Knowledge acquisition from multiple domain experts for expert systems

Loc Dinh Cung
University of Wollongong

Recommended Citation
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
Knowledge Acquisition from Multiple Domain Experts for Expert Systems

A thesis submitted in fulfilment of the requirements for the award of the degree of

MASTER OF ENGINEERING (Honours)

from

THE UNIVERSITY OF WOLLONGONG

by

Loc Dinh Cung
BE (Hons) in Electrical Engineering
The University of Wollongong (1987)

Department of Electrical and Computer Engineering
October 1988
ABSTRACT

Some aspects of the problem of knowledge acquisition from multiple experts for Expert Systems are considered using an empirical approach. In particular, the problem of knowledge acquisition from multiple experts from diverse backgrounds is of major interest.

A concept of modelling multiple experts is developed based on the model of a Human Information Processing System. This concept is then applied, together with a revised methodology for knowledge acquisition, to achieve higher efficiency in the knowledge acquisition process.

The above approach is successfully applied in the development of an expert system for the diagnosis of faulty plan view shapes of steel plates at the Slab and Plate Products Division, BHP International Group. Major aspects of the expert system are described, and practical aspects of knowledge acquisition are also presented and discussed.

To assist the knowledge engineer in a knowledge acquisition process, it is desirable to have an automated tool. An automated knowledge acquisition system is designed and implemented to fulfill this task. This system has the potential to acquire diagnostic knowledge directly from a domain expert and to use the knowledge to form a knowledge base for an expert system shell.
ACKNOWLEDGEMENT

I wish to express my deepest gratitude and appreciation to Associate Professor Tung Sang Ng for the guidance, encouragement and support which he provided throughout the course of this research, and also for his patience in reading the manuscript.

I also wish to extend my gratitude to Dr. Paul M. Carter, Dr. Roy M. Howard, Mr. Joe F. Chicharo and Dr. B. S. Perera for fruitful discussions and invaluable comments on the subject of this thesis. Special thanks are due to Dr. Karen McGraw and Captain Robert J. Hammel for useful information that they provided during the course of this research.

I also wish to thank Miss Maree Fryer for reading the manuscript, and Mrs. Julie Chin for her efficient typing. Their help throughout the course is gratefully acknowledged.

I am thankful to Ngoc Diep for the special encouragement, support and prayers.
## CONTENTS

1. **Introduction**  
   1.1 Introduction  
   1.2 Organisation of the Thesis  

2. **Literature Survey and Summary of Contributions**  
   2.1 Introduction  
   2.2 Manual Approaches to Knowledge Acquisition  
   2.3 Efforts in Automating Knowledge Acquisition  
   2.4 Summary of Contributions  

3. **Modelling Multiple Experts for Knowledge Acquisition**  
   3.1 Introduction  
   3.2 Model Creation  
   3.2.1 Model of an Expert System  
   3.2.2 Model of a Human Expert  
   3.2.3 Model of Multiple Experts  
   3.3 A Methodology for Knowledge Acquisition from Multiple Experts from Multiple Disciplinary Backgrounds  
   3.3.1 Identification of a Model for the Multiple Experts  
   3.3.2 Formation of a Multiple Expert Team  
   3.3.3 Information Retrieval from the Multiple Expert Team  
   3.3.4 Verification of the Acquired Knowledge  
   3.4 Conclusion
4. DESPLATE: A Diagnostic Expert System for Faulty Plan View Shapes of Steel Plates

4.1 Introduction
4.2 The Plate Mill
4.3 The Specific Problem
4.4 DESPLATE

4.4.1 System Overview
4.4.2 The Knowledge Base
4.4.3 Control Strategy
4.4.4 User Interface
4.4.5 Current Status

4.5 Knowledge Acquisition for DESPLATE
4.6 Conclusion

5. AKAS: An Automated Knowledge Acquisition System

5.1 Introduction
5.2 Algorithm for Acquiring Diagnostic Knowledge
5.3 Overview of AKAS
5.4 Causal Structure for Diagnosis of AKAS
5.5 Rule Generator
5.6 Interactive Interface
5.7 Conclusion

6. Conclusions and Recommendations

References

Appendix A Graphics displays from DESPLATE
Appendix B Graphics displays from AKAS
CHAPTER ONE

INTRODUCTION
1.1 Introduction

In recent years, research in the field of Artificial Intelligence (AI) has produced many important results. Among the most significant of these has been the development of a new class of computer programs known as Expert Systems (ES). These programs exhibit a very high level of performance in solving complex real-world problems within a well defined domain, they are flexible at design and robust at run time, and they are capable of providing explanations for their solutions. ES are results of an AI approach to designing software programs, they reason with symbolic information, as opposed to numeric processing, and they use inference procedures based on human knowledge rather than on algorithmic procedures. Examples of some early ES include DENDRAL (Buchanan and Feigenbaum, 1978) and MYCIN (Shortliff, 1976). DENDRAL inferred molecular structure of a substance given its mass spectrographic information. Although at the time this program was designed, an algorithm for generating all possible molecular structures existed, an exhaustive search would have been extremely expensive. DENDRAL encoded the knowledge of expert chemists into rules that controlled such a search, making it possible to obtain a satisfactory answer with a fraction of effort. Similarly, MYCIN gave consultative advice on diagnosis and therapy for blood infectious diseases that compared favourably with advice given by a human physician.

The high level of performance of an expert system comes from the specific knowledge that it contains about a particular domain. Thus, the most important task in constructing an expert system has always been the process of knowledge acquisition (KA). This process involves the acquisition and transfer of an expert's knowledge to an expert system. Experts' knowledge usually consist of facts and heuristics. The facts constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in the field. In the other hand, heuristics are mostly private, seldom discussed knowledge that characterize an expert's ability and skill in the domain.
The acquisition of facts is relatively simple, but acquiring heuristics is a much more complicated and time consuming task. A traditional knowledge acquisition process would normally involve two persons: a domain expert, who is proficient in solving problems within the domain of interest, and a knowledge engineer, who is a specialist in knowledge acquisition and in building ES. By the late 1970s, when ES applications were beginning to permeate many different fields including engineering, manufacturing, financial services, environmental sciences and many others, knowledge acquisition became recognised as an issue with ES because it turned out to be difficult, time consuming, and the number of knowledgeable domain experts was limited. As the fields of ES application continued to expand, there existed situations where a single domain expert could no longer be capable of providing adequate solutions. Consequently, there was a need for acquiring knowledge from multiple domain experts and this posed yet another challenging question to knowledge acquisition research. In addition, while there is an increasing demand for ES, efficient knowledge engineers are in extremely short supply. It would be desirable to have an alternative approach to knowledge acquisition in order to shift this task from humans to computers.

In this thesis, some aspects of knowledge acquisition from multiple domain experts are investigated. These include a concept of modelling multiple domain experts for knowledge acquisition, a practical development of an expert system using multiple experts from different disciplines, and an automated approach to the process of knowledge acquisition. The approach is to model involved domain experts, using the concept of a human Information Processing System, and use it as a guide to the knowledge acquisition process. In conjunction with the model, an appropriate methodology is employed, and finally, an automated tool may be used as an aid in such a process to improve productivity.
1.2 Organisation of the Thesis

Chapter Two presents a literature survey of major trends and techniques used in the process of knowledge acquisition. This provides background material for the work described in the subsequent chapters. The chapter characterises knowledge acquisition approaches into two major categories, and examines each in detail. Potential areas of research, that are directly related to the work in this thesis, are highlighted. A summary of contributions of this thesis is also given.

Chapter Three presents the concept of modelling a group of multiple experts for knowledge acquisition. In general, multiple experts who work on the same problem may either have expertise in a similar domain or may come from different disciplinary backgrounds. This Chapter focusses on the latter case. A methodology for knowledge acquisition from such a group of domain experts is also introduced.

Chapter Four presents the application of the concept for modelling multiple experts and the methodology for knowledge acquisition in Chapter Three to the development of DESPLATE, an expert system for the diagnosis of faulty shapes of steel plates at the Plate Mill, BHP Steel International Group, Slab & Plate Products Division, Port Kembla. The issues of knowledge representation and control strategy for DESPLATE are considered. Acquired knowledge is classified into different hierarchical levels and the use of both backward chaining and forward chaining is implemented.

In Chapter Five the design and implementation of an automated knowledge acquisition system, AKAS, are described. This system is capable of interviewing a domain expert for diagnostic knowledge. Acquired knowledge are represented by production rules which eventually form a knowledge base suitable for a commercial expert system shell. AKAS is intended to be used as a tool to assist a knowledge engineer in
parts of the KA process.

Finally in Chapter Six, conclusions and recommendations for future research are presented.
CHAPTER TWO

Literature Survey and Summary of Contributions
2.1 Introduction

The saying 'a decision is only as good as the information on which it is based' may sound familiar to most people. Similarly, the power to enhance and amplify the performance of an expert system resides in the specific knowledge of the problem domain that the system can possess (Feigenbaum and McConduit, 1983). Moreover, high performance requires that an expert system has not only general facts and principles but the specialized expertise that separates human experts from novices. Such expertise is also called heuristics. The transfer and transformation of heuristics from human experts to an expert system is known as Knowledge Acquisition (KA).

KA is currently the major bottleneck in the construction of expert systems and this directly affects the number of systems being developed. The process of KA does not simply involve information retrieval but in contrast it is a complex combination of several overlapping stages. These include identification, conceptualization, prototyping, development of the complete system, evaluation, integration (if necessary) and maintenance. In most cases a person called a knowledge engineer (KE) is required to communicate with domain experts and build a system from the start. The KE is a specialist in explicating heuristics from domain experts and prototyping an expert system that contains the knowledge. The KE then works with the experts to improve the system until it meets the required level of performance. Compared with a conventional software system analyst, the KE performs a more complicated task as he also has to analyze the thought processes of the domain expert (Hart, 1986). This approach to KA normally involves long interview sessions. With the current state-of-the-art interviewing techniques it is very time consuming and hence costly if the knowledge domain is ill-structured or if the knowledge engineer lacks the necessary communication skills (Cooke and McDonald, 1986).
As most applications of expert system technology rapidly expand to cover a large number of disciplines, a single expert can no longer be considered to be a knowledge czar. Consequently, there is a need for multiple experts either from the same domain or from inter-disciplinary backgrounds. Independent research groups have taken different approaches in dealing the problem of KA from multiple domain experts. Some researchers emphasized the importance of the early stage of KA and introduced the use of resident experts (Mittal and Dym, 1985). Others attempted to present a more systematic approach based on techniques used in psychology (McGraw and Seale, 1987). In most cases KA from multiple experts has been reported to be a much more difficult process than KA from a single expert (Standfield and Greenfeld, 1987; Woolf and Cunningham, 1987).

Because of the problems involved in handcrafting knowledge into expert systems many groups of researchers have been trying to shift the responsibility of the KA task from human to a computer. The efforts of these groups may be classified into four major areas. First, there have been endeavour in designing knowledge-based editors to facilitate the task of entering knowledge into an expert system. Also related to this area are expert system shells and programming environments that relieve the KE from programming burdens.

Second, there have been attempts to write intelligent editing programs that could assist in debugging and updating existing knowledge-based systems via interactive dialogues with domain experts. The most well known example in this area is TEIRESIAS (Davis, 1977). A similar approach has been taken by others, including Bonasso (1985) and Finin (1986).

Third, there are strong developments in the area of inductive KA. This reflects the ambition to create programmes that can learn from past examples. Many inductive
algorithms have played a role in realising this ambition. There have been positive attempts by Lenat (1976, 1983), Buchanan and Mitchell (1978), Michalski (1978, 1980). More recently, further progresses have been reported in Quinlan (1979, 1985, 1986, 1987), Carter and Catlett (1987). Inductive KA, however, still suffers from some fundamental problems: induced rules can be difficult to understand and hard to modify, and the general applicability is relatively limited (Sammut et al, 1986). These problems have motivated researchers to combine both inductive and deductive methods for KA (VanTerheyden and Chalcraft, 1987, Buntine and Stirling, 1988).

Finally, there has been much effort in creating automated systems to acquire front-end knowledge directly from domain experts, such as the work done by Leal and Pearl (1983), Boose (1984, 1985), Kahn, Nowlan and McDermott (1985). These systems exhibit some forms of intelligence even though they have not been designed to understand natural language. Natural language research is beyond the scope of this review. However, some interactive KA programmes have been reported to incorporate this capability (Phillips et al, 1985).

This survey, by its very nature, does not aim to cover completely the subject matter. In this Chapter the major approaches to KA for expert systems are considered. In particular, two aspects are examined:

1. manual approaches to KA, especially KA from multiple experts; and
2. major work attempted in shifting the responsibility of a KE to a computer.

