ISSN 0254–2757

QUÆSTIONES INFORMATICÆ

Volume 6 • Number 4 April 1989

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

QUÆSTIONES INFORMATICÆ

The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Editor

Professor D G Kourie
Department of Computer Science
University of the Pretoria
Hatfield
0083

Production

Mr Q H Gee
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

Subscriptions

The annual subscription is

	SA	US	UK
Individuals	R20	$ 7	£ 5
Institutions	R30	$14	£10

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Quæstiones Informaticæ is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.

Editorial

by

Derrick Kourie

It is my privilege to have been requested by the SAICS executive to take over the post of editor of QI from Professor Judy Bishop. I think it is in order to thank her on behalf of the readership for the fine job she has done in boosting the quality of the journal during her brief but effective term. It is also appropriate to thank the production editor, Quintin Gee, for his substantial role in producing the journal. I am grateful that he is still in the post, and for all the support and work that he continues to do.

My job as editor is directed towards the overall goal of serving the South African academic community in the various computer-related disciplines in particular, and the computer industry in general. A number of objectives which support this goal include

- ensuring that high quality papers are published, thereby providing a display window for computer-related research in South Africa
- boosting local and international circulation of the journal both within the academic community and in the computer industry at large, thereby promoting a fruitful interchange of ideas
- attempting to do this in a cost-effective fashion so that the limited financial resources of SAICS and the CSSA may be released (perhaps even modestly augmented) to promote their various other service-orientated activities.

A number of measures are planned which are intended to meet these objectives. I shall mention some of them below, while others will become manifest with the passage of time.

After much debate it has been decided to change the name of this journal from Quæstiones Informaticæ to The South African Computer Journal/Die Suid-Afrikaanse Rekenaartydskrif. It will be abbreviated to SACJ in English and SART in Afrikaans. Arguments against this name change include the conciseness and uniformity of reference in both official languages provided by QI, and a certain kind of catchiness to the name. Those in favour of the name change regard the new proposal as being more descriptive for ordinary mortals (i.e. non-Latin scholars), less pretentious, and therefore more inviting for a wider audience. The fact that the new title identifies the journal as South African is also regarded as important. Many readers would, I surmise, be fairly neutral about the name and adopt a philosophical "a rose by any name" position. Perhaps the divide is between those who opt for a high level of abstraction and information hiding, and those who feel that a measure of refinement is necessary.

Regarding the quality of papers, I shall continually strive to ensure that papers submitted are reviewed by at least two relevant and competent specialists. It is appropriate here to thank all those who have so enthusiastically reviewed papers to date. This is a time-consuming, altruistic, backroom task, with very little explicit reward. To ensure that the burden is spread more equitably, I would like to appeal to readers to suggest additional names of people who could be approached for reviewing. Names of overseas contacts would be particularly useful.

I should also like to invite as much reader-participation in the journal as possible. There are several levels at which this may be done. The most obvious is by way of letters to the editor. Many people out there have strong ideas about a variety of subjects. In the absence of a decent national network facility (perhaps someday!), please feel free to use SACJ as your soapbox.

However, it is also evident that many people read many books for a variety of purposes. Why not share these insights by submitting book reviews to the journal, particularly with respect to books which could be prescribed for courses? If there are any book publishers or distributors out there who perchance may read this editorial, perhaps you should make inspection copies to lecturers contingent on a review being provided to SACJ!

I would also encourage researchers to continue providing a steady stream of research papers to the journal. Clearly, SACJ is in competition with other international journals for your research results. However, this is not a head-on competition. While it would be sheer hubris to pretend that SACJ is precisely equivalent to one of the more prestigious overseas publications, there are considerations which argue in favour of submitting certain kinds of research to SACJ. First, SACJ will be dedicated to providing a quick turnaround in reviewing and publication. Hence, it is an ideal forum for presenting and testing interim research results, and even for quickly assuring your stamp on potentially important ideas which you hope to flesh out later. Secondly, SACJ is the obvious forum to use for locally relevant research. Finally, and quite candidly, the competition for publication in SACJ is obviously not as intense as in a more prestigious international journal. However, I need to be most explicit on the implications of this latter point.

