contain one element, the most specific description of the positive instances. Only in more complex implementations of the algorithm, for example, with multiple boundary sets, which is discussed below, does the MSV contain more than one element.

6.3.6 Inconsistent data

The question may arise of the significance of finding the lower bound at all, if it is the least upper bound of the version space which is always taken as the best hypothesis. However, the negative instances have a very important part to play in the search for the best mal-rule. Firstly, in the case where many noun phrases are observed, the bi-directional search provides a more **efficient** way of homing in on both the plausible and the best hypotheses. Secondly, a search with just the positive instances could result in more and more general hypotheses, which may be psychologically implausible. This situation might occur if the data was inconsistent and it was becoming difficult to generalise over the positive instances. The presence of a lower bound on the version space means that inconsistent data will easily be detected, where the upper bound and the lower bound cross over. When this happens, the candidate elimination process must be abandoned. Figure 6–2 illustrates this phenomena. Even a single inconsistent instance can cause this effect.

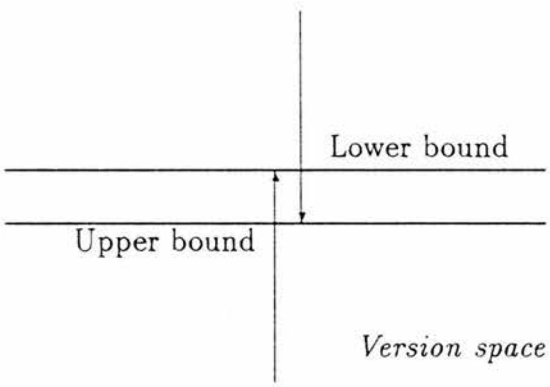


Figure 6–2: The potential effect of inconsistent data on the version space

One way of dealing with inconsistent data is to use **multiple boundary sets** (Dietterich *et al*, 1982). This is a way of keeping a record of boundary sets which correspond to some but not all of the instances observed. Using this method, as well as having an MSV representing the most specific hypothesis corresponding to all the positive instances, there is also an S_1 which is the set of hypotheses which correspond to all but one of the positive instances, and, when i is any number, an S_i which corresponds to all but i of the positive instances. Similarly, there is a G_i which corresponds to all but i of the negative instances. Thus, if the situation arises where the MSV and MGV cross over, the algorithm can try to find a hypothesis which works with all but i of the training instances, starting with $i=1$ and increasing the value of i as necessary.

This method obviously would make the candidate elimination algorithm slower to run, and would involve more memory where a lot of data was involved. It has not been implemented in the present system for two reasons: firstly, because the system is an educational tool offering feedback; and secondly, because of the potential effect of the feedback given from the system to the student. These two points will be discussed in turn.

The first point relates to the type of system involved. *ArtCheck* is a remediative system, which aims to detect and correct errors in article usage. It aims to find mal-rules which account for particularly persistent errors, based on the idea that showing the student that there is a pattern in the errors made, and explaining the basis for that pattern, will help the student to understand and correct the error more readily than if the correction of the error alone was given. The system needs to have a certain amount of confidence in the diagnosis of a mal-rule before attempting to enlighten the student. Owing to the small amount of evidence for mal-rules that is expected from the student when the diagnosis is carried out, the system will insist that it can only diagnose a mal-rule if there is no element of doubt about it. Therefore, the system assumes that there is no noise in the data. In the situation that the data is genuinely inconsistent, and the system fails to diagnose a mal-rule, the student's error will be still be corrected, but with a more straightforward explanation.

The second point relates to the feedback given by the system during the session. The system operates in two modes, GAP mode and WRITE mode. In both these modes, feedback is given to the student after each complete sentence. This feedback consists of explanations and examples relating to the error. During this process, data is being collected in case persistent errors reveal that the student may be using a mal-rule. However, the explanations are given to the student in the hope that the next time a similar noun phrase is used, the error will not be made. When the system is analysing the data to see if it can learn any mal-rules, it is not aware of the feedback that the student has received between the different usages of the noun phrases. Any such feedback may cause an apparent inconsistency in the data. Inconsistent instances are known as **false instances** (Dietterich *et al*, 1982). A false positive instance is an error which the student did not intend to make, that is, a typing error or similar mistake. This may cause the constructed mal-rule to be too general. A false negative instance is a correct noun phrase where, in similar circumstances, the student had previously made an error. If the false negative instance were to occur after the positive instances, it could be assumed that the student had benefited from the feedback given and corrected the error. In this case it is quite correct for the system to abandon the search for a mal-rule. The general intention of the system is to help the student correct errors, so whenever this is done, the system has succeeded in its aim. This means that the failure to find a mal-rule in these circumstances is not by any means a failure of the system.

6.3.7 Disjunctive concepts

It has been seen that the candidate elimination algorithm can learn rules of the form of a **conjunction** of conditions implying a certain action. However, as discussed in Section 2.1.5, one of the disadvantages of the candidate elimination technique is that it is not particularly suitable for learning **disjunctions**. This may or may not be a problem when implementing this technique in a student modelling system, depending on the domain. In any case, humans tend to avoid learning disjunctive concepts because it is difficult (Bruner *et al*, 1956), so it may not be a psychologically meaningful problem.

There are ways to amend the algorithm so that it can learn disjunctive concepts. One such way will be described below. However, in *ArtCheck*, a more domain-specific and less general way of dealing with disjunctions has been adopted, as it was easier to implement for the very occasional and unlikely cases that disjunctive concepts would be required in the article usage domain.

It is first necessary to ascertain when the problem of disjunctive concepts applies to the particular domain in question. In the English article usage domain, the first step is to consider the correct article usage rules. Referring back to Table 5-4, which gives the complete table of rules which have been implemented for the article usage domain, it can be seen that *Rule 6* on this table can be written as follows:

> Rule 6: *Certain singular, count, common nouns preceded by prepositions and of semantic category* meal, transport, place *OR* season *can occur with the zero article.*

Examples of this rule include the phrases *in spring*, *at dinner*, *by bus*, and *in prison*. This rule contains a disjunction, relating to the types of nouns involved. However, this rule could easily be expressed as four different rules, each relating to one semantic category only. Therefore, for the correct article usage rules, disjunctive rules are not a problem. Similarly, it would be possible for the system to treat the errors in this area as relating to separate mal-rules. However, in doing this, some important generalisations may be lost. Consider the following examples:

Incorrect noun phrases :

P1. **We discussed our plans over* **the breakfast**.

P2. **I go to school by* **the bus**.

P3. **John goes to school on* **the foot**.

P4. **I prefer to travel by* **the car**.

Correct noun phrases :

N1. *In* **autumn**, *all the leaves fall off the trees*

This example shows four sentences where *the* is used instead of the zero article, and one where the zero article is used correctly. In all these cases, the rule described above should have applied. Example P1 relates to a *meal noun*, examples P2, P3 and P4 relate to *transport nouns*, and example N1 relates to a *season noun*. If no disjunctions were to be allowed in the mal-rules, the system would report that the student consistently makes an error by using *the* instead of the zero article after a preposition and before a *transport noun*. However, in this case, the error in P1 could not be related to this mal-rule. It is desirable to be able to generate a rule which could combine the semantic categories which were being used in this type of situation. In order to do this, disjunctive rules must be allowed.

Another scenario which could only be expressed using disjunctive concepts, concerns the linguistic environment in which the noun phrase occurs. As has been seen, this is one of the factors determining the choice of article. Table 6–3 on page 132 gives the types of linguistic environments which errors may possibly have in common. One of these is the person, number and type of verb. The person of the verb can be categorised as *first*, *second* or *third*. Using only conjunctive rules, the system could express the fact that a student only used a certain article after a third person singular verb, for example, but would be unable to express the fact that the student only got the article **correct** after a third person verb, and made the error after first and second person verbs. While this may be an unlikely scenario, the system could be extended to allow disjunctive concepts in this case.

Some methods have been suggested for incorporating disjunctive rules into the candidate elimination and similar algorithms, and a survey of these can be seen in (Bundy *et al*, 1985). One suggested method is to form several sets of positive instances, which may or may not overlap, ensuring that the negative instances are not included in any of the different positive instance sets. Figure 6–3 shows how the positive and negative instances in the current example would be expressed with this method.

This way of envisaging the disjunctive nature of the sets obviously does not express all the common elements that the different positive instances have. An alternative method would be to show the conditions which are common to the positive instances in the intersection of the sets, and to separate the sets only at the actual point of the disjunction. In this case, there is a finer granularity involved in the disjunction. This is illustrated in Figure 6–4.

Disjunctive elements are incorporated into the mal-rule in *ArtCheck* using the method illustrated in Figure 6–4. Domain-specific information is used to trigger off a modification of the algorithm which allows disjunctive rules. Disjunctive concepts can only be expressed in certain cases, that is, where semantic categories

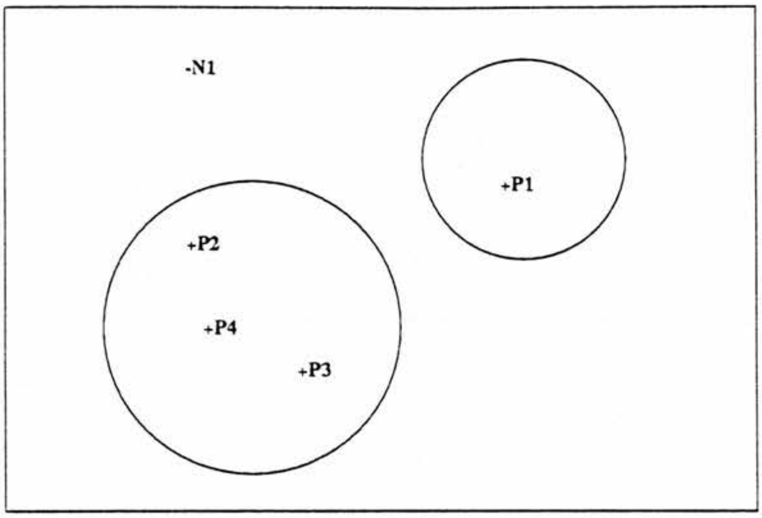


Figure 6–3: A "*meal or transport*" disjunction (1)

or tensed verbs are involved, as detailed above. In these situations, the modified algorithm allows an extra step in the process of generalising over positive instances. Normally, in the candidate elimination algorithm, the system attempts to match a new instance against the existing MSV on individual sub-conditions, to see if the sub-condition holds in both cases. If it does not, then that sub-condition takes on a new value which can hold for both instances. In the case of the sub-condition which involves considering the semantic category of the instance, the semantic categories must either match or not. If they do not match, the semantic category is instantiated to a variable. In order to allow a disjunctive concept at this stage, an intermediate step must be included. This entails the semantic category condition being instantiated to a **set of values**. Initially, this set contains two elements, the semantic category of the new instance and the semantic category of the current MSV. The upper bound is then moved nearer to the lower bound, but by a smaller step than if the semantic category had been instantiated to a variable. This is illustrated in Figure 6–5.

This method is simple to implement and captures the disjunction adequately. However, it has to be triggered off by domain-specific information. If the system were to form intermediate steps for all other conditions, the correct generalisations would not be obtained.

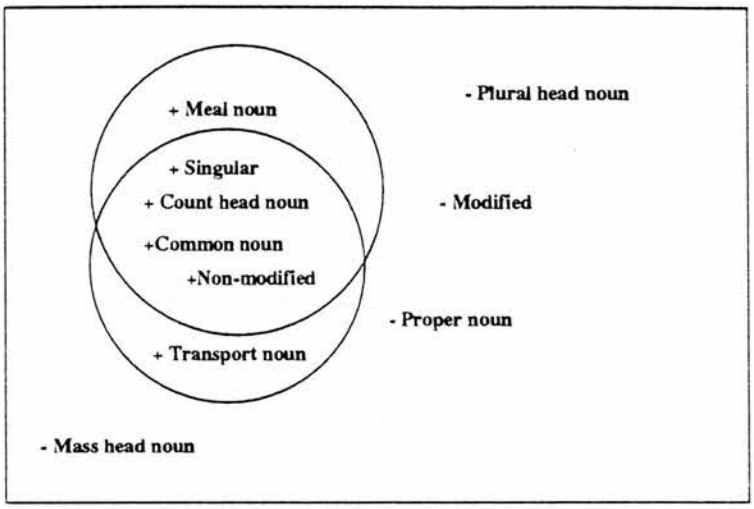


Figure 6–4: A "*meal or transport*" disjunction (2)

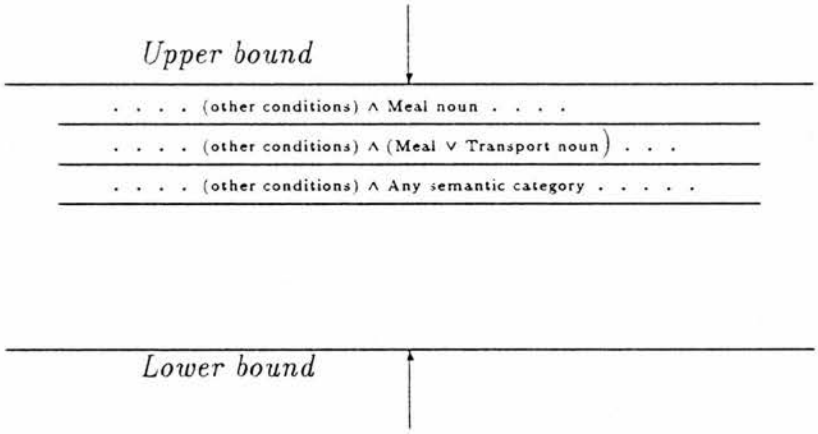


Figure 6–5: Adding a disjunctive condition in *ArtCheck*

6.4 Updating the student model

This section will describe how the information about a mal-rule which the student is believed to be using can be incorporated into the student model.

6.4.1 Filtering out unhelpful mal-rules

One of the criticisms of work involved with generating mal-rules from observing the student's behaviour is that implausible hypotheses can be generated. This can happen, for example, when the system attempts to generalise over specific instances which are not related. To address this criticism, it is necessary to ensure that implausible mal-rules are not incorporated into a student model. This can be done in two stages: firstly, before the system attempts to generate the mal-rule, and secondly, after the mal-rule has been generated and before it is added to the student model.

It has already been described in Section 6.3.3 how the errors made by the student must be seen to be related before they can be diagnosed as being examples of a common mal-rule. This is done by considering the article which is used and any common article usage rules. The mal-rules must also be checked before the student model is updated. At this stage, the mal-rules are discounted if they are **too general** and could not be seen to provide any useful information. In *ArtCheck*, a mal-rule is **too general** when the only information that the errors have in common is that they are all common, singular count nouns. Explaining such a general mal-rule would not be very informative for the student. In addition, the generation of such a general mal-rule suggests that either the relationship between the errors has been lost, for example by the inclusion of inconsistent data, or that there was not a very strong relationship between the errors at all.

Establishing a relationship between the correct noun phrases prior to the candidate elimination process should discount many of the completely general hypotheses, which would mean that the filtering out of implausible hypotheses would be less likely to be required.

6.4.2 Adding a deviation link

It was described in Section 5.4.2 that one of the links in the genetic graph is labelled a **deviation link**. This is used to link a dynamically generated mal-rule to the correct rule which should have fired instead. This helps the system to see where the student's knowledge has deviated from the system's knowledge, and give an informative explanation to the student.

The correct rule to which the mal-rule is linked must have a similar set of conditions. One of the criteria for determining whether errors can be supposed to be related to each other involves the errors having an article usage rule which should have fired in common. This is the rule that the mal-rule is said to be a deviation from. The deviation link links the mal-rule to this common article usage rule.

The example in Section 6.3.5 showed the following mal-rule being constructed.

> Mal-rule: *Where there is a singular, unmodified, common, count noun preceded by a singular form of the verb* be, **use the zero article**.

Figure 6–6 shows how this is added to the student model.

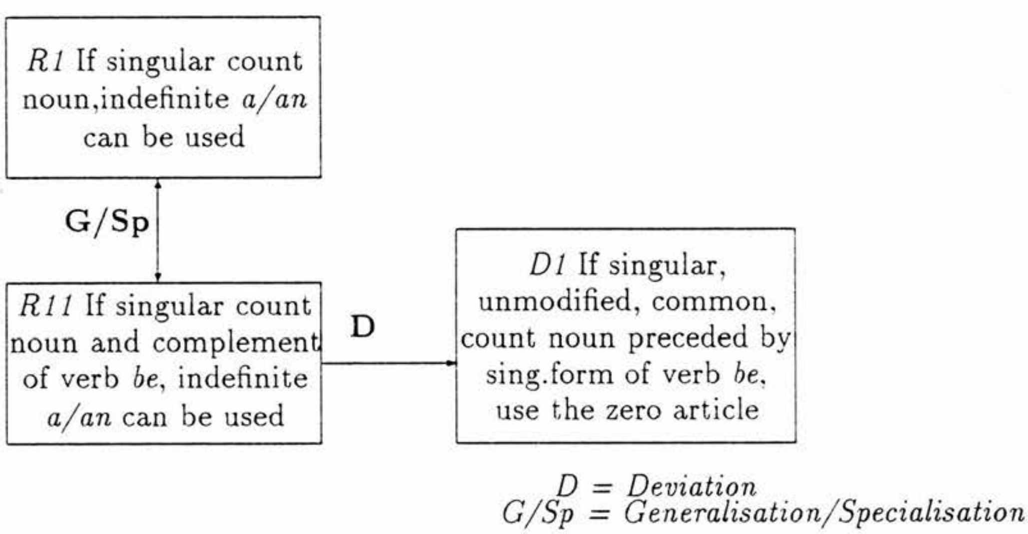


Figure 6–6: Adding a deviation link to the genetic graph

Chapter 7 will explain how the information about the student's mal-rule is used to give helpful and remediative feedback to the student.

6.5 Conclusion

Student modelling is a crucial area in Intelligent Computer-Aided Instruction. Through the effective use of a student model, the output from the system can be tailored to an individual student's needs. With the use of techniques from areas of Artificial Intelligence such as machine learning, the system can begin to model unanticipated input, rather than being reliant on the anticipation of the system designer.

Section 6.1 described the structure of the student model in the system *ArtCheck*. The student's domain knowledge is represented using a genetic graph, in which the rules of the system occur as nodes of the graph and the links of the graph show the relationship between the rules. The genetic graph is used because it is an ideal representation for dynamic student modelling, where the system is able to generate new rules in response to observed student behaviour. In this case, the rules which are generated are mal-rules which reflect the student's errors in English article usage. Section 6.2 discussed how the output from the article checking process was used to update the information in the student model, and keep an up-to-date picture of the student's current knowledge. Section 6.3 discussed the analysis of the student's errors in some detail, and the use of the candidate elimination algorithm for this purpose. It was seen that the candidate elimination algorithm adapted very easily to the article usage domain. Finally, Section 6.4 returned to the genetic graph to see how the newly generated information about a mal-rule which the student was believed to be operating with could be incorporated into the existing student model.

The next chapter describes how all the information retained in the student model can be utilised in giving tailored feedback to the student about her errors.

Chapter 7

Remediation

The aim of remediation in an Intelligent Computer-Aided Instruction (ICAI) system is to enable the user of the system to learn from any errors made. To fulfill this aim, a system must be able to understand the observed errors, and be able to communicate effectively with the user. The detection and analysis of errors has been described in Chapters 5 and 6. This chapter will be concerned with how the system communicates with the learner.

In order to remediate an error, a system needs to provide a good explanation for that error. Section 7.1 will define some criteria which can be used to determine what is and is not a good explanation, and discuss what strategies language teachers use to correct learners. Section 7.2 will describe the factors that have influenced the design of the explanation facility in *ArtCheck*. The explanation is tailored to the learner in three primary ways, relating to the learner's level of ability, learning style, and the type of error observed. In addition, the learner is given a certain amount of control over the information received. Section 7.3 will describe the implementation of the explanation facility in *ArtCheck* in more detail.

7.1 Guidelines for remediation

This section will discuss the sort of feedback which should ideally be given to users of an ICALL system. This involves considering both language learning research and explanation research to determine what should be incorporated into the system's explanation. The term **remediation** is used, because explanations are only given when the student has made an error. When the student uses an article correctly, the system simply comments to this effect.

Language learning research points out that teachers should adapt to individual students, particularly in taking into account their preferred learning style or strategy. For example, Ellis recommends that teachers should follow two guidelines:

> *"Teachers need to negotiate the learning tasks with the learners ... Good teachers ... seek to ensure that there is sufficient variety in the kinds of tasks learners are asked to undertake to satisfy all the learners at least some of the time."* (Ellis, 1992, p211)

and

> *"Teachers need to adapt the way they communicate to suit individual learners ... Accommodation to the needs and preferences of individual learners needs to be seen as part of the overall process of communicating with them."*(Ellis, 1992, p211).

Classroom teachers have a difficult task if they are to properly cater for the diversity of students within the same class. On the other hand, a computer system which interacts with students on a one-to-one basis does not have the problem of having to be all things to all students. Given the relevant information about the student, it should be able to interact with individual students in the way they prefer.

Research in explanation has been concerned with what makes a good explanation and how an explanation can be tailored to its recipient. Good explanations have been defined in various ways. For example, Kass and Finin mention three main criteria which must be satisfied to generate a good explanation: **relevancy, convincingness** and **understanding** (Kass & Finin, 1988). The first two of these refer to the content of the explanation. An explanation is **relevant** if it answers the user's questions. A **convincing** justification is one which is sound and logical, and, where possible, based on facts in which the user believes. The third criterion refers to the delivery of the explanation. Whether a user can understand the explanation depends on it being a well-organised explanation, not unnecessarily long-winded, and using terms with which the user is familiar.

An alternative set of requirements for a good explanation facility is given by Moore and Paris (Moore & Paris, 1991). This set consists of the following:

- **Fidelity**. An explanation must accurately reflect the system's knowledge and reasoning.
- **Knowledge from multiple sources**. An explanation facility must decide what is needed in the explanation and extract it from appropriate sources.
- **Naturalness**. An explanation must be well-organised and form part of a coherent dialogue.
- **Responsiveness**. A system must be able to offer alternative explanations if requested by the user.
- **Flexibility**. An explanation facility must be able to present the same information in different ways, depending on the user's knowledge and goals.
- **Sensitivity**. The explanation facility must be sensitive to the previous dialogue and the context in which the explanation occurs.
- **Extensibility**. The explanation facility must be designed in such a way that the explanation can easily be extended.
- **Portability**. The explanation facility must be portable to a variety of domains.

- **Adaptive capability** An explanation facility must be able to learn new strategies through time.

Not all of these requirements are relevant to tutoring systems, or to this domain. Remediative explanations in a tutoring context should reflect some of the teaching principles given earlier, and emphasise the flexibility and responsiveness features of the ideal explanation facility described above.

As discussed in Section 2.3, a number of developments have been made in improving the explanation facility of expert systems and intelligent tutoring systems. These developments can be divided into two areas: those that involved enhancing the expert system itself, and those that involved exploiting information held about the user in the user model. The explanation facility under discussion in this chapter incorporates developments of the latter kind, that is, the feedback to the student is individualised by exploitation of the information in the student model. The building of the student model and the interaction between the student modelling and the explanation modules is an important contributor to the effectiveness of the explanation facility.

To summarise, it can be seen that a tutoring explanation should vary its explanation according to the individual student. The student should both understand and be helped by the explanation. Finally, the student should have some control over the explanation, that is, the explanation facility should be flexible.

The next section describes how this has been achieved in *ArtCheck*.

7.2 Features of the explanation facility in *ArtCheck*

This section will describe the factors which contributed to the design of the explanation facility in *ArtCheck*. Taking into account the criteria which have been mentioned in Section 7.1, the feedback the system gives to the student will have the following features:

- It will be tailored to the student's level of expertise.

- It will be tailored to the student's learning preferences.
- The explanation given will depend on the source of the error.
- The student will have control over the length and content of the explanation given.

The first three of these elements of an explanation relate to both the **content** and the **presentation** of the explanation. The final factor refers to the control that the student has over the explanation. Each of these features will be discussed below.

7.2.1 Tailoring explanations to the student's level of expertise

A language teacher in a classroom would not give the same explanation of an error, or description of a rule, to students from the beginners' class as to those from the advanced class. Similarly, an intelligent tutoring system which is aimed at students of all levels must be able to adapt the explanations given to students of different levels of ability. The explanations given can be adapted to individual students in various ways. The vocabulary used, the complexity of the sentences used, the level of difficulty of the concepts introduced, are all factors which can be varied according to the intended recipient of the explanation.

In this system, adapting the explanation to the ability of the student is even more important, owing to the fact that the system communicates with the student in the language which is being learned, as opposed to the student's mother tongue. Therefore, explanations given to students whose general level of English is not particularly advanced should be kept as simple as possible.

The system has to take two aspects of the student's level of ability into account: the student's knowledge of the article usage rules, and the student's general ability in written English. Section 6.2.5 has described how these two aspects are combined to give four levels of ability for each student. These levels can be seen in Table 6–2 on page 126. The current level for each student is recorded in the student model. The explanation is then varied according to the student's level. The explanations become progressively more complex from Level 1 to Level 4 with respect to several criteria. These criteria include the vocabulary, the number of units of information given at one time, the type of grammatical structure used, and the amount of detail involved. Each unit of information is equivalent to the realisation of one discourse goal. The discourse goals used will be discussed in Section 7.3.1.

Level	Length	Detail	Vocabulary/grammar
1	1 unit of information given at a time	Correct error Give example and/or rule	Very simple
2	Up to 2 units of information given at a time	Correct error Give example then state rule or vice versa	Include word *article*
3	Up to 2 units of information given at a time	Gap-filling exercises to illustrate rule Give reason for error	Include some linguistic terms
4	Up to 3 units of information given at a time	Give reason for error incl. conflict between different rules	Include all linguistic terminology

Table 7–1: Levels of expertise

Table 7–1 shows how the explanations are tailored to the level of ability of the student. At Level 1, the student is asked for feedback after each unit of information, the vocabulary is kept simple, and the entire length of the explanation is kept short. At Level 2, up to two units of information are given at once, and the explanation is slightly longer, giving both the appropriate rule and an example. The order in which these are given depends on the learning style selected. At Level 3, the explanations given include more linguistic terms and more complex vocabulary, and the student can elect to try an example. At Level 4, up to three units of information are given at a time, the explanations are longer, and more technical linguistic vocabulary may be used. The gradual increase in complexity demonstrates how the student's expertise affects both the content and presentation of the explanation.

The next section will describe the effect that the student's preferred learning strategies have on the explanation generated by the system.

7.2.2 Tailoring explanations to learner strategies

Research from the area of learner strategies in language learning has shown that students use, and are aware of using, different strategies whilst learning a language. It is suggested that teachers should take into account these differences, and encourage students to be aware of the strategies they find most helpful (Rubin, 1987). In addition, teachers need to adapt the way they communicate to suit individual students (Ellis, 1992). These and other aspects of the use of learner strategies were discussed in Section 3.2.

One division of learners is made between **experiential** and **studial** learners (Ellis, 1992). Experiential learners want to immerse themselves in the language without concentrating too much on the grammar rules, whereas studial learners would rather learn about the grammar and vocabulary of a language. This is similar to the distinction between **remembering** and **communicating** discussed in (Rubin, 1987). Ellis discusses the fact that it is quite impractical for teachers in a classroom setting to divide learners up according to their preferred methods of learning and teach them accordingly. However, in one-to-one tuition, such as that offered by a computer system, this diversity can better be catered for. Difference in learner strategy is therefore something that can be taken into account when an ICALL system gives feedback to its users. This can be achieved by incorporating information about the students' preferences into the student model, as described in Section 6.2.1. The information held in the student model can then be used to vary the explanations given to the student.

In **ArtCheck**, allowance is made for two distinct learning strategies, which correspond to the experiential versus studial distinction described above. In terms of the domain of article usage, and the type of feedback given by the system, the experiential learner can be supposed to prefer exposure to **examples** which illustrate article usage, and the studial learner to prefer to learn the **rules** of article usage. Thus, in the initialisation of the student model, the student is asked to state whether she has a preference for rules or examples, or no preference. If a student indicates that she has no preference, then the default used is the rules strategy. This default was chosen because the teaching of a language with recourse to grammar rules is the more traditional approach. It was thought that if the student had no preference about learning strategy, or even did not properly understand the question, she may well expect to be given **rules** anyway.

The choice of learning strategy affects the explanation in various ways, and in conjunction with the other factors described in this chapter. It affects the **presentation** of the explanation, in that in some cases, the student may be offered the appropriate rule **and** an example, but the **order** in which these are given is dependent on the strategy chosen.

The preferred learning strategy also affects the **content** of the explanation. In the case of short explanations, the preferred type, that is, **rules** or **examples**, will be the only type of explanation given. A short explanation is given either where the student is a beginner[1] and the explanation is kept short by the system, or where the student, at any level, decides she does not want to see any more of the explanation. In addition, the student who prefers examples may, if desired, also be given gap-filling exercises relating to the current example, whereas the student who has chosen rules may be given the rule in more detail.

7.2.3 Varying the explanation according to the source of the error

Different students may make the same article usage error in the same context, but that does not necessarily imply that they made the error for the same reason. One student may make an error because she has never been taught the rule which applies in that context. Another student may know of the correct rule, but may

[1]As mentioned earlier, the beginner is given short explanations because the explanation given is not in her mother tongue.

wrongly believe that another rule which also applies in that context has priority over it. Another student may be operating with a mal-rule which she believes to have priority over the correct rule. In these situations, it would be most helpful for the students to be given different explanations.

The variation of explanations dependent on the source of the error can be achieved by utilising the information held in the student model. Sections 6.3.3 and 6.3.5 described how the system acquires information about the priority links the student has acquired and any mal-rules the student may be operating with. This information is retained and used in the generation of the explanation. The first part of an explanation will be the same regardless of the source of the error. The rest of the explanation will concentrate on the particular beliefs or lack of knowledge of the student. In order to keep the explanation simple at Levels 1 and 2, only learners at Levels 3 and 4 will receive explanations varied according to the source of the error.

The type of error is determined by following the following procedure. Firstly, the system checks to see if the system has diagnosed a mal-rule, as discussed in Section 6.3, and if so, the appropriate explanation is given. Secondly, if this is not the case, the system checks to see if the student seems to have misunderstood the priority between article usage rules which applied, and if so, gives an explanation designed to help in this case. Thirdly, if neither of the above two explanations of the error can be found, the system checks to see if the student knows the rule which has been violated, and if not, gives an explanation based on teaching that rule to the student. If none of the above cases hold, then a simple default explanation is given to the student.

7.2.4 Allowing the student to control the explanation

It was mentioned in Section 7.1 that one of the ways that teachers can improve the effectiveness of their teaching is by negotiating the learning task with the learner (Ellis, 1992). The same principle applies in Intelligent Computer-Aided Instruction. Users of ICAI systems will be happier and more enthusiastic about using a system if they feel they have some control over what they are learning (Howe, 1984). If a tutoring system persists in giving its users information which they do not need or want, then the effectiveness of the system will be considerably reduced. The same principle applies whether the information is too difficult for the student and she cannot understand it, or whether it is repeating something she knows already and is therefore irritating.

The general principle is that a student should only be presented with as much information as is wanted. After each unit of information, a student should be able to say that she has had enough, maybe because she has already realised the reason for her error. Alternatively, the student should be able to ask for more information or examples relating to a particular point. In this way, interaction between the system and the student takes place, during which the system responds to feedback from the student.

At Level 1, each unit of information is very small. This is because students at this level have said that they consider themselves to be at beginner level in their knowledge of the English language, and being required to read and digest a long piece of English can be intimidating. It also gives the student an opportunity to ask for something to be rephrased if she cannot understand some of the vocabulary. At the higher levels, progressively larger chunks of information are produced, which it is believed the student will be able to digest. At Levels 3 and 4, the student can control the content of the explanation, as she has the opportunity to try out some examples, or see more of the rule in detail.

The next section will discuss how these features have been incorporated into the system *ArtCheck* and give some examples of explanations generated by the system.

7.3 Implementation of the explanation strategy

The actual structure of the explanation generated depends on the **explanation strategy** adopted by the particular system. For the purposes of this discussion, an explanation refers to the output generated by the system in response to an error; when the student uses an article correctly, the system gives a simple message to that effect.

The explanation strategy in *ArtCheck* follows the principle that the most straightforward explanation is one that gives the answer first then follows that up with the justification (Gowers, 1986). If the justification for the correct answer is given before the answer it is less likely that the student will be able to understand it. In contrast, other ICALL systems (eg (Schwind, 1990)) do not give such a direct explanation, but prefer to give progressively larger hints at the answer. Experience with students and knowledge of the article usage domain has indicated that for this domain it is preferable to use the approach which has been described for *ArtCheck*.

Therefore, the explanation generated by *ArtCheck* consists of the following components:

- **Identification of the error**

  Firstly, the system points out to the student which noun phrase contains the error. This gives the student a chance to consider what the correct article usage should be.

- **Correction of the error**

  The system then informs the student of the correct article usage in that case.

- **Justification and exemplification of the correct article usage**

  If the student prefers to learn by rules, then the correct article usage rule is given at this stage. If the student prefers examples, then the system will give

Discourse goal	Purpose
identify error	To indicate which noun phrase contains the error.
correct error	To inform the student of the correct article.
ask for feedback	To initiate student input.
state rule	To inform the student of the applicable rule.
give example	To illustrate with an example.
teach new rule	To explain a rule in more detail.
student do example	To allow the student to try some appropriate examples.
explain mal-rule	To explain a consistent error which reflects the student's use of a mal-rule.
exemplify mal-rule	To show the student the consistency in the errors she has made.
explain rule conflict	To explain which rules have priority over others.

Table 7–2: Discourse goals in *ArtCheck*

an appropriate example. Other ways of justifying the article selection (at levels other than Level 1) include allowing the student to fill in the article in a similar example, or giving the rule in more detail.

In generating the explanation, the system selects a set of **discourse goals** which must be realised as the explanation is developed. (A discourse goal describes the effect that that part of the discourse is expected to have on the hearer or reader, as described in Section 2.3.2.) The actual discourse goals selected are dependent on the type of explanation. In addition, the discourse goals are realised in different ways depending on the level of ability of the student the explanation is intended for. Table 7–2 gives a list of the different discourse goals used in *ArtCheck* and the purpose of each of them.

7.3.1 Types of explanation

It has been seen that the actual form of the explanation is dependent on three criteria: the performance and ability of the student; the student's preferred learning strategy; and the type of error which has been observed. All of this information is derived from the student model. It has been described that in *ArtCheck* there are four levels of ability, two types of learning strategy, and up to three different sources of error. The latter variable can be increased to four sources of error, if

unknown is added to the list. This gives a total of 32 possible different types of explanation.