In the following sections detailed examinations of these aspects are presented.
2.2 Manual Approaches to Knowledge Acquisition

The process of KA has been investigated by a number of authors, for example, Buchanan (1981), Hayes-Roth, Waterman and Lenat (1983), Feigenbaum and McConduck (1983), Clancey (1984), Harmon and King (1985), Hart (1986), Bobrow, Mittal and Stefik (1986), Cooke and McDonald (1986), and Debenham (1988). The conclusion drawn from these papers is that KA is currently carried out in a painstaking, iterative and time-consuming manner. There is no standard procedure for KA except a general view that it is a complex combination of several overlapping stages where each stage is a rough characterization of the tasks involved. In Hayes-Roth, Waterman and Lenat (1983), Kahn, Nowlan and McDermott (1985) and Bobrow, Mittal and McConduck (1986) the authors attempted to summarize their experiences after constructing numerous expert systems. Other researchers emphasised on methods for acquisition of front-end knowledge (Waldron, 1985), while others introduced useful techniques for KA based on theories in psychology (Hall and Blander, 1985) or cognitive psychology (Cooke and McDonald, 1986). A brief listing of manual techniques for KA are shown in Table 2.1.

There are distinctions between building a small expert system and building a large-scale expert system (Harmon and King, 1985). In the following section we will consider the later category (which involves a more complicated process) and will examine commonly accepted stages in a manual KA process. These stages include identification, conceptualization, prototyping, development of the complete system, evaluation, integration and maintenance. They are described in more details in the following sections.
TABLE 2.1: Applicable techniques in a manual knowledge acquisition process

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Brainstorming (BS)</strong></td>
<td>This is a technique used at the initial stage of an interview session. It is aimed at helping experts and knowledge engineers to break loose from obvious or conventional solutions, to stimulate thinking and generate ideas. BS is usually followed by other techniques that help to evaluate the acquired information (McGraw and Seale, 1987a).</td>
</tr>
<tr>
<td><strong>Distinguish Goals</strong></td>
<td>Given a specific goal (a state of belief or decision) the expert is asked to specify sets of evidence which are necessary and sufficient to distinguish this goal from the other alternatives. This method provokes discussion about the precise nature of, and reasons for, the goals (Hart 86).</td>
</tr>
<tr>
<td><strong>Frequency Conditionalization</strong></td>
<td>This technique helps to determine if there are background conditions under which a particular goal is more or less likely to occur (Kahn et al, 1985). This technique may be used to evaluate acquired information.</td>
</tr>
<tr>
<td><strong>Induction Technique</strong></td>
<td>This technique helps to automatically induce rules from examples. Using this approach a KE does not have to worry about deriving rules but only to retrieve sufficient example sets from the domain expert. This topic will be discussed in great detail later in a separated section when we consider Inductive KA.</td>
</tr>
<tr>
<td><strong>Interesting Cases (IC)</strong></td>
<td>This is a technique to encourage the domain expert to describe interesting or difficult cases that he can recall (Hart 86). These cases are not necessarily typical but they are more memorable and stimulating. IC makes the expert talk more easily and thus is quite useful to start the KA process.</td>
</tr>
<tr>
<td><strong>Knowledge Level Analysis</strong></td>
<td>This analysis involves describing inference and search structure independently (Clancey 84). It separates issues of knowledge structure from representation and search. Instead of using terms like rules, actions, goals and backward-chaining, information acquired can be classified into terminological statements, relational statements and inferential statements. The prime target is to devise a better representation language that allows KE to make terms, relations and search procedures explicit.</td>
</tr>
<tr>
<td><strong>Process Tracing (PT)</strong></td>
<td>The aim of this procedure is to systematize the initial stages of KA without frequent interactions with domain experts (Waldron 85). PT is a method of determining the train of thought that allows an individual to make decision. It involves techniques like thinking aloud, retrospective verbalization, discussion protocol, and protocol analysis.</td>
</tr>
<tr>
<td><strong>Protocol Analysis (PA)</strong></td>
<td>This is the analysis of verbal reports acquired from an interview session. PA involves grouping phrases into areas of knowledge, defining inter-relationships between these areas and criteria for passing from one area to another (Newel and Simon 72, Hart 85, McGraw 87).</td>
</tr>
<tr>
<td><strong>Questionnaires</strong></td>
<td>This is a 'traditional' way of retrieving information. Even though this technique is not very effective, questionnaires are still used.</td>
</tr>
</tbody>
</table>
Reclassification

This is part of an interview session. Given a goal, acquired knowledge can be reclassified into evidences for that goal. The general sub-goals are reclassified until the evidence has been broken down into observable symptoms or facts (Hart 86).

Repertory Grid (RG)

This is based on the Personal Construct Theory which employs clinical psychotherapeutic interviewing methods (Kelly 55, Boose 84). The KE uses this technique to build a rating grid consisting of constructs and elements given by the domain expert. A construct is a bipolar characteristic which each element has to some degree, e.g., large-small, tall-short, good-bad, etc. RG allows the KE to study how an expert uses the construct to theorize, hypothesize, evaluate experimental data and reach conclusions (Hall 85). However, RG may not be suitable for retrieving deep causal knowledge, procedure knowledge, or strategic knowledge.

Talk Through (TT)

The expert is asked to describe what he has done while working on a particular task (Hart 86). Another similar approach would be retrospective verbalization in which the expert only reports the knowledge employed after completing the task. The later approach is aimed to avoid problems encountered when trying to verbalize during the task. Nevertheless, these techniques are known to be weak and thus are not very popular (Waldron 85, Cooke and McDonald 86).

Task Analysis (TA)

This is a tool for the initial stage of KA (McGraw and Seale 1987a). It is most effective with procedural knowledge. A task can be classified into a certain type. The KE can use functional analysis, timeline analysis, information flow diagram, goal analysis etc. to analyze the task under consideration. Consequently, TA can help to bound the problem domain, to provide initial framework for the knowledge base and to introduce the KE to the domain.

Test Cases (TC)

these are simple but typical cases to be tried out at an early stage of KA (Rychener 85, Bobrow 86). They help to establish the desired system behaviour in a range of typical problems. TC are normally applicable only when the KE can have frequent contacts with the expert.
Identification

This stage involves identification of suitable tasks, objectives, domain experts, end users and resources. Initially, a knowledge engineer (KE) is required to invest considerable amounts of time and effort to become familiar with the problem domain. This initial stage is very important because it dictates the success of a project. Compared to a conventional system analysis, a cost-benefit analysis may not prove feasible in a KA process for an expert system, but the potential benefits of the system should be considered (Hart 86). The KE should also decide on a set of knowledge engineering tools to be used for KA. Some recently developed tools include ESSAI, an expert system toolkit (Harvey, 1986), Knowledge Acquisition Grid (Lafrance, 1987), and AQUINAS, a knowledge acquisition workbench (Kitto and Boose, 1987).

Conceptualization

This involves the explicit development of key concepts and ideas as well as the appropriate representation paradigm for the system. The techniques applicable are brainstorming, interesting cases, process tracing, questionnaires, talk through, task analysis, and test cases (Table 2.1). All works in this stage will lead to a prototype system. There are two key questions in KA: 'what is it that the expert knows?' and 'how does he use his knowledge?'. While the former can be gradually answered throughout the KA process, the latter should be available at this stage. An incorrect answer would result in serious representation mismatch and, sometimes, it may force the KE to redesign the whole system. Once a suitable representation paradigm has been decided, the acquired knowledge should be represented in the chosen paradigm and it is useful to use a diagrammatic representation for this knowledge and make it explicit to the domain expert.
Prototyping

A first prototype system should be in operation as early as possible (Feigenbaum and McCorduck, 1983; Harmon and King, 1985). Issues of concern are choices of suitable off-the-shell packages, knowledge-engineering tools, programming languages and programming environments. If an expert system shell is to be used it may be decided at this stage based on the chosen representation paradigm. Final acceptance of such a system is also of major concern because if the system does not satisfy the user’s need, it is obviously useless. The knowledge engineer/domain expert team has to make a decision on these issues. Using chosen tools and acquired information, together with results from previous stages, the KE should be able to construct a prototype system. This system will help to keep the expert enthusiastic during the project. Experience has shown that this prototype system may be thrown away completely (Feigenbaum and McCorduck, 1983; Bobrow, 1986; Cung and Ng, 1988a) and it may be necessary to start all over again. This depends, however, on how carefully the KE initially analyzes the system. The prototype system also allows the expert to test its performance. By the end of this stage, the KE and the domain expert will have become more aware of what could be achieved when a full-scale system is developed. Once the prototype system is working relatively satisfactory the next stage can begin.

Development of the Complete System

This stage involves the expansion of a prototype into a complete system. The necessary work may involve re-arrangement of some hierarchical relationships, expanding the knowledge base, tailoring the user interface and monitoring the system performance (Harmon and King, 1985). Techniques applicable are distinguishing goals, structured-interview, knowledge level analysis, protocol analysis, questionnaires, reclassification, and repertory grid. The system can be expanded either in depth by adding rules for handling subtler aspects of particular cases, or in breadth by creating more rules on other
subproblems. The user interface can be designed to meet users' preference. A graphics interface is always advantageous because of its user-friendliness.

**Evaluation**

When the system is completed it should be tested against the objectives or specifications set out at the end of the prototyping stage. Other experts in the field may be invited to try the system, to comment, or to point out any limitations. The most likely areas of problem may be inadequate input/output characteristics, inconsistent or incomplete inference rules (for rule-based expert systems), and incorrect control strategy which causes representation mismatch (Hayes-Roth, Waterman and Lenat, 1983). Other sources may be mis-understanding or mis-conception on the part of the KE, an expert forgetting part of the solutions, or incorrect terminologies (McGraw and Seale, 1987b). It should be noted that in constructing an expert system, the KA process is an iterative one. There is always something to be changed or updated in this stage. The evaluation process may vary from asking the experts if there is something to be changed during a trial run of the system to editing flow diagrams of the knowledge base.

Normally, this stage is the last one before the system is fully commissioned. As the current trend of applications of expert systems moves from isolated off-line systems to on-line real-time system (McNurlin, 1988) the issues involved in integrating the expert system to its environment will be considered in the section which follows.

**Integration**

In order to make use of enormous databases and other equipment within the work environment it is normally necessary for the expert system to integrate with existing systems. Activities during this stage may include interfacing the system with databases, instruments, or other hardware and software. In practice, information concerning
communication capability between the system and neighbouring equipment should be
known at a very early stage, and the KE should include this information in his design. In
this stage the performance of the system should be carefully evaluated for correctness and
accuracy.

Maintenance

In conventional programs subroutine structure and control flow must be designed
explicitly to accommodate all operations. In contrast, the typical architecture of an expert
system is the one that separates the inference engine and the knowledge base. Thus, to
maintain an expert system one only needs to maintain and extend the knowledge base as
new problems arise. One of the key problems in expanding the knowledge base is the
continuing changes necessitated by new equipment, new specifications, etc. To implement
such changes it is necessary to have a trained person (who may not be the developer of the
system) whose task is to keep the knowledge base current by adding new information or
modifying existing rules.

To carry out a proper KA process the KE must have skills, both in analyzing a
given problem and in communicating with the domain expert, to acquire his knowledge
and inference strategies. The KE must also understand the tools and programming
environments available so that the system may be correctly implemented. This process is
usually long and tedious and, in many cases, the task is more than one person can handle.
It may, therefore, often involve more than one domain expert.

Knowledge Acquisition from Multiple Experts

The involvement of multiple experts (ME) in the construction of an expert system
is not something new. In fact, the first project that involved multiple experts from multiple
disciplinary backgrounds dated back to 1965. The team included a professor in genetics
(Nobel laureate), a physical chemist (the father of the birth control pill), and of course, a knowledge engineer (Feigenbaum and McCorduck, 1983). However, the problem of KA from multiple experts only became prominent in the mid-1980s when expert systems had permeated many disciplines, including engineering, science, education, business and administration (Buchanan, 1986). In complex domains such as medicine or engineering any given expert is often knowledgeable about only a small subset of the tasks in the domain (Mittal and Dym, 1985b). The expertise required, in such cases, for an expert system is normally distributed among several experts. This situation will be called multiple experts from multiple disciplinary backgrounds (Cung and Ng, 1988b). Using a single expert in such situations would result in a system with limited, biased, and perhaps erroneous information. For instance, experience with commercially successful expert systems such as R1 (McDermott, 1982) and the Dipmeter Advisor (Smith, 1984) suggests that using knowledge from a single expert can produce systems foreign to other system users. Designers of intelligent teaching systems (Woolf and Cunningham, 1987) also state that there is a need for multiple experts for systems of such nature. Similar problems may also be found in relatively simpler domains where a given expert appears to know everything. During an interview session, the expert may suffer from lack of memory, lack of concentration, misunderstanding of questions, or lack of interest. Thus there is a need for multiple experts within the same domain and this situation is known as multiple experts from single disciplinary background (Boose, 1984c). A number of approaches for dealing with KA from multiple experts is examined.

