ANALYSIS, DESIGN & EVALUATION OF MAN-MACHINE SYSTEMS 1989

53

ANALYSIS, DESIGN AND EVALUATION OF MAN–MACHINE SYSTEMS 1989

Selected Papers from the Fourth IFAC/IFIP/IFORS/IEA Conference, Xi'an, People's Republic of China, 12–14 September 1989

Edited by

BAOSHENG HU

Xi'an Jiaotong University, People's Republic of China

Published for the

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

by

PERGAMON PRESS

Member of Maxwell Macmillan Pergamon Publishing Corporation

OXFORD · NEW YORK · BEIJING · FRANKFURT

SÃO PAULO · SYDNEY · TOKYO · TORONTO

U.K.	Pergamon Press plc, Headington Hill Hall, Oxford OX3 0BW, England
U.S.A.	Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, New York 10523, U.S.A.
PEOPLE'S REPUBLIC OF CHINA	Pergamon Press, Room 4037, Qianmen Hotel, Beijing, People's Republic of China
FEDERAL REPUBLIC OF GERMANY	Pergamon Press GmbH, Hammerweg 6, D-6242 Kronberg, Federal Republic of Germany
BRAZIL	Pergamon Editora Ltda, Rua Eça de Queiros, 346, CEP 04011, Paraiso, São Paulo, Brazil
AUSTRALIA	Pergamon Press Australia Pty Ltd., P.O. Box 544, Potts Point, N.S.W. 2011, Australia
JAPAN	Pergamon Press, 5th Floor, Matsuoka Central Building, 1-7-1 Nishishinjuku, Shinjuku-ku, Tokyo 160, Japan
CANADA	Pergamon Press Canada Ltd., Suite No. 271, 253 College Street, Toronto, Ontario, Canada M5T 1R5

Copyright © 1990 IFAC

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means: electronic, electrostatic, magnetic tape, mechanical, photocopying, recording or otherwise, without permission in writing from the copyright holders.

First edition 1990

Library of Congress Cataloging in Publication Data

IFAC/IFIP/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man–Machine Systems (4th: 1989: Sian, China)
Analysis, design, and evaluation of man–machine systems, 1989: selected papers from the Fourth IFAC/IFIP/IFORS/IEA Conference, Xi'an, People's Republic of China, 12-14 September 1989/edited by Baosheng Hu.—1st ed.
p. cm.—(IFAC symposia series: 1990, no. 11)
1. Man–machine systems—Congresses. I. Hu, Baosheng. II. International Federation of Automatic Control. III. Title. IV. Series.
TA167.I33 1989a 620.8'2—dc20 89-72170

British Library Cataloguing in Publication Data

Analysis, design and evaluation of man–machine systems 1989: selected papers from the fourth IFAC/IFIP/IFORS/IEA conference, Xi'an, People's Republic of China, 12-14 September 1989.
1. Man–machine systems
I. Baosheng Hu II. International Federation of Automatic Control
III. Series
620.82

ISBN 0-08-035743-1

These proceedings were reproduced by means of the photo-offset process using the manuscripts supplied by the authors of the different papers. The manuscripts have been typed using different typewriters and typefaces. The lay-out, figures and tables of some papers did not agree completely with the standard requirements: consequently the reproduction does not display complete uniformity. To ensure rapid publication this discrepancy could not be changed: nor could the English be checked completely. Therefore, the readers are asked to excuse any deficiencies of this publication which may be due to the above mentioned reasons.

The Editor

Printed in Great Britain by BPCC Wheatons Ltd, Exeter

4th IFAC/IFIP/IFORS/IEA CONFERENCE ON ANALYSIS, DESIGN AND EVALUATION OF MAN–MACHINE SYSTEMS

Organized by
CAA–Chinese Association of Automation
P.O. Box 2827, Beijing, 100080 China

Sponsored by
International Federation of Automatic Control (IFAC)
Technical Committee on Systems Engineering
Technical Committee on Social Effects of Automation
Technical Committee on Economic and Management Systems
Technical Committee on Applications

Co-sponsored by
International Federation for Information Processing (IFIP)
International Federation of Operational Research Societies (IFORS)
International Ergonomics Association (IEA)

International Programme Committee (IPC)
G. Johannsen, FRG (Chairman)
J. L. Alty, UK
K. Bø, Norway
Chen T., PRC
Chen Z. Y., PRC
V. De Keyser, Belgium
W. J. Edwards, Australia
B. R. Gaines, Canada
R. Genser, Austria
V. G. Grishin, USSR
G. Guida, Italy
R. Haller, FRG
Hu B. S., PRC
K. Inoue, Japan
K. Kawai, Japan
T. L. Kunii, Japan
A. H. Levis, USA
N. Malvache, France
G. Mancini, Italy
L. Mårtensson, Sweden
Y. Nishikawa, Japan
J. Ranta, Finland
J. Rasmussen, Denmark
J. E. Rijnsdorp, The Netherlands
W. Rohmert, FRG
W. B. Rouse, USA
A. P. Sage, USA
R. Seifert, FRG
B. Shackel, UK
Shen R. F., PRC
T. B. Sheridan, USA
H. G. Stassen, The Netherlands
Su S. Q., PRC
Sun B. L., PRC
B. G. Tamm, USSR
H. Tamura, Japan
T. Vámos, Hungary
Wang Y. L., PRC
J. Wernstedt, GDR
R. C. Williges, USA
Xu G. Z., PRC

National Organizing Committee (NOC)
Wang Y. L. (Chairman)
Hu B. S. (Vice-Chairman)
Jia W. H.
Li R. H.
Wan B. W.
Li H. Z.
Xuan G. R.

PREFACE

This 4th IFAC Conference on Man-Machine Systems Analysis, Design and Evaluation followed closely the 3rd IFAC Conference on same topic at Oulu of Finland just after an interval of 14 months that reflected the field grown rapidly in recent years.