It was decided that the explanations at the lower two levels should be kept as simple as possible. Therefore, these explanations are not varied dependent on the source of the error. They are, however, varied according to the student's preferred strategy. This reduces the number of different types of explanation to 20.

One of the sources of error which is identified is where the student has made an incorrect assessment of the **priority** between different rules. It was felt that to understand an explanation of this error, the student would need to be at an advanced level of English. Therefore, this source of error is only described to students of level 4, and because it relates strictly to the article usage rules, only to students who have expressed a preference for learning rules. This removes 3 further possible types of explanation, giving 17 types. These are labelled from T1 through to T17 and their distribution is shown in Table 7–3.

Learning	**Level of Expertise**			
Preference	**Level 1**	**Level 2**	**Level 3**	**Level 4**
Examples	T1	T3	T5: Rule not known T6: Misconception T7: No reason	T11: Rule not known T12: Misconception T13: No reason
Rules	T2	T4	T8: Rule not known T9: Misconception T10: No reason	T14: Rule not known T15: Misconception T16: Incorrect rule priority T17: No reason

Table 7–3: Types of explanation in *ArtCheck*

Each of the above types of explanation relates to a set of discourse goals. For example, the first type of explanation, T1, which is used for a student at Level 1 who prefers examples, involves the use of the following discourse goals: **identify the error**, **ask the student for feedback** (in this case, does the student want to continue?), **correct the error** and **give a relevant example**. The use of these discourse goals in the interaction with the student is shown in Figure 7–1.

An example of an interaction following the pattern given in Figure 7–1 is given in Figure 7–2.

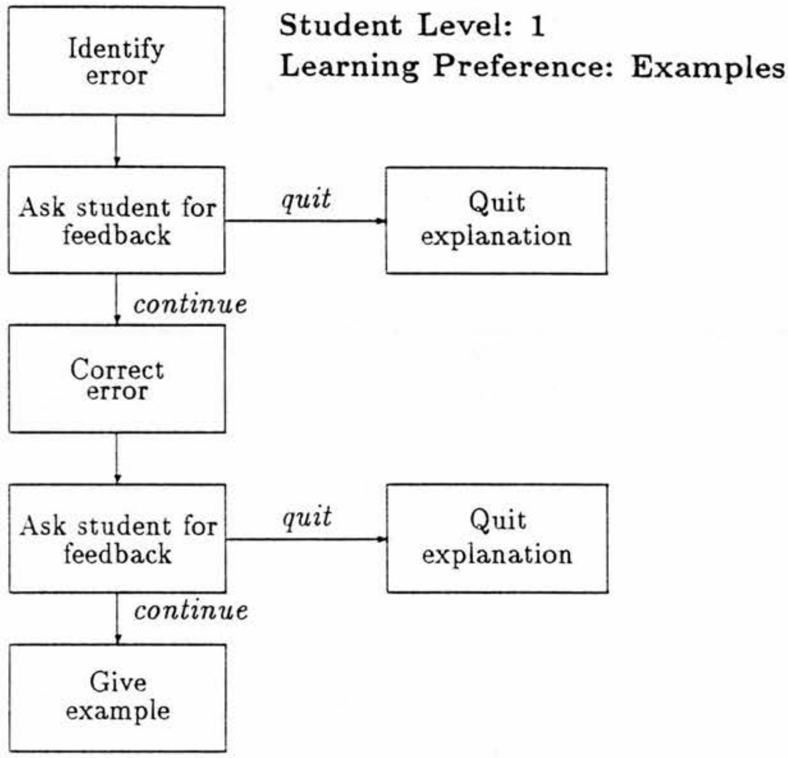


Figure 7–1: Discourse goals realised in a Type 1 explanation

Type 2 is similar to Type 1, except that a rule is stated instead of an example. **Type 3** and **Type 4** differ in that the goal **ask the student for feedback** is realised after two units of information have been given, that is, after **identify the error** and **correct the error**. The student can both learn the rule and have an example at this level.

Types 5 through to **Type 10** are used for students at Level 3. Types 5-7 refer to the examples strategy and Types 8-10 refer to the rules strategy. Figure 7–3 shows the structure of Type 5. Type 8 is similar except that initially the appropriate rule is explained to the student, after which the student can elect to try out some examples. Types 6 and 9 are explanation types which involve the explanation of a mal-rule and will be explained in Section 7.3.2. Types 7 and 10 are given when the system does not know the reason for the error. In this case the explanations consist of an example followed by the correct rule, or the rule followed by an appropriate example, depending on the learning preference. These explanations are similar to

Enter sentence >> `John is teacher.`	*student input*
`<teacher>in`	
`<John is teacher> is incorrect.`	*identify error*
Select:	*ask student for feedback*
m more	
q quit explanation	
>>>> `m`	
It should be: <a teacher> .	*correct error*
Select:	*ask student for feedback*
m more	
q quit explanation	
>>>> `m`	
An example is:	*give example*
*** Harold is a librarian.	
Continue? (y/n) >>>> y	
Enter sentence >> `Sandy is a pig.`	
Well done! No errors in this sentence!	
Continue? (y/n) >>>> `n`	

Figure 7–2: Type 1 explanation generated by *ArtCheck*

Types 3 and 4, but are tailored to a Level 3 student in terms of vocabulary etc. The structure of explanations not shown here can be found in Appendix G.

An example of an interaction with a student at Level 3 where the rule is not known (Type 5) is given in Figure 7–4.

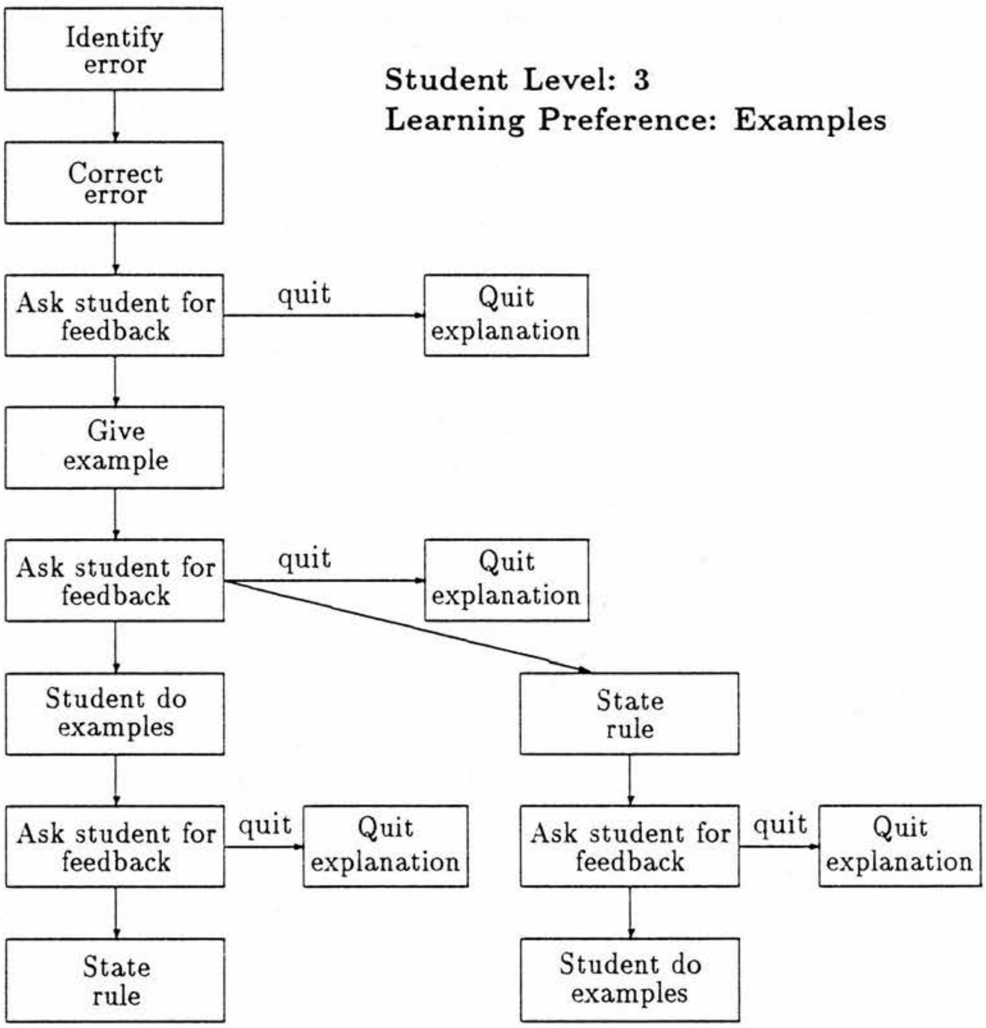


Figure 7–3: Discourse goals realised in a Type 5 explanation

```
Enter sentence >> Everest is a highest mountain.     student input

< a highest mountain> in
< Everest is a highest mountain>  is incorrect.      identify error
It should be:<the highest mountain> .                correct error

Select:     m more                                   ask student for feedback
            q quit explanation
>>>> m

An example is:                                       give example
*** John is the nicest teacher.

Select:     e try some examples                      ask student for feedback
            r explain rule
            q quit explanation
>>>> e
<<< He is ***** fastest runner >>>>                  student do examples

Choose the correct article:
            1 a
            2 an
            3 the
            4 no article
>>>> 1
Sorry, wrong answer. Do you want to try again? (y/n)>>> y
Choose the correct article:
            1 a
            2 an
            3 the
            4 no article
>>>> 3
Well done. That is the correct answer.

Select:     e try some examples                      ask student for feedback
            r explain rule
            q quit explanation
>>>> r

The rule is:                                         state rule

RULE 12 >>>

>>> Use < the > before superlative adjectives
like best and fastest.

Continue? (y/n) >>>> n
```

Figure 7–4: Type 5 explanation generated by *ArtCheck*

Types 11 through to **Type 17** are used for students at Level 4. Types 11 and 14 are similar to Types 5 and 8 described above, except that the language used is more complex, and there are more difficult examples given for the student to try. Types 12 and 15 relate to the explanation of mal-rules and will be described in Section 7.3.2. Type 16 relates to an error which has resulted from the student being confused over which rules have priority over others. This explanation is exemplified in Section 7.3.3. Types 13 and 17 are generated when the system does not have a particular reason for the error. In this case, the system gives the appropriate rule and example, tailored to a Level 4 student, as described above for Types 7 and 10.

Examples of each of these types of explanations can be found in Appendix G.

7.3.2 Mal-rules

A student is said to be operating with a mal-rule when the system is able to find a rule which accurately describes a set of related errors. A specific type of explanation is given in this case. The construction of a mal-rule was discussed in Section 6.3.5, and the example used in that section will be continued here.

To recap, the incorrect noun phrases used by the student in the example in Section 6.3.5 were as follows:

P1. **John is teacher.*

P2. **Sandy is pig.*

P3. **I am doctor.*

The mal-rule which is said to account for this data is given below:

> Mal-rule: *Where there is a singular, unmodified, common, count noun preceded by a singular form of the verb* be, **use the zero article**.

Given that **I am doctor* is the sentence which the student has just entered, the explanation given to the student in this case is given in Figure 7–5.

Enter sentence >> I am doctor.	*student input*
<doctor>in	
<I am doctor> is incorrect.	*identify error*
It should be: <a doctor>.	*correct error*
Select: m more q quit explanation	*ask student for feedback*
>>>> m	
The rule is:	*state rule*
RULE 11 >>>	
Use <a> or <an> before singular count nouns which come after the verb <to be >.	
Select: m more q quit explanation	*ask student for feedback*
>>>> m	
I have noticed that you seem to use <no article> instead of <a> or <an> before a singular count and after the verb <to be > in the singular	*explain mal-rule*
Select: m more q quit explanation	*ask student for feedback*
>>>> m	
You also said: *** Sandy is pig *** John is teacher which are similar errors.	*exemplify mal-rule*
Try one of these again:	
< Sandy is **** pig >	
Choose the correct article: 1 a 2 an 3 the 4 no article	
>>>> 1	
Well done. That is the correct answer.	
Continue? (y/n) >>>> n	

Figure 7–5: Explanation relating to a mal-rule

This example would be given to a student who has selected the learning strategy **rules**, and corresponds to Type 9 from Table 7–3. A student at Level 4 who had selected the learning strategy **rules** would receive the same explanation with the option to try some additional examples at the end. Students who had selected the learning strategy **examples** would be shown the similarity of their previous errors (**exemplify mal-rule**) before giving the actual mal-rule (**explain mal-rule**).

7.3.3 Incorrect priority links

It was discussed in Section 5.4.3 that some of the article usage rules take precedence over others when more than one rule applies. It is important for the student to understand this relationship between the rules. The situation could easily arise where the student has a good grasp of the article usage rules, and appears to be applying them correctly, but is baffled that she still is making article usage errors. One of the hard problems associated with learning to use articles correctly is which of the article usage rules take precedence over other ones. The system *ArtCheck* is able to detect that this is the case and give an appropriate and helpful explanation.

An example will be given to illustrate this point. In this example, the student has acquired **all** the article usage rules, and is at Level 4. The student then enters the incorrect sentence,

**John is a best teacher.*

In this sentence, it is the noun phrase *a best teacher* which is incorrect. The following table shows the rules which have fired in this case.

Rule	**Correct article**	**When rule applies**
1	*a/an*	Singular, count noun
11	*a/an*	Singular, count noun occurring after the verb *to be*
12	*the*	Noun modified by a superlative adjective

Enter sentence >> `John is a best teacher.`	*student input*
<a best teacher>in	
<John is a best teacher> is incorrect.	*identify error*
It should be: <the best teacher>.	*correct error*
Select: m more q quit explanation	*ask student for feedback*
>>>> `m`	
The rule is:	*state rule*
RULE 12 >>>	
Use <the> where a noun is modified by a superlative adjective.	
You may have been using these rules:	*explain rule conflict*
RULE 1 >>>	
Use <a>,<an> or <the> before singular count nouns. Use <a> or <an> when it is something new, and <the> when it is something you know.	
RULE 11 >>>	
Use <a> or <an> before singular count nouns which come after the verb <to be >.	
However, these rules do not apply in this situation as Rule 12 has priority over them.	
Select: e try some examples q quit explanation	*ask student for feedback*
>>>> `q`	
Continue? (y/n) >>>> `n`	

Figure 7–6: Explanation relating to incorrect rule priorities

To understand which rule should apply in this case, it is necessary to refer to the genetic graph, given in Figure 5–2 on page 106. Rule 11 is linked to Rule 1 by means of a specialisation link. Therefore, Rule 1, as the more general rule, can be discounted at this stage. Rule 11 is linked to Rule 12 by a priority link, indicating that Rule 12 has priority. Therefore, the article which should be used in this case is *the*.

The student in this example, who knows all the rules seen in the above table, can be assumed to have applied Rule 11 instead of Rule 12. The explanation given to the student should reflect this. The explanation generated by *ArtCheck* which corresponds to Type 16 is given in Figure 7–6.

7.4 Conclusion

Previous chapters have shown how the system is able to detect and analyse English article usage errors made by the student. The purpose of this chapter was to describe how the system *ArtCheck* used the information which it has acquired to give instruction and feedback to the student about the observed article usage errors. The explanations which the system generates are tailored to individual students by making use of information in the student model.

The chapter began by outlining the criteria which determine what is and is not a good explanation in terms of both educational and explanation research. In particular, several factors were outlined which constitute an ideal explanation facility (Moore & Paris, 1991). By considering the explanation facility in *ArtCheck* in terms of these requirements, it can be seen that it demonstrates several of the features recommended by Moore and Paris:

- **Fidelity**. The type of explanation given when a mal-rule is diagnosed reflects the exact reasoning used by the system.

- **Knowledge from multiple sources.** The explanation facility in *ArtCheck* uses domain knowledge, student knowledge and teaching knowledge in generating explanations.

- **Naturalness.** The explanations in *ArtCheck* are clear and coherent and read naturally.

- **Responsiveness.** The system is able to offer more information or examples to enhance the explanation if requested by the student.

- **Flexibility.** Explanations in *ArtCheck* are tailored to the student's knowledge and learning style.

Section 7.2 then described the three distinct types of information about the student which have an effect on the explanation which is generated. These are the **expertise** of the student, the **learning preference** of the student, and the **source of the error**. These three factors affect both the **content** and the **presentation** of the explanation. These principles were put into practice by making use of the information in the student model. In addition, the student is given some control over the length and content of the explanation.

Section 7.3 described how the incorporation of these features gives a total of 17 different types of explanation which the system *ArtCheck* can generate. Each type of explanation corresponds to a set of discourse goals. Several examples of actual explanations generated by the system were given.

The next chapter will describe the process of evaluating the system *ArtCheck*.

Chapter 8

Evaluation

An important part of the development of any ICAI system is the evaluation of the system. Evaluation serves several purposes. Firstly, it makes it clear what the implemented system can actually do. Secondly, it exposes the limitations of a system, and points to improvements which can be made. Thirdly, it indicates the educational effect which a system has on the students who use it.

The evaluation of ICAI systems is a field of research in itself, and researchers are beginning to develop system independent criteria by which all systems can be judged and recommend evaluation methodologies, (for example (Mark & Greer, 1993)). The purpose of this chapter is to describe the evaluation of the system *ArtCheck*.

Section 8.1 explains the terminology used to describe different forms of evaluation and suggests methods for carrying out the evaluation. Section 8.2 discusses which methods of evaluation were suitable for the system *ArtCheck*. Section 8.3 and 8.4 discuss the internal and external evaluation of *ArtCheck*. Finally, Section 8.5 indicates what the implications of this evaluation are and summarises the findings.

8.1 The theory of evaluation

8.1.1 Formative and summative evaluation

The two traditional forms of evaluation are known as **formative** and **summative** evaluation (Littman & Soloway, 1988). **Formative evaluation** is the process of evaluating the system while it is being developed, continually assessing whether the development of the system meets the requirements set at the design stage. **Summative evaluation** is the evaluation of the system which takes place when the system is complete, to establish the educational impact of the system. Because most ICAI systems are prototypes demonstrating new developments in this area of research, few end up being used in an educational setting (Littman & Soloway, 1988). For the purposes of this project, *ArtCheck* was developed as a research prototype, and therefore its evaluation was limited to formative evaluation.

8.1.2 Internal and external evaluation

Evaluation can also be described as either **internal** or **external**. **Internal evaluation** seeks an answer to the question:

"What is the relationship between the architecture of an ITS and its behaviour?" (Littman & Soloway, 1988, p209).

The **internal evaluation** of a system involves extensive testing to analyse how the individual modules of the system operate in practice and how they work in conjunction with one another. The purpose of the internal evaluation is to assess how well the program performs. Many of the tests are those that a system designer would expect to carry out to ensure that the program was working as originally intended.

External evaluation is concerned with the student's experience of the system and the effect it has on their knowledge of the domain. External evaluation, as described by Littman and Soloway (Littman & Soloway, 1988), can take place in the context of either formative or summative evaluation, depending on whether or not the system is a finished product. Although perhaps more associated with summative evaluation, external evaluation is essential at the formative evaluation stage as it helps the designer to guide the development of the system.

Thus, during the development of an ICAI system, formative internal and external evaluation should take place.

8.1.3 Methods of evaluation

There are several methods of evaluating systems, not all of them suitable for ICAI systems (Mark & Greer, 1993). Mark and Greer discuss seven possible evaluation techniques:

- **Proofs of correctness.** With this method, programs are theoretically **proved** to be correct. Most artificial intelligence programs are too complex to analyse in this way and therefore this method of evaluation is not suitable.

- **Criterion-based evaluation.** In this case, a set of objective criteria is developed by which the behaviour of the system can be measured. For ICAI systems, the criteria can be difficult to define, or not specific enough to give a precise measurement of the system's behaviour. However, this method may be useful in evaluating certain components of ICAI systems.

- **Expert knowledge and behaviour.** Here, the system's behaviour is checked against a human expert. Some components of an ICAI system can be tested in this way, but most components are too complex and not inspectable.

- **Certification.** This means the appraisal of an ICAI system by an independent human teacher. As with criterion-based evaluation, the problem arises of a suitable set of criteria to judge the system by.

- **Sensitivity analysis.** With this method, a component is evaluated by considering how responsive it is to different input, for example, that relating to different learner characteristics. This method may be useful for evaluating ICAI systems which claim to offer individualised instruction.

- **Pilot testing.** Pilot testing involves allowing students who are representative of the target population to test the system. It can take one of three forms: **one-to-one testing**, **small group testing**, or **field testing**. This method of evaluation is particularly suitable for formative evaluation.

- **Experimental research.** Experimental research is intended to measure quantitatively the effect that an educational interaction has on students.

Mark and Greer's overall recommendation is to use experimental research for summative evaluation and pilot testing for formative evaluation. In the next section, the discussion will move on to the method used in the evaluation of *ArtCheck*.

8.2 Evaluating individual components of *ArtCheck*

To evaluate the system *ArtCheck* it is first necessary to consider the individual components of the system and what they claim to do. Figure 8–1 shows the stages of processing in *ArtCheck*.

As the system has a clear modular structure and the stages of processing are linearly structured, the individual modules of the system can be evaluated separately, in order to consider to what extent their behaviour correlates with the aims at the design stage.

To evaluate the natural language processing component, it is necessary to consider what the aims of this module are. It is unrealistic to expect the system to be able to understand absolutely any given sentence in a natural language, and because of the problems associated with the understanding of natural language, any such system or component of a system will only be able to process a subset of natural

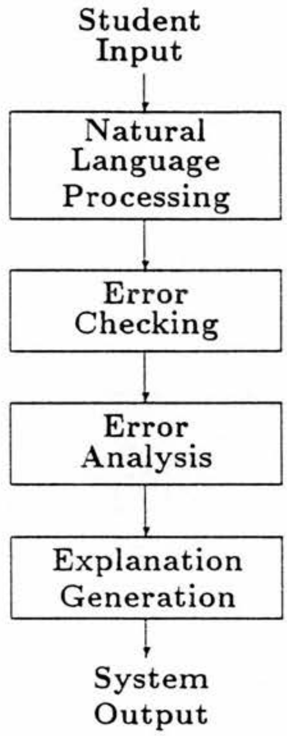


Figure 8–1: Stages of processing in *ArtCheck*

language. The task of the evaluation of this module is to identify the size of that subset. This will be done by a form of **criterion-based evaluation**, where the criteria used are an independently developed target set of sentences used to evaluate a completely different natural language tool. Criterion-based evaluation is a form of internal evaluation.

The evaluation of the error checking involves ensuring that the system's knowledge of the domain is accurate. The domain is a subset of the English language, and therefore, the error checking must be evaluated by an expert in the domain, that is, a native speaker of English. It must be determined to what extent the system has accurate knowledge of the domain. This is the method known by Mark and Greer as **expert knowledge and behaviour** (Mark & Greer, 1993) and is another form of internal evaluation.

A further example of internal evaluation of a component is the evaluation of the student modelling component, and its role in the analysis of errors. The error analysis is evaluated using **criterion-based evaluation**. In this case, the criterion that the system must meet is that the error analysis should be able to diagnose different types of errors, in the way that the system was designed.

The interface and explanation facility are evaluated by pilot testing (Mark & Greer, 1992), which is a form of external evaluation.

8.3 Internal evaluation of *ArtCheck*

This section will describe how the internal evaluation of *ArtCheck* was carried out.

Littman and Soloway suggest that an internal evaluation should set out to answer the following questions (Littman & Soloway, 1988):

- What does the ITS know?
- How does the ITS do what it does?
- What should the ITS do?

The internal evaluation of *ArtCheck* will firstly attempt to address the first two of these questions. Any limitations which are found as a result of the evaluation may provide an answer to the third question.

As discussed in Section 8.2, the internal evaluation of the system will involve considering the natural language processing facility, the error detection component, and the error analysis component in turn. The evaluation of the explanation facility is described in Section 8.4.1.

8.3.1 The evaluation of the natural language processing component

The development of a comprehensive natural language understanding component was not the primary aim of this thesis. However, if the system is to demonstrate that article checking can be achieved with free input to the system, then a natural language processing component must be included. It is a realistic expectation that a natural language processing component will only be able to process a subset of natural language, and the size of that subset depends on the amount and time and effort devoted to that part of the system. This part of the evaluation is designed to determine the extent of the subset of natural language which the system *ArtCheck* can understand.

To discover how much natural language a system can understand, it is necessary to find out the types of sentences which the system can understand, and the types that it cannot. This gives a clear idea where the boundaries of the subset of parsable natural language lie. One way of carrying out this evaluation could have been to build up a set of test sentences specifically to test *ArtCheck*, but this course of action was not chosen, because firstly, the sentences may appear biased towards demonstrating the capabilities of *ArtCheck*, rather than giving an honest assessment of its abilities, and secondly, because sets of test sentences already exist which have been used to test other natural language systems, and it is more efficient to use such facilities where available.

Therefore, for this part of the evaluation, it was decided to test the natural language processor against an existing set of test sentences from another, larger, natural language grammar (Grover *et al*, 1989). Some of these test sentences were used to test the capabilities of the natural language processing component of *ArtCheck*. The full list of sentences which were tested with *ArtCheck* is given in Appendix H. Some examples of the types of structures which are included in the *ArtCheck* grammar are given in Table 8–1.

It can be seen that *ArtCheck* can understand many different types of sentence structures. It is not able to understand questions and imperatives. However, the design of the system was such that this sort of input was not expected. The grammar could easily be extended to include structures of this kind. Other structures with which the system has problems include subordinate clauses beginning with the word *for* and sentences with a modal element in the subordinate clause. This reflects the simplicity of the grammar which is incorporated into *ArtCheck*. Thus, the evaluation of this part of the system clearly highlights its limitations, which

Type of structure	Example	Parsed/ not parsed
Wh-Questions	*Which car are you going to buy?*	Not parsed
Yes/no questions	*Can Lee abandon her?*	Not parsed
Negative sentences	*He doesn't help.*	Parsed
Adverbial clauses	*He certainly doesn't help.*	Parsed
Prepositional phrases	*He helped the abbot with some anxiety.*	Parsed
Possessives	*This mood of Lee's is not very characteristic.*	Parsed
Indirect objects	*She gives the message back to him.*	Parsed
Imperatives	*Don't help him.*	Not parsed
Subordinate clauses	*He promised her that he would help.*	Parsed
Subordinate clauses beginning with *for*	*She allows for him to be anxious.*	Not parsed
Titles	*Mr Smith is going to London.*	Parsed
Relative clauses	*The man who I saw yesterday is here.*	Parsed
Conditionals with *if*	*She might figure out if he helped.*	Parsed
Conditionals with *whether*	*She didn't take in whether he helped.*	Some parsed
Modifiers	*The big fat lazy cat is sleeping.*	Parsed
Modal verbs	*She ought to help.*	Parsed
Clitics	*He is crazy isn't he.*	Some parsed

Table 8–1: Sentences parsed by *ArtCheck*

are due to the fact that the development of a comprehensive grammar was not one of the original aims of the project. There are obviously many improvements which can be made in this area. These will be discussed in more detail in Section 9.3.1.

8.3.2 Evaluation of the error detection process

The evaluation of the error detection process involved deciding how effective the system was at looking at a parsed sentence and coming to a correct conclusion about the article usage in that sentence. In *ArtCheck*, error detection consists of two stages. The first stage is the extraction of noun phrases and any relevant structural information from the sentence. The second stage is the application of a set of article usage rules to the extracted noun phrases. This second stage also involves the issue of conflict resolution when more than one rule fires.

It is difficult to accurately evaluate these two stages because it is impossible to predict every possible noun phrase which the system could be expected to consider.

However, to evaluate this part of the system, a reasonably representative sample of sentences was used. The evaluation took place as the system was being developed.

The evaluation method used was **expert knowledge and behaviour**, as described in Section 8.1.3. The expert in this case is any native speaker of English. The evaluation can be carried out by comparing the system's decision about the article usage with that of a native speaker.

The evaluation of the error detection consisted of two stages. Firstly, it was necessary to determine in which types of noun phrases the system made a **correct** decision about the article usage. The second stage of the evaluation was to find examples of noun phrases in which the system made a **wrong** decision about the article usage, in order to find any possible limitations of this part of the system. This also involved trying to discover which part of the error detection process was unreliable. Both these stages were carried out as the system was being developed, as a way of continually improving the capabilities of the system.

As a result of the first stage, examples representing the system's ability to make correct decisions about article usage in relation to each of the article usage rules were found. Table 8–2 shows examples of incorrect noun phrases which the system was able to identify. The correspondence of the rule numbers to the actual rules is given in Table 5–5 on page 103.

Rule no.	Correct article	Example of rule violation detected by *ArtCheck*.
1	a/an	**There is* **dog** *in my garden.*
2	no article	***An dogs** *are usually very friendly.*
3	no article	**She buys* **a milk** *and bread.*
4	the	**I have a dog.* **A dog** *is called Rusty.*
5	the	**They live in* **centre of the city**.
6	no article	**Today John is going to school by* **a bus**.
7	a/an	**She buys* **the dozen eggs**.
8	a/an	**Frank runs twice* **week**.
9	a/an	**Frank runs five miles* **the day**.
10	the	***Rich** *are very lucky.*
11	a/an	**Mr Miller is* **bus driver**.
12	the	***Oldest child** *is called Steven.*
13	no article	**They have* **the three children** *and a dog.*
14	the	***Woman** *who worked at the ticket office was unhelpful.*
15	the	**All* **students** *were waiting for him.*
16	the	**He is in* **same class** *at school.*
17	the	**He is* **an only boy** *in his family.*
18	the	**On* **third day** *it rained.*
19	no article	**Mr and Mrs Miller live in* **the Glasgow**.
20	the	**Jean said there was a matinee on at* **King's Theatre**.
21	the	***Wilsons** *spent a week in Edinburgh.*
22	the	**He wants to be* **fastest runner** *in the world.*
23	the or a/an	***Tall man** *got on to the train.*
24	no article	***The life** *is hard.*
25	the	**Half* **a money** *is mine.*
26	a/an or no article	**Marcus is not such* **good swimmer** *as Gary.* **He shows such* **a determination**.
27	a/an or no article	**There is* **certain girl** *whom he likes.* **There are* **the certain books** *which he wants to buy.*
28	the or no article	**That is a* **one** *I want.* ***A one** *of them is mine.*
99	an	**Next door lives* **a old lady** *called Mrs Wilson.*

Table 8–2: Correct and incorrect noun phrases identified by the system

The next stage of the evaluation involved considering those occasions on which the error detection process does not work properly. Some examples of this are given below:

(i) *Jack has a dog called Rusty. The dog is very fond of him. His sister Janet has* **a dog** *too.*

In this case, the system would decide that the correct article to use in front of *dog* in the third sentence would be *the*. This is because rule 4 has fired:

> Rule 4: *Use* the *where a noun or semantically related noun has been mentioned before.*

The system would assume that the *dog* referred to in the third sentence is the same dog as that referred to in the first and second sentences and is therefore **given** information. However, the *dog* in the third sentence is a completely different dog. It is a well documented problem in the area of computational linguistics that it is very difficult for a computer to determine the co-reference between noun phrases, that is, which of the noun phrases in a discourse refer to the same item, and which refer to a different item. In the case of the above example, it would be necessary to have a semantic understanding of the sentences before a reliable decision about what *dog* exactly referred to in each of the sentences. In *ArtCheck*, the main emphasis of the article checking is on the available structural and lexical information available to the system. The role of the context in determining definiteness has been simplified in order to build a more or less reliable working system. Therefore, the article usage in the above sentence is outwith the scope of this system.

(ii) **Last week** *somebody crashed into my car.*

In this example, the system will apply rule 12 to the noun phrase *last week* and decide that the article **the** should apply:

> Rule 12: *Use* the *before nouns which are modified by superlative adjectives.*

The reason that Rule 12 does not apply in this case is because this is an idiomatic temporal expression. In English, there are various exceptional cases of article usage because of idiomatic expressions. The article usage in some idiomatic expressions, for example, *on foot* and *twice a week* can be correctly predicted in *ArtCheck*. However, the system has not been designed to incorporate idiomatic temporal expressions such as *last week*.

(iii) *He is* **an only child**

This is another example of an idiomatic expression. Usually before the adjective **only**, the definite article **the** is required, as in the example,

Marcus is **the only boy** *in his family.*

In this case the applying rule is Rule 18:

Rule 18: *Use* the *before nouns which are modified by ordinal numbers.*

In this case, the system will not be able to predict that it is the article *an* which is required.

(iv) *John is* **a teacher who works at Lewis's school**

Occasionally, the conflict resolution mechanism, which decides which rule has priority over another, causes the incorrect article to be selected. In this sentence, the system would predict that the definite article *the* was correct. The following rules apply to the noun phrase *a teacher.*

Rule 1: *Use* a *(or* the*) before a singular count noun.*

Rule 11: *Use* **a/an** *where a singular count noun occurs as a complement of the verb* **to be.**

Rule 14: *Use* the *before nouns which are modified by a relative clause.*

Rule 11 is a more specialised version of Rule 1, so Rule 1 can be discounted at this stage. According to the expert model, Rule 14 has priority over Rule 11, as in the sentence:

John is the teacher I like best.

This priority link works for the majority of cases. However, in the sentence given above:

John is **a teacher** *who works at Lewis's school*

this priority link causes the system to make an incorrect prediction about article usage. It is difficult to find a set of priority links which are infallible, so the occasional incorrect prediction is unavoidable.

Therefore, there are several cases, such as those exemplified above, where the system will wrongly predict the appropriate article usage. Most of these cases correspond to certain idiomatic usages of the English articles, which have not been included in the system (see Section 5.3.6) because they are very numerous and account for less common usages of the articles. Other cases include distant or complex referring expressions, where semantic information would be required to predict the correct article usage.

8.3.3 Evaluation of the diagnosis of errors

The diagnosis of errors is carried out by the student modelling component of the system. This component is responsible for identifying the **source** of the error.

The evaluation of the system's diagnosis of errors was carried out using the **criterion-based** method (Mark & Greer, 1993). In this case, the criteria for testing the diagnosis of the errors is that the system should be able to diagnose errors leading from each of the possible sources of an error.

There are three possible sources of an error in *ArtCheck*.

- The rule is not known
- The student has an incorrect view of the priority between two rules
- The student is operating with a mal-rule

The evaluation of each of these will be discussed in turn.

The student does not know the rule

The first source of error relates to the student's ignorance of an article usage rule. The system is designed to identify this particular source of error by keeping a record of the rules known by the student in the student model. The student is said to know a rule when it has been used at least three times. (The choice of this number was arbitrary.) To check that the system is diagnosing these types of errors correctly, it is necessary to make sure that firstly, the student model is being updated correctly, and secondly, that this information is being correctly abstracted from the student model at the diagnosis stage. Both these stages have been thoroughly tested, and work correctly on all occasions.