SACJ should not be seen as a soft option in the sense that quality will be sacrificed. By this I mean that on some arbitrary scale of quality measurement, if CACM contains papers above say the 95% percentile, then SACJ should fall into about a 60% percentile category. Put differently, there is clearly a gap to be filled that lies somewhere between poor, inferior drivel and outstanding research contributions – a gap which SACJ will seek to fill. Papers will therefore be rigorously reviewed, and every effort will be made to ensure that the journal is worthy of international recognition – even if such recognition does not come about immediately. This is not the impossible task that some might consider it to be. There are several South African scientific journals that already enjoy a measure of international recognition (the South African Statistical Journal – to name but one). Furthermore, it is my perception that many of our academics who travel overseas discover – perhaps slightly to their amazement – that they are well able to hold their own with academics at peer institutions. This suggests that there is probably sufficient brain power, research ability and research activity in the country to ensure that the goal of international recognition is attained.

As for the cost-effective functioning of SACJ, two points need to be made. First, SACJ will be available for a limited amount of advertising at R1000 per page and R500 per half-page. The computer industry and book publishers might wish to avail themselves of this offer, as might universities and employment agencies. Enquiries in this regard should be directed to Quintin Gee. Secondly, a modest charge per page (indicated elsewhere in this edition) will be levied on accepted research papers. This has become standard practice for most journals, the rationale being that the SACJ is one of the journals which counts for state subsidy purposes. However, the editor will have the right to waive such charges in deserving cases, as for example in the case of an author from industry whose company is unwilling to provide the financial support.

Ultimately then, SACJ will critically depend on your support. It will become what you, the reader, researcher and reviewer, make it. In a sense the South African Computer Journal will expose you, the South African Computer Academic, to the outside world without a single Latin phrase to hide behind.

A "Cooperating Expert's" Framework for Business Expert System Design

G R Finnie
Department of Computer Science, University of Natal, PO Box 375, Pietermaritzburg, 3200

Abstract

Expert system development environments based on current language constructs such as rules and frames have been criticised for approaching the task of problem solving from too low a level of abstraction. This paper describes the generic tasks framework for knowledge based systems proposed by Chandrasekaran et al. and discusses its application to the problem of financial statement analysis.

***Keywords:** expert systems, decision support systems, financial statement analysis, generic tasks.*
***Computing Review Categories:** I.2.1: Artificial Intelligence, Applications and Expert Systems*

Received October 1988, Accepted January 1989

Introduction

The development of programming languages has been categorised into a number of generations, ranging from machine code (1st) to assembler (2nd) to procedural problem solving languages like Pascal or COBOL (3rd) to non-procedural 4GLs (4th) to logic programming (5th?). Each generation step (excepting as yet the 4th to 5th) has been accompanied by a consequent leap in productivity (although some would feel that the value of 4GLs may be overstated). The expert/knowledge based system field is, in computing terms, relatively young, with problem solving being done largely in the context of the first generation of expert system tools (although again the term *second generation tools* has been used to describe the more recent development environments). Chandrasekaran and associated researchers [3,4,5,6,7] have argued that the current crop of knowledge based languages and tools operate at too low a level of abstraction for efficient problem solving, with the consequence that too much attention is focused on forcing the tools to fit the problem. They propose instead a set of tools based on the concept of generic tasks, where a specific tool is tailored to the solution of a specific class of problems.

The first section of this paper looks at some of the applications of knowledge based systems in business, and more specifically in the area of finance. This is followed by a section which outlines the generic tasks framework and some of the tools available. The next section discusses the problem of financial statement analysis by expert systems and suggests how the generic tasks approach could be used in its solution. The conclusion briefly considers the potential advantages of approaching expert systems development from a higher level of abstraction than the current tools provide.

Business Knowledge-Based Systems

In view of the high potential payoff of expert (knowledge-based) systems in certain commercial applications, their uses or possible uses in business have aroused wide interest. Several texts have been published in the area [e.g. 11,14,24] and a number of conferences or conference sessions have concentrated on the topic [e.g. 20,26]. It is apparent from the literature that there has been some success in developing useful commercial expert systems. However, some authors have questioned the value of expert systems in this area. Martins [17], for example, argues that they have proven effective only in simple applications, are weak theoretically, have high development costs and require heavy utilisation of resources.