The aim of the conference is to provide a forum for presenting and discussing the recent advances in both empirical and theoretical aspects of man-machine interaction with special emphasis on man-automation and man-computer interaction.

Preceding the conference a one day tutorial session was proposed to newcomers and presented by top researchers - Prof. H.G. Stassen and Dr. A.H. Levis from the select MMS community on topics: " Introduction to Man-Machine Systems " and "Review on Distribution Decision Making ".

The 4 plenary papers and 38 regular papers was planned to present on the conference, but due to the special event happened on June 4, 1989 at Beijing many participants cancelled their trip to China, therefore, only 3 plenary papers and 21 regular papers were actually presented on the conference. Of these presenters and participants, approximately 55% were from host country and the rest 45% from other countries still reflected an international gathering. Due to the special circumstances, the general IFAC policy with respect to only including presented papers in the Proceedings are waived in this case. The International Program Committee finally decided, in addition to 3 plenary papers,to select 24 regular papers from the 38 accepted and preprinted papers for publication in the Proceedings based on the technical subjects, qualification and geographical distribution of the papers.

I hope that the results will be beneficial to all engineers and scientists who are actively working or strongly interested in this rapidly growing field.

Baosheng Hu
Editor

此为试读,需要完整PDF请访问: www.ertongbook.com

ACKNOWLEDGMENTS

First of all, I wish to thank all authors and participants for their contributions who enriched the conference by an attitude of solidarity and friendship. I am also grateful for the excellent cooperation offered by the members of the International Program Committee (especially Prof. Gunnar Johannsen, Henk G. Stassen, James L. Alty, S.Q. Su and Dr. Alexander H. Levis).

Next, I wish to express my gratitude to China National Natural Science Foundation, Shaanxi Provincial Association of Automation, and Xi'an Jiaotong University for their financial support.

Finally, I should thank the following persons assisted diligently in the organization: Dr. Feng Z.R., Dr. Le W.L., Mr. Wang X.Y. and Ms Zhang J.H..

Baosheng Hu
Editor

CONTENTS

PLENARY PAPER I

PLENARY PAPER II

MANUAL CONTROL

MAN-MACHINE INTERACTION

PLENARY PAPER III

HUMAN PERFORMANCE

COGNITIVE PROCESSES

INDUSTRIAL PLANTS

Copyright © IFAC Man–Machine Systems, Xi'an, PRC, 1989

KNOWLEDGE ENGINEERING FOR INDUSTRIAL EXPERT SYSTEMS

G. Johannsen* and J. L. Alty**

**Laboratory for Man–Machine Systems, University of Kassel (GhK), D-3500 Kassel, FRG*
***Scottish Human–Computer Interaction Centre, University of Strathclyde, Glasgow, UK*

Abstract. The inherent difficulties involved in the process of extracting knowledge from experts are discussed and identified. Such difficulties have resulted in few expert systems progressing beyond the prototyping stage. The conflicting terminology used to describe the whole process is examined and, as a result, knowledge engineering is defined as the appropriate term for the whole process. This is then further split into knowledge acquisition and system implementation. Finally, knowledge acquisition is further subdivided into knowledge elicitation and machine induction.

The particular problems associated with the construction of expert systems in industrial control applications are discussed. Such systems are characterised by the nature of their user population, the type of support provided and whether they operate on-line or off-line. The importance of defining functionality and goals at the outset is stressed. The need for user models is also highlighted.

The various techniques used in knowledge elicitation - interviews, questionnaires, observations, protocol analyses, teachback interviewing, walkthroughs and formal techniques - are briefly reviewed. The alternative approach using machine induction techniques is also discussed. An examination is made of the competing approaches involving bottom-up and top-down techniques. The benefits resulting from the application of cognitive task analyses rather than technology-driven approaches are also stressed. Current knowledge acquisition tools such as KRITON, KADS, ACQUIST, KEATS and ROGET are reviewed.

Examples are given of the use of time line techniques in power plant knowledge acquisition, knowledge and task analyses in the construction of a failure management expert system and of the use of inductive techniques in gas-oil separator design and satellite power systems control. In the latter case, the use of qualitative modelling is highlighted.

The possibility of domain experts in industrial control carrying out their own knowledge engineering is examined but rejected as unlikely unless better tools exist. The provision of better tools is identified as one of the key factors required to simplify the knowledge engineering process.

Keywords. Knowledge engineering; knowledge acquisition; knowledge elicitation; machine induction; expert systems; supervision and control; industrial control; cognitive task analysis; qualitative modelling; knowledge acquisition tools.

INTRODUCTION

Knowledge Engineering is the process of building expert systems. Such systems are medium- to large-scale software products which are designed to solve problems of different kinds using a knowledge-based approach where the knowledge is represented in an explicit manner. They have a wide area of applicability particularly in industrial control. Hayes-Roth and others (1983), for example, have identified ten generic categories of knowledge engineering applications. These are interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction, and control. Such systems normally contain two main components (Davies, 1982) - the inference mechanism (the problem solving component) and the knowledge base (which may actually comprise a number of knowedge bases). Generally speaking, expert systems work best in narrow application domains.

Madni (1988) has provided an hierarchical classification of expert systems from a human factors perspective. One level of his classification distinguishes between expert systems with respect to different purposes:

- perfom a task
- assist in a task, and
- teach a task.

The first category deals with autonomous expert systems such as those found in autonomous robots or automation systems. The second and third categories are concerned with expert consultation systems. In addition, systems concerned with teaching have more granular uncompiled knowledge and access a pedagogical knowledge base in addition to domain knowledge bases. In industrial systems, expert systems of the first category are always embedded within the technical system whereas those in the second category can be either stand-alone or embedded.

The process of building an expert system consists of two main activities which usually overlap - acquiring the knowledge and implementing the system. The acquisition activity involves the collection of knowledge about facts and reasoning strategies from the domain experts. Usually, such knowledge is elicited from the experts by so-called knowledge engineers, using interviewing techniques or observational protocols. However, machine induction, which automatically generates more elaborate knowledge from an initial set of basic knowledge (usually in the form of examples), has also been extensively used (Michie and Johnston, 1985). In the system construction process, the system builders (i. e., knowledge engineers), the domain experts and the users work together during all stages of the process, which traditionally has involved extensive prototyping.

