The South African Institute of Computer Science
and
Information Technology

The 1997 National Research and Development Conference

Riverside Sun
Vanderbijlpark
13 & 14 November

Hosted by

Potchefstroomse Universiteit
vir Christelike Hoër Onderwys

The Department of Computer Science and Information Systems
Potchefstroom University for Christian Higher Education
Vaal Triangle Campus

SAICSIT 97

PROCEEDINGS

Edited by L.M. Venter & R.R. Lombard

The South African Institute of Computer Science
and
Information Technology

Proceedings

of the

The 1997 National Research and Development Conference

Towards 2000

Riverside Sun
Vanderbijlpark
13 & 14 November

Edited by
L.M. Venter
R.R. Lombard

ISBN 1-86822-300-0

Printed and Binded by Xerox Printers, Potchefstroom

The views expressed in this book are those of the individual authors

Foreword

This book contains a collection of papers presented at a Research and Development conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT). The conference was held on 13 & 14 November 1997 at the Riverside Sun, Vanderbijlpark. Most of the organization for the conference was done by the Department of Computer Science and Information Technology of the Vaal Triangle Campus, Potchefstroom University for Christian Higher Education.

The programming committee accepted a wide selection of papers for the conference. The papers range from detailed technical research work to reports of work in progress. The papers originate mainly from Academia, but also describe work done in and for Industry. It is hoped that the papers give a true reflection of the current research scene in Computer Science and Information Technology in South Africa. Since one of the aims of the conference is Research development, the papers were not subjected to a refereeing process.

A number of people spent numerous hours helping with the organization of this conference. In this regard, we wish to thank the members of the Organizing committee, and the Programming committee who had very little time to screen the abstracts and compile the program. A special thanks goes to the secretary of the department, Mrs Helei Jooste, whose very able work was interrupted by the birth of her first child.

Organizing Committee

Conference General Chairs
Prof. J.M. Hattingh (PU for CHE)

Organizing Chair
Prof. Lucas Venter (PU for CHE)

Organizing Committee
Mrs. S. Gilliland
Mr. J.P. Jooste
Mr. R.R. Lombard
Mrs. M. Huisman

Secretariat
Mrs. H. Jooste

Program Chair
Prof A de Waal (PU for CHE)

Program Committee
Prof D. Kourie (UP)
Prof. C. Bornman (UNISA)
Prof. L.M. Venter (PU for CHE)

Table of Contents

List of Contributors

S.A. Ajila
Department of Mathematics and Computer Science
National University of Lesotho
Roma, 180
Lesotho

L. Baart
Department of Mathematics
Vaal Triangle Campus of the PU for CHE
PO Box 1174
Vanderbijlpark, 1900

L. Barnard
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

S. Berman
University of Cape Town
Rondebosch, 7701

L. Bester
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth 6000

J.M. Bishop
Computer Science Department
University of Pretoria
Pretoria, 0002

L. Botha
Computer Science Department
University of Pretoria
Pretoria, 0002

R.A. Botha
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

B. Braude
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Wiwatersrand
Private Bag 3
Wits, 2050

T. Breetzke
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

C. Brink
University of Cape Town
Rondebosch, 7700

M. Bruynooghe
Departement Computerwetenschappen
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

S. Buffler
University of Capetown
Rondebosch, 7701

M.A. Coetzee
Department of Mathematics
PU for CHE
Private Bag X6001
Potchefstroom, 2520

R. Cools
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

E. de Preez
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

D.A. De Waal
Department of Computer Science and Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2531

B. Dekenah
The Board of Executors

M. Denecker
Departement Computerwetenschappen
Katholieke Universiteit Leuven
Celestjinenlaan 200A
B-3001 Heverlee
Belgium

M. Dunley-Owen
Department of Information Systems
University of Cape Town
Rondebosch, 7700

R. Fiqueira
University of Cape Town
Rondebosch, 7701

A. Foster
Department of Computer Science
University of Cape Town
Rondebosch, 7701

C. Gee
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Witwatersrand
Private Bag 3
Wits 2050

M. Hajek
Department of Computer Science
University of Durban Westville
Private Bag X54001
Durban, 4000

M.L. Hart
Department of Information Systems
University of Cape Town
Rondebosch, 7700