Mittal and Dym (1985) emphasized the importance of the early stages of a project when issues such as the suitability of a problem and identification of experts are decided. They suggested that the early stage is as important as later stages when knowledge representation, problem-solving strategies and programming techniques are decided. A similar approach presented in Mittal et al (1985) which stems from experience with PRIDE where the major concepts involved are identification of experts, problems, and
separability of tasks. Mittal and Dym introduce the use of *resident experts*, who are interested in collaborating with the KE, as a measure of other domain experts. Initially domain experts are chosen through discussion with as many domain experts as possible. Among the chosen experts, *resident experts* are selected. The resident experts then select a few test cases, explain to the KE what should be done, and then ask other experts to solve these cases in front of the KE. Information acquired concerning commonality of approach and difference in specialization is subsequently analyzed. The KE also analyzes the nature of the expertise that each expert possesses and try to identify the kind of problem-dependent knowledge and strategies the experts seem to use. The resultant knowledge is claimed to be better than any single expert's contribution but the method of KA appears weak.

McGraw and Seale (1987a) introduced another approach to KA from ME which used established techniques in psychology. In MEKAM (McGraw and Seale, 1987a), the authors present a six-step methodology for increasing the efficiency of working with ME. Steps involved include: deciding when to use ME, deciding how to use ME, setting up the multiple expert team (MET), preparing the MET for KA, using KA methods for ME, and finally, debriefing the MET. MEKAM uses traditional techniques, such as brainstorming, consensus decision making and nominal group technique for interviewing domain experts. This methodology was developed for use in the AI Laboratory at Texas Instruments. The focus of this is on the selection and trainings of ME and the use of appropriate techniques that assist in eliciting knowledge and managing the acquisition session. A revised methodology based on this approach to handle multiple experts from diverse disciplinary backgrounds will be presented in Chapter Three.

Woolf and Cunningham (1987) examined the need for multiple experts in building intelligent teaching systems. Here, there is no rigid methodology but rather a set of criteria for acquiring domain knowledge from ME. The authors suggested that even
though real experts are expensive, the success of a project depends critically on their availability and willingness to co-operate. Moreover, domain knowledge can be overly distributed and thus this must be acquired incrementally, prototyped, refined, augmented, and re-implemented.

PlanPower (Standsfield and Greenfeld, 1987) is one of the typical examples of the difficulties in KA from multiple experts. Started in 1982 with one KE and one domain expert, this project was completed in 1986 with a final team of 10 knowledge engineers, 12 system people and 6 domain experts. Several prototypes were completely abandoned. It is claimed to be the earliest large-scale expert system to provide personal financial plans but prompted this vital warning from the developers: "... The difficult and length of time required to develop PlanPower underscore the pressing need for better methodology ..." (Standsfield and Greenfeld, 1987).

There are tools that have potential applications in assisting KA from multiple experts. Examples include: OpusII, developed at Texas Instruments Inc. (McGraw and Seale 1987b) and Colab, developed by Xerox PARC (Stefik et al, 1987). OpusII is a simulation workstation for the development of intelligent vehicle systems, while Colab is an experimental meeting room equipped with computers to support collaborative processes in face-to-face meetings.

It is clear that KA from ME is still in its experimental state. Many methodologies and tools have been borrowed from various disciplines to enhance this process, but there is a definite need for a new approach in this field.

The following section presents the major efforts in automating the process of knowledge acquisition.
2.3 Efforts in Automating Knowledge Acquisition

Up to this point we have only surveyed methods for KA from a handcrafting approach. We next turn our attention to efforts in shifting the responsibility of the KA task from humans to computers.

At the simplest level, there are tools that have been developed to facilitate the task of entering knowledge into a system. These include expert system shells and programming environments. Shells are mostly derived from successful expert systems. A shell has the same structural and logical aspects of the original expert system, and an empty knowledge base. When using shells the KE does not have to be concerned about the programming aspect of building expert systems. Some classical shells are EMYCIN, EXPERT, ROSIE. More recently, several shells such as R1, Insight 2+, LEVEL5, EXSYS and NEXPERT have been commercially available. A programming environment is a tool which provides more flexibility than a shell. Examples include INTERLISP, GOLDEN COMMON LISP, KEE and Knowledge Craft.

Shells and programming environments normally provide good knowledge-base editing facilities. Some can check for typographic and symbolic errors (EMYCIN, INTERLISP, ROSIE). Others can check for completeness and consistency of information (KAS, PLL, EXSYS, NEXPERT). These features help to minimize errors when large knowledge bases are being constructed.

Shells are very useful in prototyping as they reduce development time. The major drawback of shells is, however, the restriction of knowledge representation frameworks that are inherent in the original expert system. For this reason most working expert systems are developed using programming environments.
A more sophisticated approach to automating KA is the creation of interactive intelligent editing programs. These programs have three major functions:

1. fix bugs in an existing knowledge base;
2. check for missing rules; and
3. check for inconsistency when new rules are entered.

A well known example of such a system is TEIRESIAS. This is a program designed to assist a domain expert in debugging and expanding the knowledge base of MYCIN via a high level dialog in a restricted domain of natural language. The principles that govern the operation of TEIRESIAS are given in the following. First, it performs KA in context. For example when an error occurs in a solution given by the system TEIRESIAS presents to its user (who is a domain expert) all the facts and knowledge the system used to reach the solution. It then asks the user to locate the area of fault. Second, TEIRESIAS builds up expectations concerning knowledge to be acquired. This allows the program to suggest a specific form of the rule that applies to the case in which the error was found. Third, TEIRESIAS dynamically forms a model of the existing rules. New rules are matched against the model to check for consistency. In effect, by using TEIRESIAS the user can codify knowledge directly into the knowledge base without the assistance of a knowledge engineer.

More recent efforts in building intelligent knowledge-base editing programs include APARSER, CHPLL and KLASSIC. APARSER is a program to assist non-Lisp programmers to add knowledge to the ANALYST system (Bonasso, 1985). CHPLL and KLASSIC are programs that help domain experts to add knowledge nodes to hierarchical frame-oriented knowledge-bases (Finin, 1986).
All the intelligent editing programs (IEP) mentioned above have some common limitations:

1. each has been designed to work with an existing, largely codified knowledge-based system where KA for the initial knowledge base has been carried out manually; and

2. considerable effort is required to construct each IEP which usually is not transferable to a different system. The use of IEP is, therefore, not cost-effective.

The search for other methods to shift the task of KA from human processes onto a computer and to capture the front-end knowledge directly from domain experts have led to two major trends of research

1. inductive knowledge acquisition; and

2. development of interactive knowledge acquisition systems.

The complexity of real world problems sometimes makes it difficult for domain experts to articulate their own knowledge. It is even more difficult to put thoughts into rules. For some problems the amount of information needed to provide a solution is extremely large. In others, the knowledge is not defined sufficiently to put into rules. In such cases the expert may find it easier to provide some examples based on past experience and machine induction may then be applied. Induction is a process of transforming scattered, ill-structured pieces of specific knowledge into a more general, well-defined form that can be used efficiently (Cohen and Feigenbaum, 1982).

Inductive knowledge acquisition proceeds in stages. Initially, a set of examples is obtained from a human expert or from real-world observations. The objects to be
classified must be explicitly enumerated. Next, the expert helps to decide what features of these objects are important for decision making. These features are referred to as attributes (also known as characteristics or factors). Based on values of the attributes the expert will provide a decision (also known as class) for each particular example. Information is usually arranged in a tabular form with explicit values of attributes and corresponding classes. An example of this taken from Carter and Catlett (1987) is shown in Table 2.2.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>account balance</td>
<td>employed monthly expense</td>
</tr>
<tr>
<td>bank 700 yes 200</td>
<td>accept</td>
</tr>
<tr>
<td>bank 300 yes 600</td>
<td>reject</td>
</tr>
<tr>
<td>none 0 no 400</td>
<td>reject</td>
</tr>
<tr>
<td>other institution 1200 yes 600</td>
<td>accept</td>
</tr>
<tr>
<td>other institution 800 yes 600</td>
<td>reject</td>
</tr>
</tbody>
</table>

An inductive algorithm will be applied to infer general rules from the given information. These rules may not necessarily represent how the expert thinks but they capture the most common characteristics of the available set of examples. To evaluate the rules it is often necessary to use a different set of examples within the same problem domain.

In inductive KA there are important factors that directly influence the quality of the final induced rules. First, the chosen examples need to be general enough to represent the problem domain. Second, it is not always an easy task to enumerate a good set of attributes from given examples. In fact, this is found to be the most critical step in inductive KA (Hart, 1986; Brew and Catlett, 1985). Poor choice of attributes may lead to
unsatisfactory rules. Finally, the inductive algorithm used should be powerful enough to generate desirable outputs. An example of such algorithms is the Iterative Dichotomiser 3 or ID3, originally developed by Quinlan to built the simplest tree that correctly represents a given set of examples (Michie, 1979). This algorithm has recently been improved to account for multi-valued attributes (Quinlan, 1985) and to cope with unknown data (this version of ID3 is known as C4, Quinlan et al, 1987).

There are many well known classical systems that have been developed using inductive techniques. For example AM (Lenat, 1976) is a program that replicated the discovery of several concepts and conjectures in number theory. BACON (Langley, 1977) is a program that can discover mathematical relationships such as Kepler's law and Ohm's law. Meta-Dendral (Buchanan and Mitchell, 1978) is a program that discovers rules describing the operation of a mass spectrometer. AQ11 (Michalski et al, 1980) produces disease-diagnosis rules for soybean diseases, and EURISCO (Lenat, 1983) is a program that attempted to learn new heuristics. More recently, there are expert systems developed using ID3 and C4. Examples include: SA, the expert system for trouble-shooting a smelter (Brew, 1985), the expert system for assessing credit card (Carter and Catlett, 1987), the expert system for botanical Key (Colier, 1987), and many other systems developed using shells such as ExpertEase, RuleMaster or 1stCLASS.

Compared to previous approaches to KA, inductive techniques have relieved the KE from many tedious tasks, cutting down the development time, and providing a promising solution to the KA bottleneck. In addition, machine induction is an established branch of Artificial Intelligence (AI) and its use has attracted attention from many AI researchers as well as AI practitioners.

However, all systems mentioned in this section have a common characteristic: they require a set of known examples. In many situations it may not be possible to meet that...
requirement. Moreover, current inductive algorithms only allow simple data types to be used, induced rules can be difficult to understand and hard to modify, and the general applicability is relatively limited (Sammut et al, 1986). These problems have motivated researchers to combine both inductive and deductive methods for KA (VanTerheyden and Chalcraft, 1987, Buntine and Stirling, 1988). These works, however, are still in a experimental stage. At this stage it is not feasible for inductive KA to replace other conventional KA techniques.

The last and, perhaps, the most interesting area of KA research is the search for better automated knowledge acquisition systems (AKAS). These systems can assist in explicating front-end knowledge directly from domain experts via interactive dialogues. Some are based solely on fixed algorithms without the understanding of natural language (ETS, MORE). Others have been designed to incorporate this capability but only within a narrow domain (For example, INKA).

The most simple system is probably the program that elicits a decision tree from an expert and produces a solution plan that recommends an action for all anticipated contingencies (Leal and Pearl, 1983). This approach centres on observed similarities between an elicitation dialog and a heuristic search on game trees which is a common practice in AI programs. This system is domain independent and does not understand natural language. However, conversations seem to follow a natural discourse due to the simplicity of the structure underlying the decision tree. Regardless of its limited capability the system exhibits three important characteristics:

(1) interactive dialog with a domain expert;
(2) elicitation of front-end knowledge; and
(3) domain independent.
Another approach to develop AKAS was taken by Boose (1984) when he designed the Expertise Transfer System (ETS) which constructs and analyzes an initial set of heuristics. This system is based on a classic theory in psychology, known as the Theory of Personal Construct (Kelly, 1955), which suggests that each individual seeks to predict and control surrounding events by forming theories, testing hypotheses, and weighing experimental evidence. Techniques applied are Repertory Grid (Table 2.1), laddering, triads and dyads, etc. ETS applies these techniques to KA using a pre-determined form of interactive dialogue. The end result is a knowledge base that can run on KS-300 or OPS5.

This approach is strongly influenced by research in psychology and cognitive science (Boose 1984). Apart from some inherent disadvantages of the underlying theory (Boose, 1985a) and lack of a capability to understand natural language, ETS has remarkable potential. It has been modified further in an attempt to deal with multiple knowledge sources (Boose, 1985b).

Another approach adopted by Kahn and McDermott of Carnegie-Mellon University is MORE (Kahn, Nowlan and McDermott, 1985), which has evolved from practical experiences in manual development of expert systems. MORE provides a mechanism for interviewing domain experts and takes a model-theoretic approach to KA, especially diagnostic knowledge. Prior to creating MORE Kahn had been involved with the construction of an expert system (called MUD) for diagnosis of drilling fluid (Kahn et al, 1984). As a result MORE may be thought of as a product of experiences gained in the development and application of MUD.

MORE uses a qualitative model of causal relations to guide the interviewing process. Such a model provides a structure for representing causal knowledge. The primary value of the system lies in its ability to identify the type of knowledge that is
likely to be diagnostically significant. This is achieved by manipulating the positive-
support and negative-support weights assigned by the domain expert to symptoms and
hypotheses. By formulating questions in this way, MORE makes the most effective use
of the expert's time. The major limitation of this system is that it has been designed
specifically for acquiring diagnostic knowledge. In this thesis a similar approach was
used to develop AKAS, an automated knowledge acquisition system (Chapter Five).
Other programs similar to MORE include MOLE (Eshelman and McDermott, 1986),
TKAW (Kahn, Breaux, Joseph and DeKlerk, 1986), SALT (Marcus, 1986) and OPAL
(Musen, Fagan, Comb and Shortliffe, 1986).