To automate the problem solving process, the relevant task knowledge in the domain of interest needs to be understood in great detail but acquiring the knowledge for expert system building is generally regarded as a hard problem. This is not surprising. As Kidd (1987) has pointed out, acquiring knowledge from an expert entails answering some really fundamental questions such as

- what is the relationship between knowledge and language?
- how can we characterise different domains?
- what constitutes a theory of problem solving?

Clancey (1986) has also pointed out that the process of extracting knowledge from an expert is not the process of transferring a mental model lying in the brain of an expert into the mind of the system builder, but the formalisation of a domain for the first time and this is inherently a difficult process. Ideally, models of conceptual structures of problem solving behaviour are required as a prerequisite to the knowledge transfer process. However, cognitive science approaches have not yet yielded sufficient information to enable a full understanding of the knowledge structures and problem solving strategies of experts to be applied so that current approaches are incomplete and often ad-hoc.

The situation is further complicated by the fact that experts often have faulty memories or provide inconsistencies. This means that separate validation of the expertise elicited from experts is essential (Chignell and Peterson, 1988). Furthermore, experts exhibit cognitive biases such as overconfidence, simplification, and a low preference for the abstract, the relative and conflicting evidence. It is therefore important to test and validate expert systems both by analysing the expertise in the knowledge base and by examining failures in actual performance. As far as possible, cognitive biases should be filtered out during the elicitation process.

Madni (1988) has taken these important points into account in his detailed view of the whole knowledge engineering process appraised from a cognitive engineering viewpoint. He has suggested the following six stages which he terms mainstream development:

- knowledge elicitation
- cognitive bias filtering
- knowledge representation and control scheme selection
- software development and integration
- system evaluation and validation, and
- advanced prototype expert system.

Stages three and four ideally should only be carried out after the elicitation and cognitive bias stages have been completed. In reality, this is not possible and our own experience as well as that of other researchers suggests that several iterations through the first five stages are required before stage six can be contemplated. Madni also proposes two additional paths of prototyping activities for demonstration and software development purposes which are to be performed in parallel to the first four stages of the mainstream development. Evaluations have to be carried out in all stages of software development.

Few systems have progressed beyond the research or prototype phase mainly because of the inherent difficulties in the knowledge acquisition process (Breuker and Wielinga, 1987).

THE TERMINOLOGY OF KNOWLEDGE ENGINEERING

There are a number of terms used to describe the expert system building process which are not well defined and appear to overlap. Such terms include knowledge elicitation, knowledge acquisition, system implementation, machine induction and even the term knowledge engineering itself. Buchanan and others (1983) define knowledge acquisition as "the transfer and transformation of problem-solving expertise from some knowledge source to a program". This definition covers the whole process including identification of the problem, its conceptualisation, formalisation, implementation, testing and prototype revision. Diederich and Linster (1989) subdivide knowledge acquisition into knowledge elicitation and an operational phase. Motta and others (1989) term the whole process knowledge engineering but subdivide it into knowledge acquisition, knowledge representation and implementation. They further break down knowledge acquisition into knowledge elicitation and data interpretation. As Motta and others state "The separation of acquisition from implementation leads to a view of knowledge acquisition as the production of an abstract architecture distinct from the implementation of the system". However, they accept that such a characterisation is also problematic since the only way of testing the knowledge is to run it so that the boundaries between acquistion and implementation can be very fuzzy.

Our view is that the process of building knowledge-based systems is essentially one of knowledge engineering and we regard the different terms as fitting together as in Figure 1.

Most authors agree over the general term knowledge engineering. We have, however, distinguished the knowledge acquisition process from the system implementation process (like Motta and others). Although it is true that the two intertwine during prototyping there are good reasons for separating them out, at least conceptually, as will become clear later on. We have also separated out elicitation (either manual or automatic) from machine induction since these acquisition techniques are quite distinct and have followed different development paths.

KNOWLEDGE ENGINEERING IN INDUSTRIAL SYSTEMS

Expert systems can be introduced into industrial systems to provide support for different classes of people such as designers, operators and maintenance personnel. In general such systems will be off-line (for designers and maintenance personnel) and on-line (for operators). The knowledge engineering task will be different for each of these applications since the tasks involved will comprise different knowledge sources and structures. One difference is that between technological/scientific knowledge and experiential knowledge. This difference was described as a "knowledge of functioning" versus a "knowledge of utilisation" by DeMontmollin and DeKeyser

此为试读，需要完整PDF请访问：www.ertongbook.com

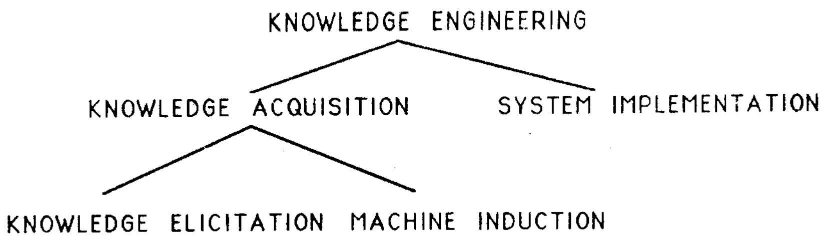

Fig. 1. Relationship between terms in knowledge engineering.

(1986). The former knowledge is used by designers and maintenance personnel whereas the latter characterises that used by operators. The adoption of a truly human-centred design approach (Alty and Johannsen, 1989; Johannsen 1990) requires designers to consider both. This means that user models are expected to be part of future knowledge-based support functions for designers (Sundström, 1988; Sundström and Johannsen, 1989).