J.M. Hattingh
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

S. Hazelhurst
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits 2050

H.A. Kruger
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

J.W. Kruger
University of the Witwatersrand
Private Bag 3
Wits, 2050

M.F. Kruger
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M.T. Lang
Eskom Information Technology Department

D. Laurie
Department of Mathematics
Vaal Triangle Campus of the PU for CHE
PO Box 1174
Vanderbijlpark, 1900

D. Lubinsky
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

R. McLeod
Saltire Software Inc.
Tigard
Oregon
U.S.A

H.J. Messerschidt
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

M. Mphahlele
Department of Computer Science
University of the North
Private Bag X1106
Sovenga, 0727

G.D. Oosthuizen
Department of Computer Science
University of Pretoria
Pretoria, 0002

J. Owen
University of Cape Town
Rondebosch, 7701

D. Petkov
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

O. Petkova
Technikon Natal
PO Box 101112
Scotsville, 3209

N. Pillay
Department of Financial Studies
Technikon Natal, Pietermaritzburg
PO Box 101112
Scotsville, 3209

L. Pluym
Katholieke Universiteit Leuven
Celestijnenlaan 200A
B-3001 Heverlee
Belgium

K. Prag
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

P. Premjeeth
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

V. Ram
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

J. Robertson
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

S. Rock
Department of Artificial Intelligence
Edinburgh University
United Kingdom

J. Roos
Department of Computer Science
University of Pretoria
Pretoria, 0002

I. Sanders
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

K. Sandrasegaran
Department of electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

C. Schoder
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Sears
Department of Mathematics
University of the Witwatersrand
Private Bag 3
Wits, 2050

E. Senior
International Center for Waste Technology
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209

N.B. Serbedzija
GMD FIRST
Rudower Chausee 5
D-12489 Berlin
Germany

S.L. Serutla
Department of Computer Science
The University of Pretoria
Pretoria, 0002

T. Steyn
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M. Thielscher
Fachgebiet Intellektik, Fachgebiet Informatik
Technische Hochschule Darmstadt
Alexanderstrasse 10
D-64283 Darmstadt
Germany

T. Thomas
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

M. Thomas
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

S. Tjasink
University of Cape Town
Rondebosch, 7700

E. Viljoen
Department of Computer Science and
Information Systems
University of South Africa
PO Box 392
Pretoria, 0001

E. Voges
University of Cape Town
Rondebosch, 7701

R. Von Solms
Faculty of Computer Studies
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth, 6000

A.J. Walker
Software Engineering Applications Laboratory,
Electrical Engineering
University of the Witwatersrand
Private Bag 3
Wits, 2050

P. Warren
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

M. Watzenboeck
University of Botswana
Private Bag 0022
Gaberone
Botswana

K.L. Wortmann
Department of Computer Science
University of Natal, Pietermaritzburg
Private Bag X01
Scotsville, 3209

Explanation Facilities in Expert Systems Using Hypertext Technology

THAYNE BREETZKE AND THEDA THOMAS

Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000
(thayne@ml.petech.ac.za or theda@ml.petech.ac.za)

Abstract

The process of delivering suitable explanations in expert systems has been a significant area of research since the inception of expert systems. Their ability to provide explanations of the reasoning process behind a conclusion in order to justify their results is an important human-like characteristic. However, a number of problems with explanation facilities in expert systems have become apparent. This article briefly describes a method of merging expert system and hypertext technology in order to improve the explanation facilities in expert systems. Essentially, this involves capturing the reasoning process employed during a consultation with an expert system and presenting it in a hypertext form thus allowing the user to browse through it freely. The hypertext links are generated by following the reasoning process used by the expert system. In addition, hypertext links to related information may be followed to obtain additional explanations of terms and concepts.

Keywords: expert systems, explanations, hypertext
Computing Review Categories: I.2.1, H.5.1, H.1.2

1. INTRODUCTION

One of the important human-like characteristics of expert systems, and a key feature of the technology, is their ability to provide explanations of the reasoning process behind a conclusion in order to justify their results. This is a critical distinguishing characteristic of expert systems over conventional systems and has long been cited as being a vital feature of interactive expert systems [1, 7]. Since not all solutions are obvious or unambiguous, advice may seem arbitrary or random without explanation facilities. However, a number of problems with explanations have become apparent. If users are to fully understand the response from an expert system, it is necessary that these problems be addressed.