In addition to the major approaches discussed above there are other trends in
developing programs that possess the capability of understanding a narrow domain of
natural language. For example, INKA (Phillips et al, 1985) is the natural language
interface to facilitate KA during the development of an expert system for electronic
instrument trouble-shooting. INKA works with a subset of English called Generalized
Language for Instrumental Behaviour (GLIB) in which only valid statements in GLIB can
be generated. The program interacts with an expert and translates acquired information
into production rules. INKA may be seen as a tool which reduces, rather than eliminates,
the involvement of the KE in the KA process. INKA is domain dependent and hence may
not be a cost-effective approach unless a large number of similar expert systems in the
same domain is to be built.

In the following section, a summary of the contributions of this thesis is given.
2.4 Summary of Contributions

The contributions of this thesis are briefly summarised below.

It was noted that for many Expert System (ES) applications the use of a single expert as knowledge czar is not sufficiently reliable. Knowledge acquisition (KA) from multiple experts, however, involves a much more difficult and time consuming process. In Chapter Three, it is shown that if a single domain expert may be modelled using the concept of a human Information Processing System, such a model may also be generalised to represent multiple domain experts. It is further shown that this concept of modelling experts may be useful when it is used in conjunction with an appropriate methodology for KA from multiple domain experts. This approach gives an insight into the works of Mittal and Dym (1985), McGraw and Seale (1987a). The model helps the knowledge engineer to conceptually separate different domains of expertise and to understand their interrelationship. The Chapter also introduces a methodology which has been tailored for the acquisition of knowledge from domain experts from different disciplinary backgrounds. The methodology emphasises not only the initial stage of selecting and training of domain experts but also the use of appropriate techniques for information retrieval in the later stages.

Chapter Four reports the application of the concept of modelling multiple experts and the methodology for knowledge acquisition described in Chapter Three to the development of DESPLATE, an expert system for the diagnosis of faulty shapes of steel plates, developed for the Slab and Plate Products Division of BHP Steel International Group. This system requires inputs of domain experts from several disciplines, namely, electrical engineering, mechanical engineering, rolling mills technology and operations. An expert system shell, called LEVEL5, is used in the development of DESPLATE.
Apart from knowledge acquisition issues, DESPLATE has many desirable features to facilitate the representation of diagnostic knowledge. In particular, knowledge is classified into several levels with different priorities, this classification is based on the frequency of occurrence of a particular fault and on the time required in fault findings. The use of both backward chaining and forward chaining is investigated and implemented to provide closer representation of how a Plate Mill expert performs diagnostic tasks.

The handcrafting approach to KA is inherently subject to numerous disadvantages. It is thus desirable to investigate the possibility of automating parts of such a process. In Chapter Five, the design and implementation of AKAS is presented and discussed. AKAS is an automated knowledge acquisition system which directly interviews a domain expert for diagnostic knowledge. It then generates production rules to represent the knowledge and forms a knowledge base for an expert system shell such as LEVELS or Insight 2+. The design of AKAS was based on a realization that diagnostic knowledge may be represented in the form of a diagnosis decision tree. The program does not incorporate natural language understanding but its conversations with users seem to follow a natural discourse.
CHAPTER THREE

Modelling Multiple Experts
for
Knowledge Acquisition
3.1 Introduction

It is found from Chapter two that for many applications of Expert Systems the use of a single domain expert is not adequate. As problem domains become more complex, knowledge possessed by a single domain expert tends to become discrete and confined to a narrow subset of the tasks. The expertise required for an expert system in such cases is normally distributed among several experts. There are different situations where difficulties may occur when acquiring knowledge from multiple experts (ME). In some cases, all domain experts are from a similar discipline, for example all pathologists. In other cases, some experts may possess different disciplinary backgrounds from others, for example, maintenance electrical engineers and mechanical engineers working on the same manufacturing site.

Problems with multiple experts from a single discipline are normally related to the inter-relationship between each individual. Different experts may have conflicting ideas and different problem-solving strategies which may equally be valid. To deal with multiple experts from different disciplines, in addition to the difficulties mentioned above, sub-areas of expertise in which different experts may specialize, concepts shared between experts and procedural knowledge that are related to each other in some hierarchical order need to be identified.

This chapter presents a concept of modelling multiple domain experts from multiple disciplines for knowledge acquisition. The model consists of a memory and a cognitive processor, based on the model of a human information processing system (IPS) of Newel and Simon (1972). From the KA perspective, the model represents two major characteristics of a domain expert

(a) a storage of domain specific knowledge; and
(b) knowledge about how the expert manipulates knowledge.
This model is similar to Nii's model of her domain expert (Feigenbaum and McConduck, 1983). In this case, the concept has been generalized for the case of multiple experts. Such a model is intended to show both the commonality of approach and differences in specialization between the involved multiple experts. Shared expertise can be used as community knowledge and different specialists are identified for further consultations if required. This is similar to observations made by Mittal and Dym (1985).

However, the model alone is not sufficient to guide the knowledge engineer in acquiring knowledge from multiple experts, there is a need for an appropriate methodology. In this chapter a methodology for knowledge acquisition will also be presented. This methodology was developed to facilitate the KA process for DESPLATE, a diagnostic expert system for faulty plan view shapes of steel plates (Cung and Ng, 1988a). The methodology is based on the same approach adopted in McGraw and Seale (1987a) except that it focuses on KA from multiple experts from diverse disciplines.

3.2 Model Creation

3.2.1 Model of an Expert System

Modelling is a familiar concept in all fields of science that require analytical approaches. Models are often used to clarify complicated concepts or to simplify tedious tasks. For example, in electrical engineering the hybrid-pi model of a transistor is used to simplify the analysis of transistor circuits. Similarly, in knowledge engineering a model can also be used to represent the internal structure of an expert system.

Figure 3.1 shows a simple model of an expert system consisting of two separate parts: an inference engine and a knowledge base. The knowledge base contains the
expert's knowledge in the form of facts and rules. The inference engine contains both inference strategies and controls that experts use to manipulate facts and rules (Davis, 1977; Hayes-Roth et al, 1983; Pau, 1986).

![Figure 3.1 Model of an Expert System](image)

The advantages of this structure are:

(a) the knowledge base can be expanded easily and therefore is highly flexible;

(b) users can view the line of reasoning and contents of the knowledge base which lead them to a better understanding of the decision making process.

### 3.2.2 Model of a Human Expert

Models of human intelligence vary from one discipline to another, each model can serve as an analytical tool for its users. Biologists view man as a brain which is constructed of neurons. Classical psychologists only study man's external behaviour as they believe internal thought processes cannot be investigated. Cyberneticists view man as a machine with many feedback networks which can be modelled mathematically. Similarly, if knowledge engineers can view man as an information processing system and study how human experts acquire and exhibit their expertise, this approach would lead to better methods of knowledge acquisition.
A simple model of a human information processing system (IPS) is shown in Figure 3.2. There are three major subsystems: the input, the output and the central cognitive system. This model was first developed by Newel and Simon (Newel and Simon, 1972) and later adapted by cognitive scientists (Harmon and King, 1985).

The cognitive processor has two major functions:

(a) encoding information into memory; and

(b) retrieving information from memory.

Information from the outside world is input into temporary buffer memories via sensors such as eyes and ears. The cognitive processor then transfers selected information into a short term working memory. This selection process is ordinarily referred to as paying attention.

Knowledge is stored within a human expert's long term memory (LTM). One way to conceptualize this is to think of a vast network of clusters of symbols called chunks. These chunks have been accumulated from years of experience in certain areas. There is a
finite time interval required before a new chunk from short term working memory can be added to LTM (Simon, 1969; Harmon and King, 1985).

The access time to retrieve information from LTM is relatively short compared with the storing time of new memory chunks. When a chunk is activated, it forms part of the short term working memory in which information can be accessed simultaneously. Only a limited number of chunks can be activated at any particular instance. When new chunks are activated, old ones will fade away. This is reflected in man's ability to concentrate on only a limited number of issues at any particular instance.

Finally, the cognitive processor transfers the information to the output subsystem which in turn activates muscles and other internal systems. This results in some observable activities.

From a knowledge acquisition perspective, in which the main concern is the process of information retrieval, two major attributes that characterize a human expert are:

(a) a memory storage which represents domain specific knowledge possessed by the expert; and

(b) a cognitive processor which symbolizes the ability to manipulate and articulate possessed knowledge.

The concept is depicted by Figure 3.3:
This model resembles the one shown in Figure 3.1, where the long term memory is associated with the knowledge base and the cognitive processor is associated with the inference engine.

3.2.3 Model of Multiple Experts

The model presented in the previous section can be generalized to cover the case of multiple experts. The idea is to give the knowledge engineer a better tool with which to analyze the knowledge acquisition process.

Consider an example of a problem which involves two experts, A and B, who come from two different disciplines. As A and B have had experience working together on different aspects of a problem they must possess two types of expertise

(i) Isolated expertise: the area of expertise known and practised by one of the experts of which the other has no significant knowledge.

(ii) Related expertise: the area of expertise that involves the knowledge and experience of both experts, each of whom can
make relatively important contributions to the solution.

From the previous section we know that the knowledge of each expert can be represented by a network of chunks embedded in long term memory. A pictorial representation of the relationship between the knowledge that is required from A and B to solve a set problem is shown in Figure 3.4. The degree of overlap indicates the amount of collaboration needed to solve the problem. This in turn gives an insight into which approach would be most appropriate for knowledge acquisition. For instance, if the overlap is significant the multiple experts could be treated as a group, or alternatively, if the overlap is negligible, information can be acquired from each expert on an individual basis.

```
Figure 3.4 Relationship Between the Expertise of A and B
```

Experts have different ways of acquiring knowledge which results in different ways of retrieving information as well as expressing opinions. Some acquire knowledge through formal and theoretical training. Others acquire it only through practical experience and rules of thumb. Some may feel comfortable with a systematic, analytical approach to a discipline, while others prefer a more intuitive approach. Consequently, it is not easy to analyze the relationship between the type of cognitive processors of different experts.
There are, however, similar operations that take place when human beings solve problems. For instance, recognition of a certain fact is equivalent to the activation of a specific memory chunk. A specific fact will trigger related facts and this is equivalent to other memory chunks being activated. The "recognize-act" or "if-then" chain reaction would finally lead to some conclusions and perhaps to some observable responses (Harmon and King, 1985). This form of cognitive behaviour can be represented using a production systems (Newel and Simon, 1972). There are other representation schemes that have been developed to simulate the way human experts manipulate their knowledge. Examples include logic, semantic networks, frames and scripts (Cohen and Feigenbaum, 1982). Consequently, the differences between one expert and another are the knowledge that one possesses and the way the expert articulates his own knowledge. If the former is represented by memory, and the later can be represented by a cognitive processor, the model of the multiple experts can thus be represented by Figure 3.5.

![Figure 3.5 Model of Multiple Experts](image)

It is believed that an optimal level of performance of multiple experts would perhaps be compared to that of a single 'super expert' who possesses the abundant knowledge of all the individuals and, yet, is able to manipulate the knowledge as effectively as a single person. Therefore, to build an ES with a high level of performance,
the knowledge engineer should carefully analyse the relationship between the knowledge each expert possesses and their ability to convey domain specific information. When this is achieved the KA process would become much more efficient.

The model of multiple experts is a conceptual one. Such a model can be developed iteratively throughout the process of KA, and it is used to guide the knowledge engineer in selecting a correct methodology for knowledge elicitation. In the following section we describe a methodology that has been developed to handle KA from multiple experts from diverse backgrounds.

3.3 A Methodology For Knowledge Acquisition From Multiple Experts From Multiple Disciplinary Backgrounds

There are many different approaches to the problem of KA from multiple experts (Bobrow et al, 1986; McGraw and Seale, 1987a; Mittal and Dym, 1985). The methodologies presented in these papers do not readily lend themselves for KA in general as the experience acquired from each expert system is quite unique. A KE should always carry out preliminary analysis of each case before applying a suitable methodology.

The following section describes a methodology for knowledge acquisition from multiple experts with different disciplinary backgrounds. Although based on the approach given in McGraw and Seale (1987a) this methodology emphasises KA from multiple experts from diverse disciplines. The concept of modelling multiple experts will be discussed in relation to the methodology and it will be shown that this concept can help to improve the effectiveness of the KA process.
The methodology consists of four stages. Firstly, we discuss factors concerning identification of a model for multiple experts. Secondly, we consider important attributes in forming the multiple expert team (MET). The most important part of the methodology is considered in the third stage where details of the KA techniques are described and finally the process of verification of the acquired knowledge will be discussed.

### 3.3.1 Identification of a Model for the Multiple Experts

The first step in building an Expert System is to select competent domain experts. The number of experts is dependent on the nature of the application. Some applications only require the expertise of a single domain expert, others require the collaboration of more than one. According to recent trends in applications in manufacturing engineering, Expert Systems usually require inputs from many domain experts, some of whom may possess different disciplinary backgrounds.