Generally, designer activity in industrial systems ranges from the computer aided design of subsystems and components to picture design for control rooms (Rouse 1986; Elzer and Johannsen, 1988). Thus for designer support, knowledge about the application domain in addition to that concerned with design procedures is needed for all these tasks (Borys, 1988). Further, ergonomic knowledge and knowledge about user behaviour is required in picture design. Hence, design activity should be supported with a set of different knowledge bases (and perhaps a user model).

Off-line knowledge-based systems are not time critical. They may utilise several knowledge sources including technical documents, reference literature, handbooks, ergonomic knowledge, and knowledge about operator personnel (for use in user modelling). Whilst their operation is not time critical they may still have to take into account operator time constraints.

The most critical and challenging industrial expert systems are those developed for system operation. They may encompass support for the automatic system as well as support for the operators and may provide heuristic control, fault diagnosis, consequence prediction, and procedural support (Johannsen, 1990). The latter is particularly suitable for consistency checking of input sequences or for operator intent recognition (Hollnagel, 1987; Shalin and others, 1988). All these support expert systems work under time constraints because they are running in parallel with the dynamic industrial process. Like off-line systems, these expert systems will depend upon a number of knowledge sources related to knowledge of functioning and knowledge of utilisation. Additional knowledge such as that of senior engineers will be required.

Whilst a support expert system for predicting the consequences of some technical failure will normally need only engineering knowledge, procedural support, diagnosis and heuristic control modules will need operational knowledge as well. Since they will also have to be integrated with the supervision and control system they will need to support numerical as well as symbolic knowledge.

In all cases, it will be necessary to define carefully the goals and functionalities of the various systems and their interdependencies at an early stage. It is also important to realise that in the industrial environment not all applications are suitable for the application of knowledge-based techniques. For example, existing numerical supervision and control systems are based upon thorough engineering methodologies and replacement by knowledge-based techniques would in most cases lead to performance degradation.

Finally, it must be realised that most industrial applications are very complex and this makes the problems of acquiring and assembling the knowledge in the industrial environment much more severe than in traditional computing domains. The elicitation and conceptualisation processes are liable to be far more complex and attempts to prove the consistency of the knowledge will be very time consuming (Shirley, 1987; Johannsen, 1989). The full process is likely to take years rather than months. In the absence of a powerful methodology, we will have to work with inadequate tools for some time to come.

TECHNIQUES FOR KNOWLEDGE ACQUISITION

The techniques used in knowledge acquisition can be broadly divided into two categories - elicitation and machine induction. Strictly speaking, there is a continuum between human-human elicitation and automatic induction. Three general principles have been proposed for the acquisition process by Gruber and Cohen (1987). They are concerned with primitives and generalisations.

The first principle prescribes that task-level primitives should be designed in order to capture important domain concepts defined by the expert. The knowledge engineer must use a language of task-level terms rather than imposing implementation-level primitives. This principle stresses the importance of separating out acquisition from implementation and will be referred to again later in the paper. These task-level primitives must be natural constructs for describing information, hypotheses, relations, and actions, in the domain expert's language. This would suggest that task analyses should be combined with knowledge analyses (Borys and others, 1987; Johannsen, 1989).

The second principle suggests that explicit declarative representational primitives are preferrable to procedural descriptions. This principle is based upon the observation that most experts more easily understand declarative representations. Formulating procedural aspects in this way "can facilitate acquisition, explanation, and maintenance". Gruber and Cohen suggest that an expert should be asked "for the parameters of a domain that affect control decisions, and then to formulate control knowledge in terms of these parameters".

The third principle requires representations at the same level of generalisation as the expert's knowledge. Experts should not be forced to generalise except when absolutely necessary and they should not be asked to specify information not available to them. An example of an oversimplified generalisation would be the requirement to categorise a process variable as high, medium or low, when the expert needs to differentiate between many more steps or even a full range of numbers.

Knowledge Elicitation

A number of techniques for knowledge elicitation are now in use. They usually involve the collection of information from the domain expert either explicitly or implicitly. Originally, reports written by the experts were used but this technique is now out of favour since such reports tend to have a high degree of bias and reflective thought. Current techniques include interviews (both structured and unstructured), questionnaires, or observational techniques such as protocol analyses and walkthroughs.

Interviews. In a structured interview, the knowledge engineer is in control. Such interviews are useful for obtaining an overall sense of the domain. In an unstructured interview, the domain expert is usually in control, however, such interviews can, as the name implies, yield a somewhat incoherent collection of domain knowledge. The result can be a very unstructured set of raw data that needs to be analysed and conceptualised. It is obviously important for the knowledge engineer to have some knowledge of the domain before wasting the valuable time of the expert. This might be obtained through textbooks, manuals and other well-documented sources. Group interviews can be useful particularly in the phase of cognitive bias filtering.

Questionnaires and Rating Scales. Questionnaires can be used instead or in addition with interviews. The interviews can be standardised in question-answer categories or questionnaires can be applied in a more formal way. However, the latter should be handled in most cases in a relaxed manner for reasons of building up an atmosphere of confidence and not disturbing the expert too much when applied in actual work situations (Borys and others, 1987).

Rating scales are formal techniques for evaluating single items of interest by asking the expert to cross-mark a scale. Verbal descriptions along the scale such as from "very low" to "very high" or from "very simple" to "very difficult" are used as a reference for the expert. The construction, use and evaluation of rating scales is described very well in the psychological and social science literature. Rating scales can also be combined with interviews or questionnaires.

Observations. Observations are another technique for knowledge elicitation. They require no or little active participation of the expert. All actions and activities of the expert are observed as accurately as possible by the knowledge engineer who makes recordings of all the observed information. A special mixture of interview and observation techniques are the observation interviews (Matern, 1984; Johannsen, 1989). Sequences of activities are observed and questions about causes, reasons, and consequences asked by the knowledge engineer during these observations. The combined technique is very powerful because the sequence of activities is observable whereas decision criteria, rules, plans etc. are elicited in addition through what-, how- and why-questions.