The relationship between expert systems (ES) and decision support systems (DSS) has also been the subject of some discussion. Turban and Watkins [28] suggest that there is a strong case for treating ES as a component or subsystem of a DSS. They argue that although there is some support for the view that ES are a special case of DSS, there are several fundamental differences separating the two areas, in particular the fact that in many cases ES make rather than support decisions. A similar view is held by Pfeifer and Luthi [23] who hold that the two concepts should be treated as complementary rather than similar. They state "the paradigm for DSS is improving (management) decision making, the one of ES problem solving". Sen and Biswas [25] argue that the ES approach could be used to establish domain independence in DSS (called XDSS) and to simplify the integration of qualitative and quantitative data. As with similar debates on nomenclature (e.g. DSS vs MIS), it is unlikely that a universally acceptable definition will emerge.

A broad range of potential commercial application areas for expert systems have been identified in the research literature. Blanning [1] has suggested four primary areas for the use of expert systems in management. These are (1) resource allocation: allocating limited resources under various constraints to several activities, (2) problem diagnosis: using stored knowledge to explain symptoms associated with problems, (3) scheduling and assignment: also under various constraints, and (4) information management: assisting managers to assemble information required to solve specific problems. Rauch-Hindin [24] identified several current uses of ES is the business and finance area. These included financial statement analysis, mergers and acquisitions, portfolio analysis, asset and liability management, expert databases and intelligent reporting systems. Some other applications which have been investigated are insurance risk assessment, tax accounting, investment planning, auditing, banking services advice, strategic decision making and the analysis of competition for marketing tasks.

The finance area has attracted particular attention for expert system development, possibly in view of the high potential return in certain applications as well as a lack of personnel with sufficiently high skills in areas like investment advice. Rauch-Hindin [24] observes however that the financial services market may develop more slowly than the industrial market, as the sector may not lend itself to incremental systems development. Much of the work in this area has been done in secret [24] in order to provide financial advantage to the developing company but a number of authors have published on their research. Mays et al. [18] describe a system based on semantic nets which establishes ways of financing large equipment proposals. Dhar and Croker [9] have considered the use of ES techniques to simplify the use of integer programming models in resource allocation problems. Methlie [19] used a rule based system built on EMYCIN to study ratio analysis in financial statement analysis. Mui and McCarthy [22] describe the design of a financial statement analyser which can use semantic nets to aid the interpretation of additional financial statement information such as footnotes. Kerschberg and Dickinson [15] developed a PC based *expert support system* for financial analysis which incorporated the concept of *intelligent spreadsheets* i.e. linking a spreadsheet with a knowledge engine. Heuer et al. [13] describe a frame-based system to provide investment advice. Duda et al. [10] have applied a functional language approach to develop a system for assessing the risk associated with business opportunities. Stansfield and Greenfield [27] have developed Planpower which generates comprehensive personal financial plans. This list is by no means exhaustive and there remain a considerable variety of financial applications for knowledge based systems which have not yet been researched or reported.

The Generic Tasks Approach to ES Development

A central theme of the research performed at the Laboratory for Artificial Intelligence Research (LAIR) at Ohio State University has been the identification of several *generic* tasks which provide the basic framework for the solution of different classes of expert system problem [3,4,5,6,7,8,12]. Chandrasekaran [7] argues that the current set of ES problem solving tools (rules, frames, semantic nets, logic programming, etc.) provide too low a level of abstraction to adequately reflect the structure of the problem under consideration. He states [7] "The available paradigms often force us to fit the problem to the tools rather than fashion the tools to reflect the structure of the problem". More *conventional* programming languages have moved through several levels of abstraction from the second generation of assembler, to the third generation problem solving languages (e.g. Fortran, Pascal) to the current 4GL tools. Chandrasekaran and his research group hold the view that most currently available AI languages are in effect the assembly languages of the field and that higher level languages and constructs are required to facilitate effective ES development.

Bylander and Chandrasekaran [5] characterise a generic task in terms of the following information:

(a) The type of problem i.e. what function does the generic task perform
(b) The representation of knowledge i.e. how should knowledge be structured to facilitate solution of the generic task.
(c) The inference strategy i.e. what inference techniques should be applied to the knowledge representation.

The research has identified six generic tasks which have been developed for the construction of expert systems. The following is a brief review of the salient features of each type of task, more detail can be found in [7]. The first four tasks arose largely from work on diagnostic reasoning while the last two came from studies of routine design problems. Routine design can be applied when the way to decompose a design problem is already known.