The student has an incorrect belief about priorities between rules

The second source of error relates to the student having a false view of the relationship between different article usage rules as regards which of the rules has priority over others. This source of error was exemplified in Section 7.3.3. Again, the information about the priority links which the student has is retained in the student model. This process has also been extensively tested, and appears to have no problems.

The student has a mal-rule

The third source of error is the most interesting part of the system and relates to the system's generation of a mal-rule when a consistent error is made. The method which is used for generating the mal-rule was explained in detail in Section 6.3.5. An example of the type of explanation given in response to this source of error was given in Section 7.3.2. The process consists of firstly, deciding what is a positive and negative instance of a mal-rule, and secondly, using the positive and negative instances to focus on a hypothesis for the reason for the error. These two distinct stages can be considered separately.

To determine the positive and negative instances of a mal-rule, certain criteria are used. Any incorrect noun phrases which are to be classified as positive instances must involve the use of the same article and have at least one article usage rule in common, which is the rule which has been violated in all the cases. For correct noun phrases to be considered as negative instances of that error, the rule which is violated in the case of the positive instances must apply. Examples of negative and positive instances were given in Section 6.3.4.

In the data which has been used to test the system, these criteria generally give good results. One difficulty which was noted occurs when a rule is consistently violated, but where the incorrect article which the student uses is not always the same.

For example, if the student makes the following errors:

> *I like to travel by* **a** *car.*
>
> *I went to school by* **the** *bus today.*

then, although both of the incorrect noun phrases in the above sentences violate Rule 6, which says:

> Rule 6: *Use* no article *in certain idiomatic prepositional phrases involving the use of nouns to do with transport, seasons, meals or institutions.*

the system would be unable to relate the two errors because they involve the use of a different article. This is because in the generation of a mal-rule, a disjunction is not allowed on the right-hand side of the rule.

The second stage of generating a mal-rule involves using the positive and negative instances to focus on a hypothesis. The way the system actually works is to use the positive instances to move the upper bound of the version space down, and the negative instances to move the lower bound up. The chosen hypothesis is taken as the least upper bound. The negative instances are used to detect inconsistencies in the data, that is, when the upper bound and lower bound cross over.

The way in which the positive instances cause the lowering of the upper bound is by generalising individual conditions to find a conjunction of conditions which account for all the positive instances so far. This is achieved by considering each of the sub-conditions in turn. This process works efficiently, without any backtracking being required.

The way in which the negative instances cause the lower bound to be made more specific is more complicated. One of the reasons for this is that there can be many lower bounds of the version space, although there will only be one upper bound (see Section 6.3.5). When a negative instance is encountered, the new lower bounds become the most general rules which can be found which exclude the negative instance. This is done by considering the present lower bound, or each of the present lower bounds, and for each of the n conditions from the most specific version (MSV) or upper bound, replacing the corresponding condition in the lower bound by a more specific version of it. This gives n new lower bounds, each with one condition made more specific. The next step is to rule out those of the n lower bounds which do not exclude the negative instance. Those remaining are the new lower bounds for the current data set. This method of generating new lower bounds works well in the majority of cases.

One of the problems with the generation of mal-rules with the candidate elimination method is that at least three instances of the error are required before the diagnosis can be made. The student may need to be interacting with the system for a little while before she makes three or more consistent errors. This is particularly a problem with the WRITE option of the system, where there is the added factor that the student may avoid constructions in which she is unsure of the article usage. In the GAP option, the exercises are fixed by the system, so there is more flexibility for the system to tailor the exercises to the student's error. This possibility is discussed in Section 9.3.4.

If the error cannot be attributed to any of these three sources, then the system gives a **no reason** verdict. This mechanism works correctly.

Therefore, the overall behaviour of the error diagnosis component is exactly as it was designed. The only limitation which has been found is that a lot of data is required before the mal-rule can be generated.

8.3.4 Summary of internal evaluation

In summary, the internal evaluation of *ArtCheck* demonstrates that the system to a large extent works as it was designed and meets the aims set at the design stage. Two of the aims given in Section 1.4 were that a set of rules should be implemented for English article usage, and that the candidate elimination be implemented in this domain as a way of dynamically generating mal-rules. Both these goals have been achieved, within the development of an ICALL system for the article usage domain.

The next section describes the external evaluation of *ArtCheck*.

8.4 External evaluation of *ArtCheck*

It was discussed earlier that the internal evaluation of *ArtCheck* involved testing the effectiveness of the detection and analysis of article usage errors. The external evaluation is more concerned with the student's experience of the system, and whether it helps to lessen the number of article usage errors made. The external evaluation in *ArtCheck* must address the following two questions:

- Did the students have a positive and useful experience of the system?
- Did the explanations of an error given to a student help her to not make that error again?

It was discussed in Section 8.1.3 that in a review of evaluation methodologies, **pilot testing** was the recommended method for carrying out formative evaluation. There are three different types of pilot testing: one-to-one testing, small group testing and field testing, which vary as to the number of students involved in the test, and the formality of the testing procedure. The external evaluation of *ArtCheck* was carried out using **small group pilot testing**. This is described as follows:

> *"A small group of students, representative of the target population, are questioned before and after system use to assess their understanding of the content taught. Such information indicates whether specific aspects of content or program use are learned or understood by students."* (Mark & Greer, 1993, p8)

In the case of *ArtCheck*, students chosen to test the system were non-native speakers of English whose native language was non-article bearing[1]. The following sections describe how the pilot testing of the system was carried out. Section 8.4.1 describes the main evaluation exercise, and Section 8.4.2 describes a supplementary evaluation exercise which was carried out and the results which came from it.

8.4.1 The main evaluation exercise

The main part of the external evaluation of *ArtCheck* involved 15 language learners from summer schools in Edinburgh taking part in a specially designed evaluation exercise. The students were felt to be representative of the sort of students for whom the system had been designed.

It was necessary to find a way of measuring the student's aptitude in article usage before and after their exposure to the system. For this purpose, the students were asked to complete a pre-test prior to using the system, and a post-test afterwards, and the results of the two tests were used to see if their exposure to *ArtCheck* caused them to make less article usage errors.

The exercise took approximately forty-five minutes to an hour. It was divided into three parts:

- The pre-test (10-15 minutes)
- Using ArtCheck (25-30 minutes)
- The post-test (10-15 minutes)

[1]**Non-article bearing languages** are those which do not have an equivalent to the English articles.

The testers

The 15 students who took part in the exercise were adults and teenagers, of varying levels of ability, and of various nationalities. All of the students had as their native language a non-article bearing language. 10 of the students were Japanese, and the other five students were Arabic, Czechoslovakian, Syrian, Turkish and Finnish respectively.

Test materials

Each pre-test consisted of 10-12 multiple choice questions, consisting of English sentences with an article missing, and a short passage to write from either a set of pictures or a selection of suggested titles. The post-test given was identical to the pre-test, with the exception that the short paragraph of free text was not included. The pre and post tests were interchanged for different students to ensure that the test itself did not give a false impression of improvement. The pre-tests and post-tests used can be seen in Appendix I.

The article usage rules were graded according to difficulty and each of the article usage rules assigned to one or more levels. All the rules which were allocated to an individual level were used in the pre-test, the post-test and the system exercise for that level. Article usage rules corresponding to each level can be seen in Table 8–3[2]. The relation of the rule numbers to the individual rules is most clearly shown in Table 5–5.

---

[2]Table 8–3 shows the domain as consisting of 25 article usage rules, which was the state of the domain when the evaluation exercise was carried out. Four other rules have since been added to account for the article usages after the words *half* and *such* and before the words *certain* and *one*. This makes the total of 29 article usage rules which are given in Table 5–5. As these four extra rules reflect less common usages of the articles, it is felt that their omission in the evaluation exercise would not have made any difference to the results.

Level	Article usage rules
Level 1:	1 2 3 4 11 12 14 19 23 99
Level 2:	3 5 6 11 12 13 14 15 18 20 21
Level 3:	7 8 9 10 15 16 17 20 21 22 24

Table 8–3: Correspondence of the article usage rules to levels of ability

The method

Each student was first asked to assess her own level of ability in English in general as **beginner**, **intermediate** or **advanced**, as described in Section 6.2.1. This assessment was used as a starting point for the system in assessing the ability of the student with respect to English articles. Depending on the student's assessment, she was given the appropriate pre-test to complete. Beginners were given a Level 1 pre-test, intermediate students a Level 2 pre-test, and advanced students a Level 3 pre-test.

Each student was then asked to use *ArtCheck*. This was done without additional instruction as much as possible, in order to gauge how self-explanatory the menus and options were to a language learner. The students' interaction with the system was observed in order to detect any problems concerning the usability of the system. The exercises offered to the student were from the GAP part of the system. The WRITE option was not used during this exercise. This decision was taken for several reasons:

- The pre-test, GAP exercise and post-test for each level tested the same article usage rules. If the student had elected to enter free input from the keyboard (by choosing the WRITE option) it is likely that she would not use all the article usage rules which corresponded to her level, and the comparison between the pre and post test would be less reliable.

- It was envisaged that the students testing the system would not be accustomed to using a computer or keyboard and might thus find it difficult to type in whole sentences. The GAP part of the system is designed so that the student only has to enter one letter or number at a time.

- The natural language processing part of the system is, as yet, unable to parse very ungrammatical English. Thus, the system might not be able to parse much of the input, or may misinterpret the input and thus confuse the student.

- The explanations given are the same for the WRITE option as for the GAP option.

During the evaluation exercise, the system selected an appropriate gap-filling exercise for the student. Two exercises are associated with each level except for Level 4, which only has one. The student was allowed to complete as many exercises as she wished, and the system kept a record of which exercises had been completed. Where a student had completed both exercises at her level, and still wished to continue, she was given an exercise at the next level up. The system moved the student up or down the levels as appropriate as described in Section 7.2.1.

After spending some time using *ArtCheck*, the student was then given a **post-test** in order to ascertain if performance when using articles had improved. The post-test covered the same material as the pre-test, testing the same subset of the article usage rules as had been tested earlier, though the rules appeared in a different order on the test. At each level, half of the students were given the (a) test as a pre-test and the (b) test as a post-test, and the other half given the (b) test as the pre-test and the (a) test as the post-test, to ensure that the tests themselves did not influence the results.

At the end of the evaluation exercise, the students were asked to comment verbally on their experience of the system.

The results

Of the 15 students who were tested, one was excluded because there was a inconsistency between the assigned level during system use and that during the pre- and post-test. The results of the remaining 14 students can be seen in Table 8–4.

Student	Nationality	Level	Pre-test score	Post-test score
1	Arabic	3	9/12	9/12
2	Japanese	1	9/10	7/10
3	Japanese	1	8/10	10/10
4	Japanese	1	9/10	8/10
5	Japanese	2	8/11	7/11
6	Japanese	2	5/11	6/11
7	Japanese	2	7/11	10/11
8	Japanese	2	9/11	10/11
9	Syrian	2	7/11	8/11
10	Turkish	2	9/11	10/11
11	Japanese	2	8/11	9/11
12	Czechoslovakian	2	8/11	10/11
13	Finnish	3	8/12	11/12
14	Japanese	3	9/12	11/12

Table 8–4: The results of the external evaluation

These results show that 10 of the 14 students had a slightly better post-test result than pre-test result, one student got the same score on both tests, and 3 students obtained a better score in the pre-test than in the post-test. The changes in the scores indicates a tendency to improvement in performance, but the small difference between the pre- and post-test results mean that a full statistical analysis would not be very revealing.

There were many examples of particular errors which occurred in the pre-test which did not occur in the post-test. For example, Student 12 made the following error in the pre-test:

Student 12: **That cat has just drunk all milk.*

The rule which was violated in this case was rule 15 which is as follows:

Rule 15: *Use* **the** *before nouns which are preceded by the word* **all**.

The student made the same error during the first system exercise, and an appropriate explanation was given. During subsequent system exercises, and in the post-test, the student was able to use the rule correctly. The student's correct use of Rule 15 in the post-test was as follows:

Student 12: *All the teachers are on holiday for six weeks.*

Taking the pre- and post-tests together, there were particular rules which seemed difficult for all the students. Two rules involving the use of the definite article with proper nouns in particular cases seemed to cause the students problems, and another rule, which is less commonly used, involving the use of the definite article where the noun is also classified as an adjective, seemed to be hard for the students. In contrast, there were four rules which all the students used correctly (Rules 8, 9, 16, 17), the first two of which involved the use of the indefinite article in temporal expressions, and the second two of which involved the use of the definite article where the adjectives **same** and **only** were used as modifiers. In general, the incidence of errors was not high, perhaps indicating that the tests were too easy for the students. On the other hand, it is probably preferable, especially during the student's exposure to the system, for the student to have a relatively high success rate in order to keep the student's confidence up.

While the students were using the system, no mal-rules were diagnosed. This was because in GAP mode, a wide selection of article usage rules are tested, and the student does not have the opportunity to make a number of consistent errors, unless she attempts a large number of exercises. In other words, not enough data was available for the system to decide whether there was a consistent mal-rule in operation. This could be remedied by making the exercises longer, and giving the students a longer period of time to use the system, or by altering the way in which the data is gathered as evidence for the mal-rule by tailoring the exercises to elicit particular errors. The latter suggestion is discussed further in Section 9.3.4.

All 3 students who fared better in the pre-test than in the post-test were part of a group of five Japanese teenagers tested together. It was difficult to determine the correct level to test them at as their spoken English was very limited, yet when it came to the test, their understanding of written English was quite good. For example, Student 2, who took a Level 1 pre-test and post-test, when using the system, progressed quickly from Level 1 up to Level 2, and completed two exercises at Level 2. Therefore, the pre-test and post-test was less related to what happened when he used the system than it might have been. The same situation was true for students 4 and 5. This is obviously a drawback of this type of testing, if the success of the testing is dependent on the accuracy of the initial assessment of the student's ability. It indicates a problem with the design of the evaluation exercise, as opposed to a limitation of the system. As an alternative, in future evaluation exercises, a diagnostic multiple-choice exercise could be given to the students prior to the evaluation exercise, so that a more accurate assessment of their level of ability could be gauged.

Some of the students tested made verbal comments about the system. Most commented that the system was easy to use, and some mentioned that the system had pointed out article usage rules with which they had not previously been familiar.

The internal evaluation described earlier included the evaluation of the role of the student modelling component of the system in the diagnosis of errors. The student modelling component has another role in addition to analysing a student's errors, which can be evaluated externally, which is to update the student's **level of expertise** based on the student's performance. This was described in Section 6.2.5. Basically, if the student makes more than 70% errors and more than 10 noun phrases have been used, then the student is put **down** a level. If, at this stage, the student has made less than 10% errors, then she is put **up** a level. These figures were obtained arbitrarily, but it was felt that the adjustments made to the student's levels when they were actually using the system were appropriate, and thus no changes have been made to these figures.

8.4.2 A supplementary evaluation exercise

A more informal external evaluation of the system was carried out by several MSc students from the Department of Artificial Intelligence at the University of Edinburgh. These students were non-native speakers of English who had experienced difficulties with articles, and were also knowledgeable about the aims and methodologies of ICAI systems. These students carried out an informal evaluation of both the GAP and WRITE options available with the system, and gave written feedback about it. The students gave both positive and negative feedback about the system; the negative feedback related more to how the system could be enhanced by the addition of extra features, than its limitations as a system. This section will list all the comments made by the testers, and where appropriate comment on how suggested improvements could be incorporated into the system.

- **Positive points of the system**

  (i) The system meets the instructional goals of detecting, analysing and correcting article usage errors.

  (ii) The system is easy to use, and the instructions given by the system are clear and helpful.

  (iii) The system is able to dynamically generate mal-rules to account for consistent errors.

  (iv) The grammar rules are expressed clearly and concisely.

  (v) The explanations are easy to understand.

  (vi) Because the system has knowledge of its domain, it could be incorporated into a more general teaching system for English.

  (vii) The two modes WRITE and GAP are complementary.

- **Feedback with respect to cosmetic improvements**

  (i) One tester pointed out that, although the system is robust with respect to unexpected responses to the prompts given, it responds to invalid input by simply prompting the student again for a response. The tester suggested that a message stating that the input was invalid should be given. This change has already been implemented.

  (ii) During the GAP option, the student is requested to indicate which article has been selected by typing in the number which corresponds to that article choice. This was to lessen the amount of typing the student would have to do. The suggestion was made to allow the actual article to be typed in, eg *the* or *an* as well, in case the student preferred to do this.

(iii) During the GAP option, a passage is presented in a read-only window with some of the articles missing. There is another window which is used for student input which gives one sentence of the passage at a time. One of the testers suggested that the article or number be typed directly into the passage to save looking at two windows. However, if the explanations to the student were still given in the main window, this would still involve the student looking at two windows. As each sentence is given in the main window anyway, the purpose of having a read-only window was purely to remind the student of the context in which the current noun phrase occurred.

(iv) Another cosmetic improvement which was suggested was to give the student a running total of the number of errors and number of correct article usages at all times, as opposed to simply at the end of each exercise. This has not been incorporated as it was felt that it might be discouraging for students who were not doing well at a particular exercise.

- **Other feedback**

(i) One of the testers suggested that the **instructional goals** of each level, that is, what the system is attempting to teach the student at that stage, be given for each level. This will be discussed further in Section 9.3.4.

(ii) The student modelling component is responsible for updating the student model and deciding if the student's level should be raised or lowered. One of the suggestions made was to inform the student when she has been put up (or down?) a level.

(iii) In the GAP option, the system asks the student for a choice of article. A suggested improvement was to include an *I don't know* option, with an appropriate explanation, to avoid forcing the students to guess if they were unsure.

(iv) One tester suggested that a greater selection of examples for each rule would help if the error was made repeatedly.

(v) As mentioned above, the GAP option requires the student to enter the choice of article for the given sentence. No justification of that choice is asked for. One of the suggestions made by the testers was to ask the student to justify the choice of article. This is discussed in Section 9.1.1.

(vi) It was also suggested that a diagnostic report of the student's strengths and weaknesses was given at the end of the exercise.

(vii) An interesting suggestion, though outwith the scope of this project, was that the system allow a greater variety of exercises by allowing teachers to input their own fill-in-the-gap exercises. This would involve *ArtCheck* having more of an authoring system role.

(viii) Another ambitious suggestion was to allow file input into the WRITE option. This would allow students to input previous written work they had done into the system in order to have their article usage errors corrected. This would involve *ArtCheck* operating in a similar way to a spelling checker, but for article usage.

(ix) The final suggestion was that the system should be available on a personal computer. Obviously, if the system was to be of any commercial use, this would be essential, but *ArtCheck* was built with the intention of being a research prototype, so this is not presently a priority.

The critical evaluation of the system by students who were knowledgeable with respect to the field and also representative of the target population for the system, provided very useful feedback. The students evaluated both the effectiveness and the design of the system. Some of the smaller improvements have been incorporated into the system as a result of the evaluation exercise. Some of the other suggestions given above will be discussed as possible future improvements of the system, in Section 9.3.

The next section summarises the results of both the internal and external evaluation of *ArtCheck*.

8.5 Lessons learned: a summary

The feedback received during the external evaluation of the system was generally very positive. Both groups of students involved in evaluation confirmed that article usage was an area of difficulty for them and were enthusiastic about experimenting with the system. Most of the students said that they found the system easy and helpful to use. The internal evaluation shows that, on the whole, the system works as it was designed. Therefore, the positive findings of the evaluation exercise can be summarised as follows:

- The system parses a sizable subset of natural language.
- The error detection and analysis generally works as it was designed.
- Most of the students who used the system showed some improvement after using *ArtCheck* for a short period of time.
- Verbal and written feedback received from all the testers was generally very positive.

One of the reasons for the formative evaluation of an ICAI system is to identify the limitations of a system, so that justified and accurate claims may be made about its capabilities and so that further improvements can be made. In the field of ICAI, suggestions for further improvements can be used both to enhance the evaluated system, and to identify areas where further research would be useful. This means that there are a number of lessons to be learned from an evaluation exercise. In the evaluation of *ArtCheck* there were several areas found where the system is limited and where improvements can be made. These can be listed as follows:

- The natural language processing component of the system is not comprehensive enough, and the program performance would be improved if more time and effort was put into developing this part of the system.

- As discussed in Section 8.4.1, the initial assessment of the students level of ability could be made more reliable. Alternatively, the pre- and post- tests could include questions testing all 25 of the rules.

- Some of the testers gave some very useful feedback about the system, as described in Section 8.4.2. Some of these design improvements suggested could easily be incorporated into the system, though others were slightly more ambitious.

- As mentioned in Section 8.3.3, in the GAP option, the exercises could be tailored more to the student's particular problems, rather than being fixed exercises for each level.

The next chapter will analyse the contribution made by this system to the field of ICAI and AI. The results of the evaluation will be used to highlight the achievements of this work, address in more detail the limitations of the system, and offer some suggestions for further research.

Chapter 9

Discussion and Further Work

The initial aims of this thesis were given in Section 1.4, and are given again here, in order that the ensuing discussion can examine the extent to which the original aims have been met. The original aims were as follows:

- To develop an ICALL system for English article usage which demonstrates the use of various Artificial Intelligence techniques.
- To implement a set of rules for the article usage domain.
- To demonstrate the use of candidate elimination and version spaces in the generation of mal-rules.
- To demonstrate the interaction of the explanation facility and the student model in the generation of explanations which are tailored to the student's level of ability and learning preferences.

The preceding chapters have given a full account of the design, development and evaluation of the system *ArtCheck*. It is the purpose of this chapter to determine to what extent the thesis has addressed these and the contribution which has been made to the field of ICALL and AI.

The thesis as a whole has addressed a specific problem in the area of second language learning, that is, the English article usage of second language learners of English. The developed system demonstrates how techniques from Artificial Intelligence can be used to build an intelligent language learning tool. This tool has three primary features: firstly, it has knowledge of its subject area; secondly, it seeks to find a reason for a student's mistakes; and thirdly, gives a response which is tailored to the individual student.

This chapter will be divided into three distinct parts. Section 9.1 will compare the system *ArtCheck* with systems which make similar claims. This will serve to highlight the strengths (and weaknesses) of *ArtCheck* and identify the areas of this research which contribute something new to this field.

Section 9.2 will define the contribution of this thesis further, by analysing the individual components which make up the contribution of the thesis. This section will act as a summary of the achievements of this research, and an analysis of the extent to which the aims of the thesis have been met.

The analysis in Section 9.2, together with the findings in Chapter 8, will identify several areas where the work which has been reported in this thesis could be developed further. Section 9.3 will describe some of the potential developments which could be made.

9.1 Comparison with related work

In this section, *ArtCheck* will be compared with two other ICALL systems with which it has something in common: firstly, with the Fawlty Article Tutor (Kurup *et al*, 1992), an ICALL system for English article usage; and secondly, with ET (Fum *et al*, 1988), a language teaching system which also claims to generate mal-rules to account for the student's errors. It will then be compared with ACM, a student modelling system which uses machine learning techniques.

9.1.1 The Fawlty Article Tutor

The Fawlty Article Tutor (Kurup *et al*, 1992) is a recently developed system which deals specifically with English article errors. As it is an ICALL system addressing the article usage domain, it will be compared in detail with the system *ArtCheck*.

The main focus of the work carried out by Kurup is on the set of article usage rules which are taught to the students. The rules are in the form of production rules, where the condition part of the rule consists of a conjunction of values for certain **dimensions**. Kurup *et al* claim that six dimensions are required to express most usages of articles. The six dimensions are as follows:

- singular/plural
- mass/count
- definite/indefinite
- first introduction/subsequent introduction
- common noun/proper noun
- specific/general

Removing implausible or unlikely combinations of these rules, Kurup et al are left with 11 rules which, they claim, account for the majority of usages of English articles. These are given in Table 9–1.

No.	Rule
1	When a common, count noun is used as a proper noun, it is preceded by *the*
2	When a common, mass noun is used as a proper noun, it takes no article.
3	Proper, definite, specific, singular, count nouns are first introduced with no article.
4	Plural, definite, count nouns are always preceded by *the.*
5	Plural, indefinite, count nouns take no article.
6	Definite, specific, subsequent references to common mass nouns are preceded by *the.*
7	Indefinite, mass nouns take no article.
8	General, definite, singular, count nouns are preceded by *the.*
9	General, indefinite, singular, count nouns are preceded by *a/an.*
10	Specific, singular, count nouns are first introduced with *a/an.*
11	Subsequent references to specific, count nouns are preceded by *the.*

Table 9–1: Kurup's rules of article usage

The Fawlty Article Tutor teaches students about articles in the following way: the student is given a scenario, a sentence or two, in which there is an article missing. The student is then invited to fill in the missing article and justify the choice of article by selecting the appropriate rule. An example of a scenario used in the system is:

> *"Once upon a time, there was a lodge in Ontario. There was a lake in front of the lodge. There were _ __ trees in front of the lake. One day, a little boy who lived near the lodge was throwing stones into the water."* (Kurup *et al*, 1992, p88)

In the example given in (Kurup *et al*, 1992), the above scenario is presented to the student, and the student correctly selects *no article*, and rule 5. If the student makes an error, the system looks to see which of the dimensions the student has misunderstood, and gives an explanation of it. If the student has got more than one dimension wrong, the system consults the student model to see if one of the dimensions has caused difficulties for the student before, and if so, teaches that dimension.

A priority queue is used to decide which rule to teach next. The choice of rule depends on which rules are already known, which rules are not known and are similar to the known rules, and the level of difficulty assigned to each rule.

The first observation when comparing *ArtCheck* with the Fawlty Article Tutor is that the focus of the two systems is different, although the domain is the same.

Kurup *et al* have concentrated on narrowing down the domain of English article usage to a finite set of 11 rules, thus, they say, removing the need for highly lexically specific rules as found in many text books.

The Fawlty Article Tutor does not have any real knowledge of the article usage domain. With each scenario offered to the student, the system already knows the dimensions of the noun in question, the correct article to be used, and the correct rule which applies (Kurup, personal communication). No natural language understanding is attempted. Because all the answers to the questions are already stored in the system, the system is not "intelligent" in this respect. The only intelligent part of the system is its comparison of the student's article choice and justifying rule with that of the system to see which of the dimensions the student may have misunderstood.

The work described in this thesis, however, has been to **implement** an already existing set of article usage rules. A traditional set was chosen for this purpose because this is what most students are familiar with. The system uses all the information at its disposal, from the parser, the lexicon and the previous discourse, to decide which is the correct article to use. Thus, this system displays knowledge of its domain. In addition, *ArtCheck* uses the student model to generate accounts of consistent errors, and tailors explanations to individual users. In the Fawlty Article Tutor, the explanations are *"tailored to the student's answer"*, that is, the system identifies the appropriate rules and dimensions to explain to the student. The rules are graded according to their level of difficulty, and the student progresses through these rules. However, the explanations do not appear to be adapted to the student's level of expertise.

The Fawlty Article Tutor's lack of knowledge of the domain is part of the design of the system and reflects the difference in aims between this system and *ArtCheck*. The most significant part of the system is the reduction of the article usage domain to eleven succinct rules (Kurup *et al*, 1992). However, these rules, as given in Table 9–1, are not easy to remember, as they contain several conditions and are sometimes very similar. For example, the rules 8 and 9 could be easily confused:

Rule 8: *General, definite, singular, count nouns are preceded by* the.

Rule 9: *General, indefinite, singular, count nouns are preceded by* a/an.

In this case, what makes the difference is that in rule 8 it is the **definiteness** of the noun that suggests the article **the** (the definite article). The other dimensions, that is, **general**, **singular**, and **count** remain the same. A similar situation arises with rules 1 and 2 (see Table 9–1), where it appears that it is only the **mass/count** distinction which affects the article. In contrast, in *ArtCheck*, although there are many more rules, and they are not laid out for the student as clearly as they are with this system, the rules are less complex, more memorable, and the rule or rules which are given highlight the factors that really affected the choice of the article. In the Fawlty Article Tutor, because the rules are not easy to memorise, the student may find it difficult to use them in other contexts, for example, when trying to communicate verbally. Therefore, outside the session with the system, such rules may simply confuse the student.

The 11 rules incorporate the six key dimensions, which were given on page 205. However, the **definite/indefinite** dimension is not very clearly defined. The word *definite* obviously has different meanings to the linguist and to the lay person. Much of the more theoretical work in the usage of articles has revolved around finding a precise definition of the word *definite*, as a key to knowing when the definite article is used, for example (Chesterman, 1991). Kurup *et al* seem to rather avoid this issue. For instance, their dimensions do not include any mention of modification as a factor affecting article usage, whereas it has been seen in Chapter 5 that it plays a part in indicating the definiteness of a noun. It is possible that their understanding of *definite* is that it includes modified noun phrases, but this is not made explicit. There is additional confusion in this use of terminology, as most students know the articles as the **definite article** *the* and the **indefinite article** *a/an*. Overall, it is not made clear in (Kurup *et al*, 1992) what they actually intend the meaning of *definite* to be.

A final problem with the *definite/indefinite* dimension is its distinction from the *first introduction/subsequent reference* dimension. These are not necessarily independent dimensions as it is often the subsequent reference to an item which makes it definite. However, this point does not appear to be acknowledged.

A positive feature of the Fawlty Article Tutor which is lacking from *ArtCheck* is that the student is asked to justify her choice of article. This is a useful feature for several reasons. Firstly, it hopefully prevents the student from guessing the correct article. Secondly, it may encourage the student to **learn** the rule by repeatedly thinking about the application of individual rules. Thirdly, it helps the student modelling component to accurately identify which rules the student has acquired. In *ArtCheck* the system can identify the rules the student knows when the student appears to use a rule successfully. In the Fawlty Article Tutor, the student actually says which rule she is using and therefore there can be no doubt about the information in the student model. Therefore, instructing the student to give a justification for the article used is a useful feature of an article checking system. The possibility of extending *ArtCheck* to incorporate this feature will be discussed in Section 9.3.

In summary, the contrasting features of *ArtCheck* and the Fawlty Article Tutor can be seen in Table 9–2.

Feature	Fawlty Article Tutor	*ArtCheck*
Interface/Tutor	* Fill-in-the-gap exercises * Student justifies choice of article	* Fill-in-the-gap exercises * Free input of English sentences
Domain knowledge	* No real domain knowledge. * Correct answer stored * Uses 11 rules based on six dimensions	* Has knowledge of article usage and can detect errors * 28 rules based on structural and contextual information
Student modelling	* Keeps track of rules the student knows	* Keeps track of rules the student knows. * Generates mal-rules to account for consistent errors
Explanations	* Explanations tailored to student's answer	* Explanations tailored to student's answer, and to level of expertise, learning style and source of error.

Table 9–2: The contrasting features of *ArtCheck* and the Fawlty Article Tutor

9.1.2 ET: generating hypotheses

This system was described in Section 3.3.2. The part of the system which is of interest here is the student modelling component and the generation of mal-rules.

ET aims to instruct students in the use of English tenses. It gives the student exercises to complete which involve selecting the correct tense, and gives a diagnostic report when it has observed enough data to confirm its hypotheses of the student's errors. It uses three techniques for modelling the student's errors: overlay modelling, a bug library, and bug construction. On observing an error, the system formulates a number of hypotheses which could possibly account for the error, by using production rules which can generate such hypotheses. It then chooses one hypothesis to concentrate on and provides further examples which would confirm that hypothesis. If the student's answers do not confirm that hypothesis, then it switches to another hypothesis and provides more examples which would confirm it. It concludes the exercise when it has found one hypothesis and confirmed it. The student's error may be modelled as the ignorance of a tense (using the overlay method), or as a stereotyped mal-rule already present in the bug library, or the system may generate a new mal-rule to account for the errors observed. This is described as **adaptive modelling**, as the system adapts its modelling technique according to the complexity of the student's errors.

The **bug generation** process is carried out by the following process:

> *"...a persistent error in the selection of a tense, eg tense t1 in place of tense t2, causes the modeler to guess that the rules for both t1 and t2 are perturbated in the student knowledge and, therefore, have to be modified in order to reflect what is happening in the student mind. There are two kinds of modifications performed on the expert rules during bug generation: the rule for the tense t1 is generalized (by removing some and-clauses and adding some or-clauses to the condition part) while the rule for the tense t2 is specialised (by adding some and-clauses and removing some or-clauses)."* (Fum *et al*, 1988, p468)

An example of this is given in Figure 9–1.

ET > EXERCISE 1:
I (live) in this house for ten years. Now the roof needs repairing.
STUDENT > **am living** *Incorrect answer*

ET > EXERCISE 2: *Student not know tense?*
I (write) the letter but I can't find a stamp.
STUDENT > **have written** *Correct answer*

ET > EXERCISE 3:
Mary (feed) the cat? - Yes, she fed him before lunch.
STUDENT > **Has Mary fed** *Correct answer*

ET > EXERCISE 4: *Student has mal-rule?*
I (study) English, Maths and Science since breakfast, but now I am tired.
STUDENT > **am studying** *Incorrect answer*

ET > DIAGNOSIS

The answers to exercises 1 and 4 are wrong.
Your knowledge about the present continuous tense *Mal-rule explained*
seems to be the following:
'When a clause describes an event which has started in the past
    and lasts up to now,
    the event still continues at the present,
    the event duration isn't short,
    the discourse context isn't formal,
    the verb accepts the 'ing-form',
then apply the present continuous tense.'

This rule doesn't take into account that:
'The present continuous is used only when the event
takes place in the present or in the future.'

Your knowledge about the present continuous tense
seems to be the following:
'When a clause describes an event completed in the past,
    the event is linked to the present situation OR
    the event takes place in an indefinite period of the past,
then apply the present perfect tense.

This rule doesn't take into account that:
'The present perfect is (also) used when the event has
started in the past and lasts up to now.'

Figure 9–1: An example of interaction with ET

The student modelling component of *ArtCheck* differs from the system ET in several ways.

Firstly, *ArtCheck* does not have a bug library or list of stereotyped mal-rules, as ET does. This is because, as mentioned in Section 2.1.3, the construction of a bug library is costly in terms of time and effort, and the resultant list of mal-rules is not always a reliable indicator of the most frequently used bugs in that domain. It is not mentioned in (Fum *et al*, 1988) how the library of stereotyped mal-rules was built up and the empirical studies involved. However, one of the other problems of having a bug library is the consequent inability to deal with unanticipated errors, and ET overcomes this disadvantage by being able to generate new mal-rules when the observed student's errors do not coincide with any of the existing ones. The problem with this, however, is that if the system is able to perform this construction of a mal-rule, it is inefficient to have to consult the library of mal-rules at each stage to see if any of the existing mal-rules match the current hypothesis. This suggests that the library of mal-rules is rendered redundant. *ArtCheck*, on the other hand, as described in Section 6.2.1, uses the overlay method to model an error which represents the student's lack of knowledge, in that, the appropriate rule is missing from the student's genetic graph, and mal-rule generation to account for errors which reflect the perturbation of the student's knowledge about article usage. This seems a slightly simpler approach to the modelling task.