Protocol analysis. Protocol analyses are useful for obtaining detailed knowledge. It can involve verbal protocols in which the expert thinks aloud whilst carrying out the task, or motor protocols in which the physical performance of the expert is observed and recorded (often on video tape). Eye movement analysis is an example of a very specialised version of this technique. Motor protocols, however, are usually only useful when used in conjunction with verbal protocols.

In a verbal protocol, the expert thinks aloud and a time-stamped recording is made of his utterances (Ericsson and Simon, 1984). In such protocols, the expert should not be allowed to include retrospective utterences. He or she should avoid theorising their behaviour and should "only report information and intentions within the current sphere of conscious awareness" (Newell and Simon, 1972). As a verbal protocol is transcribed, it is broken down into short lines corresponding roughly to meaningful phrases (see Kuipers and Kassirer, 1987 for examples of the technique). The technique can collect the basic objects and relations in the domain and establish causal relationships. From these a domain model can be built. The experience with the use of verbal protocols from the analysis of trouble-shooting in maintenance work of technicians is described by Rasmussen (1984).

It is important when using the transcription method not to allow any proposed expert systems technology (i.e., rule-based approach) to influence the selection of items. Fox, when examining failures in the performance of an expert system designed to diagnose Leukemia, noted that the expert systems technology used (in this case EMYCIN) strongly influenced the method used to "identify" useful information in the verbal protocols (Fox and others, 1987). He also comments "we are even less confident about knowledge that may be implicit or distributed in the structure of the protocols rather than concentrated in identifiable fragments".

Teachback interviewing. Another technique used in knowledge elicitation is "teachback interviewing". In this technique, the expert first describes a procedure to the knowledge engineer, who then teaches it back to the expert in the expert's terms until the expert is completely satisfied with the explanation. Johnson and Johnson (1987) describe this technique and illustrate its use in two case studies. Their approach is guided by Conversation Theory (Pask, 1974), in which interaction takes place at two levels - specific and general. The paper gives a useful set of guidelines on the strengths and weaknesses of the technique.

Walkthroughs. Walkthroughs are more detailed than protocol analysis. They are often better than protocol analysis because they can be done in actual environment which gives better memory cues. They need not, however, be carried out in real time. Indeed, such techniques are useful in a simulated environment where states of the system can be frozen and additional questions pursued.

Time lines. Time lines are tables in which several items of knowledge are contained in columns. The left column has to be filled with the time of occurrence of particularly interesting events such as failures or operator actions. Related information about the behaviour of the technical process, the automatic system and the human operators, at these times is recorded in separate columns with as much detail as is felt appropriate (Johannsen, 1989).

Formal techniques. Formal techniques include multidimensional scaling, Repertory Grids and hierarchical clustering. Such techniques tend to elicit declarative knowledge. The most commonly used is the Repertory Grid Technique (Kelly, 1955) based on personal construct theory. It is used in ETS (Boose, 1986) which assists in the elicitation of knowledge for classification type problems, and PLANET (Shaw and Gaines, 1986). In ETS, the expert is interviewed to obtain elements of the domain. Relationships are then established by presenting triads of elements and asking the expert to identify two traits which distinguish the elements. These are called constructs. They are then classified into larger groups called constellations. Various techniques such as statistical, clustering and multidimensional scaling are then used to establish classification rules which generate conclusion rules and intermediate rules together with certainty factors. The experts are interviewed again to refine the knowledge. ETS is said to save 2 to 5 months over conventional interviewing techniques. The system has been modified and improved and is now called AQUINAS (Boose and Bradshaw, 1988). To obtain procedural knowledge, techniques such as verbal protocols should be used.

Machine Induction

It is a common observation that experts have great difficulty in explaining the procedures which they use to arrive at decisions. Indeed, experts often make use of assumptions and beliefs which they do not explicitly state and are surprised when the consequences of these hidden assumptions are pointed out (Jackson, 1985). The inductive approach relies on the fact that experts can usually supply examples of their expertise even if they do not understand their own reasoning mechanisms. This is because creating an example set does not require any understanding of how different evidence is assessed or what conflicts were resolved to reach a decision. Sets of such examples are then analysed by an inductive algorithm (one of the most popular being the ID3 algorithm of Quinlan, 1979) and rules are generated automatically from these examples.

The problem with inductive techniques is that the rules induced depend both upon the example set chosen and the inductive algorithm used. There is no guarantee that the rules induced will be valid knowledge. The approach therefore normally involves a checking with the expert to see if the induced rules are reasonable. It is not uncommon to cycle a number of times through the induction process refining the knowledge base with the domain expert. Bratko (1989) gives a useful account of the techniques and the application of the ID3 algorithm. Hart (1987) has given guidelines on the appropriate use of inductive techniques:

- the technique is useful if there are documented examples or if they can be obtained easily. It is not suitable where an unpredictable sequence of observations drive the system (e.g., as in some real-time situations)
- the technique is consistent and unbiased and is very suitable for domains where rules form a major part of the knowledge representation
- induction provides the knowledge engineer with questions, results and hypotheses which form a basis for consultation with the expert
- there is no explanation for the rules produced. All output must be examined critically
- the process assumes that the example set is complete and current
- results should not be sensitive to small changes in the training set.

The inductive technique has been used for weather prediction, predicting the behaviour of a new chemical compound, diagnosing plant disease, symbolic integration, improved debt collection, and designing gas-oil separators. See Bratko and Kononenko (1987), Michalski and Chilausky (1980), and Mitchell and others (1983) for examples.

The technique is particularly useful when a great body of data exists about a process but the underlying rules are not known. Induction has been used therefore on large collections of historical process data about industrial plants in order to induce the rules of its operation. Once the rules are known the process can often be optimised. A well-known example of the use of this technique was at the Westinghouse Corporation where over $10,000,000 was saved (Westinghouse, 1984).