(1) Hierarchical Classification

The diagnostic process can be viewed largely as one of classification driven by the symptoms or known facts concerning the specific problem. In the hierarchical approach, each node in the classification hierarchy corresponds to some diagnostic hypothesis. Problem-solving proceeds in a top-down manner: the root node (highest level concept) is given initial control of the search, control then passes to the *most likely* successor node, and so on. More general concepts appear in the higher nodes of the structure while more specific concepts are placed lower in the hierarchy. Each node or concept in the hierarchy contains sufficient knowledge to enable it to decide how well this concept matches the current data. If a concept is insufficiently supported i.e. assumed not relevant to the problem, all its successors are also assumed irrelevant. The terminology adopted by the research group is that each node plays the role of a *specialist* i.e. the conceptual hierarchy is a community of specialists [7]. A blackboard system is used to provide interaction between the specialists and to maintain a record of the current state of the system.

The inference strategy employed is termed **establish-refine**. Each node attempts to establish itself. If it fails, the search structure is pruned at this point and the next node at the same level is considered. If it succeeds in establishing itself, the refinement process is started i.e. each successor attempts to establish itself. Gomez and Chandrasekaran [12] state that three types of rule can be used in each concept or node: (a) confirmatory rules which look for evidence associated with the concept, (b) exclusionary rules which can be used to eliminate the concept, (c) recommendation rules which are pieces of knowledge which suggest direction to the sub-concepts. The process continues until sufficient tip nodes (corresponding to *primitive concepts* or specific hypotheses) have been established to account for all the observations or symptoms.

(2) Hypothesis matching or assessment

During the classification process, each node or specialist must attempt to establish itself, usually under conditions of considerable uncertainty. The essential process involved is that of pattern matching of the concept against relevant data to establish a measure of *goodness of fit*. Chandrasekaran [7] argues that the matching process should be treated as a separate generic task with separate knowledge structures and inference strategy. The matcher developed by the research group for the MDX diagnostic system [12] employed a process of hierarchical symbolic abstraction. An abstraction of the data is mapped to a set of discrete qualitative measures of fit – the process is hierarchical as the final abstraction is computed from intermediate abstractions. More detail on the process is available in [4]. Alternative matching processes could be considered – the primary issue under consideration is that the problem of matching hypotheses and data can be considered as a general type of problem solving process which could be employed in a variety of contexts.

(3) Knowledge-directed information passing

A problem solving process does not operate in isolation but instead functions in the context of general knowledge of the problem domain. For example, the analysis of financial data should be considered not only for a particular set of financial statements for a particular company but also within the context of the economic environment, general principles of accounting reporting and general knowledge concerning financial issues. Inferences concerning this type of knowledge can be handled by a generic task called knowledge-directed information passing. Mittal [21] developed a specific database system for the task (PATREC) which can hold general domain knowledge together with an inference process which can be used to infer domain knowledge which is not explicitly stored in the database. The PATREC system is organized as a frame hierarchy, each frame containing either the required data or information about how the value might be obtained. Again the specific storage design and inference mechanism is not as significant as recognising the need for a specific generic task to handle this type of problem-solving.

(4) Abductive assembly

Although hierarchical classification may generate a number of hypotheses, each hypothesis tends to be developed and viewed in at least partial isolation from the others. The Ohio State Laboratory for Artificial Intelligence has proposed a process of abductive reasoning which can be applied to a set of tentative hypotheses to build a composite hypothesis which best explains the data. Each hypothesis has some set of symptoms which it can account for with some degree of certainty. The abductive assembly process develops the composite hypothesis incrementally by successively adding hypotheses until a *best* or *most economical* coverage of the total problem is obtained.

The abductive assembly process requires knowledge in the form of relations between the relevant data and the hypotheses e.g. relations such as incompatibility between the data and a hypothesis or the data being suggestive of a particular hypothesis. The inference process can be described as alternating assembly and criticism. During assembly, the search process has the goal of explaining all significant data and is driven by a means-ends heuristic strategy. Each assembly seeks to add the best hypothesis which explains the most significant portion of the remaining data to the composite hypothesis. After each assembly, the criticism phase removes any superfluous components of the hypothesis. The process continues until all the data is explained or there are no hypotheses left.

(5) Hierarchical design by plan selection and refinement

In routine design, compiled design plans are assumed to be available for each stage in design [3]. Knowledge for this generic task should be considered in two forms: (1) knowledge of the structure of the objects which are known at some level of abstraction i.e. the device components and their relationship to each other, and (2) knowledge of the design plans for each part of the structure being developed. The plans can make design choices and invoke subcomponent designs. In the Aircyl system [3], the knowledge is organised as a design specialist hierarchy. Aircyl is used for the design of air cylinders, which can be considered a routine design problem as the general structure of air cylinders is known as are the design plans for each air cylinder component.