What is needed is an explanation facility which is able to explain its reasoning in an easy-to-follow, non-cryptic, useful, meaningful and flexible manner. It should offer information that users want, presenting this material in ways that make sense in terms of the users' own experience, and should be able to engage in dialogue with its users. In an attempt to solve many of these problems the author undertook a study to investigate the merging of another popular technology, namely hypertext, with expert systems.

Hypertext can be defined by contrasting it with traditional text. Traditional text is sequential, that is, there is a linear sequence which defines the order in which the text is to be read. In contrast, hypertext is non-sequential. There is no single order defining the sequence in which the text is to be read [6]. The emphasis is placed more on the exploration of an information space rather than on the access of the information in a single sequential order. Users can move freely

through the information space and access the information that satisfies their own needs.

This article briefly describes a method of merging expert system and hypertext technology in order to improve the explanation facilities of expert systems. Essentially, this involves capturing the reasoning process employed during a consultation with an expert system and presenting it in a hypertext form thus allowing the user to browse through it freely. In addition, hypertext links to related nodes may be accessed to provide additional explanations of terms and concepts. It must be mentioned at this point that this study focused on rule-based expert systems since they are currently the most common form of expert systems [2].

2. PROBLEMS WITH EXPLANATIONS

As expert systems have progressed from research to practice, the users of these systems have progressed from almost exclusively the authors of the system to highly trained domain experts and now even novice end-users. However, the methods used to produce explanations have remained relatively constant. Consequently, a number of problems with explanations have become apparent. If users are to fully understand the response from an expert system, it is necessary that these problems be addressed. This section briefly describes these problems.

2.1 Explanations Are Often Limited and Inflexible

Explanation facilities frequently provided by rule-based expert systems are typically little more than the ability to ask *how* and *why* questions, that is, the ability to explain why a question is being asked and how a conclusion was reached. Furthermore, most facilities simply reproduce the rules that were used (or fired) in the last part of the consultation to reach the conclusion [3]. These are usually stated concisely and sometimes very briefly in their original form, making them difficult to follow by an end-user.

As an example, consider the following response from the system after the user has asked *why* it wants to know what the result of test *a* is:

IF test *a* is positive
THEN symptom *b* is confirmed

In addition, consider the response from the system after asking *how* it was determined that medicine *x* should be prescribed. The system displays the rule which inferred the fact:

IF the patient has disease *c*
THEN prescribe medicine *x*

Explanations such as these are shallow, limited and inflexible. In both cases, the system could display the chain of rules that it is trying to execute or has already executed. However, this approach often results in a long list of rules and is therefore only suitable for very small expert systems. It is critical that an explanation of a system's reasoning process is delivered in a form

that is both acceptable and useful to an end-user.

2.2 Explanations Are Often Ignored

Another problem which is closely related to the first is that explanations often tend to be ignored, possibly because they are too shallow and, thus, do not offer sufficient and suitable explanation. If users are unable to understand the explanations given then they will tend to ignore them. The result is an under-utilized explanation facility and, possibly, an expert system that fails to fully serve its purpose.

2.3 The Users' Level of Skill Is Often Ignored

Some explanations are cryptic, difficult to follow or too shallow. Explanations like these are extremely useful to the knowledge engineer, and possibly the human expert, when developing and debugging the system. However, since end-users are typically far less skilled and knowledgeable in the application domain than those who were involved in the construction of the expert system, they are less relevant to the end-user of the expert system, especially where the style of the system is that of an expert consultation or tutor. Thus, since explanations are used by a number of persons who each use them for different purposes and each have a varying level of expertise in the expert systems domain, explanations must be suitable for all the users of the expert system. This puts new demands on the explanation facility, demands that may be beyond the ability of traditional expert system explanation facilities.

The *how* and *why* explanations available in most expert systems are not sufficient to overcome the problems associated with varying levels of expertise of users. Explanation systems are still being built that ignore the fact that users may need to know more about the domain. Explanations in these systems entail a one-way flow of information from the system to the user. The lack of attention to the information needs of users is almost certain to limit the utility of any system [5]. Users need different types of explanations, for example, they may require more instructions if they are novice users, more information about something, more feedback and explanations of unfamiliar terms or concepts used in explanations, that is, explanations of the knowledge itself.