Another difference between the two systems relates to the frequency with which feedback is given to the student. *ArtCheck* gives some feedback after each of the sentences it observes, as to whether the article usage is correct or not. When it has observed enough data to generate a mal-rule it gives more detailed feedback to the student. This ensures that if there is noise in the data then the error is still remediated. In ET, the presence of noise could lead to the student having to input a great many sentences before the system admitted it could not find a suitable hypothesis. During this time, the student is not receiving any feedback or instruction from the system.

It is not made clear in (Fum *et al*, 1988) how exactly the addition and removal of clauses in the generation of a mal-rule takes place. Presumably, a considerable amount of backtracking must take place as the system endeavours to find the right combination of clauses that match the student's impression of the rules for the relevant tenses. The switching from one hypothesis to another also requires backtracking and approaching the problem from another angle. It is conceivable that in some situations, many exercises would have to be given to the student before the system was able to form a satisfactory conclusion. *ArtCheck*, however, uses a different technique, derived from the area of machine learning, which allows the system to gradually focus on the correct hypothesis without backtracking. In addition, the system can find a hypothesis to account for the error after only three errors, which may be later refined with the addition of more data.

One of the differences between ET and *ArtCheck* is that, in attempting to confirm the system's hypotheses, the system designs exercises which are tailored to the student's particular problem. *ArtCheck* unfortunately has to rely on the student making the same error again. Part of the reason for this is that in the WRITE option of the system, the student can input any sentences into the system and set exercises are not provided, and the system thus has no control over the types of errors made. However, the other part of the system, the GAP option, does provide exercises for the student, which could possibly be tailored to the student's apparent difficulty, and thus help the system to decide on the appropriate mal-rule more quickly. This option will be explored further in Section 9.3.

9.1.3 Machine learning: ACM

The system Automated Cognitive Modeller (ACM) was described in Section 2.1.5. The main point of comparison between this system and *ArtCheck* is the particular machine learning technique which is used to generate rules. In ACM, decision or discrimination trees are used, whereas *ArtCheck* uses version spaces and candidate elimination.

Decision trees are an example of a **discrimination-based** method for learning concepts or rules, the positive and negative instances serve to make a general rule more specific. Candidate elimination is an example of a technique which uses both discrimination and generalisation. The advantage of a discrimination-based method is its ability to learn disjunctive rules (Langley *et al*, 1984).

In ACM, the conditions which are learned by the system are steps in an arithmetic procedure. In choosing a path from the root to the bottom of the tree, different conditions are used to discriminate between positive and negative instances. The generated rule is the **disjunction** of the leaves of the tree which represent the positive instances.

This method was the original method implemented for the article usage domain (Sentance, 1992). However, candidate elimination, which combines generalisation and discrimination, was found to be at least as effective for this domain, so was selected for the final implementation. It was found that the candidate elimination algorithm transferred very easily to the article usage domain. This point will be discussed in Section 9.2.3.

9.2 Analysis of contribution

The previous section showed how *ArtCheck* compares with other ICALL and machine learning systems. Next, the main features of the system and the overall contribution of this thesis will be examined in more detail. This section will both summarise what has been achieved in this project, and analyse the success of individual achievements.

ArtCheck is made up of a number of components, including the expert model, which contains the article usage rules; the student modelling component, which is responsible for analysing the student's errors; and the explanation facility, which provides tailored responses for the student using the system. It would thus appear that the research described in this work could possibly contribute to a number of different areas, and that it would be easier to examine the contribution of this thesis by considering these different areas of the system separately. However, it is important to retain a picture of the work as a whole, and how it has met the aims of the thesis which were laid down in Section 1.4.

The overall aim of this work was to build a computational tool which addressed a particular problem in the area of language learning, and which was intelligent to the extent that it had knowledge of the domain and used reasoning to analyse and explain the behaviour of the students who use it. This chapter will reveal how successful this research has been in achieving its aim.

The main contributions of this work can be separated into the following areas:

- The system which has been developed has knowledge of the article usage domain.
- The genetic graph is used to represent the article usage domain and is augmented dynamically as proposed by Brecht (Wasson) and Jones.
- A traditional machine learning technique, candidate elimination using version spaces, has been successfully applied in a student modelling context.
- A generative approach is used in modelling the student's errors.
- The interaction with the student in the system is tailored to the individual student in terms of both expertise and learning strategy.

Each of these areas will be discussed in turn.

9.2.1 The system's knowledge of the domain

For the purposes of this thesis, the article usage domain is taken as consisting of a number of article usage rules, which have been taken from textbooks and grammar books used by language students and teachers. As described in Chapter 4, research into English articles has suggested a number of other ways of defining correct article usage, but from the point of view of developing a **tool** for use alongside other language learning material, it is advisable to adhere to a set of rules with which the learner is relatively familiar. The set of rules used in *ArtCheck* is given in Table 5–4.

In order to apply the article usage rules, information from a variety of sources is required. Some of this information can be obtained when the student input is parsed by the natural language processing component of the system. Allowing natural language input is often avoided in tutoring systems, firstly, because it is very complex to implement, and secondly, because the student's input is often open to misinterpretation. However, the provision of a natural language processing component is essential in the domain of article usage, because it is from understanding the student input that the system can make decisions about article usage. The natural language component of the system is not extensive enough to be able to parse any sentence in the English language, but can cope with a large enough subset to demonstrate the effectiveness of the system, as described in Section 8.3.1. The possibility of an extension to the natural language component of the system will be discussed in Section 9.3.1.

Other clues as to the correct article usage can be obtained from considering the context in which a noun phrase occurs. This includes knowing whether the noun phrase, or a semantically related item, has been mentioned before. As discussed in Section 5.3.3, the idea that the use of a noun phrase makes nouns which are semantically related to it more accessible as referents comes from the work of Sanford and Garrod (Sanford & Garrod, 1981). This theory was implemented in *ArtCheck* by linking a number of nouns in the lexicon with semantically related nouns, and made a significant contribution to the detection of errors in the article usage domain. The success of this technique is partly attributable to the fact that it was very easily incorporated into the lexicon, and demonstrates that an ICALL system can draw on research from a variety of other fields, including linguistics and psychology.

For the purposes of having an implemented set of article usage rules, the role of the context in determining the article used has obviously been simplified. A considerable amount of research in linguistics and computational linguistics has concentrated on the problems of coreference and of finding an antecedent for a noun phrase (eg (Hirst, 1981)), and this work does not claim to have dealt with this issue. As a result, the article usage rules may not be able to correctly predict the correct article in cases where a distantly related or paraphrased antecedent was involved. However, the implementation of the article usage rules shows a realistic and efficient use of the information readily available to the system, capitalising on what was available, yet not claiming to be infallible. Given these limitations, the system can be said to have knowledge of its domain, and is able to make decisions based on that knowledge.

9.2.2 The genetic graph

The topic of genetic graphs was introduced in Section 2.1.2, and the use of the genetic graph in the article usage domain discussed in more depth in Section 5.4.2. It was suggested that the genetic graph was a convenient form of representation for use in a dynamically updated student model (Brecht (Wasson) & Jones, 1988), and this was implemented in the system *ArtCheck*. In *ArtCheck*, the student's knowledge of the article usage rules is partly represented as an overlay over the expert knowledge graph, but can also include mal-rules, which are added to the genetic graph while the system is being used. Where a bug library or list of mal-rules are used, the student's deviant knowledge can still be represented on the graph, but the mal-rules in this case are already included in the graph and the graph is **static** as opposed to **dynamic**. The use of the genetic graph as a dynamic form of representation for new mal-rules demonstrates the potential of this means of representation. As far as the author is aware, the genetic graph has not been used previously to represent dynamically generated deviant rules and this work thus demonstrates the flexibility of this means of representation.

The most interesting aspect of the use of the genetic graph in the article usage domain is its role in **conflict resolution**. It has been seen in Section 5.4.3 that one of the links used for the article usage domain is the **priority link**. This link is used to indicate which of two rules would apply if they both fired at once. Conflict resolution is a very difficult area in the domain of article usage, as the distinct nature of several of the dimensions means that it is often the case that several rules fire at once. It is a difficult matter for students to understand which rules have priority over others and thus why they may be making article usage errors. The use of priority links in the article usage domain enables the system to model this aspect of the student's knowledge and comment on it if it appears to be the source of the student's error (see Section 7.3.3). The genetic graph offers an ideal means of modelling this knowledge, particularly as the available teaching material on English articles does not seem to tackle this issue at all.

The internal evaluation of the system highlighted some of the difficulties of resolving rule conflict in the article usage domain. A set of priority links is used in *ArtCheck* which represent the majority of article usages. These are mostly reliable, but there are examples, as given in Section 8.3.2, where the priority links are inadequate. A further study would be required to investigate whether subdividing the rules and introducing discriminatory information to restructure the priority links would be a more accurate alternative.

9.2.3 The generation of mal-rules with the candidate elimination technique

The use of version spaces and candidate elimination in *ArtCheck* was discussed in Section 6.3. The candidate elimination algorithm works well in the article usage domain and avoids the backtracking required in search algorithms. There are two main advantages to using this technique. Firstly, it appears that the technique, or part of it, may have some psychological validity. Secondly, this technique transferred very easily to the article usage domain. These two points will be discussed in turn.

Candidate elimination: a psychologically credible technique?

There has been some evidence that at least the generalisation part of the candidate elimination and focusing techniques is a psychologically meaningful process (Gilmore & Self, 1988). The evidence was found as a result of an experiment which was carried out to determine how people learned concepts and how they selected new hypotheses to confirm their hypotheses (Bruner *et al*, 1956). It was found that the most cognitively economical and the most frequently used strategies were **conservative focusing** and **focus gambling**. In both these methods, the first positive instance is used as a focus, and with the introduction of other positive instances, the hypothesis is generalised. Gilmore and Self discuss the fact that the experiment does not take account of negative instances, and that a learning algorithm should also include a discrimination component to incorporate negative instances.

Teachers have different methods of teaching and it is difficult to define human teaching methods in a way that facilitates a comparison with artificial teaching methods (Mark & Greer, 1992). It is difficult to say whether teachers in general look for consistencies in students' errors, and whether they are interested in finding an explanation for such consistencies. If they do not, then the system's diagnosis of mal-rules is in addition to what a teacher would do, and thus generates extra information which may be useful to the student. However, if the teacher does tend to look for consistencies in her student's errors, the results of this experiment offer evidence to suggest that the teacher may do it using an algorithm of the type implemented in *ArtCheck*.

The suitability of the algorithm for the article usage domain

It was suggested in Section 9.1.3 that an alternative method which has been used for generating mal-rules, the decision tree method, is not suitable for the article usage domain. As a result, in the development of *ArtCheck*, this method was abandoned in favour of the candidate elimination algorithm. In contrast, this method transferred very easily to the article usage domain. No amendments were necessary to the basic algorithm. What is interesting is the features of this particular domain which made it suitable for this technique. The condition part of a rule in the article usage domain consists of a conjunction of instantiated features. It is the nature of the features which describe the domain which appear to be the deciding factor in whether the candidate elimination technique is suitable. The features of the article usage domain can be categorised as being of two types. The first type includes those which take one of any number of values, though usually only two, and where the values are simple items. An example of this type of feature is the feature **number**, which can be instantiated as **singular**, **plural**, or left as a variable. The other type of feature in the article usage domain is that which has values which are more complex. An example of this type of feature is the **linguistic environment** of the noun phrase, which can be a kind of verb, or a preposition, etc. In this case, what may appear in the rule is that the immediately preceding linguistic environment of the noun phrase is a verb (any verb) or that it is a singular verb, or that it is a particular verb, for example *be*, with a specific person, tense and number. In other words, there are different **degrees** of specialisation within the individual feature.

These types of features occur in both rules and mal-rules in the article usage domain. The reason that the candidate elimination algorithm works so well in this domain is that both these types of features are suitable for an algorithm where a partial ordering of feature values is required. For example, in the case of the simple feature values, the feature is either instantiated or uninstantiated. In the generation of a mal-rule, a feature being uninstantiated means that it is not included in the resultant mal-rule. In the case of complex feature values, the feature can be instantiated to whatever degree is necessary. Thus, when considering positive and negative instances of a mal-rule, the feature will be instantiated sufficiently to account for all the positive instances which have been observed, and exclude the negative instances. Therefore the article usage domain is ideally suited to the candidate elimination technique. The conclusion is that any rule-based domain in which the features lend themselves to partial ordering would be equally suitable for this technique.

Limitations of candidate elimination

Self (Self, 1990) discusses an attempt to use the **focusing** algorithm in a collaborative learning system. As discussed in Section 2.1.5, focusing is a similar technique to candidate elimination. Self comments that there were several difficulties found during the project, some of them more related to collaborative learning, and some of them specific problems encountered when applying the focusing algorithm. For example, one point is that the algorithm cannot take account of the student's wider general knowledge, and that the closed world assumption which might hold in machine learning systems cannot be applied to educational systems. Another point is that students need to have a reason for learning concepts, and that conceptual learning should be embedded within a problem setting. However, neither of these two points apply to the system *ArtCheck*. Firstly, although there is a problem with regard to the students general or world knowledge when it comes to checking the article usage (see Section 5.3.4), this is not an issue when the system is generating mal-rules. Secondly, in *ArtCheck*, the student **is** learning the domain rules within a problem setting, and the system is doing the learning of the **mal-rules**, to model the student's difficulty. Therefore, outwith the context of a collaborative learning system, this objection does not apply.

One point which does apply to *ArtCheck* is that a considerable amount of data may be required before the student gives the system enough information with which to generate a mal-rule. This situation could be improved by tailoring the exercises to obtain more related data. This will be discussed in Section 9.3.4.

In summary, the candidate elimination technique is a useful, possibly psychologically credible technique for generating mal-rules. It is particularly suitable for the article usage domain, as demonstrated by its implementation in the system *ArtCheck*.

9.2.4 Validity of the generative approach

The most important feature of the implemented system from an Artificial Intelligence viewpoint is its ability to generate mal-rules to account for consistent errors made by the student. This is achieved using machine learning techniques.

There have been several criticisms of the idea of computationally generating analyses of errors (Laurillard, 1990; Hennessy, 1990). A representative view is given by Laurillard:

> *"A generative model is restricted to remediating that class of error that can be adequately described as procedural or syntactic, such as errors made by novice programmers, and that is a small subclass of the errors we need to address in most subjects."* (Laurillard, 1990, p53).

Laurillard discusses the difference between **deep-level processing** and **surface-level processing**. The difference is easy to understand with respect to the mathematical domain, as a student's blind observation of arithmetic rules (surface-level processing) may not mean she has any understanding of **why** she is doing what she is doing (deep-level processing). Similarly, a young child may be able to recite the numbers 1 to 10, without knowing what the numbers mean at all. In an ICAI context, the purpose of modelling the student is to help her understand why she is making mistakes. Therefore, to really help the student, the system must form a **deep-level** understanding of her errors. Laurillard's objection is that the **generation** of mal-rules only reflects the surface behaviour and therefore is not adequate for the purposes. As an alternative, she suggests that an empirical study would give more information about students' learning.

The fact that a deep-level understanding of the student's errors is desirable is not disputed. However, whether the generation of a mal-rule reflects the deep-level or surface-level processing depends on the domain which is under discussion. For different domains it is necessary to specify what the deep and surface-levels of processing actually are.

In this case, the domain is article usage, and it may be argued that the rules of article usage represent the student's surface-level processing, as they reflect what the student **does** in terms of choosing an article, as opposed to what the student **thinks** about the choice of an article. So what is the deep-level processing in the article usage domain? The study discussed in Section 4.3.2 (Herranen, 1978), in which the English article usage errors of Finnish students were analysed, resulted in some hypotheses about why the students were making particular sorts of errors. Most of these errors were attributed to language transfer[1]. However, the results of

[1] Language transfer does indeed have an effect on second language learning, as dis-

the study, though interesting, were only hypotheses, and may not have necessarily reflected the actual thought processes of individual students. The problem with attempting to find a deep-level representation of errors, is that it is difficult to be confident about the level of accuracy of the hypothesis.

Another problem with deep and surface-level processing, is that usually, though dependent on the domain, it is discussed as if it were a strict two-tier system, when it is not. In other words, there may be a fuzzy area between the two levels. Laurillard discusses the fact that the generation of mal-rules is able to reflect errors made by faulty reasoning. It could be argued that this is what is done in *ArtCheck*. In this case, it is harder to state categorically whether deep or surface-level processing is involved.

The alternative to the generation of mal-rules is to carry out empirical analyses for all domains. However, empirical analyses are costly in terms of time and effort. More significantly, the results of empirical analyses are not always consistent (Payne & Squibb, 1990). So even this time-consuming method may not be reliable. In contrast, the system *ArtCheck* is able to look at what the student **does** when using articles, and explain this to them. This has been achieved using domain-independent techniques which were easily and successfully implemented. The system then has a certain amount of information to give to the student about her errors. This information accurately reflects what the student has done, rather than a deeper-level approach which may rather inaccurately hypothesise what she may have thought while she was doing it. If the student is not able to express her own thought processes she may have difficulty understanding an explanation based on cognitive processes. This is particularly the case when the student is being tutored in the target language. Obviously, it is an advantage to obtain the best possible analysis of the student's errors. However, this system demonstrates

---

cussed in Section 3.1.3. This was not reflected in this system because the aim was to build a system which could be of use to students of all different nationalities, rather than just a particular group of students. The original motivation for the system stemmed from the idea that students with no articles in their language suffer from a **lack of positive transfer** (Odlin, 1989).

an efficient way of obtaining a lot of information about a student's errors without an empirical analysis.

In summary, this thesis does not claim that the mal-rules generated reflect the deepest level of cognitive processing of the student. What it claims is that the information which is gathered from constructing the mal-rule is useful to the student because it shows the student the consistency of her errors, in the hope that her awareness of this will enable her to correct these errors.

9.2.5 The system-student interaction

It has been discussed in Section 2.3.3 and Chapter 7 how the interaction between the system and the student should be tailored to the individual student. This ensures that the student receives the right amount of material, at the right level of difficulty, and presented in the right way. One of the aims in *ArtCheck* was to implement an explanation facility which had this ability for the article usage domain. The evaluation exercise described in Section 8.4 demonstrated that this was found to be effective and useful. The feedback from this part of the system was very positive, and included several suggestions for enhancing the system. However, these suggestions were mostly in the form of add-on features, rather than replacing parts of the existing system.

One of the interesting things about the explanation facility in *ArtCheck* is that it tailors the explanation of the error to the **learning strategy**. Section 3.2 discussed the reasons for doing this, and Section 7.2.2 discussed how it was implemented. The aim of that particular part of the system was to reflect two main learning strategies, as would be preferred by two different types of learner. Obviously there are many more learning strategies, but this simplified approach was used to demonstrate how research from the area of second language learning education could be incorporated into a working system. Another reason for doing this was to make students think about how they preferred to learn, thus encouraging a more active role for the student in the learning process. In many ways, using two different learning strategies was somewhat of an experiment, and the evaluation exercise showed that the students were fairly happy with this part of the system. However, if future systems are to focus on this area of the student's learning preferences, it must be made much more sophisticated. The importance of learning strategies could be explained to the student, and more choices of learning styles offered. Alternatively, students could try out different learning styles for the same material, to discover for themselves how they preferred to learn. However, *ArtCheck* has demonstrated how a system could adapt to different students' learning styles, in a way that a classroom teacher is not able to. It also illustrates how second language learning tools should be designed with the expert knowledge of second language learning teachers and educational experts in mind.

The final point about the way the system interacts with the student refers to the nature of the system itself. *ArtCheck* is a system which detects and corrects errors, and coaches the student by picking up on the student's weak points. As a system it has more of a **remediative** role than a **tutorial** role. The system's knowledge of the domain consists of the sort of article usage rules which are used in classrooms and textbooks. This means that the system is designed to be used as a tool, rather than a teaching system in its own right. The reason for this is partly to do with the domain, because students learn little bits about article usage as they learn more of the English language. It is unlikely that a teacher would sit down with a class of beginners and teach them everything there was to know about articles. Therefore, different students using *ArtCheck* will have had a different degree of exposure to English articles. This is another reason why the system must be tailored to students with different levels of ability.

9.3 Directions for further research

In addition to evaluating the results of the research described in this thesis, it is possible to identify several areas in which the work could be either improved upon, or developed further. Thus, one of the benefits of carrying out research is that it provides a pointer to further research.

In this section, several different areas of further research will be discussed. Some of the limitations of the system have been mentioned in earlier chapters, and there is obviously scope for more work in these particular areas. Other areas of the discussion will focus on more general and far-reaching research which could be explored as a result of the present work. The discussion in this section will cover the following areas:

- Improvements to the natural language understanding component
- The computational model of article usage
- Maintenance of the student model
- The acquisition of a set of frequently-used mal-rules
- Improvements to the tutorial component

Each of these areas will be discussed in turn.

9.3.1 Improvements to the natural language understanding component

It has been discussed in Sections 5.5.2, 8.3.1 and 9.2.1 that the natural language processing component of the system can understand only a limited subset of natural language and this is an obvious disadvantage if the system is intended to accept free input from real students. One of the problems highlighted in Section 5.5.2 is that the grammar only contains a minimal number of features, such as are sufficient for the article usage domain, and the result of this is that far too many different parses are found for more complex sentences. The system at present has the unenviable task of trying to discern what is a plausible parse without access to semantic information. There are two possible solutions to this problem.

The first solution would be to considerably enhance the grammar by including many more semantic features, such as those included in the Alvey Natural Language Tools Grammar (Grover *et al*, 1989).

This would mean that information about the selectional restrictions of verbs, that is, a more precise definition of the context in which different types of verbs were allowed to occur, would be included in the grammar, and a smaller number of more accurate parses would be found.

Another possible solution would be to avoid the many problems of building a tailor-made natural language understanding program, by using an existing natural language system instead, for example, the Alvey Natural Language System, of which the grammar mentioned earlier is a part. Such systems have been built specifically for the purpose of understanding natural language, using several man-years of research, and are obviously more effective than the simple natural language processor presently available in *ArtCheck*. However, the problem with this course of action is that such a tool may not necessarily be compatible with the present system, and more time and effort may be required to link the two together, perhaps eventually culminating in a rewritten version of *ArtCheck*. It may be suggested that all ICALL systems which intend to accept natural language as input should therefore be built on top of comprehensive natural language tools, as exemplified by (Schwind, 1990) and (Kurtz *et al*, 1990). However, the other drawback to this course of action is that attaching an ICALL system to a natural language tool will probably greatly increase the computing resources required to run the system. This implies that there is a trade-off to a certain extent between the flexibility of a natural language interface and the size of the resultant system.

In developing ICALL systems which require the student to communicate with the system in the target language rather than the student's mother tongue, there is an additional complication with regards the natural language interface. The system is only interested in the article usage errors made by the student, but the student may make many other grammatical errors. Although the system is not interested in analysing these errors, it has to be able to make some sense of the student's input. It was described in Section 5.5.3 that the present system attempts to do this by asking the student for a general classification of unknown words.

Another approach to parsing ill-formed input which could be implemented in this system is the **multi-level relaxation** used in XTRA-TE (Kurtz *et al*, 1990). This system was described in Section 3.3.2. With multi-level relaxation, the system responds to an unsuccessful parse, by first relaxing the syntactic agreements restrictions, then the semantic agreement restrictions, and then both, until a parse is found. This requires a mechanism whereby the different types of agreement can be switched on and off. If this were to be built into *ArtCheck* this may be a more successful way of accepting ill-formed input.

An alternative technique for parsing ill-formed input combines both top-down and bottom-up parsing (Mellish, 1989). If the bottom-up parser is unsuccessful in finding a parse, a top-down parse is attempted, until a point is reached where there is a contradiction between the category being looked for by the top-down parser, and that which was found by the bottom-up parser. The system can then hypothesise about what the correct parse should be.

To be able to reliably parse ill-formed input, a combination of these techniques could be used. One disadvantage of incorporating these techniques into the parser might be that the parser would become very slow.

9.3.2 Maintenance of the student model

As it stands, *ArtCheck* offers the student explanations for any article usage errors which have been detected, and records any mal-rules which the system believes the student is operating with. The next stage for any human teacher would be to notice when the student is not making the error any more, that is, when the misconception has been successfully remediated. If the explanations are helpful, it would be expected that the student should learn from them. The student model should then be able to reflect this. At present, when a student uses an article usage rule correctly where she had previously made errors, *ArtCheck* is unsure whether the student has corrected the error as a result of the explanations given, or whether there is inconsistent data. This works in so far as it prevents mal-rules being recorded where they do not exist, but a better facility for updating the student model would be preferable. One suggestion might be to use truth maintenance techniques in the student model to continually check whether the student still has the mal-rule, but it is not clear exactly how these would work. However, the maintenance of the student model is an important issue and this would be an interesting direction for further research.

9.3.3 The acquisition of a set of frequently-used mal-rules

The system *ArtCheck* does not consult a library of mal-rules containing frequently used mal-rules. Each student is treated individually, and the mal-rules are generated from the observation of the student's errors. There is some evidence from the algebra domain that a given set of mal-rules may not be applicable for different groups of students (Payne & Squibb, 1990). On these grounds, therefore, there does not appear to be any need for the system to remember the mal-rules which it has found and build up a library of mal-rules over time. It is just as easy for the system to generate the mal-rule in response to the observed errors.

However, there may be a justification for remembering the diagnosed mal-rules on other grounds. It may be worthwhile to examine whether there was a set of frequently-used mal-rules for the article usage domain, by keeping a record of all the mal-rules which had been found. Studies such as that by Herranen (Herranen, 1978) (discussed in Section 4.3.2) have found particular areas of article usage which students find difficult, and attempted to give accounts of the consistent errors. If the *ArtCheck* system were to be used for a considerable length of time, with a number of different students, it might also be able to draw up a picture of the types of consistent errors which students were making, and retain the system's analysis, in terms of mal-rules, for those errors. This could be an interesting contribution to the research on article usage errors. It could easily be implemented by retaining the student's mal-rules after they had finished using the system.

9.3.4 Extending the tutorial component

As mentioned in Section 9.2.5, *ArtCheck* has a mainly **remediative** as opposed to a **tutorial** role. Its main job is to detect and correct errors. Therefore, there are many enhancements which could be made to the system by promoting the tutorial part of the system.

Firsly, the system could set out the **instructional goals** for each exercise, by explaining to the student what they should have achieved by the end of the exercise. After the exercise, the system would then be able to inform the student of the extent to which those goals had been met.

Secondly, the system could explain and teach rules without waiting for the rule to be violated. At present, if the student gets all of the article usages correct, she may not ever be taught any of the rules. The system will assume that she already knows the rules. However, it may be helpful for her to have the option of seeing the appropriate rule if so desired. This change could easily be incorporated into the system.

The system could also offer more tuition on the priorities between the article usage rules. It was discussed in Section 9.2.2 that different rules often fire at the same time, and it is difficult for students to know which rules take priority over others. At present, the system explains the relationship between the rules to the student when it believes that it is a misunderstanding of the priorities between the rules which has caused the error. The system could be extended to give more tuition on this when teaching the students the article usage rules. In addition, an interesting piece of further research might also be to carry out an empirical study to see how pervasive this source of error is amongst language learners.

One of the points raised in Section 8.5 was that in the GAP option, the exercises could be tailored more to the student's particular errors. At present, there are two GAP exercises for each level, which cover the subset of article usage rules which the student should know at that level. During each exercise, each of the rules is tested once or twice. The system was designed in this way in order to give the student an overall view of article usage, including some of the less common usages. However, the system could alternatively be designed so that when the student made an error, the next part of the exercise focussed on the particular difficulty which the student had. This would give the system more data with which to generate mal-rules, and it is this method which is used in ET (Fum *et al*, 1988).

More importantly, it could be seen as making better use of the student's time with the system, by concentrating in more depth on the errors the student has made. The disadvantage would be that the system may not be able to give the student as much opportunity to use a wide range of article usage rules. An alternative suggestion might be to leave the GAP exercises as they are at present, but follow them with **follow-up exercises** which tried to identify more accurately what the student's difficulties were. This suggestion would be preferable because one of the advantages of the GAP exercises is that they are given in the form of a coherent story, which allows the student to take into account the discourse history of the noun phrases which are being used.

Finally, one of the design improvements which was suggested by the students taking part in the evaluation of the system (see Section 8.4.2) was that when all the GAP exercises were exhausted, the teacher should be able to enter new passages into the system, with gaps in the appropriate places. This would mean that an infinite number of new exercises could be used with the system, and would mean that students could spend longer with the system without exhausting all the prepared material. To implement this enhancement of the system, the system would obviously have to have an interface for the teacher to carry out the "authoring", and the teacher would have to test the new exercise to ensure that the system knew all the vocabulary and could parse the sentences. However, this could be carried out in conjunction with the improvements to the natural language processor suggested in Section 9.3.1, and would be an interesting piece of further work to carry out.

There are therefore many improvements which could be made to the tutorial component. However, because students may have had different exposure to teaching on article usage before using the system, even students at the same general level of English, it may not be appropriate to alter the style drastically to a "give a lesson then practice" type of system. A slight redressing of the balance to include more teaching, and less emphasis on the student's errors, may be sufficient.

9.4 Conclusion

This chapter has discussed the contribution and the achievements resulting from the research described in this thesis. *ArtCheck* is an implemented remediative system which demonstrates a number of Artificial Intelligence techniques. Section 9.1 compared various aspects of *ArtCheck* with other systems which have attempted to do something similar. *ArtCheck* compared favourably with The Fawlty Article Tutor (Kurup *et al*, 1992) and ET (Fum *et al*, 1988), other ICALL systems. In addition, the candidate elimination technique used for generating mal-rules in *ArtCheck* seemed to be more suitable for the article usage domain than the alternative discrimination-based method used in ACM (Langley *et al*, 1984). Section 9.2 considered each of the main features of the system, and examined to what extent it was a contribution to the field of ICALL. Section 9.3 discussed several possibilities for enhancements of the system, and pointed to several interesting areas of further research.

The final chapter of this thesis offers a few concluding remarks.

Chapter 10

Conclusion

This thesis has demonstrated how Artificial Intelligence techniques can be incorporated into an ICALL system for English articles. The aim of the project, as laid down in Chapter 1 was to develop an ICALL system for English article usage which had both knowledge of its domain and a student modelling component which would allow it to adapt to the student's behaviour.

Thus, this thesis describes a system, *ArtCheck*, which has knowledge of the domain of English article usage, and offers remediation for the article usage errors of second language learners of English. This system demonstrates the use of a student modelling component which interacts with the explanation component of the system to give tailored remediative responses, and shows how machine learning techniques can be used by an ICALL system to generate an account for the student's errors.

The different aspects of the system were described in individual chapters. Chapters 2 and 3 gave an overall picture of the research which served as a background to the work described here, and described research on which the present work was built. Chapters 4 and 5 described the domain of English article usage and how knowledge of article usage was built into the ICALL system *ArtCheck*. Chapter 6 described the student modelling component of the system and Chapter 7 the explanation facility. Chapter 8 described the process of evaluating the developed system. In Chapter 9, the contribution of the project to the research in this area was considered, with pointers to further work which could be carried out.

There are several other points which can be made in conclusion with respect to the implementation of the system *ArtCheck*.

It was described in Chapter 5 that *ArtCheck* has knowledge of the article usage rules and applies them using the syntactic, morphological, lexical and discoursal information which it has at its disposal. The rules implemented reflect a **structural** and **contextual** approach to article usage. The structural approach (Yotsukura, 1970) involves looking at the type of noun phrase and its linguistic environment. The contextual approach determines definiteness by looking for identical and semantically related (Sanford & Garrod, 1981) noun phrases in the preceding discourse. Obviously, the role of context in determining definiteness is more complex than has been recognised in this thesis, but the linguistic problems of coreference and finding distant antecedents are considered to be outwith the scope of this thesis.

In implementing a set of article usage rules, an issue has been highlighted which is not addressed in text books and grammar books on the subject, and that is, the difficulties associated with resolving conflict between article usage rules which fire at the same time. A set of priority links between the rules have been developed by extensively testing the system. It is important to address this issue in the article usage domain, although, given the inconsistency of natural language, it is difficult to find a completely reliable set of priority links.

The system has the ability to generate mal-rules in response to the student's errors. The use of the generative approach means that the student's errors do not have to be anticipated in advance, and the student model is not static. It also maximises the use of the information available to the system. The mal-rules which are generated mirror the surface-level behaviour of the student, and may possibly reflect a deeper-level problem which the student has with article usage. Other empirical analyses of article usage may give more insight into the particular deep-level processing problems that students have, but these would be specific to particular groups of students. This thesis demonstrates a method that is easy to implement, offers an analysis of students' behaviour, and can be used by any students with article usage problems.

The system generates explanations which are tailored to the student's level of expertise and learning preferences, and to the source of the error made. This demonstrates that the interaction between the student modelling and the explanation modules is an important contributor to the effectiveness of the explanation facility. The system draws on research from the area of second language learning in incorporating the student's learning preferences into the student model. There are many suggestions in the literature about strategies that students use. In this system, two particular learning strategies are modelled, which affect the content of the explanation given to the student. The purpose of this was to illustrate how learning strategies could work in an ICALL system. *ArtCheck* has been evaluated, as described in Chapter 8, with satisfactory results. Some of the limitations, as well as the positive points, of the system were highlighted and some of the testers gave some constructive feedback on possible improvements to the system.

The system operates in two modes, GAP mode and WRITE mode. The GAP mode works well and could be used as a plausible tool for students experiencing difficulties in article usage. Some suggestions were given in Section 9.3 as to how this part of the system could be further developed. The WRITE mode is more ambitious and demonstrates how the system attempts to analyse the article usage errors in any sentence of English with which it is presented. The problems with this part of the system reflect the hard problems associated with natural language processing research. However, it is advantageous for the student to be able to check their own article usage by asking the system to evaluate sentences which they have composed.

To conclude, the contribution which this thesis offers to the area of Artificial Intelligence and Education, particularly to the area of Intelligent Computer-Aided Language Learning was described in Section 9.2 and can be summarised as follows:

- The system which has been developed has knowledge of the article usage domain.
- The genetic graph is used to represent the article usage domain and is augmented dynamically as proposed by Brecht (Wasson) and Jones.

- A traditional machine learning technique, candidate elimination using version spaces, has been successfully applied in a student modelling context.

- A generative approach is used in modelling the student's errors.

- The interaction with the student in the system is tailored to the individual student in terms of both expertise and learning strategy.

Thus, *ArtCheck* has contributed something to the field of ICALL in that is demonstrates an implementation of an area of natural language which has not been attempted before. It illustrates the use of machine learning in ICAI tools, and it demonstrates the implementation of learning strategies in an ICALL system. The result is an ICALL system which can recognise, analyse and respond to English article usage errors made by second language learners of English. It is hoped that more research in this area will lead to more intelligent tools in the area of language learning, which will enhance the learning experience of many language learners.