The control process works recursively to establish a complete design by initially selecting a specialist corresponding to an object component. The specialist selects a specific plan which may in turn suggest the activation of other specialists. If a plan fails, the reasons for the failure may cause a higher level specialist to change the plan and retry the problem.

(6) State abstraction

In many practical problems, an expert system user is interested in predicting the consequences of actions i.e. the well-known *what-if* capability of decision support systems. Typically, solution of this type of problem requires some form of simulation.In simplified form, the knowledge structure should be capable of representing the relationship between any change of state in any subsystem and the consequent state change in the system immediately containing the subsystem. These relationships could be represented as a hierarchy of *conceptual specialists* [7]. The control process operates in a bottom-up manner by tracing the effect of a specific state change through the various affected subsystems until the effect of the change can be determined at the required level of interest.

A variety of languages and tools have been developed by the Ohio State University Laboratory for Artificial Intelligence Research group to support the encoding of generic tasks. The process of hierarchical classification has resulted in the development of the language CSRL (Conceptual Structures Representation Language) [6]. The language HYPER [4] uses structured matching to measure the fit of a hypothesis to a problem situation. DSPL [3] was developed to cater for the process of object synthesis by selection and refinement. The knowledge-directed database system PATREC [21] has been used for knowledge-directed information passing while a language for abductive assembly (PEIRCE) is under development. These languages could considerably simplify the encoding of expert system tasks by facilitating problem definition and programmed solution at the correct level of abstraction.

An Example From Financial Statements Analysis

Methlie [19] has observed that the financial position of a company has much in common with the health of a patient where the financial statements can be considered as indicators of the financial health of the firm. However, as in medical diagnosis, the analysis and interpretation of the financial statements is a non-trivial task. The information must be consolidated and organised into a compact understandable form. A primary tool for this organisation is ratio analysis, in which the ratios between specific financial values can be used as indicators of possible malfunction in a given sector of the organisation. An example is the current ratio which is simply the ratio of current assets to current liabilities. A poor current ratio could indicate that either current assets or current liabilities are unsatisfactory. Further investigation might reveal low levels of assets like accounts receivable, inventory and cash or high levels of various types of debt.Six groups of financial ratios may be used in analysing the financial state of an organisation [29]:

(a) Liquidity ratios are a measure of the firms ability to meet its short-term commitments from liquid assets.
(b) Leverage ratios are used to indicate the capacity to service long term debt.
(c) Activity ratios illustrate how assets are being utilised.
(d) Profitability ratios indicate net return on sales and assets.
(e) Growth ratios which assess the firm's ability to maintain its economic position.
(f) Valuation ratios which assist in focussing on the goal of maximising the value of the firm.

If the simple determination of ratios was sufficient to determine the financial state of a company, there would be little need for knowledge based systems. However, the analysis is complicated by several issues. Ratios should not be considered in isolation but should be studied in the context of factors like industry averages, the general state of the economy and demographic considerations. This type of comparison requires considerable skill and general financial knowledge e.g. the knowledge that a particular industrial sector is under stress. Much of this type of data is qualitative rather than quantitative. Factors such as the relative size of the company should also be considered. Ratios should not be treated individually but should be considered together i.e. certain ratios may reinforce or negate decisions based on other ratios. Another complicating factor is that ratios should be analysed over time i.e. trends and discontinuities should be noted and related to such issues as past business decisions.

However, probably the most complex problem for a knowledge based system to handle is that the meaning of the financial data can vary according to the accounting principles and explanatory statements included in the financial data. Rauch-Hindin [24] points out that the relatively rare skill of a successful financial statement analyst is "generally characterised by their ability to read a financial statement's footnotes" where footnotes are defined as explanations of the criteria used to prepare the financial statement. Interpretation of the footnotes establishes the financial environment in which the analysis should proceed. The FSA system of Mui and McCarthy [22] is intended to cater for the preliminary data gathering or intelligence phase of a financial analysis (called the familiarisation phase). A major component of this system, a frame based representation of footnotes, is responsible for interpretation of the footnotes to establish the *general financial knowledge* required for effective analysis of the financial data. The FINEX system of Kerschberg and Dickinson [15] however ignores this issue of interpreting the additional financial statement data and assumes that spreadsheet models of the statements can provide sufficiently accurate ratios for analysis. This system is however interesting for its use of a semantic net to define the relationships between various financial concepts and the ratios. The semantic net also provides a control strategy for eliciting data by questioning the users.