3. DESIRABLE FEATURES OF AN EXPLANATION FACILITY

The primary goal of an explanation facility is to provide clear explanations, regardless of the types of explanations it provides. Explanations must be clear in order to be understandable and useful. It is beneficial to construct a list of desirable features of explanations from the problems identified in the previous sections. These can then serve as guidelines for the development of an explanation facility that satisfies this primary goal. Table 1 shows the list of these guidelines.

TYPES OF EXPLANATIONS The facility should deliver justification explanations because they are more effective than strategic explanations. *How* and *why* explanations are still applicable, although, if available, they should not be shallow. A session log or a reasoning trace can be provided to assist the developer, but lengthy lists can be less useful to the expert system user.
UNDERSTANDABILITY The system should be able to explain why it is asking a question in such a manner that the user can understand the explanation clearly. The system should be able to explain how it reached its conclusions in such a manner that the user can understand the explanation clearly.
EXPLANATION COMPLEXITY The explanations should be deeper than simple answers to *how* and *why* questions. The system should provide more than the simple reproduction of a rule in its original form. The inclusion and separation of different kinds of knowledge is necessary. The system should allow a user to access additional domain knowledge. The system should cater for the varying level of expertise of its users. The process of explanation should be a dialogue between the user and the system.
EASE OF EXPLANATION DEVELOPMENT AND MAINTENANCE The development and maintenance of explanations must be carried out easily and with minimum involvement from the developer. The development and maintenance of additional domain knowledge must be carried out easily and with minimum involvement from the developer.

Table 1 Guidelines for the development of a suitable explanation facility.

4. A HYPERTEXT EXPLANATION FACILITY

For most explanation systems, the explanation process involves a one-way flow of information to the user. This is in direct contrast to human interaction in which the process of explanation is highly interactive. This section briefly discusses an approach to explanations in expert systems

which involves the merging of hypertext technology, an already proven and highly interactive technology, with expert systems. In doing so, it tries to address many of the problems identified with explanation facilities.

4.1 An Overview of this Approach to Explanations

Essentially, this approach involves capturing the reasoning process employed by the expert system during a consultation, presenting it in a hypertext form and allowing the user to browse through it freely. Furthermore, the approach enables additional domain knowledge to be more fully incorporated into the hypertext explanations, and thereby enabling the expert system to more effectively support the users information needs. In this way, the user is able to selectively obtain information and explanations about terminology and keywords used in explanations.

As shown in Figure 1, the problem-solving knowledge, in the form of production rules, is used when conducting a consultation with the user. During the consultation the expert system constructs a reasoning chain representing the rules it has tried and/or fired. When the user requests an explanation, language techniques of expression (rhetoricals) are used to express the reasoning to form a browsable hypertext network which serves as the final explanation. In addition, this approach is able to convey an understanding of the expert system's domain by allowing access to additional domain knowledge from within this network to provide further explanation of pertinent topics.

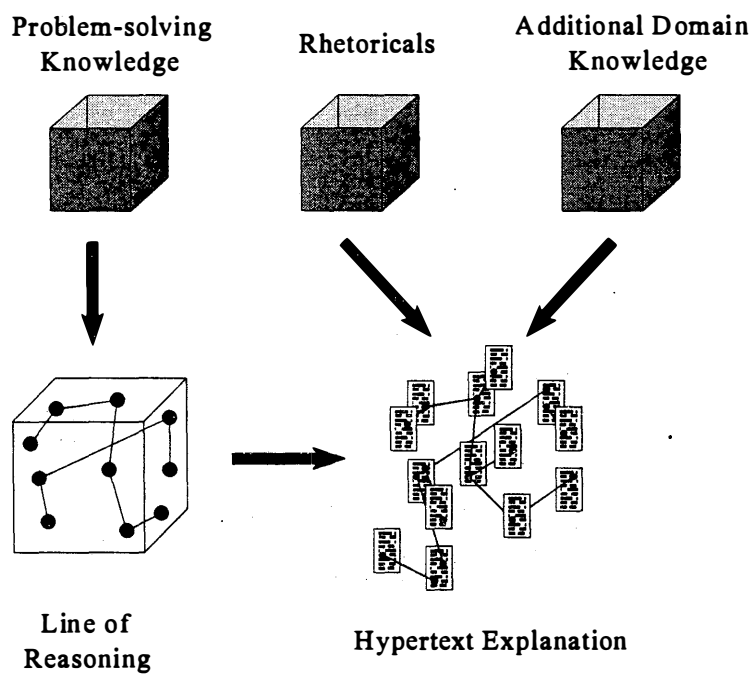

Figure 1 The delivery of hypertext explanations in expert systems.