Various techniques have been proposed for selecting domain experts. These range from interviewing each expert individually to giving a sample problem to a group of nominated experts and using their performance in this problem to select the most suitable persons (Mittal and Dym, 1985; McGraw and Seale, 1987). In dealing with experts from different disciplinary backgrounds the important factors are:

(i) **Domain Expertise:** the person must possess a level of knowledge commensurate with other recognized colleagues in the same field; and

(ii) **Communication Ability:** the person must be willing to co-operate and be able to communicate relatively well with the knowledge engineer and other experts.
During this initial stage the knowledge engineer and the selected domain experts should work together to identify the problem under consideration and to set an overall goal for the project. This defines specific limits on the required domain expertise (Hayes-Roth, 1983). If ES shells are to be used, issues concerning desirable features such as user friendliness, accessibility and portability should also be introduced to the experts. If convenient, sample Expert Systems developed using selected shells should also be demonstrated. Simple explanations of how these systems work would normally result in greater enthusiasm from the experts.

During early meetings with domain experts, it is more important for the KE to pay attention to the way the experts manipulate their knowledge than to detailed information. For example, there are people who exhibit the 'modus ponen' type of inference quite explicitly while others tend to tell lengthy stories of events that happened a day before. This type of information helps the KE to form a good picture of how each expert approaches a given problem and to choose the most suitable knowledge representation and inference method to match the experts' behaviour. Consequently, the representation mismatch between the way experts state their knowledge and how it must be represented in a knowledge base can be minimized.

Through identification and separation of the domain expertise, the KE can identify the sub-domains of expertise that each expert seems to possess and may gradually construct a model for each expert and then derive the model for the multiple experts (refer to section 1.2 and 1.3, respectively). The complete model showing the interactions between the experts' knowledge will only be obtainable during the actual process of acquiring knowledge. However, the information gathered at this stage would help the knowledge engineer to decide whether to use the experts individually or in a group. For instance, the amount of 'related expertise' required to solve a set problem would give a
good indication of whether to treat the experts as a group, or alternatively, to acquire information from each expert individually.

A relatively simple model of the domain experts should be formed by the end of this stage. Used as an analytical tool the model helps the KE in choosing the most effective approach to knowledge acquisition. The model can be developed further in the stages which follows. Its applications will be discussed later. The next step is to form a multiple expert team (McGraw and Seale, 1987) and to prepare the team for information retrieval.

3.3.2 Formation of a Multiple Expert Team (MET)

In a typical knowledge acquisition process, the sequence of questions asked by the KE would fall into the 'WHAT-HOW-WHY' pattern, that is:

a. Identify specific problem(s) to be solved,

e.g. "What do you want to diagnose?"

"What do you want to design?" etc.

b. Extract problem solving techniques of experts

e.g. "How do you diagnose 'this'?"

c. Finally, retrieve some explanations of why certain conclusions are made

e.g. "Why do you use this test to detect that fault?"

Similarly, to maximize the chance of retrieving the most relevant information, MET members must be prepared to know what information they are expected to provide, how the information would be acquired, and preferably why the acquisition process should be carried out in the proposed method. It is advantageous to focus MET members' attention
on relevant issues before the actual process of information retrieval. If the experts are well aware of what they are expected to contribute they may be more active not only in answering specific questions but also in extending answers to relevant issues.

Clarification to the MET members of all techniques to be used in the next stage is important as it encourages them to ask questions such as why selected techniques should be used. This has a twofold purpose: (i) to gain consensus from the experts; and (ii) to create a carefree atmosphere necessary for information retrieval in the next stage.

At the end of this stage, the MET members are ready for information retrieval. There is a direct trade off between efforts spent in this stage and the time required for the subsequent stages. The more clearly the domain experts know what they have to supply, the less number of iterative steps will be needed to refine the knowledge base.

3.3.3 Information Retrieval from the Multiple Expert Team

During this stage, the multiple expert team members are brought together to provide specific problem solving expertise. Essential questions to be answered are:

1. What does each expert know about solving the particular problem at hand?
2. How do MET members normally solve a given problem: individually or together?
3. Why is the problem solved the way the MET members suggest?

The process is divided into two phases.

Phase one: Brainstorming (Osborn, 1953)

Initially, Question 1 will be answered using a technique called 'brainstorming'. This is designed to stimulate ideas and to encourage the experts to provide as
much information as possible on the subject under consideration. Redundancy in the information obtained could be useful.

The rules for brainstorming are as follows:

Rule 3.1 Brainstorming

(i) Prior to the meeting, the knowledge engineer provides each member of the multiple expert team (MET) with the problem(s) to be solved.

(ii) At the meeting, the MET members are invited to call out their ideas. No evaluation is made during this period.

(iii) When the rate of presentation of new ideas seems to fall off, the knowledge engineer can start revising the information with the MET members and may then proceed to the next phase.

Phase two: Group Evaluation Technique (GET)

The underlying principle in this stage is to concentrate only on problems that require elaborations of more than one domain expert. The major activities involved in this phase are:

(i) The knowledge engineer presents to the MET members all the available information. This information includes techniques or strategies used by the multiple expert team members to solve the problem under consideration.

(ii) The MET members classify the information into two major groups:
a. Isolated Expertise: Information in this category will be dealt with outside the meeting. The concerned experts will be approached individually using conventional methods (Hart, 1986; McGraw and Seale 1987a and 1987b).

b. Related Expertise: The involved experts are invited to join an evaluation process.

The model of the experts may be further updated to keep a record of the related expertise identified during this process. Knowing the ability of each person involved the knowledge engineer can carefully conduct the process at the most appropriate level to suit those involved. The rules for Group Evaluation Techniques are as follows:

Rule 3.2: Group Evaluation Technique

(i) The knowledge engineer specifies the information to be evaluated.

(ii) At the first meeting, the MET members are asked to nominate the attributes that must be observed when evaluating the results. The knowledge engineer should lead a discussion in such a way to result in a consensus among the experts.

(iii) The inter-relationship between the information should then be investigated. Relationships such as concurrence, precedence, similarity or conflict between all the information supplied needs to be sorted out thoroughly.

(iv) The evaluation process can be carried out based on the results of (i) and (ii).
(v) Special cases which do not conform to (ii) should also be considered.

(vi) The knowledge engineer records the results as well as the lines of reasoning. Questions may be asked to clarify certain points.

(vii) Steps (iii) to (vi) can be carried out for different subsets of the information obtained in Phase 1.

(vii) At the end of each session, the acquired information must be revised by the MET members. This information will be subject to verification in the following stage.

### 3.3.4 Verification of the Acquired Knowledge

In this stage the information acquired is exposed to the experts, both participants and non-participants, for thorough verification. The aims are twofold: (i) to enhance the validity of the information; and (ii) to allow domain experts to familiarize themselves with the way their knowledge is represented explicitly, thereby reducing representation mismatch.

Initially the acquired information should be transcribed into a readable form, for instance a 'knowledge tree', and then given back to each multiple expert team member for assessment. Collaboration between the knowledge engineer and each individual at this stage would be extremely useful in achieving the following goals:

a) corrections to incorrect terminologies or misinterpretations of certain ideas and concepts;

b) acquisition of more detailed explanations; and

c) identification of missing or incomplete subsets of information.
The model of the experts showing the specialized areas of expertise of each expert and interactions between these areas would be very valuable at this stage. The knowledge engineer can use this to gain a good understanding of each expert and to develop a more effective approach to later stages of knowledge acquisition.

Feedbacks from experts are subject to group evaluation as discussed in Stage 3. This interactive process can be carried out until the information obtained is satisfactory. Acquired information can now be exposed to the 'consultants' (refer to Stage 1) for further assessments. Feedback from these, both contributions and criticisms, will then be subject to discussion using GET. The amount of time involved in the whole process would depend upon the working conditions of the multiple experts, the working conditions of the knowledge engineer, and most importantly, the ability of the experts in expressing themselves logically and explicitly. Poor expression will lead to incorrect interpretation which will eventually result in more iterative steps.

It would not be preferable to alter the way the experts present their opinions. Nevertheless, it would be advantageous in the long run for experts to learn how to convey their knowledge rather than for the knowledge engineer to interpret every single bit of their knowledge. This could partially be achieved if, after an initial analysis of the experts' problem solving strategy as well as their ability to convey their knowledge, the knowledge engineer consistently uses an appropriate method to convey information to and to receive information from the domain experts.

At the end of this stage the knowledge base should be ready for encoding into an Expert System.
3.4 Conclusion

A concept of modelling a group of domain experts from multiple disciplines for knowledge acquisition has been presented. Discussion centred on how the model is applied to a particular methodology for KA from multiple experts, and a methodology has also been introduced. It is shown that such an approach can assist the KE to speed up the KA process.

The development of this approach to KA was partially initiated in the course of developing DESPLATE. The modelling of multiple experts together with the methodology presented above was found to be a success when it was applied to develop DESPLATE. In the next chapter, a detail description of this project is presented.
CHAPTER FOUR

DESPLATE

The Diagnostic Expert System

for

Faulty Plan View Shapes of Steel Plates
4.1 Introduction

This chapter describes the development of DESPLATE (Cung and Ng, 1988a), an expert system for the diagnosis of faulty plan view shapes of steel plates in the Plate Mill of BHP Steel International Group, Slab & Plate Products Division, Port Kembla, Australia.

The Plate Mill is an old mill which requires professional experiences, or so-called heuristics, to maintain satisfactory operation. During the last decade the mill has undergone significant upgrades through application of modern state-of-the-art technology. The number of experts is small while the need for their expertise is great. Thus, there is a strong incentive to develop an expert system which will make their valuable expertise available to less experienced operators and hence assist them in their fault diagnosis of the mill.

Due to the complex nature of the environment of the Plate Mill the project required contributions of domain experts from several disciplines. These include electrical engineering, mechanical engineering, rolling mills technology and operations. Knowledge acquisition from multiple experts from different backgrounds is not a trivial process. In general, it requires considerable time as well as the use of a correct methodology. The concept for modelling multiple experts and the methodology outlined in chapter 3 for KA were used in developing DESPLATE. Practical aspects of this development are described in this chapter.

Section 4.2 presents a brief overview of the production process at the Plate Mill. While in Section 4.3 problems that motivated the development of DESPLATE are described, Section 4.4 discusses a number of issues that arose in the course of developing DESPLATE. In particular, the classification of knowledge and the use of both forward
and backward chaining techniques are described. Finally, Section 4.5 presents discussion on the practical aspects of the approach to knowledge acquisition.

4.2 The Plate Mill

To understand the extent of the problem of diagnosing faults in the Plate Mill it is necessary to have a brief overview of the setting and the production process in the mill. The Plate Mill first started production in 1964. It was a traditional rolling mill heavily dependent on operators to produce high quality products. However, during the last decade the mill has been gradually upgraded through installations of new equipment, computer control systems, new control pulpits and by the introduction of new rolling practices. The function of the mill is to roll re-heated slabs into plates of scheduled thickness and dimension (Kelly, 1984). The final products are of rectangular plan view shapes having a width of up to 3300mm and a thickness of 180mm.

A schematic layout of the mill is shown in Figure 4.1. The mill had two major rolling stands, denoted No.1 and No.2. These are equipped with large rolls (up to 508mm in diameter) which are driven by high power electric motors (up to 4470kW). Roll gaps and forces were maintained by large hydraulic cylinders.

![Figure 4.1 Schematic Layout of the Plate Mill](image-url)
Typical operation of the mill can be described as follows: Slabs coming from the reheating furnace initially go through the No.1 Stand whose function is to reduce the slabs to the required width. The slabs proceed through the No.2 Stand, where finishing passes are carried out to produce required slab thickness, and finally through the Hot Plate Leveller which gives each plate a smooth finishing surface.

The mill is controlled by sophisticated computer systems with provision for manual over-ride. A top down picture of the control system is outlined in Figure 4.2. The operation of the control system is as follows: after the Mill Control Computer (MCC) receives schedules of products from the main frame computer it first performs all necessary calculations such as mill gap settings, number of passes, etc., then passes the results of these calculations to the Automatic Gauge Control (AGC) computers which directly controls the hydraulic systems in both Stands. The AGC also controls other equipment such as motors and decoders and it communicates with these through Programmable Logic Controllers (PLC). The MCC also communicates with an auxiliary computer which is used to control the Hot Plate Leveller and the Cooling Bed.

![Diagram of the Computer Control System of the Mill](image-url)
The mill has been designed for automatic operations. However, manual over-ride is also provided to allow operators to maintain control over situations in which computers cannot operate effectively. These include unexpected conditions such as temperature deviations in the furnace, incoming faulty slabs, etc.

While the implementation of the computer control system has greatly improved the productivity of the mill, as well as the quality of products, operation has also become more complex and thus faults are harder to identify. The impact of new technology is yet to be fully comprehended as only a limited number of senior staff are fully knowledgeable of the changes in the system. With their invaluable experience and knowledge acquired over years of practice, these people have quickly absorbed the new concepts, adapted themselves to new changes and are considered to be experts in their fields.