The view that a company financial position and a patient's health are analogous suggests that a knowledge based system for the analysis of financial statements could be similar in structure to those used in medical diagnosis. Mui and McCarthy use a framework for financial statement analysis based on a protocol analysis performed by Bouwman [2] which divides the decision process into the phases of familiarising and reasoning. Familiarisation involves data gathering and searching the environment to identify key values in the statements and potential problem areas. Successful familiarisation requires interpretation of footnote data. The reasoning phase utilises not only the ratios but also requires the type of knowledge discussed above e.g. knowledge of the economy and the industry.

The generic tasks approach suggest a possible system design framework for financial statements analysis (FSA) which allows the process to be subdivided into a number of co-operating experts, each expert comprising a specific generic task. The initial phase of financial statement analysis could be considered a form of diagnosis in which the system would be attempting to ascertain whether anything is wrong in any part of the company. This could be approached using the generic task of hierarchical classification. A partial classification hierarchy is outlined in Figure 1 with the diagnostic knowledge of FSA distributed throughout the hierarchy. The establish-refine problem-solving paradigm could be applied to drive the classification process. The ratios node would attempt to establish if a financial problem exists, if so, the refinement would consist of attempting to establish each successor i.e. to determine whether the problem lies in short-term obligations, long term debt requirements, asset use, profitability or growth (or some combination of these). Each of these concepts would then attempt to establish itself before refining its successors. For example, if a problem is suspected in long-term debt servicing, the focus would the shift to attempting to establish either a balance sheet or income statement class of problem (or combination). The hierarchical classification technique provides a suitable control regime for effective data gathering i.e. the data gathering is performed at each point of diagnostic focus.

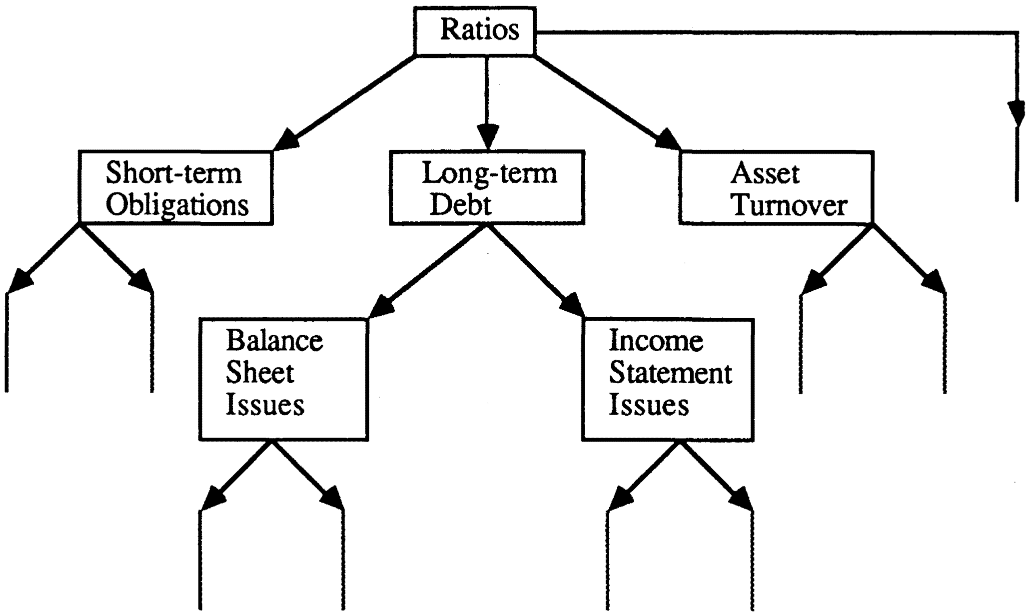

Figure 1 Hierarchical Classification for Financial Statement Analysis

The process of establishing a specific concept (node in the hierarchy) can be approached using the generic hypothesis matching task. The technique can employ various forms of uncertainty, one possible approach being the qualitative technique used by the HYPER system [4]. The task essentially involves matching a concept to specific data to establish a *goodness of fit* measure. The matching process concentrates on the accumulation of evidence which either supports or rules out a specific diagnosis with some qualitative or quantitative measure of confidence. For example, in establishing short term liquidity problems, attention would be focussed on determining indicators such as the current ratio and the acid test ratio. If the focus was on long term income statement problems, evidence collection would be aimed at data like fixed charge coverage and cash flow coverage.