Consider the line of reasoning shown in Figure 2. When translated into English it reads as

follows:

> The top-level goal was to prove **A**. Rule **4** was used to prove **A**. However, to prove **A** it was first necessary to prove **B**, **C** and **D**. Rules **8**, **10** and **6** were used to prove **B**, **C** and **D** respectively. However, the proof of **B** required the proof of **E** and **F**. Proof of **E** and **F** was obtained from the user's response to two questions posed by the expert system. To prove **C**, it was first necessary to prove **G**. **G** was proved from the user's response to a question. To prove **D** it was necessary to first prove **H** and **I**. Proof of **I** was obtained from the user's response to a question while Rule **12** proved **H** after **J** and **K** were proved from the user's response to two questions.

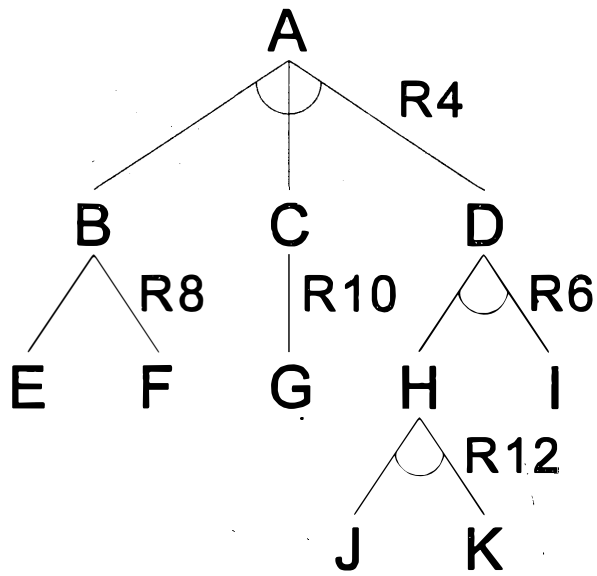

Figure 2 An example of an expert system's line of reasoning.

This description of the line of reasoning explains precisely how the expert system determined **A**. However, a description like this can be difficult to follow, confusing and overwhelming, especially when many rules were applied to reach the conclusion. The approach to explanation suggested in this study tries to convey this information and additional domain knowledge to the user in an effective manner so that it is easier to comprehend. This is achieved by modelling the expert system's reasoning in a node and link structure. Thus, the above example can be divided into a number of segments with each segment containing pointers to related segments. Together, the segments and pointers form a hypertext network with the line of reasoning serving essentially as the framework of the network. Once the segments and pointers become part of the network the segments are referred to as nodes and the pointers are referred to as links. Additional domain knowledge segments residing in the hyperbase augment this network by participating in relationships with existing nodes in the network. The participating knowledge segments are also referred to as nodes.

An example of a hypertext network generated from the line of reasoning illustrated in Figure 2

is shown below in Figure 3.