4.3 The Specific Problem

After re-heated slabs are rolled into plates they are transferred to the Plate Finishing Area where they are cut to customized dimensions. The ideal plan view shape of a plate which minimizes the waste is a perfect rectangle. However, perfect plates rarely exist. There are unwanted operating conditions that cause the products to fall into one of the following faulty categories: camber, off-square, taper, concave or convex end, etc. Figure 4.3 illustrates exaggerated plan view of the plates for these conditions. Among them camber occurs quite frequently and is the most difficult shape to diagnose. In many occasions, as faulty plates fail to meet certain requirements they have to be cut into smaller dimensions and re-directed to another customer. This is a costly exercise and is of great concern to the Company.
Because of complex mill settings, problems may arise from various sources such as:

- electrical failures,
- mechanical breakdowns or wear,
- operational errors,
- pre-rolling conditions such as incorrect temperature, faulty shapes, variable slab thickness before rolling, etc.

Consequently, fault diagnosis often requires the expertise of experts from different disciplines. In the past, attempts have been made to reduce the fault diagnosis time. However, despite large budget allocations and considerable effort spent in preparing documentation for the mill, it was found to be very difficult to present the fault diagnosis in a suitable form. This was because once a fault had occurred there was little or no time to refer to the documentation. In such situations, experts have to be called in regardless of time. This was often inconvenient and time consuming, especially if the fault occurred late at night.
A remedy to the above situation is to have a program that captures the knowledge of experts in a certain field and makes this knowledge available to less experienced people within that field. An expert system to assist in fault diagnosis is thus a sensible solution. If the expert system is equipped with diagnostic knowledge of recognized domain experts then it can become a very valuable tool in helping to reduce stoppage time and thus improves productivity. DESPLATE is such a system and was developed to assist in locating possible causes for a particular faulty shape and to suggest appropriate remedies.

4.4 DESPLATE

In this section an overview of DESPLATE is presented initially. This is followed by a description of the knowledge base architecture, an examination of the control structure of the system and finally a description of the user interface.

4.4.1 A System Overview

The structure of DESPLATE structure is shown in Figure 4.4. It consists of knowledge base, an inference engine, and an interactive user interface.
The knowledge base contains the domain experts' heuristics in the forms of facts and production rules. It is sub-divided into six separate knowledge bases. Each one corresponds to one particular faulty shape. The inference engine uses knowledge stored in the knowledge bases to perform diagnostic tasks by emulating the reasoning process of human experts. A combination of forward-chaining and backward-chaining has been applied to improve the system performance. Graphics routines are also used to clarify information that may be too difficult to describe in words. The use of graphics has certainly enhanced the expressiveness and user-friendliness of the user interface.
4.4.2 The Knowledge Base

A diagrammatic view of the available knowledge bases is depicted in Figure 4.5.

![Diagram of Available Knowledge Bases]

Figure 4.5 An Overview of the Available Knowledge Bases

An initial visual aid is provided by dispatcher knowledge base which allows the user to identify the type of faulty shape of a the given plate. The system software then calls the appropriate knowledge base which pinpoints possible faults and recommends (on most occasions) appropriate corrective measures.

To minimize the time required to diagnose possible faults the following two principles were used in the conceptual design of the system:

1. diagnostic knowledge was arranged in such a way that the time required to search for faults was minimized. This was achieved in part by carrying out simpler tasks before the more time consuming generate and test procedures were done.

2. priority was given to faults that occurred most frequently.

Implementation of these principles required that the knowledge base for diagnosis of camber patterns be classified into seven hierarchical levels. These levels included:
Visually Detectable Faults, Operational Errors, Shapes Before Rolling, Mill Zeroing-Test, Mill Conditions-Mechanical, Modulus Test, and Mill Conditions-Electrical. These levels were determined by the involved domain experts. An example of the knowledge levels is illustrated in figure 4.6 where the following abbreviations are used:

Obs: Observations, carefully chosen facts or symptoms that helped the user to identify certain 'families' of faults.

Tests: Special tests or actions normally carried out by experts to generate further information necessary to identify a particular fault.

Faults: Probable causes of the particular problem which is being solved.

Figure 4.6 An Example of the Knowledge Levels

In a typical consultation session (see Figure 4.6) DESPLATE first prompts its user with a set of possible observable facts that may have been noticed by the user prior to or during the session. Step by step responses from the user results in DESPLATE coming up with a set of most likely causes and appropriate corrective actions. The system may then ask the user to perform certain tests or measurements and to provide results to
DESPLATE. Analysis of the results is undertaken and appropriate interpretation is provided. If no positive symptoms are being found in Level 1 the system continues to search for faults in Level 2. However, the system may allow the user to jump from one level to another (lower priority level) if there are positive indications that a particular fault can be found in the later level or some tests in the later level can be used to confirm findings in the former. For example

If "Roll chuck moves from side to side with change in direction of rotation" is identified at Level 1, "Mill Zeroing Test" at Level 5 may be required to determine if "Roll cross-over" has occurred.

Apart from the concept of knowledge levels, part of the knowledge base for diagnosis of off-square patterns were also partitioned. *Meta-rules* were applied to determine which knowledge-chunk should be accessed next. In effect, this caused the system to bear some similarity with a structured production system (Pau, 1986). Figure 4.7 illustrates this concept, where Rule 2.0.1 causes Rule 2.1 family to be activated while Rule 2.2 family is inhibited.
The knowledge bases for taper No.1, No.2, No.3 and No.4 are simpler and resemble Camber's knowledge base to a certain extend. We now consider the control structure of the system.

4.4.3 Control Strategy

There are two basic control strategies for expert systems which are known as 'forward chaining' and 'backward chaining'. In forward chaining the reasoning proceeds from data or symptoms to a conclusion. Given data or symptoms the system makes appropriate deductions until the conclusion is reached. The process is similar to pruning a decision tree. In backward chaining the process is reversed. A certain goal is initially
assumed and the inference engine seeks relevant data to prove it. If the goal turns out to be false the system can reject initial assumption and begins again with another one.

Experience with DESPLATE shows that neither of these strategies alone seems to adequately simulate the way human fault finders appear to exercise their expertise (Merry, 1983). For diagnosis in an environment like the Plate Mill, where sources of faults are widespread experts first rely on observable symptoms to distinguish between areas of fault by rejecting negative possibilities. When a particular domain is identified relevant tests can then be used to identify the fault. However, it is not always possible to explicitly prove that a particular fault exists. In such cases experts have to make some educated guesses to reach appropriate conclusions.

To simulate this type of activity a combination of both strategies has been used. The mixed strategy works as follows:

- Start with forward chaining to narrow down the search.

- When the search domain is reasonably narrow a certain fault can be assumed and if all pre-conditions required to prove that fault are known, backward chain to prove it.

- If it is not possible to carry out backward chaining (perhaps due to missing subsets of knowledge) forward chaining is continued.

Implementation of the concepts is achieved by using the LEVEL 5 expert system shell which supports both forward and backward chaining. The changes in strategy are controlled by careful arrangement of production rules within the knowledge base (LEVEL 5, 1987).
An example of the type of rule that generates forward chaining is as follows:

RULE 2.0.1 For speed mismatch between slew rolls
IF Slab twists while running over a set of slew rolls
AND Observe more closely and select \ Bar slews when broadsiding
THEN Check for faults when start and stop
AND NOT Check for speed mismatch

If the antecedents (pre-conditions) of this rule are satisfied, the inference engine would attempt to match contextual data, or information specific to a particular situation, to a pattern or template described by the Rule 2.0.1. In this case the simple fact "Check for faults when start and stop" will be activated. Consequently there are seven other rules that may be subject to activation as they all start with the same activated fact. Each rule is similar to Rule 2.2.1 shown below.

RULE 2.2.1 For faults when start and stop
IF Check for faults when start and stop
AND Run the roll in short bursts, you find \ A slow starting roll
AND Check sideguards \ Roll is not rubbing on sideguards
THEN Problem with bearings

In Rule 2.2.1, the fact "Problem with bearings" has been explicitly defined as a GOAL in the the Rule Language of LEVEL 5. Backward chaining is thus applied where the goal "Problem with bearing" is pursued and the inference engine will search for all antecedents that support that goal.

4.4.4 User Interface

The basic user interface of DESPLATE is that provided by LEVEL 5. In addition, it has been enhanced with graphics written in Turbo Pascal. Examples include plan view
shapes of faulty plates, animations of a moving slab illustrating correct (and recommended) rolling procedures etc., which are shown in Appendix A.

4.4.5 Current Status

Development of DESPLATE commenced in January 1987. At present it is deployed at three locations within the Plate Mill, including the No.2 Stand Central Control Room. Total of about 200 production rules are contained in the knowledge bases, while the source files occupy approximately 200 Kbytes of memory. The system runs on IBM PC AT, XT or compatibles.

In a recent demonstration, the system was found to be satisfactory. The solutions and recommendations it provided to experienced operators were compatible with their expectations. DESPLATE is now being used by Stand 2 operators in the diagnosis of three common faulty shapes: Camber, Off-square and Taper. The system is found to be most useful in reminding operators of things they forget and in guiding them through a systematic approach to fault-finding.

4.5 Knowledge Acquisition For DESPLATE

Some practical aspects of KA for DESPLATE are highlighted in this section. The process of selecting domain experts and issues that arose during the formation of the multiple expert team will be discussed. An example of the type of results obtainable from an information retrieval session will be given. Finally the process of verifying acquired knowledge and testing of the system will be examined.

In order to introduce the DESPLATE project to key personnel of the Plate Mill a seminar was organized during which a general overview of expert system technology was
given. A previously developed expert system was also shown to the Plate Mill personnel to demonstrate how an expert system works. Because the mill was rather complex in technical and administrative terms a co-ordinator was appointed to set up communication links between the Plate Mill personnel and the author. This co-ordinator had an overall knowledge of the rolling process and of the developments of the mill. In addition, he was an excellent administrator and very co-operative.

As mentioned in Chapter 3 there are many different techniques that can be used to select domain experts. When developing DESPLATE a sample problem was given to a group of experts nominated by the co-ordinator. Based on their performance in tackling this problem the most suitable people was selected. The sample problem was to construct a diagnostic expert system for off-square patterns which the domain experts unanimously agreed was the most straightforward case. Regular visits to the Plate Mill was made to gain a better view of the problem domain as well as a more accurate evaluation of the domain experts. Various techniques such as talk through, retrospective verbalization (Table 2.1) were used to test the experts' communication skills. Questionnaires were also used to test their ability to articulate their knowledge in writing. Consequently, a group of multiple experts was chosen with one member from each discipline. The remaining domain experts were invited to be consultants. Their contributions were valuable at a later stage.

During this period a conceptual model of the chosen experts was formed. The relation between the use of the model and a fixed methodology for KA is illustrated in Figure 4.8. The model provides the KE with some guidelines as to which methodology is to be applied and which technique to be applied for further questioning. The KE then analyses the domain experts' responses and update the model accordingly. This loop was repeated iteratively throughout the remaining KA process. In summary, the model acts as
a meta-level knowledge to guide the author in selecting the most effective methodology and techniques for knowledge elicitation.

Figure 4.8 The Relation Between the Use of a Model and a Methodology for KA

Using the model developed, the inter-relationship between the memories (Section 3.2.3) indicated that there were three major disciplines which were so strongly related to each other that the author had to use the experts as a group. It was also evident from analyzing the cognitive processors that a simple production system was relatively sufficient to represent the experts' inference process. The use of knowledge trees was also found to be a better tool for the experts to represent and to edit their knowledge during interactions.

Thus, the multiple expert team (MET) was formally formed and the author presented to the team members the techniques that were to be used for information retrieval (Section 3.2.3). The reasons for using them and what would be expected from the experts' responses were clearly specified. The team decided on the camber pattern as the next target. In the first one-and-half-hours meeting with the MET thirty probable
causes of camber patterns were obtained and classified into six hierarchical groups. These causes formed a basis for later evaluations. The meetings were organized by the co-ordinator depending on both our demand and the experts' availability. After each meeting the author developed and/or updated knowledge trees that represented the acquired information. The author also approached the team members individually if so required. Copies of the revised knowledge trees were always handed back to the MET members together with a brief notice on goals to be achieved at the next meeting.

A prototype system was immediately set up as soon as the knowledge trees were found to be reasonably complete. The system was subsequently tested and evaluated. All criticisms of the system were accepted and changes were immediately made to satisfy the experts. From our experience with DESPLATE the number of changes made at this stage was not significant as most conceptual errors had already been detected while editing the knowledge trees. Graphics routines for the user interface were also evaluated at this stage and they were modified quite significantly to suit the preference of Plate Mill personnel. This feature was later found to be one of the most important factors in the success of the system. The system, together with the final knowledge trees, was then exposed to non-participant experts for further comments. These comments were recorded and presented to the MET members for evaluations. In most cases they turned out to be valuable reminders for members of aspects that might have been omitted.