References

Adams, Richard. (1972). *Watership Down*. Harmondsworth, Middlesex: Puffin.

Arnold, J., Haavista, A., Kalleen, M-L., Nikkanen, L. and Suurpää, A-L. (1988a). *You Too 7*. Espoo: Weilin and Göös.

Arnold, J., Haavista, A., Kalleen, M-L., Nikkanen, L. and Suurpää, A-L. (1988b). *You Too 8*. Espoo: Weilin and Göös.

Bialystok, Ellen. (1983). Some factors in the selection and implementation of communication strategies. In Færch, Claus and Kasper, Gabriele, (eds.), *Strategies in interlanguage communication*. New York: Longman.

Brecht (Wasson), Barbara and Jones, Marlene. (1988). Student models: The genetic graph approach. *International Journal of Man-Machine Studies*, 28:483–504.

Brown, J. S. and Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2:155–192.

Brown, J. S. and VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, 4:379–426.

Bruner, J. S., Goodnow, J. A. and Austin, G. A. (1956). *A study of thinking*. New York: Wiley.

Bundy, Alan, Silver, Bernard and Plummer, Dave. (1985). An analytical comparison of some rule-learning programs. *Artificial Intelligence*, 27:137–181.

Burton, Richard R. (1982). Diagnosing bugs in a simple procedural skill. In Sleeman, D. and Brown, J. S., (eds.), *Intelligent Tutoring Systems*. London: Academic Press.

Carberry, Sandra. (1988). Modelling the user's plans and goals. *Computational Linguistics*, 14(3).

Carr, Brian and Goldstein, Ira P. (1977). Overlays: A theory of modelling for computer aided instruction. AI Memo 406, MIT.

Carr, Brian. (1977). WUSOR II: A computer aided instruction program with student modelling. AI Memo 417, MIT.

Catt, Mark and Hirst, Graeme. (1990). An intelligent CALI system for grammatical error diagnosis. *Computer Assisted Language Learning*, 3.

Chesterman, Andrew. (1977). Error analysis and the learner's linguistic repertoire. In Sajavaara, Kari and Lehtonen, Jaakko, (eds.), *Jyväskylä Contrastive Studies 4: Contrastive Papers*. Department of English, University of Jyväskylä.

Chesterman, A. (1991). *On Definiteness: A study with special reference to English and Finnish.* Cambridge: Cambridge University Press.

Chin, David N. (1989). KNOME: Modelling what the user knows in UC. In Kobsa, Alfred and Wahlster, Wolfgang, (eds.), *User Models in Dialog Systems.* Berlin : Springer-Verlag.

Christophersen, P. (1939). *The articles. A study of their theory and use in English.* Copenhagan: Munksgaard.

Clancey, William J. and Letsinger, Reed. (1981). Neomycin: Reconfiguring a rule-based expert system for application to teaching. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 829–836.

Clancey, William J. (1987). *Knowledge-Based Tutoring: The GUIDON program.* MIT Press.

Close, R. A. (1972). *A Reference Grammar of English.* Longmans.

Cohen, Robin, Jones, Marlene, Sanmugasunderam, Amar, Spencer, Bruce and Dent, Lisa. (1989). Providing responses specific to a user's goal and background. *International Journal of Expert Systems*, 2(2):135–162.

Corder, S. Pitt. (1967). The significance of learner's errors. *International Review of Applied Linguistics*, 5. Reprinted in: Schumann, J. H. and Stenson, N. (1974). New Frontiers in Second Language Learning. Massachusetts: Newbury House Publishers.

Corder, S. Pitt. (1974). Idiosyncratic Dialects and Error Analysis. In Richards, J C, (ed.), *Error Analysis: Perspectives in Second Language Acquisition.* London: Longman.

del Soldato, Teresa. (1992). Motivational planning. In Brusilovsky, P. and Stefanuk, V., (eds.), *Proceedings of the East-West Conference on Emerging Computer Technologies in Education*, pages 293–298. Moscow: ICSTI.

Dietterich, T, London, B, Clarkson, K and Dromey, G. (1982). Learning and inductive inference. In Cohen, P. and Feigenbaum, E., (eds.), *The Handbook of Artificial Intelligence: Vol III.* Los Altos: William Kaufmann.

Dillenbourg, P. and Self, J. (1992). A framework for learner modelling. *Interactive Learning Environments*, 2(2):111–137.

Dulay, H C and Burt, M K. (1973). Should we teach children syntax? *Language Learning*, 23:245–57.

Dulay, H C and Burt, M K. (1974). You can't learn without goofing: an analyis of children's second language errors. In Richards, J C, (ed.), *Error Analysis: Perspectives in Second Language Acquisition.* London: Longman.

Dulay, Heidi, Burt, Marina and Krashen, Stephen. (1982). *Language Two*. New York: Oxford University Press.

Ellis, Rod. (1985). *Understanding second language acquisition*. Oxford: Oxford University Press.

Ellis, Rod. (1992). *Second Language Acquisition and Language Pedagogy*. Clevedon: Multilingual Matters.

Elsom-Cook, Mark. (1988). Guided discovery tutoring and bounded user modelling. In Self, John, (ed.), *Artificial Intelligence and Human Learning*. Chapman and Hall Computing.

Escott, J. A. (1988). Problem solving by analogy in novice programming. Unpublished M.Sc. thesis, Department of Computational Science, University of Saskatchewan.

Færch, Claus and Kasper, Gabriele. (1983). Plans and strategies in foreign language communication. In Færch, Claus and Kasper, Gabriele, (eds.), *Strategies in interlanguage communication*. New York: Longman.

Finin, Timothy W. (1989). GUMS - A general user modelling shell. In Kobsa, Alfred and Wahlster, Wolfgang, (eds.), *User Models in Dialog Systems*. Berlin : Springer-Verlag.

Fum, Danilo, Giangrandi, Paolo and Tasso, Carlo. (1988). ET: An intelligent tutor for foreign language teaching. In *Proceedings of ITS-88, Montreal*.

Gazdar, Gerald and Mellish, Chris. (1989). *Natural Language Processing in Prolog*. Addison-Wesley.

George, H. V. (1972). *Common Errors in Language Learning. Insights from English*. Rowley, Mass.:Newbury House.

Ghemri, Lila. (1992). A cognitive framework for second language error diagnosis. In Frasson, C., Gaultier, C. and McCalla, G. I., (eds.), *Proceedings of ITS 92*. Berlin: Springer Verlag.

Gilmore, David and Self, John. (1988). The application of machine learning to intelligent tutoring systems. In Self, John, (ed.), *Artificial Intelligence and Human Learning*. Chapman and Hall Computing.

Gilmore, David J. (1986). Concept learning: alternative methods of focussing. In *Proceedings of International Meeting In Advances in Learning*. Les Arcs, France.

Goldstein, Ira P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In Sleeman, D. and Brown, J. S., (eds.), *Intelligent Tutoring Systems*. London: Academic Press.

Gowers, Sir Ernest. (1986). *The Complete Plain Words.* London: Penguin, *Revised by* Greenbaum, Sidney and Whitcut, Janet.

Grover, Claire, Briscoe, Ted, Carroll, John and Boguraev, Bran. (1989). The Alvey Natural Language Tools Grammar. Technical report, Computer Laboratory, University of Cambridge, No. 162.

Guillaume, G. (1919). *Le Problème de l'article et sa solution dans la langue française.* Paris:Hachette.

Halasz, F. and Moran, T. P. (1982). Analogy considered harmful. In Moran, T. P., (ed.), *Eight short papers in user psychology,* Cognitive and Instructional Sciences Series, CIS-17. Palo Alto: Xerox Palo Alto Research Center.

Halliday, M. A. K. (1967). Notes on transitivity and theme in English, Part 1. *Journal of Linguistics,* 3:37–81.

Hawkins, John A. (1978). *Definiteness and Indefiniteness. A Study in Reference and Grammaticlaity Prediction.* London: Croom Helm.

Hennessy, Sara. (1990). Why bugs are not enough. In Elsom-Cook, Mark, (ed.), *Guided Discovery Tutoring: A Framework for ICAI research.* London: Paul Chapman Publishing.

Herranen, Tauno. (1978). Errors made by Finnish University students in the use of the English article system. In Sajavaara, Kari, Lehtonen, Jaakko and Markkanen, Raija, (eds.), *Jyväskylä Contrastive Studies 6: Further Contrastive Papers.* Department of English, University of Jyväskylä.

Hewson, John. (1972). *Article and noun in English.* The Hague: Mouton.

Hirst, Graeme. (1981). *Anaphora in natural language understanding: a survey.* Springer-Verlag.

Horwitz, Elaine K. (1987). Surveying student beliefs about language learning. In Wenden, Anita and Rubin, Joan, (eds.), *Learner Strategies in Language Learning.* Englewood Cliffs, N.J.: Prentice Hall.

Howe, J. A. M. (1984). Artificial intelligence. In Unwin, Derick and McAleese, Ray, (eds.), *The encyclopeadia of educational media communications and technology.* Westport: Greenwood Press, second edition.

Hunt, Earl B., Marin, Janet and Stone, Philip J. (1966). *Experiments in Induction.* New York: Academic Press.

Jespersen, O. (1949). *A modern English grammar on historical principles. Part VII : Syntax.* London :Allen and Unwin. (This work was completed after Jespersen's death by Niels Haislund.).

Johnson, W. Lewis. (1986). *Intention-Based Diagnosis of Novice Programming Errors.* Morgan Kaufmann.

Kass, Robert and Finin, Tim. (1988). The need for user models in generating expert system explanations. MS-CIS-88-37, University of Pennsylvania.

Kellerman, Eric. (1984). The empirical evidence for the influence of the L1 in interlanguage. In Davies, Alan, Criper, C and Howatt, A P R, (eds.), *Interlanguage.* Edinburgh: Edinburgh University Press.

Kleinmann, Howard. (1977). Avoidance behaviour in second language acquisition. *Language Learning*, 27:93–107.

Kurtz, Barry L, Chen, Li and Huang, Xiuming. (1990). An English-Chinese language learning system using adaptive correction and multipass error diagnosis. *Artificial Intelligence in Education*, 1(4).

Kurup, Maya, Greer, Jim E. and McCalla, Gordon I. (1992). The Fawlty Article Tutor. In Frasson, C., Gaultier, C. and McCalla, G. I., (eds.), *Proceedings of ITS 92.* Berlin: Springer Verlag.

Kwansy, S. C. and Sondheimer, N. K. (1981). Relaxation techniques for parsing ill-formed input. *AJCL*, 7(2):99–108.

Lado, R. (1957). *Linguistics across Cultures.* Ann Arbor: University of Michigan.

Langley, Pat and Ohlsson, Stellan. (1984). Automated cognitive modeling. In *Proceedings of the AAAI*, pages 193–197.

Langley, Pat, Ohlsson, Stellan and Sage, Stephanie. (1984). A machine learning approach to student modelling. Technical report, Carnegie-Mellon University, CMU-RI-TR-84-7.

Laurillard, Diana. (1990). Generative student models: The limits of diagnosis and remediation. In Elsom-Cook, Mark, (ed.), *Guided Discovery Tutoring: A Framework for ICAI research.* London: Paul Chapman Publishing.

Leech, Geoffrey. (1989). *An A-Z of English Grammar and Usage.* Edward Arnold.

Littman, David and Soloway, Elliott. (1988). Evaluating ITSs: The cognitive science perspective. In Polson, Martha C. and Richardson, J. Jeffrey, (eds.), *Intelligent Tutoring Systems.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Mann, W. (1983). An overview of the PENMAN text generation system. In *Proceedings of the National Conference of Artificial Intelligence*, pages 261–265.

Mark, Mary A. and Greer, Jim E. (1992). Methods for intelligent tutoring system evaluation. In Brusilovsky, P. and Stefanuk, V., (eds.), *Proceedings of the East-West Conference on Emerging Computer Technologies in Education*, pages 204–209. Moscow: ICSTI.

Mark, Mary A. and Greer, Jim E. (1993). Evaluation methodolgies for intelligent tutoring systems. *Journal of Artificial Intelligence in Education.*

McCoy, Kathleen F. (1988). Reasoning on a highlighted user model to respond to misconceptions. *Computational Linguistics*, 14(3).

McDonald, D and Pustejovsky, J D. (1985). Description directed natural language generation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 799–805.

McKeown, Kathleen. (1985). *Text Generation: using discourse strategies and focus constraints to generate natural language text.* Cambridge:Cambridge University Press.

McKeown, Kathleen, Wish, Myron and Matthews, Kevin. (1985). Tailoring explanations for the user. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 794–798.

Mellish, Chris S. (1989). Some chart-based techniques for parsing ill-formed input. In *Proceedings of the Association of Computational Linguistics*, volume 27.

Mitchell, Tom M. (1977). Version spaces: A candidate elimination to rule learning. In *Proceedings of the Fifth International Joint Conference of Artificial Intelligence*, pages 305–310.

Mitchell, Tom M. (1982). Generalization as search. *Artificial Intelligence*, 18:203–226.

Mitchell, Tom, Utgoff, Paul and Banerji, Ranan. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. In Michalski, R S, Carbonell, J G and Mitchell, T M, (eds.), *Machine Learning: An Artificial Intelligence Approach*, volume 1. Palo Alto, CA: Tioga.

Moore, Johanna D. and Paris, Cécile L. (1991). Requirements for an expert system explanation facility. *Computational Intelligence*, 7.

Moore, Johanna D. and Swartout, William R. (1988). Explanation in expert systems: A survey. Technical report, Information Sciences Institute, University of S.California.

Morik, Katharine. (1985). User Modelling, Dialog Structure and Dialog Strategy in HAM-ANS. In *Proceedings of the 2nd Conference of the European Chapter of the Association for Computational Linguistics*, pages 268–273, Geneva, Swizerland.

Naiman, N., Fröhlich, M., Stern, H. and Todesco, A. (1978). *The good language learner.* Toronto, Ontario: Institute for Studies in Education.

Odlin, Terence. (1989). *Language Transfer. Cross-linguistic influence in language learning.* New York: Cambridge University Press.

Oller, John and Redding, Elcho. (1971). Article usage and other language skills. *Language Learning*, 21(1):85–95.

O'Malley, J. Michael and Chamot, Anna Uhl. (1990). *Learning Strategies in Second Language Acquisition.* Cambridge University Press.

Paris, Cécile L. (1988). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3).

Payne, Steven J. and Squibb, Helen R. (1990). Algebra mal-rules and cognitive accounts of error. *Cognitive Science*, 14(3):445–481.

Pusack, J. P. (1983). Answer-processing and error correction in foreign language CAI. *System*, 11:53–64.

Quillici, Alex, Dyer, Michael and Flowers, Margot. (1988). Recognising and responding to plan-oriented misconceptions. *Computational Linguistics*, 14(3).

Quinlan, J. Ross. (1983). Learning efficient classification procedures and their application to chess endgames. In Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., (eds.), *Machine Learning: An Artificial Intelligence Approach*, volume 1. Palo Alto, CA: Tioga.

Quinlan, J. Ross. (1986). The effect of noise on concept learning. In Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., (eds.), *Machine Learning: An Artificial Intelligence Approach*, volume 2. Los Altos, CA: Morgan Kaufman.

Quirk, R., Greenbaum, S., Leech, G. and Svartvik, J. (1972). *A Grammar of Contemporary English.* London : Longman.

Rich, Elaine. (1979). User modelling via stereotypes. *Cognitive Science*, 3:329–350.

Ritchie, G., Russell, G., Black, A. and Pulman, S. (1992). *Computational Morphology.* Cambridge, Mass.: MIT Press.

Rubin, Joan. (1987). Learner strategies: Theoretical assumptions, research history and typology. In Wenden, Anita and Rubin, Joan, (eds.), *Learner Strategies in Language Learning.* Englewood Cliffs, N.J.: Prentice Hall.

Sanford, A. J. and Garrod, S. J. (1981). *Understanding Written Language:Explorations of comprehension beyond the sentence.* John Wiley&Sons.

Schuster, Ethel and Finin, Tim. (1986). VP^2 : The role of user modelling in correcting errors in second language learning. In Cohn, A. G. and Thomas, J.R., (eds.), *Artificial Intelligence and Its Applications.* New York: Wiley.

Schuster, Ethel. (1986). The role of native grammars in correcting errors in second language learning. *Computational Intelligence*, 2:93–98.

Schwind, Camilla B. (1990). An intelligent language tutoring system. *International Journal Man-Machine Studies*, 33:557–579.

Self, John. (1987). Student models: what use are they? In Ercoli, P. and Lewis, R., (eds.), *Artificial Intelligence Tools in Education.* North Holland.

Self, John. (1990). Bypassing the intractable problem of student modelling. In Frasson, Claude and Gauthier, Gilles, (eds.), *Intelligent Tutoring Systems: At the crossroads of Artificial Intelligence and Education.* Norwood, NJ : Ablex.

Selinker, Larry. (1974). Interlanguage. In Schumann, John H. and Stenson, Nancy, (eds.), *New frontiers in Second Language Learning.* Massachusetts: Newbury House Publishers.

Selinker, Larry. (1992). *Rediscovering interlanguage.* New York: Longman.

Sentance, Sue. (1992). Analysing misconceptions in the domain of second langauge learning. In Brusilovsky, P. and Stefanuk, V., (eds.), *Proceedings of the East-West Conference on Emerging Computer Technologies in Education*, pages 287–292. Moscow: ICSTI.

Shortliffe, Edward H. (1976). *Computer Based Medical Consultations: MYCIN.* Elsevier North Holland Inc.

Singleton, D. M. (1981). Language transfer: a review of some recent research. Technical report, Trinity College, Dublin.

Sleeman, Derek H. (1982). Assessing aspects of competence in basic algebra. In Sleeman, D. and Brown, J. S., (eds.), *Intelligent Tutoring Systems.* London: Academic Press.

Sleeman, D. (1983). Inferring (mal) rules from pupil's protocols. In Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., (eds.), *Proceedings of the International Machine Learning Workshop.*

Sleeman, Derek H. (1987). PIXIE: A shell for developing intelligent tutoring systems. In Lawler, R. W. and Yazdani, M., (eds.), *Artificial Intelligence and Education*, volume 1, pages 239–265. Ablex, Norwood, NJ.

Swartout, William R. (1981). Explaining and justifying expert consulting programs. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 815–822.

Szwedek, A. J. (1976). *Word order, sentence stress and reference in English and Polish.* Alberta: Linguistic Research, Inc.

Wallis, J. W. and Shortliffe, E. H. (1982). Explanatory power for medical expert systems: studies in the representation of causal relationships for clinical consultations. *Methods of Information in Medicine*, 21:127–136.

Wang, Yang and Garigliano, Roberto. (1992). An intelligent language tutoring system for handling errors caused by transfer. In Frasson, C, Gaultier, C and McCalla, G I, (eds.), *Proceedings of ITS 92.* Berlin: Springer Verlag.

Weischedel, Ralph M. (1983). Meta-rules as a basis for processing ill-formed input. *AJCL*, 9(3–4):161–177.

Wenden, Anita. (1987). How to be a successful language learner: Insights and prescriptions from l2 learners. In Wenden, Anita and Rubin, Joan, (eds.), *Learner Strategies in Language Learning.* Prentice Hall International (UK) Ltd.

Westlake, P., Stoneham, D., Nurmi, J., Leppänen, R., Pitkänen, E-L., Hirvenoja, T. and Häkinen, I-S. (1988). *OK English.* Porvoo: WSOY.

Wick, M. R. and Slagle, J. R. (1989). An explanation facility for todays expert system. *IEEE Expert*, 4(1).

Wick, Michael R. and Thompson, William B. (1992). Reconstructive expert system explanation. *Artificial Intelligence*, 54(1-2).

Yotsukura, Sayo. (1970). *The Articles in English: A Structural analysis of usage.* Hague: Mouton & Co.

APPENDICES

Appendix A

System interface

ArtCheck runs on a Sun-4 workstation, using X-Windows. Most of the dialogue with the user is carried out using one main window. Subsidiary windows are used to display passages of text for the GAP option. The GAP and WRITE options will be illustrated in Section A.3 and Section A.4. Section A.2 will show the initial screens given when the student first uses the system.

A.1 Running the system

The student starts up the system by typing the command:

Xartcheck

A window icon then appears, which the student then clicks on to open the window. The following sections show the screens which are displayed at various stages.

A.2 Introduction to *ArtCheck*

After starting up the system, the screen showed in Figure A–1 is displayed.

If the student has not previously used the system, she is then asked for some preliminary information in order to initialise the student model, as shown in Figure A–2.

```
  ***********************************************************
 *                                                           *
*                                                             *
*                                                             *
*       A      RRR   TTTTTT  CCCC   H  H   EEE   CCCC  K  K   *
*      A A     R R   TTTTTT  C      H  H   E     C     K K    *
*     A   A    RR      TT    C      HHHH   EEE   C     KK     *
*    AAAAAAA   R R     TT    C      H  H   E     C     K K    *
*    A     A   R  R    TT    CCCC   H  H   EEE   CCCC  K  K   *
*                                                             *
*                                                             *
 *                                                           *
  ***********************************************************


                    A R T C H E C K


           This program will help you learn about
           using articles correctly.


           There are two types of exercise offered, one
           where you choose a missing article in a sentence,
           and the other where you type sentences and the
           program corrects them.


           When the program expects you to enter something,
           you will see the prompt '>>>>'.
           After you have typed your answer, press <RETURN>
```

Figure A–1: Introduction to *ArtCheck*

```
Firstly, what is your name?

    >>>> brodie

Secondly, how good is your English?

          b      beginner
          i      intermediate
          a      advanced

     >>>> i
```

```
Some people prefer to learn rules about articles.
Other people prefer to see examples of how articles
are used.  Which do you prefer?

          r      rules
          e      examples
          n      no preference

     >>>> r
```

Figure A–2: System initialisation

```
******************************************************
*              M A I N   M E N U                     *
*                                                    *
*                                                    *
*   Choose a type of exercise or quit:               *
*                                                    *
*                                                    *
*         gaps      fill in the gaps                 *
*         write     type in whole sentences          *
*         quit      leave the system                 *
*                                                    *
*                                                    *
******************************************************

>>>>  gaps
```

Figure A–3: Main menu in *ArtCheck*

The main menu for the system, as given in Figure A–3, is then displayed. Students are then asked to choose to do the GAP exercise, or the WRITE exercise.

A.3 The GAP option

If the student selects "gaps", then the system enters GAP mode. The instructions for the GAP mode are given in Figure A–4.

```
You will be given some exercises which
involve filling in the missing article.
Click on the window marked "EXERCISE" to
see a short passage with the missing articles
marked with "****".  The program will then
ask you to fill in each missing article.
```

Figure A–4: GAP mode instructions

The system selects an appropriate gap-filling exercise for the student. Two exercises are associated with each level except for Level 4, which only has one.

Each exercise appears in full in a small box in the corner of the screen so that the student can read it through. The missing articles, where the student has to fill in the gaps, are marked with ****. The system then steps through each sentence in turn, asking the student to fill in the correct article. If the student gets the answer correct, the student model is updated, and the system proceeds on to the next sentence. If the student's answer was incorrect, the system gives an interactive explanation based on the student's ability and learning preferences.

The student is allowed to complete as many exercises as she wishes. The system keeps a record of which exercises have been completed. Where a student has completed both exercises at her level, and still wishes to continue, she is given an exercise of the next level up. The system moves the student up or down the levels as appropriate as described in Section 7.2.1. Although this is a gap-filling exercise, the system does not know in advance what the missing articles are. Instead, it treats each sentence, with the article which the student has typed in inserted at the appropriate place, as if it were a new sentence typed in at the keyboard (though taking into account the discourse history). It parses the sentence and detects and analyses the article errors before generating the explanation.

A.4 The WRITE option

In the WRITE option, the student is prompted to enter sentences of English on a particular subject, or as a short story of their choice. The subject options are kept as general as possible, as they are simply given to give the students some ideas. The vocabulary in the lexicon is not restricted to particular domains indicated by the choices of subjects given.

The instructions for the write option are given in Figure A–5.

```
You will see a list of possible subjects.
Choose one, and write a short paragraph about
that subject.  The program will tell you about
any article errors after each sentence.
Each sentence must end with a full stop.
```

Figure A–5: WRITE mode instructions

The menu of options in WRITE mode are shown in Figure A–6.

The enusing interaction between the system and the student is shown in Appendix B.

```
*****************************************************
*      W R I T E - A - S T O R Y    M E N U         *
*                                                   *
*                                                   *
*     Choose a subject or quit:                     *
*                                                   *
*                                                   *
*        1       Your stay in Edinburgh             *
*        2       Your favourite place in Edinburgh  *
*        3       Your home town                     *
*        4       Somebody you know                  *
*        5       A short story of your choice       *
*                                                   *
*     exit       EXIT to main menu                  *
*                                                   *
*                                                   *
*****************************************************


>>>>  2
```

Figure A–6: WRITE mode menu

Appendix B

A sample session

In this sample session, the student is of Level 2 and selects the GAP mode. The passage she is shown (in a separate window) is as follows:

> **John's journey.**
>
> John is **** teacher. He works in **** best school in **** Glasgow. The school has **** 900 students. It is in ***** middle of the city. John teaches history. He is also the head of **** fifth year.
>
> Today, John is going to **** school by **** bus. His car is in **** garage being serviced. He waits for ***** hour before the bus comes. Then the bus gets stuck in **** traffic jam. He gets to **** school one hour late. He is late for **** first lesson. All **** students are waiting for him.

During the session shown here, the student progresses part of the way through the passage, and then decides to try the WRITE mode for a while.

```
===> John is ****  teacher.
                                   Choose the correct article:
                                         1   a
                                         2   an
                                         3   the
                                         4   no article
                                         e   exit to main menu

                                    >>>>  4
<teacher> in
<John is teacher> is incorrect.
```

```
It should be: <a teacher>.

      Select:
             m   more
             q   quit explanation

      >>>>  m

The rule is:

 >>>  Use <a> or <an> before singular count nouns which come
      after the verb <to be>.

      Select:
             m   more
             q   quit explanation

      >>>>  m

An example is:

  ***  Harold is a librarian.


===> He works in ****  best school
                                              Choose the correct article:
                                                     1   a
                                                     2   an
                                                     3   the
                                                     4   no article
                                                     e   exit to main menu

                                                >>>>  3

Well Done!  <He works in the best school> is correct.


===> in ****  Glasgow.
                                              Choose the correct article:
                                                     1   a
                                                     2   an
                                                     3   the
                                                     4   no article
                                                     e   exit to main menu

                                                >>>>  3
```

```
<the Glasgow> in
<in the Glasgow> is incorrect.

It should be: <Glasgow>.

      Select:
             m   more
             q   quit explanation

      >>>>  m

The rule is:

 >>>  Usually you dont need an article before a proper noun.

      Select:
             m   more
             q   quit explanation

      >>>>  m

An example is:

  ***  My name is John.


===> The school has ****  nine hundred students.

                                         Choose the correct article:
                                                1   a
                                                2   an
                                                3   the
                                                4   no article
                                                e   exit to main menu

                                          >>>>  4

Well Done!  <The school has nine hundred students> is correct.
```

```
===> It is in ****  middle of the city.
                                        Choose the correct article:

                                               1    a
                                               2    an
                                               3    the
                                               4    no article
                                               e    exit to main menu

                                          >>>>  4


<middle> in
<it is in middle of the city> is incorrect.

It should be: <the middle>.

      Select:
             m   more
             q   quit explanation

      >>>>  m


The rule is:

 >>>  If the noun comes before <of> you can use <the>.

      Select:
             m   more
             q   quit explanation

      >>>>  q



===> John teaches history.
===> He is also the head of ****  fifth year.

                                        Choose the correct article:

                                               1    a
                                               2    an
                                               3    the
                                               4    no article
                                               e    exit to main menu

                                          >>>>  e
```

```
****************************************************
*            M A I N   M E N U                     *
*                                                  *
*                                                  *
*  Choose a type of exercise or quit:              *
*                                                  *
*                                                  *
*        gaps      fill in the gaps                *
*        write     type in whole sentences         *
*        quit      leave the system                *
*                                                  *
*                                                  *
****************************************************

>>>>  write


          You will see a list of possible subjects.
          Choose one, and write a short paragraph about
          that subject.  The program will tell you about
          any article errors after each sentence.
          Each sentence must end with a full stop.


****************************************************
*      W R I T E - A - S T O R Y   M E N U         *
*                                                  *
*                                                  *
*     Choose a subject or quit:                    *
*                                                  *
*                                                  *
*       1      Your stay in Edinburgh              *
*       2      Your favourite place in Edinburgh   *
*       3      Your home town                      *
*       4      Somebody you know                   *
*       5      A short story of your choice        *
*                                                  *
*     exit     EXIT to main menu                   *
*                                                  *
*                                                  *
****************************************************
```

```
    >>>>  2


Enter sentence >>  My favourite place is Edinburgh Castle.


Well done!  No errors in this sentence!
Continue? (y/n)       >>>>  y


Enter sentence >>  I go there once month.

<month> in
<I go there once month> is incorrect.

It should be: <a month>.

      Select:
             m   more
             q   quit explanation

      >>>>  m

The rule is:

 >>>  After a <how often> word use <a> or <an>.

      Select:

             m   more
             q   quit explanation


      >>>>  m


An example is:

  ***  Once a day.


Continue? (y/n)       >>>>  y


Enter sentence >>  It is best castle I have been to.
```

```
<best Castle> in
<It is best Castle I have been to> is incorrect.

It should be: <the best Castle>.

      Select:

             m   more
             q   quit explanation
```

```
      >>>>  m

The rules which apply are:

 >>>  Use <the> before adjectives like <most> and <best>.
 >>>  You can use <the> where the noun is followed by
      a relative clause.

      Select:
             m   more
             q   quit explanation

      >>>>  m

Some examples are:

  ***  John is the nicest teacher.
  ***  I like the house with the red door.


Continue? (y/n)      >>>>  1
INVALID INPUT  - PLEASE TRY AGAIN!
      >>>>  In the castle there are a cannons.
INVALID INPUT  - PLEASE TRY AGAIN!
      >>>>  y



Enter sentence >>  In the castle there are a connons.

I do not know the word <connons>.
Please type it again >>>  cannons.

OK - I know it now.
```

```
<cannons> in
<In the Castle there are a cannons> is incorrect.

It should be:

      Select:

            m   more
            q   quit explanation

      >>>>  m
```

```
The rule is:

 >>>  Use no article or the before a plural noun.

      Select:
            m   more
            q   quit explanation

      >>>>  m


An example is:

  ***  John likes bananas


Continue? (y/n)       >>>>  n
```

Appendix C

Data Collection

Three data collection exercises were carried out during the course of this project. Two were aimed at students and one at teachers of English as a second language. Section C.1 gives the exercise and results of the exercise carried out in Oulu involving 38 advanced learners of English. Section C.2 gives a multiple choice exercise which was carried out in Edinburgh involving 15 students who were on a year's exchange from Helsinki University. Finally, Section C.3 gives a questionniare which was sent to Finnish teachers of English.

C.1 The Oulu exercise

C.1.1 The exercise

COMPREHENSION

Instructions: Read through the passage below a couple of times, then, **without reference to the passage,** *answer the questions overleaf.*

How to save a Rainforest

Clive Wicks, Senior Conservation and Development Executive, of the World Wide Fund for Nature

The conservation and rehabilitation of tropical forests is WWF-World Wide Fund for Nature's top priority.

Over 50% of the forests that existed at the beginning of the century have been destroyed. This rate of destruction is increasing day by day. It is estimated that by the year 2030 nearly all the forests outside the protected areas will have been destroyed with the exceptions of the Amazon and Congo basins. Even those will disappear before the middle of the next century unless action is taken now.

WWF believes that the future of the forests and the future of mankind are closely linked together. We have already severely damaged the ability of our planet to support humans, animals and plants. Man himself has become an endangered animal in many countries.

When the forests were destroyed, the top soil disappeared, climate changed and the ability to grow crops was also destroyed. Over 7 million hectares of land in Africa is threatened by desertification. In addition, if the predictions of climate change, caused by the destruction of the forest and by the burning of fossil fuels, are accurate then by the year 2010 we may also have to deal with 200 million refugees from countries such as Bangladesh and the South Sea Islands that are being swamped as the seas rise.

WWF belives that the rainforests could be saved if action is taken now.

We must help countries that have rainforest to help themselves.

In order to save the forests that remain we must understand the current problems.

The causes of destruction vary in different parts of the World. In Africa the main causes are the rapid population explosion combined with logging. In South America the main cause is the conversion of Tropical forest lands to cattle ranches and the population explosion, in South East Asia the main cause of destruction is logging. The recent WWF publication "Timber from the South Seas" clearly indicates that Japan is the leader in logging in South East Asia. An International Tropical Timber Organisation (ITTO) investigation in 1989 showed that less than 1% of all the logging operations in the tropics are sustainable.

The future of the forest must be considered on both a global and a regional basis, not purely a national basis. The destruction of forests in one country can lead to death and destruction in another country. Examples of this are the Himalayas and Ethiopia. When the trees in Nepal are destroyed people in India and Bangladesh are killed. When the water catchment areas for the Nile in Ethiopia, Zaire and Uganda are destroyed people are killed by floods or starvation in Egypt and Sudan.

We must stop blaming the poor, often landless, people who destroy the forest. Instead we must help them. In many cases they do not understand the forest or how to live in it. Land distribution is very unfair. In some countries the rich own over 80% of the land, and worst of all, they own nearly all the good land which forces the poor to live on the poor or marginal land.

Most countries with tropical forest are very poor and at present they are often forced to sell their logs at low prices in order to pay their national debt or even to meet their basic funning expenses. No country should be forced to log its forests in an unsustainable way to pay its debt or meet its day to day needs. All logging must be sustainable.

Probably the most important thing that the forest produces is water, not wood. Water is the essence of life on earth. The forest's ability to provide water in a controlled way is unreplaceable. The forest's ability to recycle water and to control, not only the micro climate, but also to contribute to macro climates must be costed. The mature forest acts as a carbon sink. Burning releases all the CO2.

The cost of protecting the forest must be paid for by the developed as well as the developing world.

QUESTIONS

1. What is the message of this article by the WWF?

Answer:

2. Why do people destroy the rainforests?

Answer:

3. What are "logging operations"?

Answer:

4. What are the effects of rainforest destruction?

Answer:

5. Why must the future of the forest be considered on a global level?

Answer:

6. What is the most important function of the rainforest, according to the author?

Answer:

7. What do you think should be done?

Answer:

MULTIPLE CHOICE QUESTIONS

Instructions: For each question, tick the sentence which you think is correct

1. A We spent the morning visiting a Tower of London.
   B We spent the morning visiting the Tower of London.
   C We spent the morning visiting Tower of London.