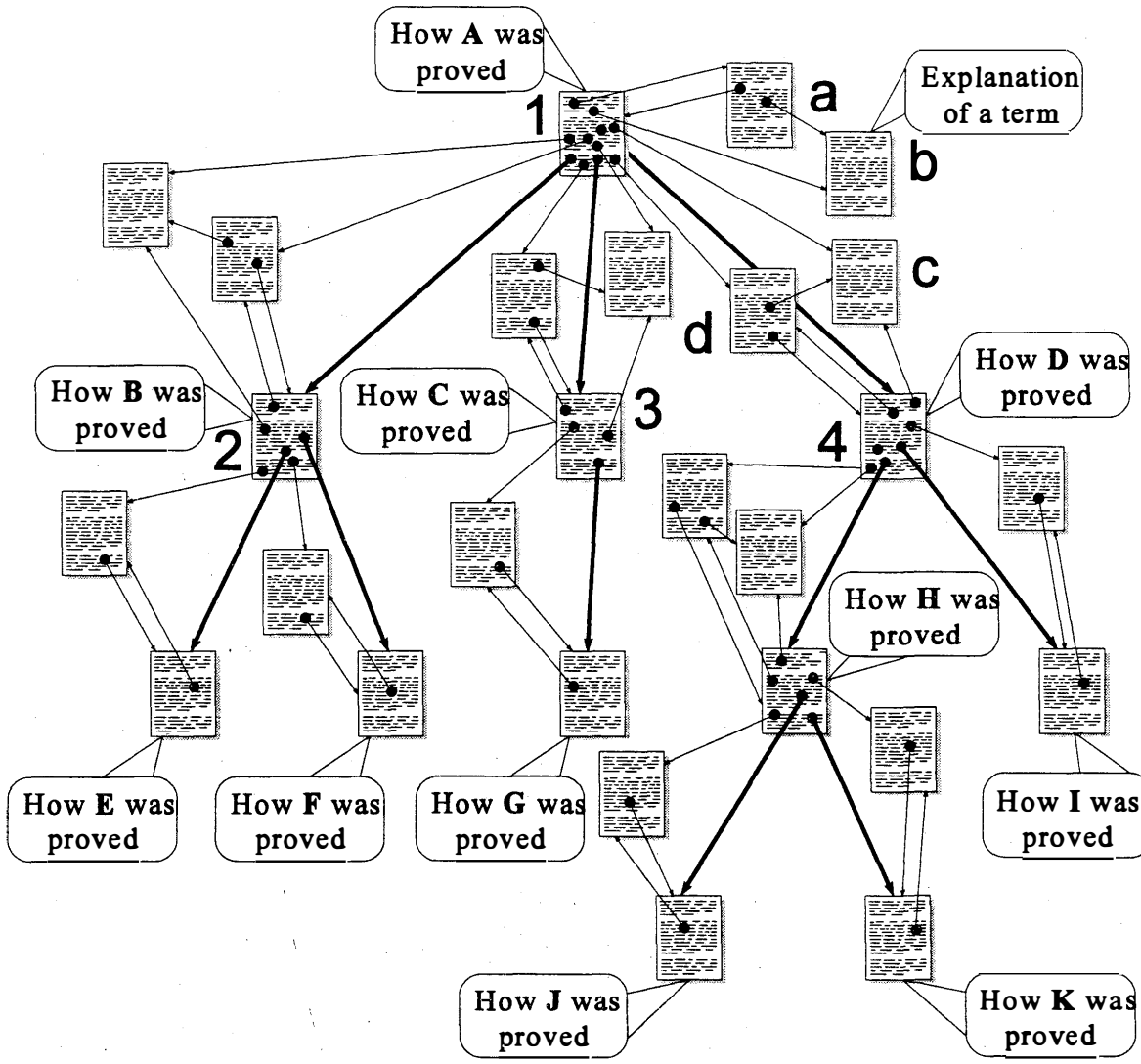

Figure 3 An example of a hypertext network generated from the line of reasoning shown in Figure 2.

An arrow reveals the presence of a link from the node at the tail end of the arrow to another node pointed to by the arrow. In addition, the arrows shown in bold indicate which part of the network is based on the expert system's line of reasoning. Non-bold arrows indicate links to nodes that convey additional domain knowledge. Notice how the line of reasoning shown in Figure 2 serves as the framework for the hypertext network and that additional nodes are present to support this framework by providing additional domain knowledge.

The node labelled **1** explains how **A** was proved and that the proof of **B**, **C** and **D** (shown in Figure 2) was used to prove **A**. An explanation of how **B**, **C** or **D** was proved can be obtained

by following a link from node **1** to node **2**, **3** or **4** respectively. Node **2** explains how **B** was proved and that the proof of **E** and **F** was used to prove **B**. Similarly, node **3** explains how **C** was proved and that the proof of **G** was used to prove **C**. In this way the expert system's reasoning can be explored in as much detail as is necessary.