A similar approach was taken to build knowledge bases for off-square patterns (a revised version) and the four different taper patterns. The longer the experts worked in the project, the more proficient they became. The approach to the problem adapted to solve the problem of KA from multiple experts was found to be an interesting, productive and correct one. This view was also shared by the management of Chief Electrical Engineering Department of the Company (Evans, 1987).
4.6 Conclusion

This chapter has reported the development of a diagnostic expert system, DESPLATE, for the Plate Mill, BHP Steel International Group, Slab & Plate Products Division, Port Kembla. DESPLATE can assist less experienced operators to diagnose the off-square, camber and taper shapes. The system also incorporates graphics routines written in Turbo-Pascal to provide users with more user-friendly interface. DESPLATE has been found by experienced operators to be capable of providing systematic guidance in searching for faults. It is now deployed at the No.2 Stand (major rolling stand) control room of the Plate Mill.

The project involved contributions from multiple domain experts from multiple disciplines. An application of an appropriate methodology for knowledge acquisition has also been presented with reference to the project. This approach was found to be particularly effective in the case of DESPLATE and it is believed that it would be useful for projects of a similar nature.
CHAPTER FIVE

AKAS

An Automated Knowledge Acquisition System
5.1 Introduction

In this chapter the design and implementation of the automated knowledge acquisition system, denoted by AKAS, is described. AKAS has evolved from the development of DESPLATE during which knowledge acquisition (KA) tasks were carried out manually. Although the methodology and techniques described in Chapter Three has been applied to improve the efficiency of the KA process, it would be desirable to have an automated tool that would make the process even less time-consuming and therefore less costly. Such a system may be used to achieve the following goals:

(i) reduce time required for a knowledge engineer (KE) to learn enough about the problem domain in order to converse effectively with domain experts;

(ii) eliminate time required for domain experts to come to trust the KE enough to provide useful information without feeling insecure about their employment; and

(iii) automate some interview procedures that are routines and time-consuming.

The motivation for AKAS arose not only from the above drawbacks in manual approaches to knowledge acquisition, but also from an investigation of common characteristics of diagnostic knowledge. Under many circumstances diagnostic knowledge can be represented in a similar form to a diagnosis decision tree (Lister, 1988) in which the knowledge is bound together according to some causal relationship. Moreover, most approaches to diagnosis are evidential, that is, they rely on a description of a piece of evidence or a test result to confirm the existence of a particular causal event.
AKAS was designed to be an interactive system for knowledge acquisition (refer to Chapter 2). The system would have the potential of relieving the KE from the elicitation of front-end knowledge, and also of acquiring knowledge from a member of a multiple expert team when such knowledge is classified as *isolated expertise* (Section 3.2.3 and 3.3.3).

AKAS is similar in some respects to the Expertise Transfer System (ETS, Boose, 1984), MORE (Kahn et al, 1985), or other systems that belong in the same category as outlined in Chapter 2. AKAS provides a mechanism for interactive interviewing domain experts and transforming results into production rules. However, AKAS differs from ETS in that it aims at acquiring deep causal diagnostic knowledge. The approach taken to develop AKAS was based on an understanding of the causal structure of diagnosis rather than on a theory from psychology. Unlike MORE, which provides a diagnostic interpreter to act as a diagnostic shell, AKAS rules are formatted into a knowledge base for use by expert system shells such as LEVEL 5 or Insight 2+.

The chapter is organized as follows: in Section 5.2 a description of the algorithm underlying AKAS is provided. Section 5.3 presents an overview of AKAS. In Section 5.4 a detailed examination of the causal structure for diagnosis of AKAS is undertaken. The rule generating capability of AKAS is presented and discussed in Section 5.5; while the interactive interface of AKAS is considered in Section 5.6. Finally, an example of how the programme works is provided and discussed.

5.2 Algorithm For Acquiring Diagnostic Knowledge.

Apart from the methodological aspects of knowledge acquisition from multiple experts described in previous chapters, it was found that diagnostic knowledge could be
represented in a similar form to a diagnosis decision tree (Lister, 1988). This encouraged an investigation of a simple algorithm for automating the knowledge acquisition process.

Diagnostic knowledge is knowledge required to identify, or to confirm, the existence of a particular causal event. In the simplest case this type of knowledge normally consists of two basic elements: a single symptomatic event, denoted as \textit{obs}, the detection of which indicates the occurrence of a causal event, denoted as \textit{cause}. This is illustrated in Figure 5.1.

In a more complex situation in which there are multiple causal levels, a first level \textit{cause} may in turn be the result of deeper \textit{causes} each of which is associated with an appropriate \textit{obs}. (Figure 5.2).
In DESPLATE the most complex causes have six causal levels. An equivalent depth is found in MYCIN (Buchanan and Shortliffe, 1984; Buchanan, 1988).

To acquire knowledge for a simple diagnosis a domain expert may initially be asked to nominate all possible causes of the problem under consideration and then to provide a suitable obs to differentiate one cause from another (Section 3.2.2). This process is referred to as the acquisition of basic diagnostic elements.

For a complex case in which there are deep causal levels the above process may be applied repeatedly. Initially, a set of first-level diagnostic elements is obtained. Then for each first level cause a set of second-level diagnostic elements is elicited and so forth until the final cause and obs are obtained. Here final cause is a loosely defined term. A cause is said to be final whenever the user's knowledge is exhausted or there is a suitable remedy for that cause.

Acquired knowledge is weighted qualitatively to determine its importance in a diagnosis. For instance, every first level cause is assigned a value to represent its significance in causing the problem under consideration. Effectively, this approach results in a hierarchical system which bears some similarities to the knowledge base of Camber patterns in DESPLATE (Section 4.4.2). An example of this approach is presented at the end of this chapter.

5.3 Overview of AKAS

Based on the algorithm described in Section 5.2, AKAS automates the acquisition of diagnostic knowledge directly from domain experts. AKAS proved to be effective, its success being due to its underlying algorithm. To enhance AKAS' performance a
considerable amount of time was spent in improving its user friendliness and its rule generating capability.

AKAS has the potential of relieving a knowledge engineer (KE) from parts of the knowledge acquisition tasks. First, AKAS can explicate front-end knowledge from a domain expert (before any meeting is required) and this knowledge can be very useful to the KE. Second, AKAS can help the domain expert to articulate and evaluate their diagnostic knowledge via interactions with the program. Third, if there are multiple experts involved as was the case of DESPLATE, AKAS can assist the KE to interview the corresponding domain expert whenever a sub-set of knowledge is identified as isolated expertise.

At the global level of description, AKAS can be characterized as having a causal structure for diagnosis, a rule generator and an interactive interface. The causal structure of AKAS is based on the algorithm described in the previous section. This structure controls the behaviour of AKAS and through interactions with a human expert it determines the final structure of the created knowledge base. AKAS' rule generator converts entered knowledge into production rules and formats rules into a suitable knowledge base for a commercially available expert system shell. AKAS makes no attempt to understand natural language but it takes advantage of its underlying structure and user responses to enhance its user friendliness. AKAS is examined in more detail in the remaining sections of this chapter.

5.4 Causal Structure for Diagnosis

As AKAS interviews a domain expert, it attempts to build a causal network by mapping the expert's responses into the following diagnostic elements: causes, obs and
remedies. A cause, as defined in Section 5.2, represents a causal event whose identification will be the result of a diagnosis. An example is mal-functioning of equipment. An obs represents a symptomatic event that helps to detect the cause. A remedy denotes an immediate action that would be taken by the expert in the given situation. It is not always possible to obtain an effective corrective action, but from experience with DESPLATE it was found that experts always gave valuable suggestions once faults were detected.

Next, AKAS attempts to interpret the expert's line of reasoning by linking the diagnostic elements together. For a simple diagnosis in which there is only one causal level, the line of reasoning is represented in the manner illustrated by Figure 5.3.

![Figure 5.3 AKAS' Representational Structure of Knowledge in a Simple Diagnosis](image)

In a more complex situation where there are multiple causal levels, causes are then classified into either intermediate causes or final causes. A cause is called intermediate or final depending on the causal level in which it resides. The actual number of causal levels is interactively determined by the expert's responses to AKAS. An intermediate cause serves as a leverage for obtaining other causes which may include a final one. To represent the line of reasoning in this situation all intermediate obs are linked together and to the final elements as illustrated in Figure 5.4.
This structure facilitates the identification of the final cause which is the aim of the diagnosis process. It also eases the rule generation process to be discussed in the next section. Here the following assumptions have been made: (i) the first level causes are simplest and easiest to detect; and (ii) the level of complexity in diagnosis is increased with the depth of the causal levels. These assumptions are explained by AKAS to the domain expert prior to starting the interviewing session.

5.5 Rule Generator

AKAS is capable of acquiring diagnostic knowledge directly from a domain expert and also of constructing production rules to represent the knowledge. As a rule generator, AKAS performs the following tasks:

(i) rule construction and evaluation; and
(ii) formatting rules into a suitable knowledge base for an ES shell.

The process of acquiring knowledge from an expert and the task of constructing rules are related to each other. As soon as the first cause and obs are obtained they are used to form a production rule that represents the user's line of reasoning. For example:

If it is true that

The Slab twists while running over a set of slew rolls [obs 1]

Then the cause for off-square problem could be

A problem due to speed mismatched [cause]
Notice that the user's responses are shown in italic. As AKAS proceeds to deeper causal levels, additional symptomatic events or test results are acquired.

If it is true that

*The slab twists while running over a set of slew rolls* [obs 1]

And if it is also true that

*The slab only twists when doing sizing passes* [obs 2]

And if it is also true that

*OIS and DIS generator voltage readings differ by more than 5 volts* [obs 3]

Then the cause could be

*Unbalanced magnetic amplifiers* [cause 3]

And a remedy is

*Call electrician to check and balance magnetic amplifiers* [remedy]

In this example there is only one explicit cause in the rules followed by a remedy. The remaining intermediate causes are implicit and stored internally as modes in a diagnosis decision tree. These causes will appear later when the rule is put into a knowledge base.

Rule evaluation is part of rule construction. Whenever new diagnostic elements are added to the rule, AKAS presents to the expert its interpretation of the new line of reasoning and requests an approval. This approach is aimed at stimulating the user's retrospection and their responses reinforce the validity of the rule. AKAS allows for corrections to be made to all parts of the rule until the expert is completely satisfied with it. AKAS does not understand natural language so it must rely on its user's intelligence to evaluate created rules. While our approach seems to work effectively with a small number of rules, it may not prove feasible to preserve the consistency of a significantly larger amount of acquired knowledge.
Finally, the created rules are formatted into a suitable form for LEVEL 5 and Insight 2. This process involves some alterations to suit specifications of the target system. The prime concern in AKAS is to arrange the rules in such a way that they reflect the experts approach to the problem. This is chiefly done by assigning weighing values in the process of entering knowledge as mentioned in Section 5.2. This issue will be discussed further in the next section.

5.6 Interactive Interface

One of the most important characteristics of AKAS is its interactive responses to users. Although the program has no capability to understand natural language, the conversation seems to follow a natural discourse. This is because of the simplicity of the underlying structure of AKAS for diagnostic knowledge which has been discussed in previous sections. Moreover, users' answers are mapped directly into next questions in the form of a cause, an obs, or a remedy. So far two kinds of causes have been determined: intermediate causes and final causes. In addition there are five kinds of obs: observable facts, typical symptoms, test names, test results, and assumptions. AKAS interactively searches for an appropriate obs in the following manner: once a particular cause is obtained, AKAS requests the user to nominate an observable fact or a typical symptom consequent to the existence of the cause. If the user cannot provide such information, AKAS continues to ask for the name of a specific test or a measurement which would result in the identification of the cause. If the user is still unable to supply the information required, the program would make an assumption based on the knowledge about the cause. Such an assumption is created by mapping the cause into one of AKAS' existing "canned" text, for example:

If it is true that you suspect roll Cross-Over has occurred.
Here, "If it is true that you ... has occurred" is a "canned" text, it is combined with the user's answer "roll Cross-Over" to form an assumption which will act as an antecedent (pre-condition) in a rule. As usual, this assumption will be included in the new line of reasoning and the user will have a chance to modify it.

An example is given in the following section to show detailed interactions between AKAS and a domain expert in the diagnosis of plate-related problems. The problem in this example is off-squareness of plates and AKAS attempts to explicate knowledge of what the causes are and of how to detect them. For simplicity and in order to show the main features of AKAS, only the major parts of the interview session are given. The domain expert's responses are in italic (which represents a different colour used by AKAS), where the author's comments are in square-brackets ([]).

[After an initial introduction and explanation of the basic concepts and how to use the system, AKAS starts by locating the problem to be diagnosed.]

What problem do you want to diagnose?
> Off-square

Is it true that you want to diagnose off-square problem? (Y/N)
> Y

Please state the most direct cause of off-square problem
> Speed mismatch between rolls

Please enter a weighing factor (1 to 9)
> 7

Do you know any other cause of off-square problem? (Y/N)
> Y
What is it?
> Turn-up slab

Please enter a weighing factor (1 to 9)
> 4
... 
Do you know any other cause of off-square problem? (Y/N)
> N

[After the first level causes have been acquired, the user is allowed to check for correctness and consistency. The causes are listed in decreasing order of weighing factor.]