Another interesting use of inductive techniques which will have a wide application in industrial control is its use in conjunction with a qualitative model of the process. This was first carried out in the analysis of electro-cardiograms (Lavrac and others, 1985). A qualitative model of the domain is built. Then, components are failed and the consequences on measurable parameters determined for this failure. The process is repeated for each component and this builds up a complete set of examples of failure. The examples are then used as input to the ID3 algorithm and the rules governing the failure are induced. These form the basis for a diagnostic expert system. The technique will be discussed in more detail when we examine the application of inductive techniques to satellite power system diagnosis.

BOTTOM-UP OR TOP-DOWN?

There are two competing views about the knowledge acquisition task which might be described as bottom-up and top-down. The bottom-up proponents aim to prise data and concepts out of the expert and then iteratively refine it. Feigenbaum for example has described knowledge acquisition as "mining those jewels of knowledge out of their (*the experts*) minds one by one" (Feigenbaum and McCorduck, 1983). The implication is that deeper mining will reveal more relevant knowledge, but this assumes that there is a simple relationship between what is verbalised by experts and what is actually going on in their minds. Hayes-Roth and others (1983) claim that the building of expert systems "is inherently experimental" and is therefore characterised by rapid prototyping which is essentially a bottom-up process. The basic assumption underlying this bottom-up approach is that an expert system is based upon a large body of domain specific knowledge and that there are few general principles underlying the organisation of the domain knowledge in an expert's mind. However, the existence of underlying principles and causal relationships (Davies, 1983) may be an indication that expert knowledge is more domain independent than was assumed by Feigenbaum (1979). Breuker and Wielinga (1987), for example state that "In our experience over the past three years in analysing eight widely different domains a number of concepts have invariably recurred, such as 'procedure', 'process', 'quantification object'... and 'identification object'...... Such concepts are abstractions of real world knowledge". So "expert behaviour that is seemingly domain-specific may originate from higher level problem solving methods which are well-structured and have some degree of domain independence". Domain-independent aspects to the problem solving process have been observed by Pople (1982) in medical diagnosis tasks.

Breuker and Wielinga strongly support the top-down alternative and claim that there is a crucial step missing in the prototyping approach between the identification of the relevant characteristics of the domain and selection of solution methods, that of "the interpretation of the data into some coherent framework, a model, schema or canonical form". They equate it to the knowledge level of Newell (1980) or the "missing level" of Brachman (1979) in semantic network analysis. They propose five levels of knowledge analysis - identification, conceptualisation, epistemological, logical and implementational, and have developed these ideas into a knowledge acquisition methodology called KADS (Knowledge Acquisition and Documentation Structuring, Breuker and Wielinga, 1985).

An example of the application of the technique to insurance underwriting is given in Hayward and others, 1988.

A COGNITIVE TASK ANALYSIS APPROACH

Roth and Woods (1989) identify "failing to appreciate the demands of the task" as a major reason for the failure in current expert systems developments. They identify the iterative refinement approach (Hayes-Roth and others, 1983) used almost universally during the knowledge acquisition phase as the main cause. From a small prototype, the full system is developed through iterative refinements until the final delivery system is produced. They claim that "the amount of time and resources typically available for systems development in industry does not allow for the long term evolution of systems entailed in the iterative refinement approach" and point out that "architectures which are built based on consideration of a core set of examples will often not have the necessary structural hooks and processing mechanisms to deal with new cases that have complex aspects that had not been represented in the original set" (Bachant and McDermott, 1984). The correct handling of new cases then requires major restructuring of the knowledge rather than fine tuning. Experts often state rules to which there are exceptions, which are not usually revealed until much later.

They further point out that systems designed from a core set of examples often result in oversimplified representation of goals and constraints and this results in the optimisation of one dimension of the user's problem at the expense of ignoring other goals. One example from process control given in Roth and Woods concerned the design of an AI system to support operators in the start-up procedure for a boiler. The AI developers had originally concentrated upon a single goal - that of preventing shut-down - however the operators, in reality, had other goals to meet as well (shut-down could be caused by other sources). Thus, there were circumstances where sub-optimal performance on the boiler level goal was appropriate. They claim that their up-front analysis of the demands of the complete task enabled a much more realistic system to be built (Woods and Roth, 1988).

They suggest a multi-phase progression from initial informal interview techniques (to derive a preliminary mapping of the semantics of the domain), to more structured knowledge elicitation techniques (to refine the initial semantic structure), to controlled experiments designed to reveal the knowledge and processing strategies utilised by domain practitioners.

The first phase gives preliminary cognitive description of task to guide further analysis. It is important here not to home in on specific rules. One possibility is to get the experts to provide an overview presentation (Gammack and Young, 1985). Only when an overview of the semantics of the application has been developed can more structured techniques be used.

The second phase concentrates on how practitioners perform their tasks, thus, there is emphasis on observation and analysis of actual task performance. It will involve techniques such as critical incident review, discussion of past challenges, or the construction of test cases on which to observe the experts at work. During this phase, Roth and Woods also recommend the use of "expert panels" to obtain a corpus of challenging cases to identify critical elements and strategies for handling them.

The third phase uses observational techniques under controlled conditions to observe expert problem solving strategies. The practitioner is observed and asked to provide a verbal commentary. The task can be deliberately manipulated, for example, by forcing the expert to go beyond reasonably routine procedures. In some cases, the expert himself controls the information gathering. Alternatively, it is controlled by the observer. Each approach provides useful information, the former provides data on the diagnostic search process and the latter on the effect (or bias) of particular types of information on expert interpretations. Another useful technique is to compare the performance of experts with different levels of expertise, so as to isolate what factors really account for superior performance.

Roth and Woods make a strong case for a cognitive task analysis approach as compared with a technology-driven approach where knowledge acquisition concentrates upon AI representation mechanisms (e.g., rules and frames).