Hierarchical classification and hypothesis matching require the interpretation of financial statement inserts and footnotes as well as general data on the industry and the economy. The process of knowledge directed data passing could be used for the intelligent retrieval of this data. The frame based approach used in PATREC can be used to hold general domain knowledge with each slot in a frame containing either the required knowledge or information about how to determine the required knowledge. Procedural attachment could be used to generalise the knowledge gathering process. Mittal et al. [21] also discuss techniques for the representation and processing of temporal knowledge which could be of value, inter alia, in the processing of trend information. Within the context of financial statement analysis, the classification and matching processes could be investigating the hypothesis of problems in the long-term debt structure of the company. However, the effect of leverage depends, inter alia, on the state of the economy and the degree of risk averseness of the investors. When the economy is in a down cycle, firms with low leverage ratios have a lower risk of loss but they also have lower probable returns if the economy enters an up-swing. The reverse is true for firms with high leverage. Knowledge concerning the relationship between leverage, as measured by the ratios, and factors such as the economy or the risk profile could be held in a frame-structured database for retrieval as required.

The classification process, in concert with matching and information passing, could produce a number of possible hypotheses to account for the firms financial state. The hypotheses would in most cases not be independent. The abductive assembly process could be used to assemble a composite hypothesis which best explains the data i.e. a hypothesis which provides a minimal covering set to explain all the financial symptoms. As a simple example, to explain a low profit margin on sales, the hypothesis of high costs could be supported (rather than that of low sales prices). To explain high interest expenses, there could be a supported hypothesis of excessive investment in plant and equipment. However, the cause of the high input costs and hence the low profit margin could be the high interest charges.

Conclusion

The generic tasks approach appears to have some potential for the efficient development of business knowledge based systems. Many business problems do not readily lend themselves to expert system design based on existing tools, which tend to provide a relatively uniform and low level set of representations. The higher level of abstraction proposed by Chandrasekaran et al could provide the ability to attack larger problems by providing a framework for problem decomposition as well tailoring the method to the problem. An example could be the area of strategic planning, in which a wide variety of qualitative and quantitative sources of data and techniques are needed. This group also argue that the generic tasks approach assists in providing a clearer explanation of the problem solving employed than the current tools, a critical component of any expert system. Bylander and Chandrasekaran [5] also claim that the approach will simplify knowledge acquisition by suggesting a different knowledge acquisition strategy for each kind of reasoning.

If the current set of knowledge base tools can be considered the equivalent of assembler level programming for expert systems, dealing with the problem solving at a higher level of abstraction has the potential of providing the same leap forward in knowledge base computing that languages like Fortran and COBOL provided over assembler coding. The building of sophisticated expert systems is currently a very slow process, even with development environments, and any advances which could dramatically improve the rate of development are worth serious investigation.

References

[1] R W Blanning, [1985], Expert Systems for Management: Research and Applications, *Journal of Information Science*, 9, 153-162.

[2] M J Bouwman, [1983], Human Diagnostic Reasoning By Computer: An illustration From Financial Analysis, *Management Science*, 29, 653-672.

[3] D C Brown and B Chandrasekaran, [1986], Knowledge and Control for a Mechanical Design Expert System, *IEEE Computer*, **19** (7), 92-100.

[4] T Bylander and T Johnson, [1988], Structured Matching, Tech Report, The Lab for AI Research, Ohio State University, Columbus, Ohio.

[5] T Bylander and B Chandrasekaran, [1987], Generic tasks for Knowledge-Based Reasoning: the "Right" Level of Abstraction for Knowledge Acquisition, *Int J Man-Machine Studies*, 26, 231-243.

[6] T Bylander and S Mittal, [1986], CSRL: A Language For Classificatory Problem Solving and Uncertainty Handling, *AI Magazine*, **7** (3), 66-77.

[7] B Chandrasekaran, [1986], Generic Tasks In Knowledge-Based Reasoning: High-Level Building Blocks for Expert System System Design, *IEEE Expert*, **1** (3), 23-30.