2. A The dogs are my favourite pets.
   B Some dogs are my favourite pets.
   C Dogs are my favourite pets.

3. A I am very fond of the nature and particularly like the birdwatching.
   B I am very fond of nature and particularly like birdwatching.
   C I am very fond of the nature and particularly like birdwatching.

4. A Roses are much more beautiful than daffodils.
   B The roses are much more beautiful than daffodils.
   C The roses are much more beautiful than the daffodils.

5. A The importance of family in our society is overestimated.
   B The importance of a family in our society is overestimated.
   C The importance of the family in our society is overestimated.

6. A An aeroplane has revolutionised travel.
   B The aeroplane has revolutionised travel.
   C Aeroplane has revolutionised travel.

7. A The bread is a very nutritious food.
   B Some bread is a very nutritious food.
   C Bread is a very nutritious food.

8. A If you have a fever you must stay in bed.
   B If you have a fever you must stay in the bed.
   C If you have a fever you must stay in a bed.

9. A The olive grows only in warm climates.
   B Olive grows only in warm climates.
   C An olive grows only in warm climates.

10. A The English is a difficult language to learn.
    B An English is a difficult language to learn.
    C English is a difficult language to learn.

....

Finally, it would help me if you could complete this section on the length of time you have spent learning English. Thank you.

Total number of years spent learning English:

Years learning English at University level:

Examinations passed:

C.1.2 The results

Table C–1 shows the results of the comprehension exercises, in terms of how many times each article usage rule was used correctly and incorrectly. The correlation between the rule numbers and the actual rules was given in Table 5–5 on page 103. Table C–2 gives a summary of the results from the multiple choice exercise, showing how many students got each question correct, and what the distribution of the answers was. Table C–3 shows more specifically, for individual students, which questions they answered correctly and incorrectly.

Rule	Correct usages	Incorrect usages
1	33	4
2	260	3
3	141	12
4	190	11
5	73	8
6	4	0
10	6	0
12	5	0
14	24	2
15	8	1
17	1	0
19	17	1
21	0	1
22	69	17
23	57	4
24	7	1
Total	895	65

Table C–1: Article usage rules applying in comprehension answers

Questions	1	2	3	4	5	6	7	8	9	10
Correct answers	**B**	**C**	**B**	**A/C**	**C**	**B**	**C**	**A**	**A**	**C**
A	0	1	0	38	19	2	0	38	9	0
B	37	0	31	0	9	16	0	0	25	0
C	1	37	7	0	10	20	38	0	4	38
Total correct	37	37	31	38	10	16	38	38	9	38

Table C–2: Multiple choice results: Distribution of answers

Questions	1	2	3	4	5	6	7	8	9	10	
Correct Answers	**B**	**C**	**B**	**A/C**	**C**	**B**	**C**	**A**	**A**	**C**	No. Correct
1	.	.	.	.	A	C	.	.	B	.	7
2	.	.	.	.	A	C	.	.	C	.	7
3	.	.	C	.	.	A	.	.	B	.	7
4	.	.	.	.	B	C	.	.	B	.	7
5	.	.	.	.	B	.	.	.	.	.	9
6	.	.	.	.	.	.	.	.	.	.	10
7	.	.	.	.	A	C	.	.	B	.	7
8	.	.	.	.	A	C	.	.	B	.	7
9	.	.	C	.	A	C	.	.	B	.	7
10	.	.	.	.	A	C	.	.	B	.	6
11	.	.	.	.	A	C	.	.	B	.	7
12	C	.	C	.	A	C	.	.	B	.	7
13	.	.	.	.	B	.	.	.	B	.	9
14	.	.	.	.	.	C	.	.	.	.	9
15	.	.	.	.	.	C	.	.	B	.	8
16	.	.	.	.	A	C	.	.	B	.	7
17	.	A	.	.	A	.	.	.	.	.	8
18	.	.	.	.	A	C	.	.	B	.	7
19	.	.	.	.	.	.	.	.	B	.	9
20	.	.	.	.	B	C	.	.	.	.	8
21	.	.	C	.	A	C	.	.	B	.	6
22	.	.	.	.	B	C	.	.	B	.	7
23	.	.	C	.	.	C	.	.	B	.	7
24	.	.	C	.	.	.	.	.	.	.	9
25	.	.	.	.	B	C	.	.	B	.	7
26	.	.	.	.	B	.	.	.	C	.	8
27	.	.	.	.	A	.	.	.	B	.	8
28	.	.	.	.	.	.	.	.	.	.	10
29	.	.	.	.	B	A	.	.	.	.	8
30	.	.	.	.	A	.	.	.	.	.	9
31	.	.	.	.	.	.	.	.	B	.	9
32	.	.	.	.	B	C	.	.	B	.	7
33	.	.	.	.	A	.	.	.	B	.	8
34	C	.	.	.	A	.	.	.	C	.	7
35	.	.	.	.	.	.	.	.	B	.	9
36	.	.	C	.	A	.	.	.	B	.	7
37	.	.	.	.	A	C	.	.	B	.	7
38	.	.	.	.	A	.	.	.	C	.	10

Table C–3: Multiple choice results: Incorrect answers for each student

C.2 The Edinburgh exercise

C.2.1 Multiple choice exercise

MULTIPLE CHOICE QUESTIONS

Instructions: For each question, tick the sentence which you think is correct

1. A We spent the morning visiting a Tower of London.
   B We spent the morning visiting the Tower of London.
   C We spent the morning visiting Tower of London.

2. A The dogs are my favourite pets.
   B Some dogs are my favourite pets.
   C Dogs are my favourite pets.

3. A I am very fond of the nature and particularly like the birdwatching.
   B I am very fond of nature and particularly like birdwatching.
   C I am very fond of the nature and particularly like birdwatching.

4. A An aeroplane has revolutionised travel.
   B The aeroplane has revolutionised travel.
   C Aeroplane has revolutionised travel.

5. A The bread is a very nutritious food.
   B Some bread is a very nutritious food.
   C Bread is a very nutritious food.

6. A If you have a fever you must stay in bed.
   B If you have a fever you must stay in the bed.
   C If you have a fever you must stay in a bed.

7. A The olive grows only in warm climates.
   B Olive grows only in warm climates.
   C An olive grows only in warm climates.

8. A My favourite sport is skiing.
   B My favourite sport is the skiing.
   C My favourite sport is some skiing.

9. A The dolphins are the most intelligent creatures next to man.
   B Some dolphins are the most intelligent creatures next to man.
   C Dolphins are the most intelligent creatures next to man.

10. A Glasgow is a largest city in Scotland.
    B Glasgow is the largest city in Scotland.
    C Glasgow is largest city in Scotland.

11. A The secretaries are annoyed because photocopier is broken.
    B The secretaries are annoyed because the photocopier is broken.
    C The secretaries are annoyed because a photocopier is broken.

12. A People who live in the Netherlands speak Dutch.
    B People who live in Netherlands speak Dutch.
    C People who live in some Netherlands speak Dutch.

13. A Earth moves round sun.
    B The earth moves round the sun.
    C An earth moves round a sun.

14. A We discussed our plans for the day over the breakfast.
    B We discussed our plans for the day over a breakfast.
    C We discussed our plans for the day over breakfast.

15. A My young children do not like reading.
    B My young children do not like the reading.
    C My young children do not like a reading.

16. A Please meet me in the bookshop in the centre of town.
    B Please meet me in bookshop in the centre of town.
    C Please meet me in a bookshop in the centre of town.

17. A Telephone can be an expensive means of communication.
    B The telephone can be an expensive means of communication.
    C A telephone can be an expensive means of communication.

18. A The English is a difficult language to learn.
    B An English is a difficult language to learn.
    C English is a difficult language to learn.

....

Finally, please answer the following questions:

1. Do you think English articles(*the*, *a/an*) are particularly difficult for Finnish students? If so, why is this?

2. Do you still find it difficult to remember which article to use in some cases?

3. Are there any particular areas of article usage which you have difficulty with or find confusing? If so, which? (Please give examples if you can, either from the questions above, or your own).

4. For how many years have you been learning English?

C.2.2 Multiple choice results

This section shows the results from the multiple choice exercise shown above which was given to 15 students in Edinburgh. Table C–4 gives a summary of the results from the multiple choice exercise. It shows how many students got each question correct, and what the distribution of the answers was. Table C–5 shows more specifically, for individual students, which questions they answered correctly and incorrectly.

Questions	1	2	3	4	5	6	7	8	9
Correct answers	**B**	**C**	**B**	**B**	**C**	**A**	**A**	**A**	**C**
A	0	0	1	1	0	13	4	15	2
B	13	0	8	8	0	0	10	0	0
C	2	15	6	6	14	1	1	0	13
Total correct	13	15	8	8	14	13	4	15	13
Questions	10	11	12	13	14	15	16	17	18
Correct answers	**B**	**B**	**A**	**B**	**C**	**A**	**A**	**B**	**C**
A	0	1	12	1	8	14	14	8	0
B	0	12	3	14	0	1	0	5	0
C	15	2	0	0	7	0	1	2	15
Total correct	15	12	12	14	7	14	14	5	15

Table C–4: Multiple choice results: Distribution of answers

Questions	1	2	3	4	5	6	7	8	9	
Correct answers	**B**	**C**	**B**	**B**	**C**	**A**	**A**	**A**	**C**	
1	.	.	.	C	.	.	.	.	.	
2	.	.	.	.	.	.	B	.	.	
3	C	.	C	.	.	.	B	.	.	
4	.	.	.	C	.	.	B	.	.	
5	.	.	.	.	.	.	.	.	.	
6	.	.	.	.	.	.	B	.	.	
7	.	.	C	.	.	.	C	.	.	
8	.	.	A	A	.	.	B	.	.	
9	.	.	.	.	.	.	.	.	A	
10	.	.	C	C	.	.	B	.	A	
11	.	.	.	.	.	.	.	.	.	
12	.	.	C	C	.	C	B	.	.	
13	.	.	.	C	.	.	B	.	.	
14	C	.	C	C	.	.	B	.	.	
15	.	.	C	.	DK	DK	B	.	.	

Questions	10	11	12	13	14	15	16	17	18	Total
Correct answers	**B**	**B**	**A**	**B**	**C**	**A**	**A**	**B**	**C**	correct
1	.	C	.	.	A	.	.	A	.	14
2	.	.	.	A	A	.	.	A	.	14
3	.	.	.	.	.	.	.	.	.	15
4	.	A	B	.	.	.	.	A	.	13
5	.	.	.	.	.	.	.	.	.	18
6	.	.	.	.	A	.	.	.	.	16
7	.	.	.	.	.	.	.	.	.	16
8	.	.	.	.	A	.	.	C	.	13
9	.	.	B	.	.	.	.	C	.	15
10	.	.	.	.	A	.	.	A	.	12
11	.	.	.	.	.	.	.	.	.	18
12	.	.	.	.	A	.	.	A	.	12
13	.	.	.	.	A	.	.	A	.	14
14	.	.	.	.	A	.	.	A	.	12
15	.	C	B	.	.	B	C	A	.	9

Table C–5: Multiple choice results: Incorrect answers for each student

C.3 Teachers' questionnaire

PART ONE

PERUSKOULU (9-16)

1. At what age are students first taught about the English article?

2. What rules of article usage are students taught initially? (If you can, give a list of rules, for example, *"The indefinite article is used for new information"*, or *"The zero article is usually used for mass and plural count nouns"*).

3. What are the common difficulties students have with articles at this stage? (Please give examples of typical errors).

LUKIO (16-18)

4. What aspects of article usage are generally taught at a later stage?
(Give details of rules introduced to the more advanced student, for example, *"The definite article is used before superlatives"*, or *"The indefinite article is used within expressions to do with rate, speed, quantity and time"*)

5. What particular difficulties do students have with articles at a more advanced stage? (Please give examples of typical errors).

BOTH

6. What do you find are generally successful strategies for correcting the errors which you have described above?

7. Are there any strategies for correcting errors which you have tried which seemed to be unsuccessful?

8. Are there any particular errors which you find are persistent and difficult to correct?

9. At what stage do you introduce the distinction between mass and count (uncountable and countable) nouns?

10. When teaching articles, do you make a distinction between reference to things in general (eg Dogs are very lovable animals) and reference to specific things (eg I have a very lovable dog), and if so, at what stage?

11. Do you ever teach articles by contrasting with the Finnish ways of showing indefiniteness and definiteness (eg word order, partitive case)?

12. Are there any particular aspects of teaching articles which you feel are not dealt with properly by the available text books?

PART TWO

Below are some examples of typical errors that may be made by students learning English. Please indicate (by ticking the appropriate column) whether a 13-year old, a 15-year old, or a 17-year old student would be likely to make these errors. Leave the column blank where you find that students in that age-group no longer make that type of error.

ERROR *(Corrected version in brackets)*	AGES OF STUDENTS *(Tick age-group(s) making error)* 13 15 17

The indefinite article:

1. a orange
(an orange)

2. a hour
(an hour)

3. an bicycle
(a bicycle)

4. My father is teacher
(My father is a teacher)

5. I'd like dozen apples, please.
(I'd like a dozen apples, please)

ERROR *(Corrected version in brackets)*	AGES OF STUDENTS *(Tick age-group(s) making error)* 13 15 17

The definite article:

6. Earth moves round the sun.
(The earth moves round the sun)

7. Helsinki is capital of Finland.
(Helsinki is the capital of Finland)

8. John is best player in the team.
(John is the best player in the team)

9. Please close window!
(Please close the window!)

10. Aeroplane has revolutionised travel.
(The aeroplane has revolutionised travel)

11. The President of United States of America is George Bush.
(The President of the United States of America is George Bush.)

12. French cook beautiful food.
(The French cook beautiful food)

13. Mr Smith spent the night at Grand Hotel.
(Mr Smith spent the night at the Grand Hotel)

ERROR *(Corrected version in brackets)*	AGES OF STUDENTS *(Tick age-group(s) making error)* 13 15 17

The zero article:

14. a bicycles
(bicycles)

15. a bread
(bread)

16. The sugar is bad for your teeth.
(Sugar is bad for your teeth)

17. The dogs are very lovable animals.
(Dogs are very lovable animals)

18. In summer, there are the flowers in the garden.
(In summer, there are flowers in the garden)

19. You can't buy the love.
(You can't buy love)

20. I went to Helsinki by the train.
(I went to Helsinki by train)

21. What did you do at the school today?
(What did you do at school today?)

Any other types of error not exemplified here?

PART THREE

As the questionnaire in its present form is only being sent to a limited number of teachers, it would be helpful to note any comments/criticisms you have so that it can be refined as necessary.

1. How long (roughly) has it taken you to complete this questionnaire?

2. Do you feel this is an excessive amount of time for such a questionnaire?

3. Is this subject area one you are interested in?

4. Are there any ways you feel this questionnaire could have been presented better (for example, making it completely multiple-choice)?

5. Any other comments:

Appendix D

Knowledge base for article usage

D.1 The article usage rules

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% THE RULES
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Name        : rule
%%% Arguments   : 1. Rule number
%%%               2. Noun phrase:
%%%                  (i)   Head noun
%%%                  (ii)  Proper noun
%%%                  (iii) Number
%%%                  (iv)  Count/mass
%%%                  (v)   Semantic
%%%               3. Article
%%%               4. Context of noun
%%%               5. If and how modified
%%% Called By   : apply_rules
%%% Description: Rules of English article usage
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 1
%
% If the noun is a singular count noun, 'a' is used in the absence of
% any other rules applying.

rule(1,[HN,neg,singular,count,Sem],ADet,_,Mod):-
    check_a_or_an(HN,Mod,ADet).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 2
%
% If the noun is a plural count noun, it is OK to use zero.

rule(2,[_,neg,plural,count,_],zero,_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 3
%
% If the noun is mass, it is OK to use zero.

rule(3,[_,neg,_,mass,_],zero,_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 4
%
% If the head noun has been mentioned before, use 'the'

rule(4,[HN,_,_,_,_],the,_,_):-
    in_context(HN).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 5
%
% If the head noun is modified with a pp beginning with prep 'of',
% use 'the'.

rule(5,_,the,_,Mod):-
   member([pp(sem:of),of],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 6
%
% If the noun is a special case (ie meals, time, seasons, institutions
% etc, then when this noun appears after certain prepositions,
% no article is used.

rule(6,[_,neg,singular,_,Semantic],zero,Context,Mod):-
  \+ member([a(sem:Sem),_],Mod),    % doesn't count if there's an
  member([p(sem:_),Prep],[Context]), % adjective
  exceptions(Semantic,Prep).
rule(6,[_,neg,singular,_,Semantic],zero,Context,Mod):-
  \+ member([a(sem:Sem),_],Mod),    % doesn't count if there's an
  member([pp(sem:_)|_],[Context]),  % adjective
  exceptions(Semantic,Prep).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 7
%
% if the noun is preceded by a plural quantity word
% such as 'hundred' or 'dozen', use 'a\an'

rule(7,[HN,neg,plural,count,quantity],Det,_,Mod):-
   \+ member([a(sem:number),_],Mod),    % no adjectives
   check_a_or_an(HN,Mod,Det).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 8
%
% If the article is preceded by a frequency word, use 'a'.
% eg 'once a day'

rule(8,[HN,neg,singular,count,time],Det,[adv(sem:frequency),Adv],Mod):-
        check_a_or_an(HN,Mod,Det).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 9
%
% In expressions of rates, eg five miles a day, use 'a'.


rule(9,[HN,neg,singular,count,time],Det,[n(_,_,_,_,sem:rate),_],Mod):-
          check_a_or_an(HN,Mod,Det).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 10
%
% If the noun is also classified as an adjective, use 'the'

rule(10,[HN,neg,plural,_,_],the,_,_):-
   word(a(_),HN,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 11
%
% If the head noun is a singular count noun complement of the verb 'be',
% use 'a'.

rule(11,[HN,neg,singular,count,_],Det,[bv(_,_,_),_],Mod):-
      check_a_or_an(HN,Mod,Det).
rule(11,[HN,neg,singular,count,_],Det,[_,[bv(_,_,_),_]],Mod):-
      check_a_or_an(HN,Mod,Det).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 12
%
%  Where a superlative adjective is used, use 'the'

rule(12,[HN,neg,_,_,_],the,_,Mod):-
   member([a(sem:sup),_],Mod).
rule(12,[HN,neg,_,_,sup],the,_,_).  % the oldest, the youngest

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 13
%
% No article used before a number.

rule(13,[_,neg,_,count,_],zero,_,Mod):-
   member([a(sem:number),Adj],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 14
%
% Use 'the' before a noun modified by a relative clause.

rule(14,_,the,_,Mod):-
   member([rel],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 15
%
% Use 'the' after 'all'.
% (all is the context)

rule(15,_,the,[X,all],_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 16
%
% Use 'the' before 'same'.

rule(16,_,the,_,Modified):-
   member([a(_),same],Modified).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 17
%
% Use 'the' before 'only'.

rule(17,_,the,_,Mod):-
   member([a(_),only],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 18
%
% Use 'the' before ordinals

rule(18,_,the,_,Mod):-
   member([a(sem:ord),_],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 19
%
% Use zero usually for a proper noun.

rule(19,[HN,pos,_,_,_],zero,_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 20
%
% Use 'the' where proper noun is also common noun

rule(20,[HN,pos,singular,_,_],the,_,_):-
   word(n(pn:neg,number:singular,_,_,_),HN,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 21
%
% Use 'the' where pn is plural name.

rule(21,[HN,pos,plural,_,_],the,_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 22
%
% Use 'the' where there is an indexical noun.

rule(22,[HN,neg,singular,_,indexical],the,_,_).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 23
%
% When noun is modified by an adjective, follow other rules, but 'the'
% is allowed.

rule(23,_,the,_,Mod):-
   member([a(sem:Sem),_],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 24
%
% certain singular count nouns can occur in subject position with no
% article in certain contexts (but can't specify that here)

rule(24,[HN,neg,singular,count,_],zero,subject,nil):-
   member(HN,[man,woman,age,family,cost,part,life,word]).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 25
%
% Use 'a' or 'the' after 'half' before a sing count noun
% Use the' after half in other cases
% (half is the context)
% eg half the bread, half the bottle, half a bottle

rule(25,[_,neg,singular,count,_],[a,the],[X,half],_):-!.
rule(25,_,the,[X,half],_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 26
%
% Use 'a/an'  after 'such' before a sing count noun  eg such an animal
% Use zero after 'such' in other cases   eg such beauty

rule(26,[HN,neg,singular,count,_],Det,Context,Mod):-!,
   list_member([adv(sem:_),such],Context),
   check_a_or_an(HN,Mod,Det).
rule(26,_,zero,Context,_):-
   list_member([adv(sem:_),such],Context).
% 'such' may be parsed as an adjective.
rule(26,[HN,neg,singular,count,_],Det,_,Mod):-!,
   member([adv(sem:_),such],Mod),
   check_a_or_an(HN,Mod,Det).
rule(26,_,zero,_,Mod):-
   member([adv(sem:_),such],Mod).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 27
%
% When noun is modified by the adjective 'certain', use 'a'.
% unless mass or plural noun, when use zero

rule(27,[_,_,singular,count,_],a,_,Mod):-
   member([a(sem:Sem),certain],Mod),!.
rule(27,[_,_,plural,_,_],zero,_,Mod):-
   member([a(sem:Sem),certain],Mod).
rule(27,[_,_,_,mass,_],zero,_,Mod):-
   member([a(sem:Sem),certain],Mod).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rule 28
%
% When noun is 'one', can be zero or the

rule(28,[one|_],[zero,the],_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Other predicates called by the rules
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: exceptions/2
% Description: Called by Rule 6 to decide if a noun
%              is of a particular semantic category, and
%              occurring with a particular preposition,
%              and thus idiomatically occurs with the
%              zero article.

exceptions(season,Prep):-
   member(Prep,[in,after,during]).
exceptions(institution,Prep):-
   member(Prep,[to,at,after,before]).
exceptions(time,Prep):-
   member(Prep,[by,at]).
exceptions(meal,Prep):-
   member(Prep,[over,after,at,during,without]).
exceptions(transport,Prep):-
   member(Prep,[by]).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: check_a_or_an/2
% Description: Decides if the indefinite article takes the
%              form 'a' or 'an' by looking at next word

check_a_or_an([CN|_],[],Det):-  % compound noun
   check_form(CN,Det).
check_a_or_an(_,Mod,Det):-
   member([a(sem:_),Adj],Mod),  % check what is first adjective
   check_form(Adj,Det).
check_a_or_an(HN,_,Det):-
   check_form(HN,Det).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: check_form/2
% Description: Decides if the indefinite article takes the
%              form 'a' or 'an' by looking at next letter

check_form(Var,[a,an]):-
   var(Var).
check_form(FirstWord,a):-
   name(FirstWord,[117,110,105|Rest]).  % word begins uni* a university
check_form(FirstWord,a):-
   name(FirstWord,[117,115,101|Rest]).  % word begins use* a use
check_form(FirstWord,a):-
   name(FirstWord,[117,116,105|Rest]).  % word begins uti* a utility
check_form(FirstWord,an):-
   name(FirstWord,[104,111,117,114]).   % word begins hour*  an hour
check_form(FirstWord,an):-
   name(FirstWord,[H|_]),               % vowels
   member(H,[97,101,105,111,117]),!.
check_form(_,a).                        % consonants

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name: in_context(Noun)
% Description: checks to see if the noun has been asserted
%              in the discourse history database

in_context(Noun):-
   word(_,Noun,Root),
   ( disc_hist(Root,Num,context),   % part of context
     Num>=0
   ; disc_hist(Root,Num,used),      % mentioned in previous sentence
     Num>=1
   ).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

D.2 Edges of the genetic graph

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%%      THE GENETIC GRAPH: EDGES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% The genetic graph consists of the article usage rules and the
%%% edges.  There are two types of links/edges:
%%%
%%% 1. The generalisation/specialisation link.
%%%    This link works both ways but only one direction is overtly
%%%    specified.
%%% 2. The priority link
%%%    This is a one-way link, indicating the priority of one rule
%%%    over another if both fire. The direction is:
%%%    edge(LessPriority,MorePriority,_).

%%% SPECIALISATION LINKS

edge(1,8,specialisation).
edge(1,9,specialisation).
edge(1,11,specialisation).
edge(1,25,specialisation).
edge(1,26,specialisation).
edge(1,27,specialisation).
edge(11,26,specialisation).
edge(2,13,specialisation).
edge(2,27,specialisation).
edge(3,26,specialisation).
edge(21,20,specialisation).
edge(23,16,specialisation).
edge(23,17,specialisation).
edge(23,18,specialisation).


%%% PRIORITY LINKS

edge(1,4,priority).
edge(1,5,priority).
edge(1,6,priority).
edge(1,12,priority).
edge(1,14,priority).
edge(1,15,priority).
```

```
edge(1,16,priority).
edge(1,17,priority).
edge(1,18,priority).
edge(1,19,priority).
edge(1,20,priority).
edge(1,22,priority).
edge(1,28,priority).
edge(2,4,priority).
edge(2,7,priority).
edge(2,10,priority).
edge(2,12,priority).
edge(2,15,priority).
edge(2,21,priority).
edge(2,25,priority).
edge(2,26,specialisation).
edge(2,27,specialisation).
edge(3,14,priority).
edge(3,15,priority).
edge(3,25,priority).
edge(3,27,specialisation).
edge(4,7,priority).
edge(4,8,priority).
edge(4,9,priority).
edge(4,19,priority).
edge(4,15,priority).
edge(4,16,priority).
edge(4,17,priority).
edge(4,6,priority).
edge(4,18,priority).
edge(4,12,priority).
edge(4,13,priority).
edge(4,26,priority).
edge(4,27,priority).
edge(6,16,priority).
edge(6,12,priority).
edge(11,5,priority).
edge(11,12,priority).
edge(11,15,priority).
edge(11,14,priority).
edge(11,16,priority).
edge(11,17,priority).
edge(11,18,priority).
edge(14,7,priority).
edge(14,8,priority).
edge(14,9,priority).
edge(14,27,priority).
edge(19,20,priority).
```

```
edge(19,21,priority).
edge(23,11,priority).
edge(23,12,priority).
edge(23,13,priority).
edge(23,26,priority).
edge(23,27,priority).
edge(24,14,priority).
edge(24,4,priority).
edge(24,23,priority).
edge(28,16,priority).
edge(28,17,priority).
edge(28,18,priority).
edge(28,12,priority).


%%% ADDITIONAL LINKS FOR WRITE MODE

%%% ADDITIONAL SPECIALISATION LINKS

edge(1,4,specialisation).
edge(1,12,specialisation).
edge(1,15,specialisation).
edge(1,16,specialisation).
edge(1,17,specialisation).
edge(1,18,specialisation).
edge(1,20,specialisation).
edge(1,22,specialisation).
edge(1,23,specialisation).
edge(1,28,specialisation).