KNOWLEDGE ACQUISITION TOOLS

A large number of tools for supporting the knowledge acquisition process have been developed in the academic environment and some of these have been mentioned already. The general aim of all these tools is to minimise the number of iterations needed for the whole knowledge engineering process by bridging the gap between the problem domain and the implementation. Boose and Gaines (1988) give a brief summary of the main tools under development and provide a summary. Some tools endeavour to make the process fully automatic. KRITON (Diederich and others, 1987), for example, has a set of procedures pre-stored - interviews, incremental text analysis, and protocol analysis. Repertory Grids are used to pull out declarative knowledge. An intermediate knowledge representation system is suggested for supporting the knowledge elicitation techniques. The knowledge representation scheme involves a propositional calculus for representing transformations during the problem solving process and a descriptive language for functional and physical objects. This is then translated semi-automatically into the run-time system but this commits the knowledge engineer to a particular representation. Other tools (for example KADS and ACQUIST) merely provide a set of tools to aid a more methodological approach. Thus, KADS aims only to produce a document describing the structure of the problem in the form of a documentation handbook.

KRITON supports only bottom-up knowledge acquisition but KADS supports both top-down and bottom-up approaches. KADS supports bottom-up through a hypertext protocol editor (PED) and hierarchies are developed and manipulated by a context editor (CE). Top-down is supported by a set of interpretation models each describing the meta-level structure of a generic task.

The KADS methodology is based upon the following principles:

- knowledge and expertise should be analysed before the design and implementation starts, i.e., before an implementation formalism is chosen
- the analysis should be model driven as early as possible (see also Su, 1988)
- expert problem solving should be expressed as epistemological knowledge
- the analysis should include the functionality of the prospective system
- the analysis should be breadth-first allowing incremental refinement

- new data should only be elicited when previous data has been analysed
- all collected data and interpretations should be documented.

The approach produces a four layer model of expertise (Hayward and others., 1988):

- definition of the domain concepts and their static relationships
- definition of relations arising in a task context which are concerned with dynamics and are expressed in the inference structure
- specification of how the available inferences can be used to undertake a particular task
- definition of how the task level may be controlled. This is the least developed part of the model.

KEATS-1 (Motta and others, 1988) provided a Cross Reference Editing Facility (CREF) and a Graphical Interface System (GIS), to support data analysis and domain conceptualisation. CREF organises the verbal transcript text into segments and collections and GIS allows the knowledge engineer to draw and manipulate domain representations on a sketch pad. In KEATS-2, these have been replaced by ACQUIST, a hypertext application for structuring the knowledge from the raw text data. Fragments from the data are collected round concepts, concepts are factored into groups, and groups into meta-groups. Links can then be defined between any of these entities. The emerging structure is displayed graphically. ACQUIST provides support for both bottom-up approaches (fragments to concepts to groups to meta-groups) and top-down approaches (using what are called coding sheets on which a "caricature of the observed behaviour of the domain expert" is captured). In this approach, the knowledge engineer uses a predefined abstract model to guide the knowledge acquisition process. Use of such models (even if incomplete or inadequate) can dramatically improve the knowledge acquisition process. The coding sheet is a set of hypertext cards.

The knowledge acquisition tool by Strothotte and Sack (1988) is based on the assumption that it is often quite natural for domain experts to express themselves through diagrams. These diagrams and the related dialogue allow the expert to transfer the knowledge in a way often used among humans. The diagrams are drawn with lines by using a simple graphical editor. All details which are important for describing certain objects have to be included. Then, the tool extracts features from the diagram by applying computational geometry and image processing algorithms. Clarifying questions are then asked by the computer to the expert about features which have to be further specified. Knowledge about the objects and their relationships are derived and stored in the final knowledge base together with the diagram itself. Thus, information content of diagrams can be entered semi-automatically into the knowledge base. Diagrams can be re-used if necessary. Strothotte and Sack stated that their knowledge acquisition tool needs to be combined with a tool for textual knowledge in all those domains which allow to describe only some types of knowledge in a diagrammatic way. A further limitation may be that the expert will be forced to overspecify irrelevant details.

A further knowledge acquisition tool is ROGET (Bennett, 1985). It conducts a dialogue with a domain expert in order to acquire his or her conceptual structure. ROGET gives advice on the basis of abstract categories and evidence. Initial conceptual structures are selected on this basis. Only a small set of example systems were tested.

The use of Pathfinder networks for knowledge acquisition was proposed by Esposito and Dearholt (1988). It is a tool for the identification of conceptual structures with a sophisticated interactive graphics system for network display and manipulation. Experts are asked to make simple similarity judgements and answer specific questions. This graph-theoretic tool has been used in investigations with network models of human semantic memory utilising estimates of psychological distance. Path Algebra techniques (Alty and Richie, 1985) provide a more generalised tool for such approaches.

The systematic acquisition of knowledge about the fault behaviour of a technical system was suggested by Narayanan and Viswanadham (1987). A procedure involves the development of a hierarchical failure model with fault propagation digraphs and cause-consequence knowledge bases for a given system. It uses the so-called augmented fault tree as an intermediate knowledge representation. Fault propagation digraphs describe the hierarchical structure of the system with respect to faults in terms of propagation. The cause-consequence knowledge bases characterise failures of subsystems dependent on basic faults by means of production rules. The knowledge acquisition process can be reduced to defining parameters required by the knowledge representation scheme and transforming human expertise into these parameter values. The augmented fault tree is a conceptual structure, which describes causal aspects of failures as in conventional fault trees but additionally also probabilistic, temporal and heuristic information. The production rules of cause-consequence relations are derived from the augmentated fault tree by decomposing it into mini fault trees. The proposed methodology has reached a relatively high level of formal description. However, it cannot yet deal with inexact knowledge by using ranges of parameters. An example of a failure event in a reactor system is given.

ELICITATION EXAMPLES

The application of the knowledge elicitation techniques for industrial expert systems will be shown with two examples. The first is the task and knowledge analysis in power plants performed with using observation interviews, questionnaires, and time lines, and the second is the knowledge and task analysis performed in parallel to the construction of a failure management expert system for space systems.