Nodes **a**, **b**, **c** and **d** are four of the many nodes present which convey supplementary domain information, for example, explanations of terms or concepts present in other nodes. Thus, if a user is unfamiliar with a certain term used in an explanation, he or she can simply follow a link to an explanation of the term (assuming an explanation exists for it). For example, while examining the reasoning process, a physician who is unfamiliar with the terminology used or with the details of a newly identified disorder can obtain greater insight by requesting additional information.

An expert system which is capable of generating explanations in a hypertext form requires an architecture that differs to that of a traditional expert system in a number of ways:

i. It needs to allow for the storage and maintenance of the additional domain knowledge. Since this knowledge is unable to be represented in the same form as that of the problem-solving knowledge (that is, by using rules) extensions to the knowledge base are necessary to accommodate this knowledge.

ii. It needs to allow for the retrieval of the expert system's line of reasoning and additional domain knowledge to generate explanations in a hypertext form.

iii. It needs to be able to present the hypertext explanations in a clear and unambiguous way and process user requests.

The authors have developed a structured rule-based expert system shell, called *HyperMagic*, which implements this approach to explanations in expert systems. It supports a knowledge base that, in addition to storing problem-solving knowledge, contains knowledge about the terminology and concepts, or more specifically the variables and values, used in this problem-solving knowledge. These knowledge segments become accessible to the user when he or she browses a hypertext explanation. The knowledge base also holds the mask text database which contains information used by the explanation facility to generate more English-like rules, rather than simply displaying rules in their original form.

The explanatory module and the user interface collectively form the consultation facility and were designed to provide interactive dialogue with the user. One of the most important components of the architecture is the hypertext explanation generator which is located in the explanatory module of the consultation facility. Its function is to make use of the expert system's line of reasoning, the domain knowledge in the hyperbase and the information in the mask text database, if available, to generate explanations in a hypertext form. The approach used by the hypertext explanation generator to create the links within the nodes it generates is similar to that used to create the links in the documents in the World Wide Web (WWW). Once generated, these explanations are passed to the hypertext presentation module which is responsible for presenting them to the user and handling user requests to follow additional links.

4.2 The Types of Explanation Facilities

The expert system shell offers two types of explanation facilities which can be initiated by the user via the user interface of the expert system:

i. A *how* explanation facility which explains how a variable received its value.
ii. A *why* explanation facility which explains why a question is being asked.

These explanation facilities are discussed in the following sections, with examples of screens given in Figures 4 and 5. In all the examples that follow, the presence of a link is indicated using underlined text. Note that in each node generated by the hypertext explanation generator the variable names and values that are shown, or any text segments associated with the variables or values, are designated as links to nodes conveying their related knowledge segments, unless no related knowledge segment exists. Thus, users are able to ask the system to provide additional domain knowledge about a variable, value or a text segment associated with a variable or value by simply selecting it. For example, if the user selects the word *modem*, a node would be displayed containing the knowledge segment associated with the variable *modem*.

4.2.1 The *HOW* Explanation Facility

When initiated by the user, this facility displays a list of all the facts already known and allows the user to select one to obtain an explanation of how it was inferred. If the selected fact was inferred from the user's response to a question then this is explained to the user, otherwise a hypertext network is created based on the expert system's line of reasoning to form the collective explanation. Each node present in that segment of the network that is based on this line of reasoning is called a *how* node and explains how a fact was inferred or, more specifically, how a variable acquired its value. The *how* node lists the facts used to infer this fact as well as the rule that was applied, unless the fact was determined from the user's response to a question in which case this is stated. Alongside each fact listed is a *how* link that points to another *how* node explaining how that fact was inferred. Thus, by moving from one *how* node to another the user is able to explore the expert system's line of reasoning in as much detail as is necessary to determine what facts were used to reach a conclusion, how each fact was inferred, how the facts depend on one another, and their significance in the final conclusion.

For example, consider an expert system that tries to suggest the most suitable modem for a user. The system displays the following when asked how the best modem was determined:

The following facts have already been determined:

The **type of computer** that you intend using is a **desktop computer (how)**.
You intend sending and receiving only **small volumes of data** (**how**).
The best **location for the modem** is **external** (**how**).

According to rule 1f, as shown below, it can be inferred that the best **modem** is an **external modem**:

If the **type of computer** that you intend using is a **desktop computer** and you intend sending and receiving only **small volumes of data** and the **best location for the modem** is **external** then the best **modem** is an **external modem**.