%%% ADDITIONAL PRIORITY LINKS

edge(1,8,priority).
edge(1,9,priority).
edge(1,11,priority).
edge(1,24,priority).
edge(1,25,priority).
edge(1,26,priority).
```

Appendix E

Natural language processing

E.1 The grammar

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%                THE  GRAMMAR
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Top level sentence

s  --->  [np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed)].
s  --->  [advp,np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed)].
s  --->  [pp(sem:Form),np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed)].
s  --->  [rel(number:N,sex:S),vp(person:P,number:N,sex:S,
          verb_form:tensed)].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NOUNS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% np -> det np pp

np(person:3,number:N,sex:S,case:_)  --->
[det,nb(number:N,sex:S),pp(sem:_)].
% np -> nb pp
np(person:3,number:N,sex:S,case:_)  --->
       [nb(number:N,sex:S),pp(sem:_)].
% np -> det nb
np(person:3,number:N,sex:S,case:_)  --->
       [det,nb(number:N,sex:S)].
% np -> nb
np(person:3,number:N,sex:S,case:_)  --->
       [nb(number:N,sex:S)].
```

```
% np -> det ap nb
np(person:3,number:N,sex:S,case:_)  --->
        [det,ap(sem:_),nb(number:N,sex:S)].
% np ->ap nb
np(person:3,number:N,sex:S,case:_)  --->
[ap(sem:_),nb(number:N,sex:S)].
% np ->ap(such) det  nb
np(person:3,number:N,sex:S,case:_)  --->
        [ap(sem:such),det,nb(number:N,sex:S)].
%
% A rule for 'all the...'
%
np(person:P,number:N,sex:_,case:Case)  --->
         [qdet,np(person:P,number:N,sex:_,case:Case)].
%
% The partitive rule,eg 'some of the...'
%
np(person:P,number:N,sex:_,case:Case)  --->
         [qdet,pp(sem:of),np(person:P,number:N,sex:_,case:Case)].
%
% Possessives and Compound Nouns
%
% (Flat structure to avoid left recursion)
%
% np -> det n(poss) np
np(person:3,number:N,sex:S,case:Case)  --->
         [det,n(pn:_,number:N1,sex:_,type:_,sem:_),poss,
          np(person:3,number:N,sex:S,case:Case)].
% np -> n(poss) np
np(person:3,number:N,sex:S,case:Case)  --->
[n(pn:_,number:N1,sex:_,type:_,sem:_),poss,
         np(person:3,number:N,sex:S,case:Case)].
% no following noun:  eg. this mood of Lee's
% np -> det n(poss)
np(person:3,number:N,sex:S,case:_)  --->
         [det,n(pn:_,number:N1,sex:_,type:_,sem:_),poss].
% np -> n(poss)
np(person:3,number:N,sex:S,case:_)  --->
         [n(pn:_,number:N1,sex:_,type:_,sem:_),poss].
%
% possessives with adjectives
%
% eg The wheel on the fat man's big car fell off
% np -> det ap n(poss) np
np(person:3,number:N,sex:S,case:Case)  --->
         [det,ap(sem:_),n(pn:_,number:N1,sex:_,type:_,sem:_),poss,
          np(person:3,number:N,sex:S,case:Case)].
```

```
% np -> ap n(poss) np
np(person:3,number:N,sex:S,case:Case)  --->
          [ap(sem:_),n(pn:_,number:N1,sex:_,type:_,sem:_),poss,
           np(person:3,number:N,sex:S,case:Case)].
% np -> det ap n(poss)
np(person:3,number:N,sex:S,case:Case)  --->
         [det,ap(sem:_),
          n(pn:_,number:N1,sex:_,type:_,sem:_),poss].
% np -> ap n(poss)
np(person:3,number:N,sex:S,case:_)  --->
         [ap(sem:_),n(pn:_,number:N1,sex:_,type:_,sem:_),poss].
%
% compound nouns
%
% np -> det n np
np(person:3,number:N,sex:S,case:Case)  --->
         [det,n(pn:_,number:_,sex:_,type:_,sem:_),
          np(person:3,number:N,sex:S,case:Case)].
% np -> n np
np(person:3,number:N,sex:S,case:Case)  --->
         [n(pn:_,number:_,sex:_,type:_,sem:_),
          np(person:3,number:N,sex:S,case:Case)].
% np -> det n np
np(person:3,number:N,sex:S,case:Case)  --->
         [det,ap(sem:_),
          n(pn:_,number:_,sex:_,type:_,sem:_),
          np(person:3,number:N,sex:S,case:Case)].
% np -> ap n np
np(person:3,number:N,sex:S,case:Case)  --->
         [ap(sem:_),n(pn:_,number:_,sex:_,type:_,sem:_),
          np(person:3,number:N,sex:S,case:Case)].
%
% nbar
%
% nb -> ap n
nb(number:N,sex:S)  --->
         [ap(sem:_),n(pn:_,number:N,sex:S,type:_,sem:_)].

% titles
nb(number:N,sex:S)  --->
         [n(pn:pos,number:N,sex:S,type:_,sem:_),
          n(pn:pos,number:N,sex:S,type:_,sem:_)].
nb(number:N,sex:S)  --->
         [titlep(number:N),n(pn:pos,number:_,sex:S,type:_,sem:_),
          n(pn:pos,number:_,sex:S,type:_,sem:_)].
nb(number:N,sex:S)  --->
         [titlep(number:N),n(pn:pos,number:_,sex:S,type:_,sem:_)].
```

```
nb(number:N,sex:S)  --->
[n(pn:_,number:N,sex:S,type:_,sem:_)].
nb(number:N,sex:S)  --->
[n(pn:_,number:N,sex:S,type:_,sem:_),
rel(number:N,sex:S)].                    % relative clause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% VERBS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% vp -> vb
vp(person:P,number:N,sex:_,verb_form:V)  --->
        [vb(person:P,number:N,verb_form:V)].
% vp -> vb advp
vp(person:P,number:N,sex:_,verb_form:V)  --->
        [vb(person:P,number:N,verb_form:V),advp].
% vp -> vb advp np
vp(person:P,number:N,sex:_,verb_form:V)  --->
        [vb(person:P,number:N,verb_form:V),advp,
         np(person:_,number:_,sex:_,case:object)].
% vp -> advp vp
vp(person:P,number:N,sex:_,verb_form:V)  --->
        [advp,vp(person:P,number:N,sex:_,verb_form:V)].
% vp -> vb vp(inf)
vp(person:P,number:N,sex:_,verb_form:V)  --->
        [vb(person:P,number:N,verb_form:V),
         vp(person:P,number:N,sex:_,verb_form:infinitive)].
% vp -> vb vp(part)
vp(person:P,number:N,sex:_,verb_form:V)  --->
         [vb(person:P,number:N,verb_form:V),
          vp(person:P,number:N,sex:_,verb_form:part)].
%  vp -> vb pp
vp(person:P,number:N,sex:_,verb_form:V)  --->
         [vb(person:P,number:N,verb_form:V),pp(sem:_)].
%  vp -> vb advp pp
vp(person:P,number:N,sex:_,verb_form:V)  --->
         [vb(person:P,number:N,verb_form:V),advp,pp(sem:_)].
% vp -> vb pp advp
vp(person:P,number:N,sex:_,verb_form:V)  --->
         [vb(person:P,number:N,verb_form:V),pp(sem:_),advp].


%
% vbar
%
% vb -> v
vb(person:P,number:N,verb_form:V)  --->
         [v(person:P,number:N,verb_form:V)].
```

```
% vb -> aux vb
vb(person:P,number:N,verb_form:V)  --->
        [aux(person:P,number:N,verb_form:V),
         vb(person:P,number:N,verb_form:part)].
% vb -> mv vb(inf)                    % must see
vb(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),
         vb(person:P,number:N,verb_form:infinitive)].
% vb -> dv vb(inf)                    % does see
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),
         vb(person:P,number:N,verb_form:infinitive)].
% vb -> mv neg vb(inf)
vb(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),neg,
         vb(person:P,number:N,verb_form:infinitive)].
% vb -> dv neg vb(inf)              % does not see
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),neg,
         vb(person:P,number:N,verb_form:infinitive)].
% vb -> bv
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V)].
% vb -> hv
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V)].
% vb -> dv
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V)].
% vb -> bv neg
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),neg].
% vb -> hv
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),neg].
% vb -> dv
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),neg].
% vb -> v np
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object)].
% vb -> bv np
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object)].
% vb -> hv np
```

```
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object)].
% vb -> dv np
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object)].
% vb -> v advp np
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),advp,
         np(person:_,number:_,sex:_,case:object)].
% vb -> bv advp np
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),advp,
         np(person:P,number:N,sex:_,case:object)].
% vb -> hv advp np
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),advp,
         np(person:_,number:_,sex:_,case:object)].
% vb -> dv advp np
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),advp,
         np(person:_,number:_,sex:_,case:object)].
% vb -> v pp
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         pp(sem:_)].
% vb -> bv pp
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),
         pp(sem:_)].
% vb -> hv pp
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),
         pp(sem:_)].
% vb -> dv pp
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),
         pp(sem:_)].
% vb -> bv ap(sem:_)
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),
         ap(sem:_)].
% vb -> bv ap pp
vb(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),
         ap(sem:_),pp(sem:_)].
```

```
% vb -> v ap
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         ap(sem:_)].
% vb -> v ap pp
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         ap(sem:_),pp(sem:_)].
% vb -> v np vp        % she inspires her student to write
vb(person:P1,number:N1,verb_form:V1)  --->
        [v(person:P1,number:N1,verb_form:V1),
         np(person:P2,number:N2,sex:S2,case:object),
         vp(person:P2,number:N2,sex:S2,verb_form:infinitive)].
% vb -> v np pp      % she puts the dog in the kennel
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object),pp(sem:_)].
% vb -> v np np      % she gives  the dog a bone
vb(person:P,number:N,verb_form:V)  --->
        [v(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object),
         np(person:_,number:_,sex:_,case:object)].
% vb -> hv np pp     % she has a tree in the garden
vb(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object),
         pp(sem:_)].
% vb -> dv np pp   % she does her homework in her bedroom
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),
         np(person:_,number:_,sex:_,case:object),pp(sem:_)].
%  vb(inf) -> to vb(inf)
vp(person:P,number:N,verb_form:infinitive) --->
        [to,vb(person:P,number:N,verb_form:infinitive)].
% vb -> mv vb(inf) pp
vb(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),
         vb(person:P,number:N,verb_form:infinitive),pp(sem:_)].
% vb -> dv vb(inf) pp
vb(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),
         vb(person:P,number:N,verb_form:infinitive),pp(sem:_)].
% vb -> mv vb(inf) pp
vp(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),neg,
         vb(person:P,number:N,verb_form:infinitive),pp(sem:_)].
% vb -> dv vb(inf) pp
```

```
vp(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),neg,
         vb(person:P,number:N,verb_form:infinitive),pp(sem:_)].

%
%  Auxiliary category for auxiliaries and modal verbs
%
% aux -> bv                                 % she is closing
aux(person:P,number:N,verb_form:V)  --->
         [bv(person:P,number:N,verb_form:V)].
% aux -> hv                                 % she has closed
aux(person:P,number:N,verb_form:V)  --->
         [hv(person:P,number:N,verb_form:V)].
% aux -> dv                                 % she does close
aux(person:P,number:N,verb_form:V)  --->
         [dv(person:P,number:N,verb_form:V)].
% aux -> mv                                 % she would close
aux(person:P,number:N,verb_form:V)  --->
         [mv(person:P,number:N,verb_form:V)].
% aux -> hv bv
aux(person:P,number:N,verb_form:V)  --->        % has been closing
         [hv(person:P,number:N,verb_form:V),
         bv(person:_,number:_,verb_form:part)].
% aux -> mv bv
aux(person:P,number:N,verb_form:V)  --->        % would be closing
         [mv(person:P,number:N,verb_form:V),
          bv(person:_,number:_,verb_form:infinitive)].
% aux -> mv hv
aux(person:P,number:N,verb_form:V)  --->        % would have closed
         [mv(person:P,number:N,verb_form:V),
          hv(person:_,number:_,verb_form:infinitive)].
% aux -> mv hv bv
aux(person:P,number:N,verb_form:V)  --->        % would have been closing
         [mv(person:P,number:N,verb_form:V),
          hv(person:_,number:_,verb_form:infinitive),
          bv(person:_,number:_,verb_form:part)].
%
%  Negative sentences
%
%
% aux -> bv neg                       % she is not closing
aux(person:P,number:N,verb_form:V)  --->
         [bv(person:P,number:N,verb_form:V),neg].
% aux -> hv neg                                 % she has not closed
aux(person:P,number:N,verb_form:V)  --->
         [hv(person:P,number:N,verb_form:V),neg].
% aux -> dv neg                                 % she does not close
```

```
aux(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),neg].
% aux -> mv neg                               % she would not close
aux(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),neg].
% aux -> hv neg bv
aux(person:P,number:N,verb_form:V)  --->        % has not been closing
        [hv(person:P,number:N,verb_form:V),neg,
         bv(person:_,number:_,verb_form:part)].
% aux -> mv neg bv
aux(person:P,number:N,verb_form:V)  --->            % would not be closing
        [mv(person:P,number:N,verb_form:V),neg,
         bv(person:_,number:_,verb_form:infinitive)].
% aux -> mv neg hv
aux(person:P,number:N,verb_form:V)  --->  % would not have closed
        [mv(person:P,number:N,verb_form:V),neg,
         hv(person:_,number:_,verb_form:infinitive)].
% aux -> mv neg hv bv
aux(person:P,number:N,verb_form:V)  --->  % would not have been closing
        [mv(person:P,number:N,verb_form:V),neg,
         hv(person:_,number:_,verb_form:infinitive),
         bv(person:_,number:_,verb_form:part)].


%
%  Adverbial phrases in auxiliaries
%
% aux -> bv advp                              % she is quietly closing
aux(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),advp].
% aux -> hv advp                              % she has quietly closed
aux(person:P,number:N,verb_form:V)  --->
        [hv(person:P,number:N,verb_form:V),advp].
% aux -> dv advp                              % she does quietly close
aux(person:P,number:N,verb_form:V)  --->
        [dv(person:P,number:N,verb_form:V),advp].
% aux -> mv advp                              % she would quietly close
aux(person:P,number:N,verb_form:V)  --->
        [mv(person:P,number:N,verb_form:V),advp].
% aux -> hv advp bv
aux(person:P,number:N,verb_form:V)  --->        % has quietly been closing
        [hv(person:P,number:N,verb_form:V),advp,
         bv(person:_,number:_,verb_form:part)].
% aux -> mv advp bv
aux(person:P,number:N,verb_form:V)  --->       % would quietly be closing
        [mv(person:P,number:N,verb_form:V),advp,
         bv(person:_,number:_,verb_form:infinitive)].
% aux -> mv advp hv
```

```
aux(person:P,number:N,verb_form:V)  --->  % would quietly have closed
        [mv(person:P,number:N,verb_form:V),advp,
         hv(person:_,number:_,verb_form:infinitive)].
% aux -> mv advp hv bv
aux(person:P,number:N,verb_form:V)  --->% would quietly have been closing
        [mv(person:P,number:N,verb_form:V),advp,
         hv(person:_,number:_,verb_form:infinitive),
         bv(person:_,number:_,verb_form:part)].

%
%  Negatives and adverbials in the auxiliary
%
% aux -> bv neg advp                  % she isnot quietly closing
aux(person:P,number:N,verb_form:V)  --->
        [bv(person:P,number:N,verb_form:V),neg,advp].
% aux -> hv neg advp                  % she has not quietly closed
aux(person:P,number:N,verb_form:V)  --->
         [hv(person:P,number:N,verb_form:V),neg,advp].
% aux -> dv neg advp                  % she does not quietly close
aux(person:P,number:N,verb_form:V)  --->
         [dv(person:P,number:N,verb_form:V),neg,advp].
% aux -> mv neg advp                  % she would not quietly close
aux(person:P,number:N,verb_form:V)  --->
         [mv(person:P,number:N,verb_form:V),neg,advp].
% aux -> hv neg advp bv
aux(person:P,number:N,verb_form:V)  --->   has not quietly been closing
         [hv(person:P,number:N,verb_form:V),neg,advp,
          bv(person:_,number:_,verb_form:part)].
% aux -> mv neg advp bv
aux(person:P,number:N,verb_form:V)  --->  % would not quietly be closing
         [mv(person:P,number:N,verb_form:V),neg,advp,
          bv(person:_,number:_,verb_form:infinitive)].
% aux -> mv neg advp hv
aux(person:P,number:N,verb_form:V)  --->  % would not quietly have closed
         [mv(person:P,number:N,verb_form:V),neg,advp,
          hv(person:_,number:_,verb_form:infinitive)].
% aux -> mv neg advp hv bv      % would not suddenly have been closing
aux(person:P,number:N,verb_form:V)  --->
         [mv(person:P,number:N,verb_form:V),neg,advp,
          hv(person:_,number:_,verb_form:infinitive),
          bv(person:_,number:_,verb_form:part)].
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The following rules handle unbounded dependencies in NP topicalization
% and relative clauses:
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% s -> np vp rel
s  ---> [np(person:P,number:N,sex:S,case:C),
         vp(person:P,number:N,sex:S,verb_form:tensed),
         rel(number:N,sex:S)].
% s -> advp np vp rel
% first she said that only the most expensive seats were left.
s  ---> [advp,np(person:P,number:N,sex:S,case:C),
         vp(person:P,number:N,sex:S,verb_form:tensed),
         rel(number:N,sex:S)].
%
%  relative clauses
%
% rel -> wh s (slash np)
rel(number:N,sex:S)  --->
[wh(sex:S),
         s(slash:np(person:_,number:N,sex:S,case:_))].
% rel -> s (slash np)
rel(number:N,sex:S)  --->
        [s(slash:np(person:_,number:N,sex:S,case:_))].
% rel ->  wh vp
% the man who was sitting on her doorstep was John
rel(number:N,sex:S)  --->
        [wh(sex:S),
         vp(person:_,number:N,sex:S,verb_form:tensed)].

%
% s (slash: XP)
%
% s(slash:XP) -> np vp(slash:XP)
s(slash:XP)  --->
        [np(person:P,number:N,sex:S,case:subject),
         vp(person:P,number:N,sex:S,verb_form:tensed,slash:XP)].
% vp (slash:XP) -> bv
vp(person:P,number:N,sex:S,verb_form:V,slash:np(person:_,
        number:N,sex:S,case:object))  --->
        [bv(person:P,number:N,verb_form:V)].
% vp (slash:XP) -> vb
vp(person:P,number:N,sex:S,verb_form:V,slash:np(person:_,
          number:_,sex:S,case:object))  --->
         [vb(person:P,number:N,verb_form:V)].
```

```
% vp (slash:XP) -> vb vp(inf)
vp(person:P1,number:N1,sex:S1,verb_form:V1,slash:np(person:P2,
         number:N2,sex:S2,case:object))  --->
        [vb(person:P1,number:N1,verb_form:V1),
         vp(person:P2,number:N2,sex:S2,verb_form:infinitive)].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PREPOSITIONAL PHRASES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pp -> pb
pp(sem:Form) ---> [pb(sem:Form)].
% pp -> pb conjp pb
pp(sem:_) ---> [pb(sem:_),conjp,pb(sem:_)].

% pb -> p np
pb(sem:Form) ---> [p(sem:Form),np(person:_,number:_,sex:_,case:object)].
% pb -> p
pb(sem:Form) ---> [p(sem:Form)].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ADJECTIVAL AND ADVERBIAL PHRASES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ap -> ab
ap(sem:Sem) ---> [ab(sem:Sem)].
% ab -> adet ab
ab(sem:Sem) ---> [adet(sem:Sem),ab(sem:Sem)].
% ab -> a
ab(sem:Sem) ---> [a(sem:Sem)].         % lazy dog
% ab -> adv ab
ab(sem:Sem) ---> [adv(sem:_),ab(sem:Sem)].   % very lazy dog
% ab -> a ab
ab(sem:Sem) ---> [a(sem:_),ab(sem:Sem)].     % big bad lazy dog

% advp -> adv
advp ---> [adv(sem:_)].
% advp -> adv np
advp ---> [adv(sem:frequency),np(_,_,_,case:object)].
% advp -> adv conjp pp
advp ---> [adv(sem:_),conjp,pp(sem:_)].    % happily and in a great
                                           % mood, he went to work
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   CONJUNCTIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

conjp ---> [conj].
conjp ---> [conj,neg].

% use of conjunctions
%
% s -> np vp conjp vp       John went outside and started the car
s  --->  [np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed), conjp,
          vp(person:_,number:_,sex:S,verb_form:_)].
% s -> np vp conjp s    John went outside and Paul started the car
s  --->  [np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed), conjp,s].
% s -> advp np vp conjp  vp
s  --->  [advp,np(person:P,number:N,sex:S,case:subject),
          vp(person:P,number:N,sex:S,verb_form:tensed),conjp,
          vp(person:_,number:_,sex:S,verb_form:_)].
% s -> pp np vp conjp vp
s  ---> [pp(sem:Form),np(person:P,number:N,sex:S,case:subject),
         vp(person:P,number:N,sex:S,verb_form:tensed),conjp,
         vp(person:_,number:_,sex:S,verb_form:_)].
% s -> advp np vp conjp s
s  ---> [advp,np(person:P,number:N,sex:S,case:subject),
         vp(person:P,number:N,sex:S,verb_form:tensed),conjp,s].
% s -> pp np vp conjp s
s  ---> [pp(sem:Form),np(person:P,number:N,sex:S,case:subject),
         vp(person:P,number:N,sex:S,verb_form:tensed),conjp,s].
% np -> np conjp np           John and Carol went outside
np(person:P,number:plural,sex:S,case:Case) --->
        [np(person:P,number:_,sex:_,case:Case),conjp,
         np(person:_,number:_,sex:_,case:Case)].
% use flat-ish structure as for possessives
% np -> np vp pp conjp np
np(person:3,number:N,sex:S,case:Case)  --->
        [det,nb(number:N,sex:S),
         pp(sem:_),conjp,np(person:P2,number:N2,sex:S2,case:Case)].
% np -> nb pp conjp np
np(person:3,number:N,sex:S,case:Case)  --->
        [nb(number:N,sex:S),pp(sem:_),conjp,
         np(person:P2,number:N2,sex:S2,case:Case)].
% np -> det np conjp np
np(person:3,number:N,sex:S,case:Case)  --->
        [det,nb(number:N,sex:S),conjp,
```

```
         np(person:P2,number:N2,sex:S2,case:Case)].
% np -> nb conjp np
np(person:3,number:N,sex:S,case:Case)  --->
        [nb(number:N,sex:S),conjp,
         np(person:P2,number:N2,sex:S2,case:Case)].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TITLES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

titlep(number:plural)--->[title,conjp,title].
titlep(number:singular)--->[title].


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of grammar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

E.2 Extract from the lexicon

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% EXTRACT FROM THE LEXICON
%%

%% Nouns

n(pn:neg,number:singular,sex:_460894,type:count,sem:transport)--->[bus].
n(pn:neg,number:singular,sex:_460867,type:count,sem:(transport,
  [engine,wheel,door,vehicle,garage,automobile,steering,key,driver]))--->
n(pn:neg,number:singular,sex:_460873,type:count,sem:nil)--->[family].
n(pn:neg,number:singular,sex:_460894,type:count,sem:nil)--->[swim].

%% Proper nouns
n(pn:pos,number:singular,sex:_460873,type:count,sem:(city,
  [city,population,people,mayor,council]))--->[edinburgh].
n(pn:pos,number:singular,sex:male,type:count,sem:nil)--->[gary].

%% Pronouns

np(person:3,number:plural,sex:_460894,case:subject)--->[she].
np(person:3,number:plural,sex:_460894,case:object)--->[them].

%% Verbs

bv(person:3,number:singular,verb_form:tensed)--->[is].
bv(person:_460836,number:_460856,verb_form:infinitive)--->[be].
dv(person:_460842,number:_460862,verb_form:infinitive)--->[do].
dv(person:2,number:singular,verb_form:tensed)--->[do].
dv(person:1,number:singular,verb_form:tensed)--->[do].
dv(person:_460842,number:plural,verb_form:tensed)--->[do].
hv(person:_460842,number:_460862,verb_form:infinitive)--->[have].
mv(person:_460849,number:_460869,verb_form:tensed)--->[might].
v(person:_460863,number:_460883,verb_form:infinitive)--->[swim].
v(person:1,number:singular,verb_form:tensed)--->[swim].
v(person:2,number:singular,verb_form:tensed)--->[swim].
v(person:_460863,number:plural,verb_form:tensed)--->[swim].
v(person:_460863,number:_460883,verb_form:infinitive)--->[want].
v(person:2,number:singular,verb_form:tensed)--->[want].
v(person:1,number:singular,verb_form:tensed)--->[want].
v(person:_460863,number:plural,verb_form:tensed)--->[want].
```

```
%% Prepositions

p(sem:nil)--->[in].
p(sem:nil)--->[of].

%%
%% Adjectives
%%

a(sem:nil)--->[fast].
a(sem:nil)--->[happy].

%%
%% Adverbs
%%
adv(sem:nil)--->[better].

%%
%% Determiners
%%
det --->[a].
det --->[his].
det --->[the].

%%
%% Titles
%%
title ---> [mr].
```

Appendix F

A worked example

This appendix shows how a sentence of English is processed by the various stages of the system, and the results generated.

This example shows a student from Level 2, whose learning preference is for **examples**, who inputs a sentence containing one article usage error.

Stage 1: Student input

Gary is fastest swimmer in his family.

Stage 2: Parse found

In this case, there were 2 different parses found, corresponding to the attachment of the prepositional phrase at different levels. This did not affect the system's decision about article usage, so only one of the two parses is shown below.

```
[s,
  [np(person:3,number:singular,sex:_5529,case:subject),
    [nb(number:singular,sex:_5529),
      [n(pn:pos,number:singular,sex:_5529,type:count,sem:nil),gary]
    ]
  ],
  [vp(person:3,number:singular,sex:_5529,verb_form:tensed),
    [vb(person:3,number:singular,verb_form:tensed),
      [bv(person:3,number:singular,verb_form:tensed),is],
      [np(person:3,number:singular,sex:_5647,case:object),
        [nb(number:singular,sex:_5647),
          [ap(sem:sup),
            [ab(sem:sup),
              [a(sem:sup),fastest]
            ]
          ],
```

```
          [n(pn:neg,number:singular,sex:_5647,type:count,sem:nil),swimmer]
        ]
      ]
    ],
    [pp(sem:nil),
      [pb(sem:nil),
        [p(sem:nil),in],
        [np(person:3,number:singular,sex:_5766,case:object),
          [det,his],
          [nb(number:singular,sex:_5766),
            [n(pn:neg,number:singular,sex:_5766,type:count,sem:nil),family]
          ]
        ]
      ]
    ]
  ]
]
```

Stage 3: Noun phrases extracted

There are three noun phrases extracted from this sentence. The first two noun phrases are preceded by the zero article. The third noun is preceded by the determiner *his*. Determiners which are personal pronouns and demonstrative pronouns are not dealt with by *ArtCheck*, so the third noun phrase will not be included in the article checking. It will, however, be added to the discourse model, so it can be recognised as given information if used again.

The following table shows the information extracted from the parse.

Det	HN	PN?	Number	Count	Env	Modifier	Semantic cat.
zero	*Gary*	yes	singular	count	subject	nil	nil
zero	*swimmer*	no	singular	count	*is*	*fastest*	nil
other_det	family	no	singular	count	prep *in*	nil	nil

Stage 4: Results of article checking

The article usage rules are then applied. Rule 11 is a more specialised version of Rule 1, and Rule 12 has priority over Rule 11. The following table shows the results of the application of the rules:

Correct/ Error	Noun details	Det. used	Rules firing	Rules chosen	Det. chosen
Correct	*Gary*, proper noun, etc.	zero	19	19	zero
Error	*swimmer*, modified, etc.	zero	1,11,12	12	the

Stage 5: Student model updated

At this stage, the student model is updated with any rules and edges the student appears to have used or acquired. In addition, the correct and incorrect noun phrases are also retained in the student model.

In this case, the rule:

```
user.rule(19,1)
```

is added to the student model, indicating that Rule 19 has been used once correctly.

Stage 6: Explanation generated

The explanation produced is Type 5 (see Appendix G), which is used for a student of Level 2, whose learning preference is for examples, where the error is caused by the student not knowing the rule. The explanation generated is given below:

```
<fastest swimmer> in
<Gary is fastest swimmer in his family> is incorrect.

It should be: <the fastest swimmer>.

      Select:
             m   more
             q   quit explanation

      >>>> m

An example is:

  ***  John is the nicest teacher.

      Select:
             m   more
             q   quit explanation

      >>>> m

The rule is:

 >>>  Use <the> before adjectives like <most> and <best>.
```

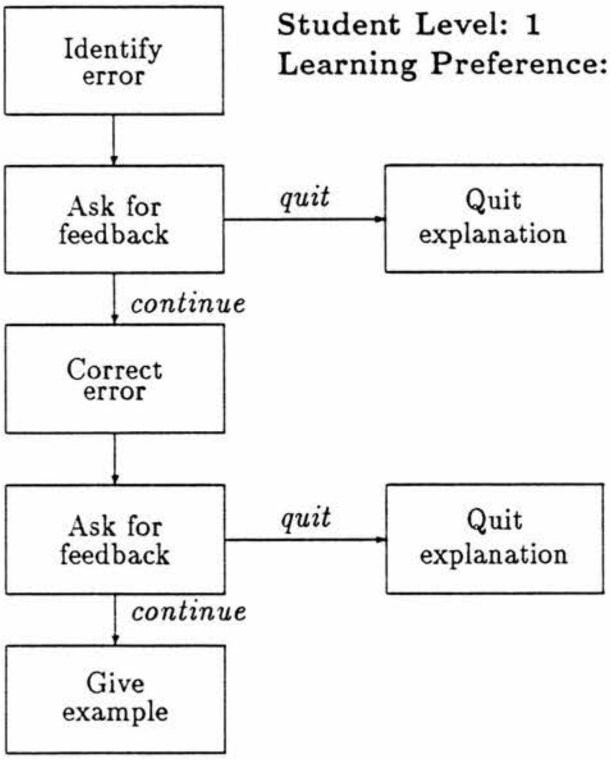
Appendix G

Structure of Explanations

There are 17 different types of explanations, as given in Table 7.3.1. This appendix shows the structure of each type of explanation, and the discourse goals which may be realised as the explanation progresses.

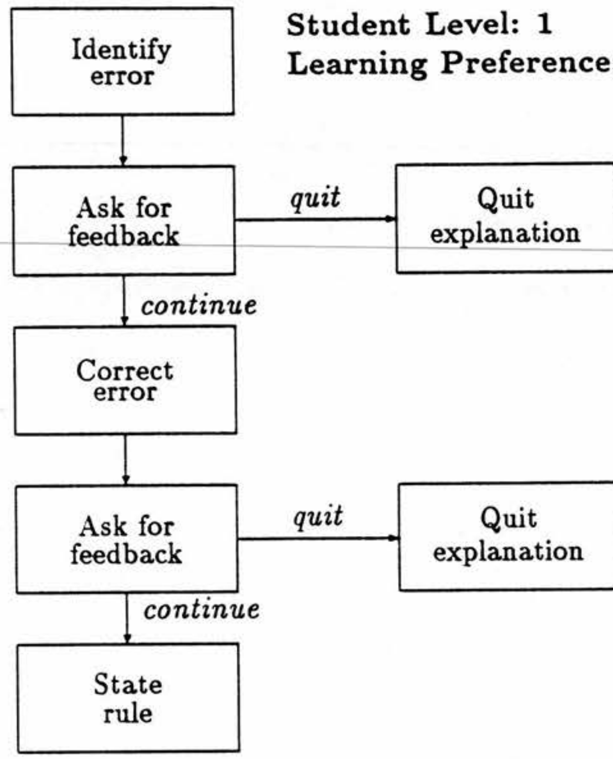
Explanation type 1

Student Level: 1
Learning Preference: Examples



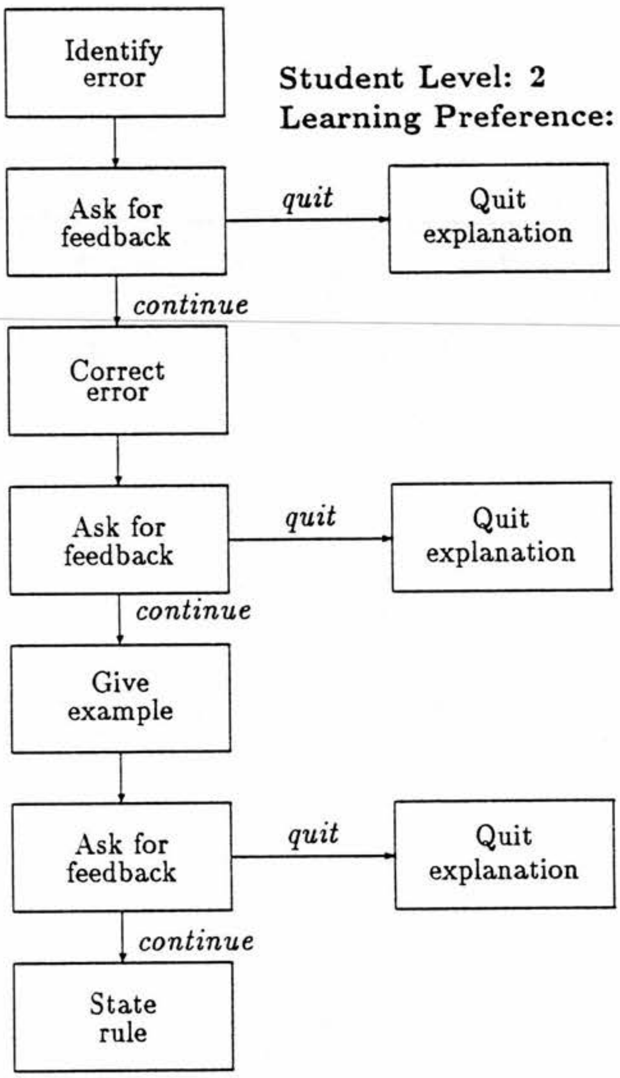
Explanation type 2

Student Level: 1
Learning Preference: Rules



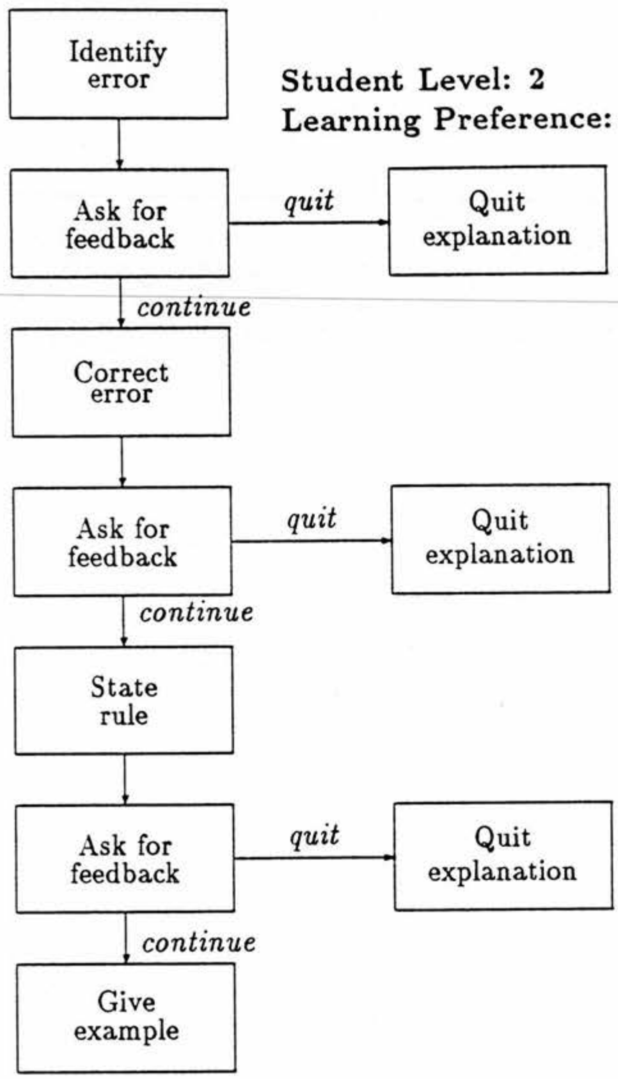
Explanation type 3

Student Level: 2
Learning Preference: Examples



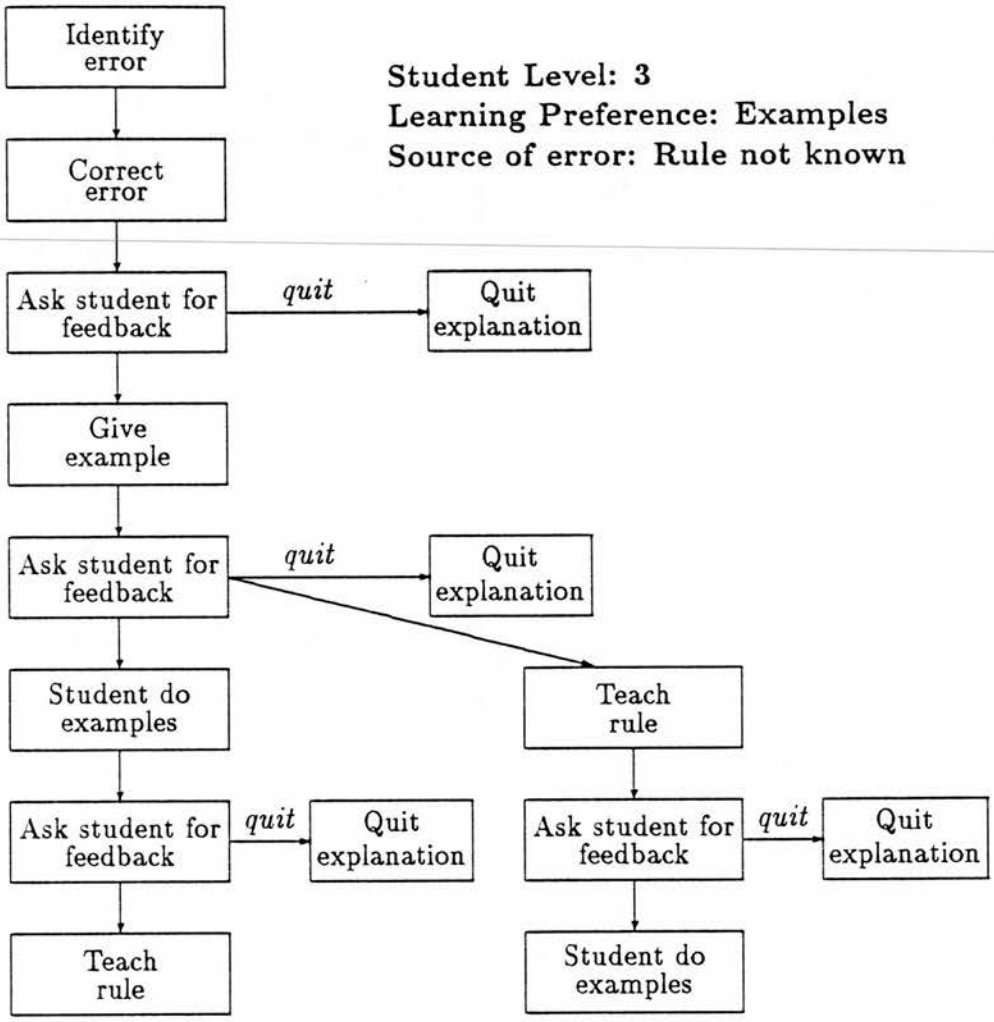
Explanation type 4

Student Level: 2
Learning Preference: Rules



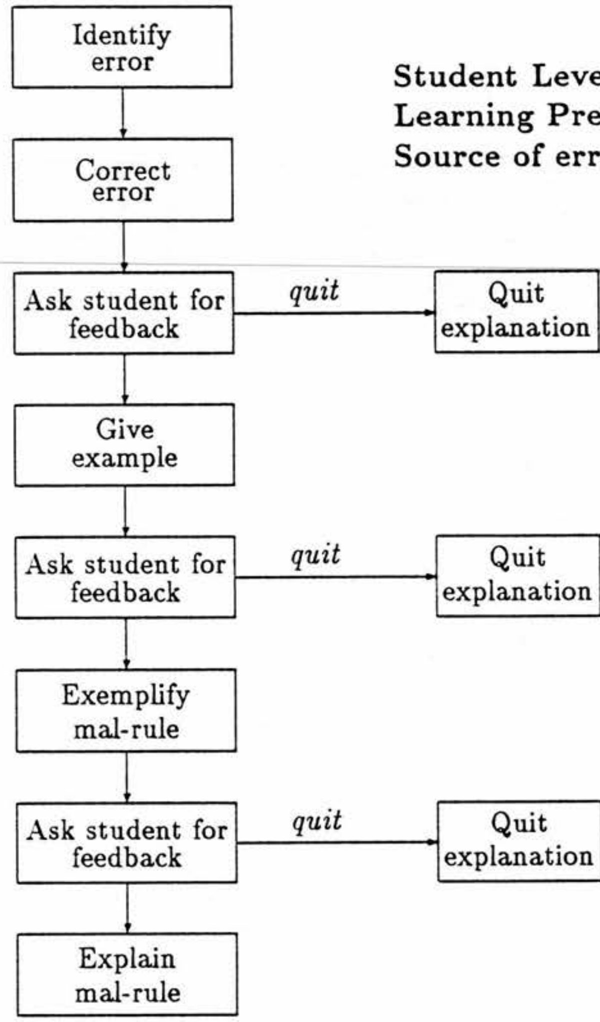
Explanation type 5

Student Level: 3
Learning Preference: Examples
Source of error: Rule not known

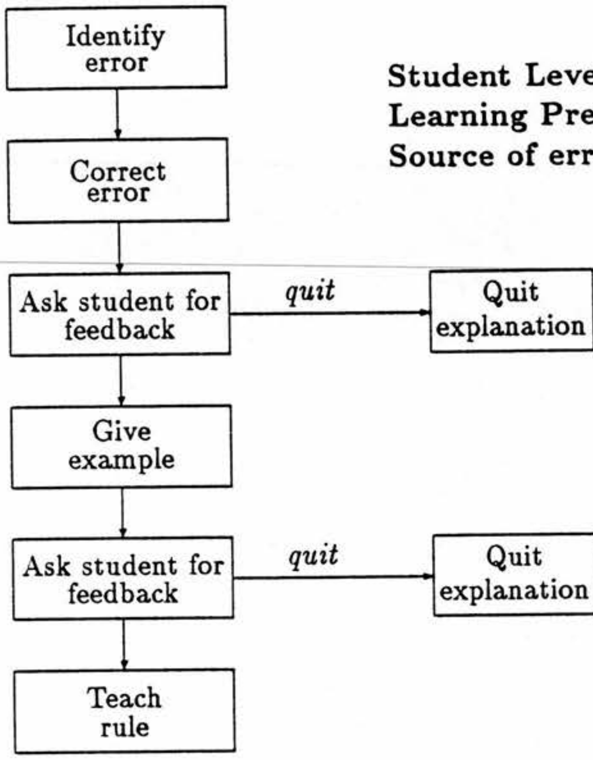


Explanation type 6

Student Level: 3
Learning Preference: Examples
Source of error: Student has mal-rule