Knowledge Analysis in Power Plants

An extensive task and knowledge analysis has been performed in coal-fired power plants and in a power plant school by the first author of this paper and his research group (Borys and others, 1987; Johannsen and others, 1987; Johannsen, 1989; Sundström, 1989). The work is part of the ESPRIT-GRADIENT project on "Graphics and Knowledge Based Dialogue for Dynamic Systems" which is partially supported by the Commission of the European Communities and is performed in cooperation between the research groups of the two authors of this paper together with two industrial companies from Germany and Denmark and a Belgian university.

A thermal power plant consists basically of a water-steam cycle involving a boiler, turbines, condenser and feedwater system, as well as a generator. The automation system or supervision and control of the plant is hierarchically organised into drive, group and control levels. When higher levels of the automation system fail, the shift leader needs to operate the plant with less automation on the lower levels. It is intended to support the human operators in these situations using expert systems. Several cooperative expert systems are developed within the whole research consortium, mainly for diagnosis of causes of failures, prediction of consequences of failures, knowledge-based alarm handling, procedural support of operator behaviour and plan recognition with operator input evaluation. Also, a graphical expert system will be developed. It will be based on intelligent graphical editors which contain knowledge-based support functions for graphical picture designers.

The task and knowledge analyses were performed during a period of three years in several power plants during day and night shifts as well as in a power plant school. Observation interviews, questionnaires, and time lines were used as elicitation techniques. During later stages of the elicitation process, the analyses were restricted to failure situations in the pre-heating system. The knowledge was collected for two reasons - to build a diagnostic expert system for supporting operators and to construct a user model which will form part of an intelligent graphical editor to support designers. After using interviews and observations with power plant operators and operational engineers, a number of different questionnaires were applied. A state-oriented questionnaire with a total number of 29 questions helped to build frames of knowledge for each substate of the plant. The questionnaire was structured into the five groups of substate description, activities, mental models of the operator, effects of activities, and suggestions for improvement with respect to the operator's work. Another questionnaire was designed to capture expert strategies and knowledge representations available to operators in failure situations. This failure-oriented questionnaire was based on a general separation of each failure situation into several phases of fault management such as detection, diagnosis, localisation, compensation and correction (Johannsen, 1988).

Time lines were used in later stages of the task and knowledge analyses. They were mainly applied in the power plant school where it is possible to freeze a system state of interest in the simulator, measure the time of occurrence and collect the related knowledge from all available sources without any pressure. For the diagnostic expert system, knowlege collected and formalised using the time lines approach includes alarm messages, affected components, parameters, actions of the supervision and control system, and human operator actions. The knowledge elicitation for the user model of the intelligent graphical editor is concerned with decision alternatives and associated information search behaviour of the operator (Sundström, 1989). It is related to the information processing goals of categorisation of states, choice of actions, and evaluation of outcomes. This information was also elicited using time lines. The information gathered in each of the two types of time line is different but both types are related to each other with respect to time for the same failure situation. Each of these time lines can be viewed as a kind of intermediate knowledge representation on paper. This technique is a useful tool for further knowledge formalisation and for knowledge implementation. Time lines can easily be discussed with domain experts before any implementation needs to be accomplished. In case of this project, the time lines, as well as first prototype implementations, were evaluated by a power plant instructor (the domain expert) together with two researchers (the knowledge engineers).

Construction of Failure Management Expert Systems

Another example of the application of knowledge elicitation techniques was given in the construction of failure management expert systems by Malin and Lance (1987). A knowledge and task analysis was performed in parallel with the construction of an expert system for failure management in a space station prototype device. A device for removing carbon dioxide from cabin air was selected. The expert system is called FIXER (Fault Isolation Expert to Enhance Reliability).

The knowledge engineering process for developing FIXER was performed by three persons: an expert in life support systems, a cognitive scientist and a consulting knowledge engineer. The prototype was developed through close cooperation between the systems expert and the cognitive psychologist who was the main knowledge engineer. The knowledge for the trouble-shooting expert system was based on device design information and experience with similar devices. Operational knowledge about trouble-shooting failures in the device was not used. The goals, tasks, knowledge, methods, and design decisions involved in constructing the failure management expert system were intensively observed and analysed. Thus, observations were the selected knowledge acquisition technique for the investigation of the whole design process. Five design tasks were observed and analysed:

- allocation of failure management functions and interface definition
- analysis of failure events, fault modes, and effects
- selection and construction of measurements and test procedures
- analysis of fault-symptom patterns and construction of diagnosis procedures, and
- construction of procedures for failure effects management and maintenance.

Furthermore, tasks for the revision of the failure management software were observed and evaluated. The development and analysis effort for FIXER required about 30 full working days for each person, the domain expert as well as the knowledge engineer, distributed over a time of five months. The software revisions were performed one year later and required half a day for the expert and seven days for the knowledge engineer.

One of the conclusions drawn from the experience with this cooperative knowledge elicitation technique is that "the knowledge acquired from the expert should include much more than the rules and procedures for failure management". The expert's choice of strategies, and supporting analyses and models need also to be represented explicitly in order to deal with limitations of the failure management expert system and the need for later revisions. Further, it was observed during the whole design process that mental models of the device and its behaviour were important. Such models are under development by Sundström and Johannsen (1989).

INDUCTION EXAMPLES

Two examples will be given of the use of inductive techniques in the industrial environment. - its use in British Petroleum in the design of gas-oil separator plant, and its use at the European Space Agency for the design of expert systems in satellites.

Use of Induction by Engineers

The gas-oil system assists engineers to design gas-oil separators. The underlying hydrocarbon production separation process is quite complicated relying on a variety of knowledge sources such as that in manuals, codes of practice, space limitations, and by the crude oil quantity and the gas quality required. Key factors included the delivery system, a user friendly interface involving graphics design, and interfacing to existing FORTRAN routines. Gas-oil is a large system - containing over 2,500

此为试读,需要完整PDF请访问: www.ertongbook.com