Figure 4 An example of a *how* node.

If the *how* link next to *small volumes of data* is selected, the following is displayed explaining how it was determined that the user does not need to send small volumes of data:

The following facts have already been determined:

You do not need to **send large volumes of data** (**how**).
You do not intend using the modem for **online communication** (**how**).

According to rule 5b, as shown below, it can be inferred that you intend sending and receiving only **small volumes of data**:

If you do not need to **send large volumes of data** and you do not intend using the modem for **online communication** then you intend sending and receiving only **small volumes of data**.

Figure 5 An example of a *how* node.

Recall that throughout this exploration process hypertext links to related domain information exist that allow the user to obtain additional information about the variables and values used in the explanations. Thus, by selecting the *modem* link shown in Figure 4, the user can obtain an explanation of what a modem is.

4.2.2 The *WHY* Explanation Facility

The *why* explanation facility is initiated by the user during a consultation to ask the expert system why it is asking a certain question. The system uses its line of reasoning up until the point at which the facility was initiated to construct a hypertext network to serve as the collective

explanation. In addition to *how* nodes, the part of the network based on the line of reasoning includes *why* nodes which explain why the system is trying to determine a value for a certain variable. Each *why* node corresponds to one of the goals being pursued by the inference engine and shows the fact that it is trying to determine and the rule that it has selected to try and determine it. Alongside each fact is a *why* link that points to another *why* node that explains why the system is trying to determine that fact. By moving from one *why* node to another the user is able to explore why the determination of each pursued fact is significant to the final conclusion.

For example, when asked whether the user has a free power supply point the user may ask why this question is being asked. The following is a typical response:

> I am asking you this question to determine if the best **location for the modem** is **internal** (**why**).
>
> According to the current rule (rule 2a), if the **preferred location** of the modem is **external** and a **free power supply point** is not available then the best **location for the modem** is **internal.**

Figure 6 An example of a *why* node.

The *why* link can be selected to ask why the system is trying to determine if the best location for the modem is internal. Thus, the user is able to determine which facts have already been determined, how they were determined, how they depend on one another, why the system is trying to infer a certain fact and how each fact that it is trying to infer will affect the final conclusion. Hypertext links to related domain information exist that allow the user to obtain additional information about the variables and values used in the explanations.

5. CONCLUDING REMARKS

This study explored the possibility of producing explanations using hypertext technology to utilize the interactive and easy-to-use nature of the technology to strengthen the explanatory power of an explanation facility and, at the same time, enhance the presentation of expertise and additional domain knowledge. This approach to explanations facilitates the complete freedom of browsing the reasoning behind an expert system's conclusion or question in as much detail as is necessary by asking *how* and *why* questions. In addition, the facility allows access to additional domain knowledge thereby catering for users with different levels of expertise.

The approach presented here provides a useful improvement over the conventional approach to explanations in expert systems. This study can be viewed as an initial step toward formally proposing a concrete and sound hypermedia explanation facility.

REFERENCES

[1] Berry, D, Gillie, T and Banbury, S. (1995). **What Do Patients Want to Know: An Empirical Approach to Explanation Generation and Validation**. Expert Systems with Applications. Vol. 8, No. 4, pp. 419-428.

[2] Brown, C and O' Leary, D. (1995). **Introduction to Artificial Intelligence and Expert Systems.** http://wwwbus.orst.edu/faculty/brownc/es_tutor/es_tutor.html.

[3] Edwards, J. (1991). **Building Knowledge-Based Systems - Towards a Methodology.** Pitman Publishing, Great Britain.

[4] Eliot, L. (1994). **The Best Explanation of All.** AI Expert, Vol. 9, No. 12, pp. 13-15.

[5] Forsythe, D. (1995). **Using Ethnography in the Design of an Explanation System.** Expert Systems with Applications. Vol. 8, No. 4, pp. 403-417.

[6] Nielson, J. (1995). **Multimedia and Hypertext: The Internet and Beyond**. Academic Press, Inc., U.S.A.

[7] Wick, M. (1992). **Expert System Explanation in Retrospect: A Case Study in the Evolution of Expert System Explanation.** The Journal of Systems and Software. Vol. 19, No. 2, pp. 159-169.