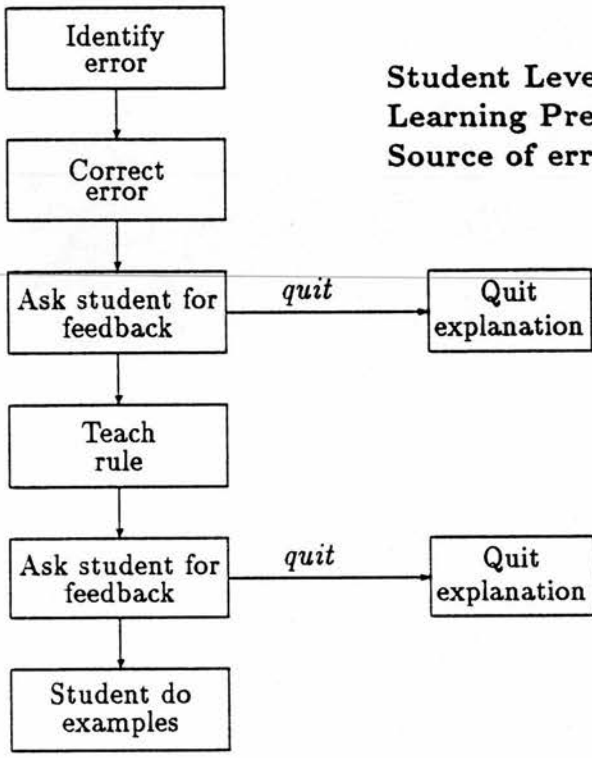


Explanation type 7



Student Level: 3
Learning Preference: Examples
Source of error: Reason not known

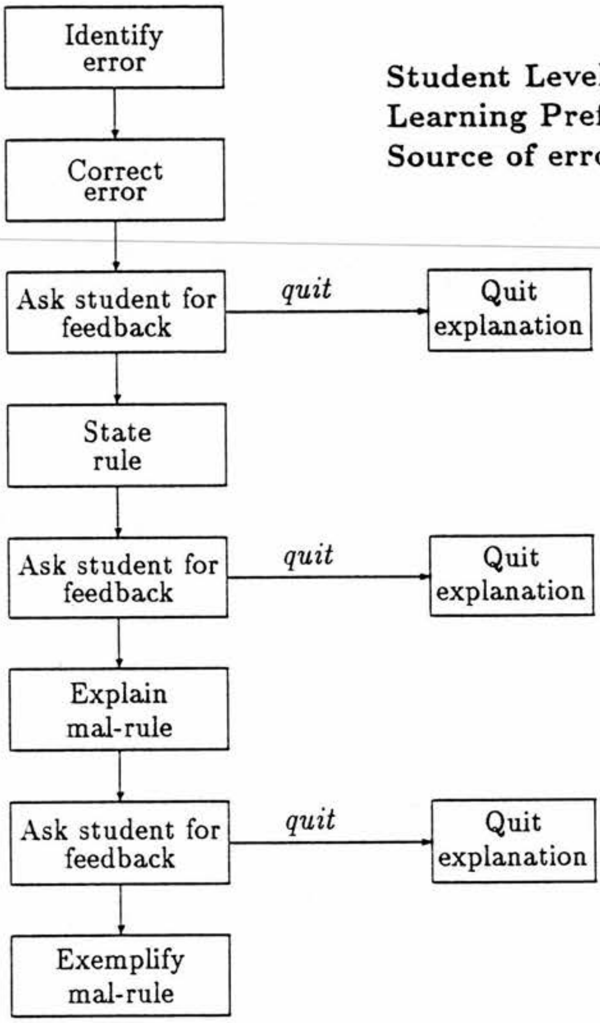
Explanation type 8



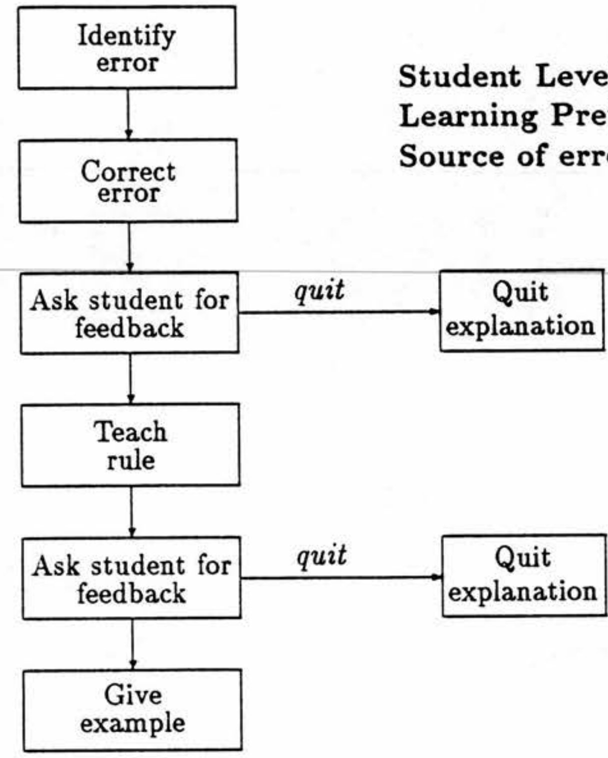
Student Level: 3
Learning Preference: Rules
Source of error: Rule not known

Explanation type 9

Student Level: 3
Learning Preference: Rules
Source of error: Student has mal-rule



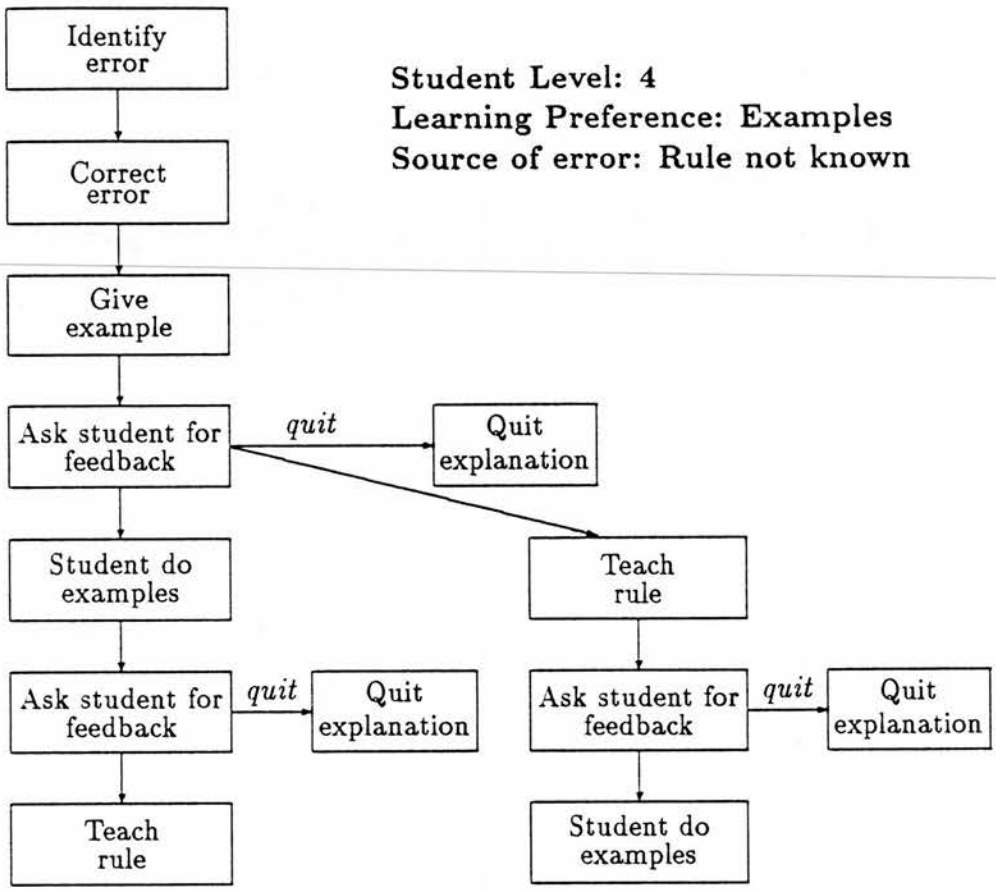
Explanation type 10



Student Level: 3
Learning Preference: Rules
Source of error: Reason not known

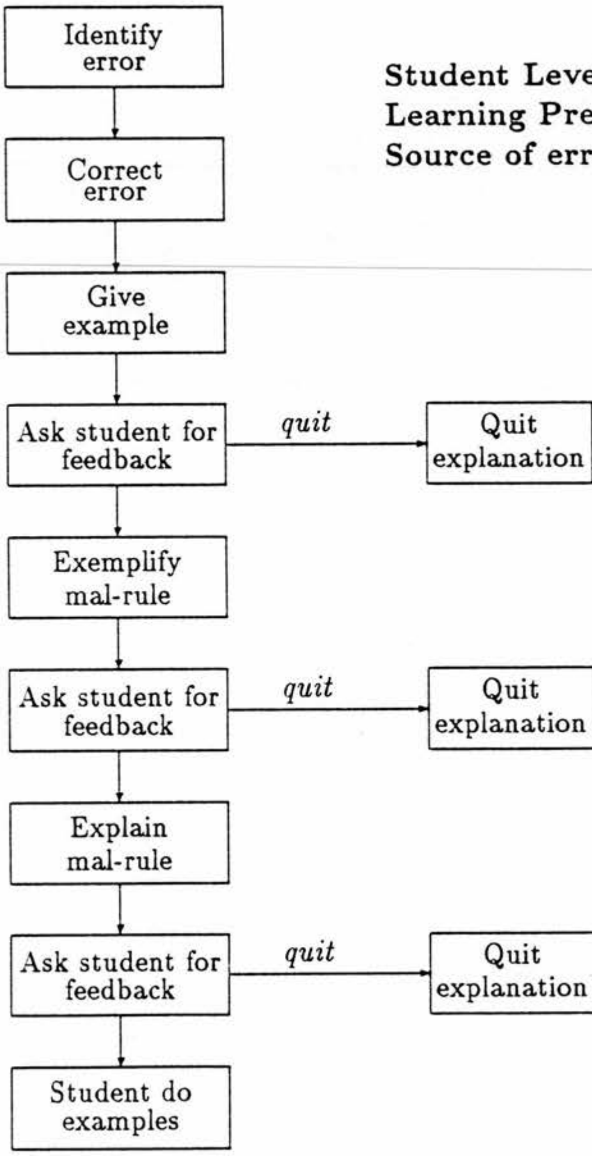
Explanation type 11

Student Level: 4
Learning Preference: Examples
Source of error: Rule not known



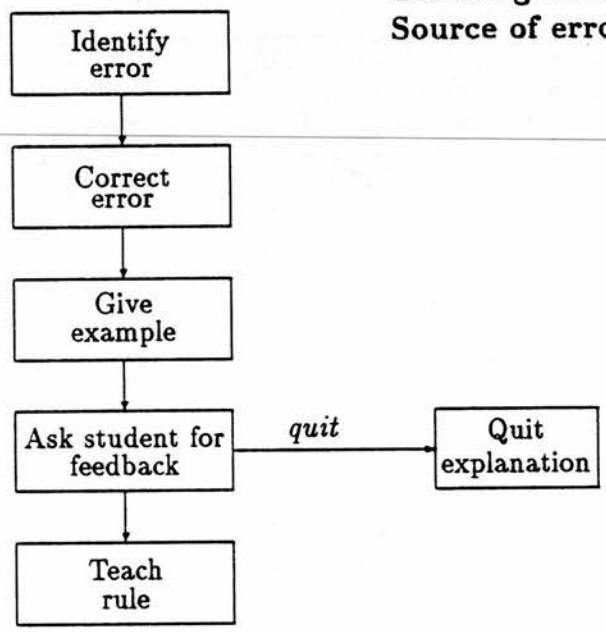
Explanation type 12

Student Level: 4
Learning Preference: Examples
Source of error: Student has mal-rule



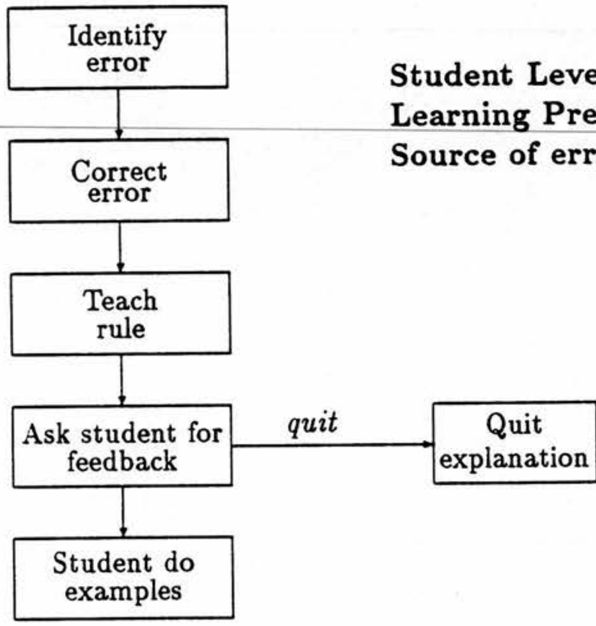
Explanation type 13

Student Level: 4
Learning Preference: Examples
Source of error: Reason not known



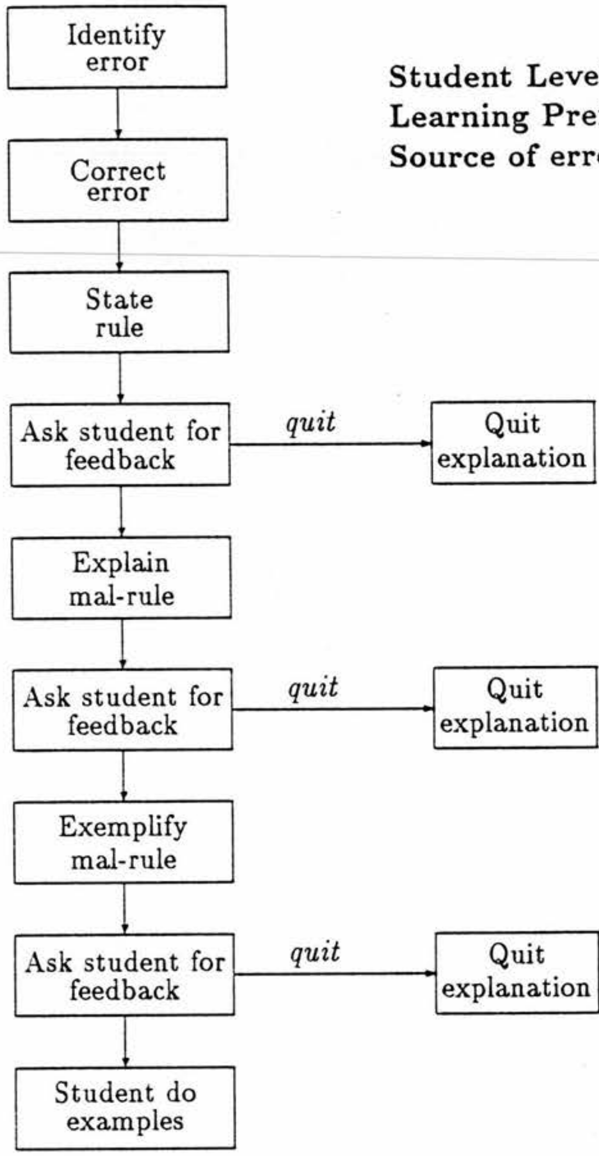
Explanation type 14

Student Level: 4
Learning Preference: Rules
Source of error: Rule not known

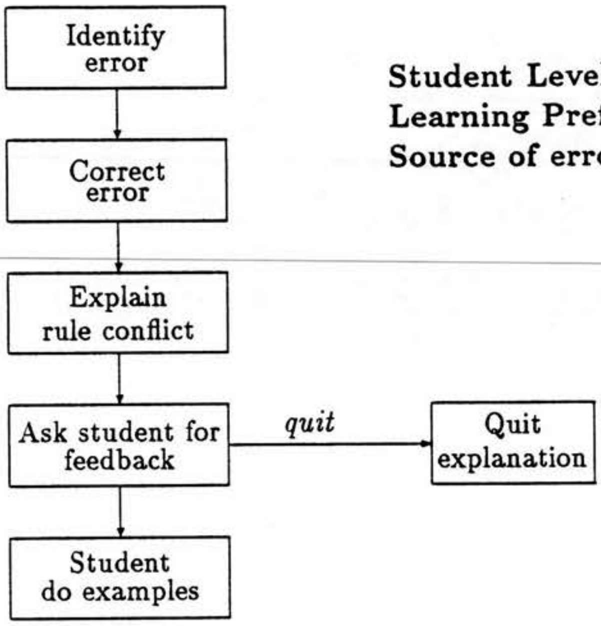


Explanation type 15

Student Level: 4
Learning Preference: Rules
Source of error: Student has mal-rule

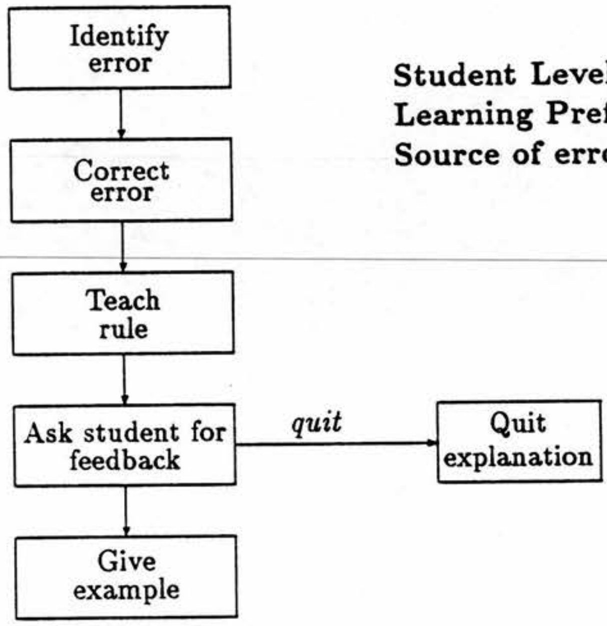


Explanation type 16



Student Level: 4
Learning Preference: Rules
Source of error: Rule conflict

Explanation type 17



Student Level: 4
Learning Preference: Rules
Source of error: Reason not known

Appendix H

Evaluation of natural language processor

H.1 Sentences parsed by *ArtCheck*

He doesn't help.
He certainly doesn't help.
He confidently accepted their conditions.
He accepted their conditions confidently.
Confidently he accepted their conditions.
In the abbey he helped the abbot.
He helped the abbot in the abbey.
In an anxious mood he helped the abbot.
He helped the abbot in an anxious mood.
With some anxiety he helped the abbot.
He helped the abbot with some anxiety.
Without a doubt he helped the abbot.
He helped the abbot without a doubt.
Without a doubt but with some anxiety he helped the abbot.
He helped the abbot without a doubt but with some anxiety.
Confidently but not without some anxiety he helped the abbot.
He helped the abbot confidently but not without some anxiety.
He is crazy isn't he.
He isn't crazy is he.
He helps doesn't he.
He doesn't help does he.
He can't help can he.
He can help can he.
He put it there.
He put it here.
He helped them down at the abbey.

He helped them because of their age.
This mood of Lee's is not very characteristic.
He couldn't help hearing that admission of the abbot's.
He helps.
He helps out.
She abandons him.
She asks the abbot out.
She asks out the abbot.
She asks him out.
She asks out him.
She gives him a message.
She gives him it.
She gives him it back.
She gives him the message back.
She gives him back it.
She gives him back the message.
She agrees with him.
She carries on with him.
She acquits him of it.
She gives it back to him.
She gives back it to him.
She gives the message back to him.
She gives back the message to him.
She answers to him for her actions.
She answers for her actions to him.
She comes down on him for his actions.
He turned it from a doubt into an anxiety.
He bartered his abacus with them for their message.
He bartered it with them for her.
He falls into the abbey.
He gets on its back.
He gets under the abbey.
He ended up at the abbey.
He ended up in the abbey.
She puts it beside him.
It costs him his abacus.
It set him back his abacus.
He acts well.
He acts resolutely.
The message comes across well.
He comes across resolutely.
He comes out resolutely.
This augurs well.
This augurs well for him.
He acquits himself well.
She anticipates that he will help.
She anticipates he will help.

She let on that she knows him.
She let on she knows him.
It appears that she knows him.
It appears she knows him.
It turns out that she knows him.
That he didn't help mattered.
It mattered that he didn't help.
It mattered he didn't help.
He promised her that he would help.
He promised her he would help.
That he apologized amuses her.
It amuses her that he apologized.
She agrees with him that he should help.
It dawned on him that he ought to help.
She gets through to him that he should help.
That Lee helps matters to her.
It matters to her that Lee helps.
She arranged for him to help.
She arranged with him for him to help.
She arranged that he help.
He petitioned them that they let him appeal.
He petitioned them they let him appeal.
She asks who helps.
She figured out who helped.
She doesn't know about what matters to him.
She asks whether he helps.
She asks if he helps.
She couldn't figure out whether he helped.
She might figure out if he helped.
She didn't take in whether he helped.
They advised him what he should accept.
They advised him who would help him.
They asked him whether he had accepted.
They asked him if he had accepted.
I would appreciate it if you could help me.
She dictated to him whether they would accept.
It dawned on him what he should do.
She asked what to give him.
She couldn't figure out whether to help.
She reflected on whether to help.
She arranged with him whether to do it.
He acts the host.
He appears an able host.
He looks an able host.
He ended up abbot.
He turned out an able abbot.
She appears busy.

It appears certain to amuse her that he apologized.
That he apologized appears certain to amuse her.
She looks busy.
She begins to help.
It is beginning to rain.
That he won't help appears to amuse her.
It turns out to matter that he was crazy.
That he was crazy turns out to matter.
She agrees to apologize to him.
She set out to help him.
It must begin raining.
That he was crazy started amusing her.
It carried on raining.
He could do with being more confident.
It could do with raining.
She anticipated being able to help.
She figured on abandoning him.
She was banking on being able to help.
She might get around to helping him.
He will get caught.
He will get looked at.
She promised him to help.
He strikes me as crazy.
He strikes me as an able host.
She acknowledged him an acquaintance.
She knows him to be crazy.
She couldn't bring herself to help him.
She looks to him to help her.
It falls to her to help him.
She appeals to him to help her.
He comes down on us to help him.
It hurts her to abandon him.
She anticipated him helping.
She anticipated it hurting her to abandon him.
She figured on him helping.
She was banking on him helping.
She puts him off helping.
She lets him off helping.
She could hear him apologizing.
It ended in him helping.
He gives himself over to helping the abbot.
She lets Lee help her.
She makes Lee help her.
She sees him fall over.
She hears him accept their conditions.
She looks at him fall.
She looks at it amuse Lee that she helps.

She gets him accepted.
She gets him looked at.
She sees him accepted.
She sees him looked at.
It hurts to fall.
To fall hurts.
She does help.
To help does amuse her.
She ought to help.
To help ought to amuse her.
She ought to help.
To help ought to amuse her.
She has helped him.
She is helping.
She is amused that he helped.
That he is crazy is amusing her.
That he is crazy is acknowledged by her.
She is an acquaintance.
She is not the host.
She is in the mood.
She is not in the mood.
She is crazy.
She is not crazy.
There is an abbot in the abbey.
It is the abbot who dictates messages.
It is to the abbot that he gives the message.
He is agreed with.
He is carried on with.
He is carried with on.
That he helps is acknowledged by her.
He is amused that she helps.
It is acknowledged to help that he accepted.
That he accepted is acknowledged to help.
It is acknowledged certain to help that he accepted.
That he accepted is acknowledged certain to help.
She helps busily.
She busily helps.
She helps in the abbey.
She is busily helping.
She helps busily and with abandon.
She does not help.
Mr Smith is going to London.
The big fat lazy cat is sleeping.
The man who I saw yesterday is here.

H.2 Sentences not parsed by *ArtCheck*

Don't help him.
Do be more resolute.
Help me.
Apologize to him.
He can't help can't he.
He ought to help isn't he.
He hasn't abandoned her mustn't he.
Because he was scared Lee didn't arrange to help.
Since I am not appreciated I don't choose to help.
Lee wasn't able to help although he had promised that he would.
It cost much anxiety.
It may come about that he will help.
It may come about he will help.
She may have him on that she knows him.
He bet her his abacus that he could make her blush.
He bet her his abacus he could make her blush.
She arranged with him that he help.
She arranged with him what they would see.
She worked out what to give him.
She knows about who to see.
She asked whether to help.
They asked him who to help.
They advised him whether to accept.
She arranged with him what to do.
She started off eager.
It started off convenient that he should help.
That he should help started off convenient.
He turns out to have been crazy.
She ended up crazy.
She may begin being resolute.
To appear confident carried on being easy.
To appear confident could do with being easy.
He appears to her to be crazy.
It appears to her to be raining.
That he will help appears to her to be certain.
She arranged with him to give him the message.
It strikes me as conceivable that he would help.
She acknowledged it necessary that he help.
She made him out crazy.
She counts him crazy.
She sanded it down normal.
She condemned him as crazy.
She put him down as crazy.
She made him out to be crazy.

She allows for him to be anxious.
She allows for there to be doubts about the abbot.
She may draw on him to help her.
She looks to it to amuse him that she helps.
She appeals to there to be doubts.
To abandon him hurts her.
Do help her.
Do be less happy.
She may help.
To help may amuse her.
To be crazy has amused her.
For her to help us is inessential.
For her to help us is not inessential.
She is busy and in the mood.
She is not busy and in the mood.
He was made out to be crazy.
That he accepted was made out certain to help.
Which car are you going to buy?

Appendix I

Evaluation : Test Materials

This appendix includes the pre- and post-tests given to students in the exercise described in Section 8.4.1. The tests Multiple Choice (1a) and (1b) were given to students at Level 1. The tests Multiple Choice (2a) and (2b) were given to students at Level 2. The tests Multiple Choice (3a) and (3b) were given to students at Level 3. The story writing exercise was given to all students as part of the pre-test, after the multiple choice, in order to elicit additional article usage errors.

Multiple Choice (1a)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a***,* **an***,* **the***, or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5. I have a friend called ____ John.
Which is the missing article? **a** **an** **the** **no article**

- -

1. Rosie is reading a magazine. She finds ____ magazine very interesting.
Which is the missing article? **a** **an** **the** **no article**

2. Jack is eating ____ apple.

Which is the missing article? **a** **an** **the** **no article**

3. I decided to buy ___ black coat.
Which is the missing article? **a** **an** **the** **no article**

4. Sally is ___ doctor.
Which is the missing article? **a** **an** **the** **no article**

5. Ben Nevis is ___ highest mountain in Scotland.
Which is the missing article? **a** **an** **the** **no article**

6. Peter has ___ two brothers.
Which is the missing article? **a** **an** **the** **no article**

7. One day I saw ___ giraffe in my garden.
Which is the missing article? **a** **an** **the** **no article**

8. I usually drink ___ beer when I go out.
Which is the missing article? **a** **an** **the** **no article**

9. ___ man who is standing over there is my father.
Which is the missing article? **a** **an** **the** **no article**

10. There are ___ cats in my garden.
Which is the missing article? **a** **an** **the** **no article**

Multiple Choice (1b)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a, an, the,** *or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5. I have a friend called ____ John.
Which is the missing article? **a an the no article**

- -

1. Andrew is ____ tallest boy in his class.
Which is the missing article? **a an the no article**

2. Jane Smith has ____ four cats.
Which is the missing article? **a an the no article**

3. ____ red car came speeding round the corner.
Which is the missing article? **a an the no article**

4. Peter found ____ mouse in his kitchen.
Which is the missing article? **a an the no article**

5. ____ film that I saw last night was called Star Trek VI.
Which is the missing article? **a an the no article**

6. There was ____ book on the table.
Which is the missing article? **a an the no article**

7. She doesn't like ____ coffee.
Which is the missing article? **a an the no article**

8. This is ____ interesting book.
Which is the missing article? **a an the no article**

9. ____ cows eat grass.

Which is the missing article? **a** **an** **the** **no article**

10. Sheila's father bought her a horse. She rides ____ horse at the weekend.
Which is the missing article? **a** **an** **the** **no article**

Multiple Choice (2a)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a, an, the,** *or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5 I have a friend called ____ John.
Which is the missing article? **a** **an** **the** **no article**

- -

1. Don't stand in ___ middle of the road!
Which is the missing article? **a** **an** **the** **no article**

2. James is going to ____ school today.
Which is the missing article? **a** **an** **the** **no article**

3. I bought ___ fourth car I went to see.
Which is the missing article? **a** **an** **the** **no article**

4. He was rushed to ___ Royal Infirmary.
Which is the missing article? **a** **an** **the** **no article**

5. Is that a map of ___ West Indies?
Which is the missing article? **a** **an** **the** **no article**

6. All ____ teachers are on holiday for six weeks.
Which is the missing article? **a** **an** **the** **no article**

7. Is the desk made of ____ wood?

Which is the missing article? **a** **an** **the** **no article**

8. For four years I was ___ student at Edinburgh University.
Which is the missing article? **a** **an** **the** **no article**

9. This really is ___ most childish behaviour!
Which is the missing article? **a** **an** **the** **no article**

10. ___ teacher I used to like has now left.
Which is the missing article? **a** **an** **the** **no article**

11. There are ___ ninety nine pages in this book.
Which is the missing article? **a** **an** **the** **no article**

Multiple Choice (2b)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a, an, the,** *or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5. I have a friend called ____ John.
Which is the missing article? **a** **an** **the** **no article**

- -

1. This is ___ second time this has happened to us.
Which is the missing article? **a** **an** **the** **no article**

2. Linford Christie was ____ fastest 100m runner at the Olympics.
Which is the missing article? **a** **an** **the** **no article**

3. ____ restaurant that they went to was closed.
Which is the missing article? **a** **an** **the** **no article**

4. Edinburgh Castle stands at the top of ___ Royal Mile.
Which is the missing article? **a** **an** **the** **no article**

5. I think ___ Smiths have gone away on holiday.
Which is the missing article? **a** **an** **the** **no article**

6. That cat has just drunk all ____ milk.
Which is the missing article? **a** **an** **the** **no article**

7. Would you like ___ milk in your coffee?

Which is the missing article? **a** **an** **the** **no article**

8. Paul drank ___ eight pints of beer.
Which is the missing article? **a** **an** **the** **no article**

9. When Jean was ___ air hostess she travelled all over the world.
Which is the missing article? **a** **an** **the** **no article**

10. Tricia lives at ___ end of this street.
Which is the missing article? **a** **an** **the** **no article**

11. We are travelling around Europe by ___ car.
Which is the missing article? **a** **an** **the** **no article**

Multiple Choice (3a)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a, an, the,** *or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5. I have a friend called ____ John.
Which is the missing article? **a** **an** **the** **no article**

- -

1. All ____ peaches I bought were rotten.
Which is the missing article? **a** **an** **the** **no article**

2. I drink two pints of milk ____ day.
Which is the missing article? **a** **an** **the** **no article**

3. ____ life is short.
Which is the missing article? **a** **an** **the** **no article**

4. It is unusual to see such ____ talent.
Which is the missing article? **a** **an** **the** **no article**

5. It is very hard for ____ elderly to manage in winter.
Which is the missing article? **a** **an** **the** **no article**

6. You are ____ only person I can trust.
Which is the missing article? **a** **an** **the** **no article**

7. Last time I was in Britain I visited ___ Tower of London.
Which is the missing article? **a** **an** **the** **no article**

8. There are ___ thousand pages in this book.
Which is the missing article? **a** **an** **the** **no article**

9. We went to ___ same place on holiday last year.
Which is the missing article? **a** **an** **the** **no article**

10. If you are lucky you can go on holiday twice ___ year.
Which is the missing article? **a** **an** **the** **no article**

11. I would like to visit ___ United States of America.
Which is the missing article? **a** **an** **the** **no article**

12. ___ sun is millions of miles away from us.
Which is the missing article? **a** **an** **the** **no article**

Multiple Choice (3b)

Name: ....
Pre/Post test? ....

Instructions: *You will be given some sentences with one of the articles (***a**, **an**, **the**, *or* **no article***) left out. You must decide what the article should be and circle your answer. An example is given below:*

EXAMPLE:

5. I have a friend called ____ John.
Which is the missing article? **a** **an** **the** **no article**

- -

1. Tricia works in a day centre for ____ homeless.
Which is the missing article? **a** **an** **the** **no article**

2. I have spent all ___ money I brought with me.
Which is the missing article? **a** **an** **the** **no article**

3. If I won ___ million pounds I would give it all away.
Which is the missing article? **a** **an** **the** **no article**

4. ____ age is not important.
Which is the missing article? **a** **an** **the** **no article**

5. That is ___ only answer to the question.
Which is the missing article? **a** **an** **the** **no article**

6. The angry mob held a protest outside ___ Town Hall.
Which is the missing article? **a** **an** **the** **no article**

7. Jonathan usually goes to church at least once ___ week.
Which is the missing article? **a** **an** **the** **no article**

8. This car can go at ninety miles ___ hour.
Which is the missing article? **a** **an** **the** **no article**

9. Tom has lived in ___ same house for thirty years.
Which is the missing article? **a** **an** **the** **no article**

10. We went to see ___ Stone Roses in concert last night.
Which is the missing article? **a** **an** **the** **no article**

11. The population of ___ world is growing all the time.
Which is the missing article? **a** **an** **the** **no article**

12 I have always said she is such ___ likeable person.
Which is the missing article? **a** **an** **the** **no article**

Story Writing

Name: ....

Instructions: ***Please write a SHORT paragraph on ONE of the subjects given below. Do not spend more than 5-10 minutes on this part of the exercise.***

Subjects:

- Your stay in Edinburgh
- *OR* Your favourite place in Edinburgh
- *OR* Your home town
- *OR* Somebody you know
- *OR* The Olympics