We are considering the causes of off-square problem. The possible causes that you have given are:
1. Speed mismatch between the rolls  WF = 7
2. Turn-up slab  WF = 4

Are you happy about the weighing factors assigned to these causes?
> Y

[The main function of these factors is to classify the causes into correct order of importance. If answering 'No' the user is allowed to change the weighing factors until completely satisfied. For the above answer AKAS continues to search for the required obs.]

One cause of off-square problem is speed mismatch between rolls. Can you identify "speed mismatch between rolls" by observing it? (Y/N)
> N
Is there a typical symptom of "speed mismatch between rolls"? (Y/N)
> Y

Please describe the symptom that helps you to identify "speed mismatch between rolls"
> A slab twists while running over a set of slew-rolls

This is how your reasoning is understood:
The given problem is off-square problem
If it is true that a slab twists while running over a set of slew-rolls
Then the cause of off-square problem could be speed mismatch between rolls

Do you wish to change (or reword) any sentence? (Y/N)
> N

[If answering 'Y' the user is allowed to modify any chosen sentence until completely satisfied; an example of which will be shown later. Here, AKAS continues to search for either a remedy or deeper level causes.]

Is there an obvious remedy for "speed mismatch between rolls"? (Y/N)
> N

Is there anything that may cause "speed mismatch between rolls"? (Y/N)
> Y

What is the most direct cause of "speed mismatch between rolls"?
> Consecutive rolls run at different speeds
Please enter a weighing factor (1 to 9)
> 6

Do you know any other cause of "speed mismatch between rolls"? (Y/N)
> Y

What is it?
> Consecutive rolls accelerate or decelerate differently

Please enter a weighing factor (1 to 9)
> 5
...
Do you know any other cause of "speed mismatch between rolls"? (Y/N)
> N

One cause of speed mismatch between rolls is "consecutive rolls run at different speeds". Can you identify "consecutive rolls run at different speeds" by observing it? (Y/N)
> N

Is there a typical symptom of "consecutive rolls run at different speeds"? (Y/N)
> Y

Please describe the symptom that helps you to identify "consecutive rolls run at different speeds"
> The slab only twists when doing sizing passes or broadsiding

This is how your reasoning is understood:
The given problem is off-square problem
If it is true that a slab twists while running over a set of slew-rolls
Then the cause of off-square problem could be *speed mismatch between rolls*

And if it is also true that *the slab only twists when doing sizing passes or broadsiding*

Then the cause could be *consecutive rolls run at different speeds*

Do you wish to change (or reword) any sentence? (Y/N)  
> N

Is there an obvious remedy for "consecutive rolls run at different speeds"? (Y/N)  
> N

Is there anything that may cause "consecutive rolls run at different speeds"?  
(Y/N)  
> Y

What is the most direct cause of "consecutive rolls run at different speeds"?  
> *Unbalanced magnetic amplifiers*

Please enter a weighing factor (1 to 9)  
> 8

Do you know any other cause of "consecutive rolls run at different speeds"? (Y/N)  
> N

One cause of consecutive rolls run at different speeds is unbalanced magnetic amplifiers. Can you identify "unbalanced magnetic amplifiers" by observing it?  
(Y/N)  
> N
Is there typical symptom of "unbalanced magnetic amplifiers"? (Y/N)

> N

Can you carry out any test (or measurement) to find out if "unbalanced magnetic amplifiers" has occurred? (Y/N)

> Y

What test do you use?

> Check O/S and D/S generator voltages

To confirm that "unbalanced magnetic amplifiers" has occurred, what result do you expect from check O/S and D/S generator voltages?

> readings differ by more than 5 volts

This is how your reasoning is understood:

The given problem is off-square problem

If it is true that a slab twists while running over a set of slew-rolls
Then the cause of off-square problem could be speed mismatch between rolls
And if it is also true that the slab only twists when doing sizing passes or broadsiding
Then the cause could be consecutive rolls run at different speeds
And if it is also true that from check O/S and D/S generator voltages, result is readings differ by more than 5 volts
Then the cause could be unbalanced magnetic amplifiers

Do you wish to change (or reword) any sentence? (Y/N)
Is there an obvious remedy for "unbalanced magnetic amplifiers"? (Y/N)

> Y

Please give the most appropriate remedy (if possible)

> Call electrician to check and balance the mag. amp. as soon as possible

As the result rule No.1 has been constructed based on the acquired information. AKAS continues to search for further information about consecutive rolls accelerate or decelerate differently and turn-up slab, additional rules will eventually be formed. At the end of the interview session, AKAS creates a knowledge base for LEVEL5 by putting all the rules into the following format:

TITLE A Knowledge Base for Diagnosis of Off-square Problem

SUPPRESS ALL  {Turn off the default displays of LEVEL5 }

{ Goal }
1. Unbalanced magnetic amplifiers
2. A roll is rubbing on sideguards
3. Faulty or tight bearings
4. Turn-up slab
...
{Rules}
RULE 1
IF a slab twists while running over a set of slew-rolls
{Which may mean that speed mismatch between rolls}
AND the slab only twists when doing sizing passes or broadsiding
{Which may mean that consecutive rolls run at different speeds}
AND O/S and D/S generator voltages readings differ by more than 5 volts
THEN Unbalanced magnetic amplifiers
AND DISPLAY pc Unbalanced magnetic amplifiers

RULE 2
IF a slab twists while running over a set of slew-rolls
{Which may mean that speed mismatch between rolls}
AND the slab only twists when starting and stopping
{Which may mean that consecutive rolls accelerate or decelerate differently}
AND there is a slow starting and fast stopping roll with shiny marks on it.
Sparkings occur occasionally
THEN A roll is rubbing on sideguards
AND DISPLAY pc A roll is rubbing on sideguards

RULE 3
IF a slab twists while running over a set of slew-rolls
{Which may mean that speed mismatch between rolls}
AND the slab only twists when starting and stopping
{Which may mean that consecutive rolls accelerate or decelerate differently}
AND there is a slow starting and fast stopping roll with no shiny marks on it and no sparking
THEN Faulty or tight bearings
AND DISPLAY pc Faulty or tight bearings

RULE 4
IF You can visually detect a turn-up slab
THEN Turn-up slab
AND DISPLAY pc Turn-up slab

...
DISPLAY pc A roll is rubbing on sideguards
POSSIBLE CAUSES
A roll is rubbing on sideguards
REMEDY
Contact mechanical maintenance

DISPLAY pc Faulty or tight bearings
POSSIBLE CAUSES
Faulty or tight bearings
REMEDY
Contact mechanical maintenance

DISPLAY pc Turn-up slab
POSSIBLE CAUSES
Turn-up slab
REMEDY
Notify furnace foreman of turn-up problem. Keep sideguards close to slab to minimize slewing. This helps to reduce the problem.

END

The knowledge used in this example was obtained from Plate Mill personnel, and it has been put into a suitable format to be compiled and run using LEVEL5 or Insight 2+. 
5.7 Conclusion

This chapter has presented the design and implementation of AKAS, an automated system for the acquisition of diagnostic knowledge. It has been shown that even though the underlying algorithm is simple and the system does have some weak points, AKAS has tremendous potential in assisting a knowledge engineer in knowledge acquisition, especially when the tasks are repetitive or the wanted knowledge is possessed by a single expert. Experience from several trial runs revealed that the level of interaction provided by AKAS is sufficiently friendly for obtaining deep causal knowledge. In most cases the user only needed a short demonstration before fruitful results were obtained.

Nevertheless, there are more developments that can be done to improve AKAS' performance: The structure of diagnostic knowledge is rigid and thus may not allow a more flexible interaction with the user. Further, AKAS is illiterate and blindly accepts answers which may be syntactically or grammatically incorrect. The current remedy is to rely on the user's corrections but this may exhaust the user's patience as a large number of information is being elicited. To solve this AKAS needs to have a capability to understand natural language. A more detail suggestion for the future developments of AKAS will be given in the next Chapter.
CHAPTER SIX

Conclusions and Recommendations
To complete this thesis it is appropriate to summarise the contributions which have been presented in the previous chapters, to point out some remaining problems and to suggest ways in which the subject may be further developed in the future.

In Chapter Two, a literature survey of major trends and techniques for knowledge acquisition has been presented. Approaches to Knowledge Acquisition (KA) were characterised into two major categories: manual and automated (or computer-aided) approaches. In the former category, major stages of developing an expert system were reviewed, and commonly used KA techniques were summarised. Among the problems encountered in this tedious approach, KA from multiple experts was of major interest. Even though there had been different methods suggested by various research groups to tackle this problem, it was found to remain the most important and challenging area of research. The later category concentrated on using computer programmes to partially replace human beings in KA tasks. This approach led to research and development of automated knowledge acquisition systems. These systems were categorised into four major groups based on their functions and each group was examined in detail.

To tackle the problem of KA from multiple experts Chapter Three presented the concept of modelling a group of multiple domain experts from multiple disciplines. The concept was shown to be useful for a knowledge engineer when it is used in conjunction with an appropriate methodology for KA. Such a methodology for the acquisition of knowledge from domain experts from different disciplinary backgrounds was introduced. The methodology emphasised the initial stage of selecting and training of domain experts as well as the use of appropriate techniques for information retrieval in the later stages.

Chapter Four reported the application of the above concept for modelling multiple experts and the methodology for KA to the development of DESPLATE, an expert system
for the diagnosis of faulty shapes of steel plates. The system could assist less experienced operators to diagnose major problems that occurred in the production of plates at the Plate Mill, BHP Steel International Group, Slab & Plate Products Division, Port Kembla. The system also incorporated graphics routines, written in Turbo-Pascal, to provide users with an user-friendly interface. DESPLATE had been found by experienced operators to provide useful guidance in search for faults.

To provide DESPLATE with a satisfactory performance, the issues of knowledge representation and control strategy for DESPLATE were also considered. Efforts were made to arrange acquired knowledge into hierarchical levels to reflect the experts' approach to diagnosis. In addition, both backward chaining and forward chaining were implemented.

Chapter Five presented the design and implementation of an automated knowledge acquisition system, AKAS, which was capable of directly interviewing a domain expert for diagnostic knowledge. Acquired knowledge was represented by production rules in a format suitable for compilation by a commercial expert system shell. The design of AKAS was based on a view that diagnostic knowledge could be represented in the form of a diagnosis decision tree. The system was used as a tool to assist a knowledge engineer in part of the KA process.

Among the above works, the following areas may be subject to future developments:

1. The model of multiple experts presented was a simple and conceptual one. The development process of such a model was very much dependent on the developer's perception. A tendency in modelling would logically be from simple to complex, and
CONCLUSIONS AND RECOMMENDATIONS

from conceptual to detailed. Further work may therefore be done to devise a more systematic and algorithmic approach to the development of the model. On the other hand, the applied methodology has been aimed at one specific subset of the KA bottleneck, i.e. KA from multiple domain experts from multiple disciplinary backgrounds. This methodology could be generalised to cover other situations in which multiple experts possess a similar disciplinary background. Once this is done, the resultant methodology would provide a more general guidance to KA from multiple experts.

2. DESPLATE is presently an off-line expert system. Its approach to problem solving is to rely on familiar observable phenomena or known results of routine testing procedures. Such an approach is sometimes inadequate because observations of the same symptom are not always identical, a slightly varied phenomenon may easily be overlooked by operators. Moreover, not all possible phenomena are covered by the system. It is thus desirable to improve the reliability of DESPLATE by incorporating on-line data. These data may include both measured and scheduled plan-view width and thickness of slabs under rolling. The measured width and thickness of a slab may be used to evaluate the quality of the slab at different stages of rolling. In practical situations, it has been noticed that operators could readily make decision based on these measured data (which were available to be used by other processes within the Plate Mill). Scheduled width and thickness may be used for this purpose.

3. There are further developments that may improve AKAS' performance. The structure of diagnostic knowledge is rigid and thus may not allow a more flexible interaction with the user. Further, AKAS is illiterate and blindly accepts answers which may be syntactically or grammatically incorrect. The current remedy is to rely on the user's corrections but this may exhaust the user's patience in cases a large amount of information is being elicited. Incorporating a restricted ability to interpret natural language could be the next step. In addition, the system could be made more robust by
incorporating the ability to identify, acquire and represent different kinds of reasoning (Kornell, 1987).


24. Cung, L.D. and Ng, T.S. (1988b); 'Modelling Multiple Experts For Knowledge Acquisition', submitted for publication.

25. Davis, R. (1977a); 'Interactive Transfer of Expertise: Acquisition of New Inference Rules', *IJCAI 1977*.


Appendix A

Graphic Displays from DESPLATE
Figure A.1 Camber Shape

Figure A.2 Display of A Correct Procedure For Re-Broadsiding Slabs
Appendix B

Graphic Displays from AKAS
Figure B.1 Title Page

Figure B.2 An Overview of The Knowledge Base
At each causal level, you will be asked for two types of information:

**Cause**: This represents either a simple cause of the problem under consideration or a specific fault that can readily be fixed.

**Obs**: This represents either an observable fact, a test result, or any other means of detecting the above cause.

Figure B.3 Detailed Definition of Each Element