[8] B Chandrasekaran and M Tanner, [1986], Uncertainty handling in Expert Systems: Uniform vs Task-Specific Formalisms, in L Kanal and J Lemmer (eds), *Uncertainty in Artificial Intelligence*, Elsevier Science Publishers, 35-46.

[9] V Dhar and A Croker, [1988], Knowledge-Based Decision Support in Business: Issues and a Solution, *IEEE Expert*, 3, 53-62.

[10] R O Duda, P E Hart, R Reboh, J Reiter and T Risch, [1987], SYNTEL: Using a Functional Language for Financial Risk Assessment, *IEEE Expert*, **2** (3), 18-31.

[11] C J Ernst, (ed), [1988], *Management Expert Systems*, Addison-Wesley Publishing Company.

[12] F Gomez and B Chandrasekaran, [1984], Knowledge Organization and Distribution for Medical Diagnosis, in Clancey and Shortliffe (eds), *Readings in Medical Artificial Intelligence*, Addison-Wesley Publishing Company.

[13] S Heuer, U Koch and C Cryer, [1988], Invest: An Expert System for Financial Investments, *IEEE Expert*, **3** (2), 60-68.

[14] C W Holsapple and A B Whinston, [1987], *Business Expert Systems*, Richard D Irwin Inc.

[15] L Kerschberg and J Dickinson, [1988], FINEX: a PC-based Expert Support System for Financial Analysis, in C J Ernst (ed), *Management Expert Systems*, Addison-Wesley Publishing Company, 111-133.

[16] P Levine, J Ch Maillard and J Ch Pomerol, [1987], DECIDEX, a Multi-expert System for Strategic Decisions, in H G Sol, C A Takkenberg and P F de Vries Robbe (eds), *Expert Systems and Artificial Intelligence in Decision Support Systems*, D Reidel Publishing Company.

[17] G R Martins, [1984], The Overselling of Expert Systems, *Datamation*, **3** (18), 76-80.

[18] E Mays, C Apte, J Griesmer and J Kastner, [1987], Organizing Knowledge in a Complex Financial Domain, *IEEE Expert*, **2** (3), 61-70.

[19] L B Methlie, [1987], On Knowledge-Based Decision Support Systems For Financial Diagnosis, in C W Holsapple and A B Whinston, (eds), *Decision Support Systems: Theory and Applications*, Springer-Verlag, 335-371.

[20] G Mitra, [1986], *Computer Assisted Decision Making*, North Holland.

[21] S Mittal, B Chandrasekaran and J Sticklen, [1984], PATREC: A Knowledge-Directed Database For a Diagnostic Expert System, *IEEE Computer*, **1** (9), 51-58.

[22] C Mui and W E McCarthy, [1987], FSA: Applying AI techniques to the Familiarization Phase of Financial Decision Making, *IEEE Expert*, **2** (3), 33-41.

[23] R Pfeifer and H-J Luthi, [1987], Decision Support Systems and Expert Systems: a Complementary Relationship, in H G Sol, C A Takkenberg and P F de Vries Robbe (eds), *Expert Systems and Artificial Intelligence in Decision Support Systems*, D Reidel Publishing Company.

[24] W B Rauch-Hindin, [1985], *Artificial Intelligence Applications in Business, Science, and Industry* vol 2: applications, Prentice-Hall.

[25] A Sen and G Biswas, [1985], Decision Support Systems: An Expert Systems Approach, *Decision Support Systems*, 1, 197-204.

[26] H G Sol, C A Th Takkenberg and P De V Robbe, [1987], *Expert Systems and Artificial Intelligence in Decision Support Systems*, D Reidel Publishing Company.

[27] J L Stansfield and N R Greenfield, [1987], Planpower: A Comprehensive Financial Planner, *IEEE Expert*, **2** (3), 51-60.

[28] E Turban and P R Watkins, [1986], Integrating Expert Systems and Decision Support Systems, in R H Sprague and H J Watson (eds), *Decision support Systems: Putting Theory into Practice*, Prentice-Hall.

[29] J F Weston and E F Brigham, [1981], *Managerial Finance*, (7th ed), The Dryden Press.

NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. These may include research, review and exploratory articles of interest to the journal's readers. The preferred language of the journal will be English, although papers in Afrikaans or other congress languages of IFIP will not be precluded. Typed manuscripts for review should be submitted in triplicate to:

Professor D G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Pretoria

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible)
  - author's initials and surname
  - author's affiliation and address
  - an abstract of less than 200 words
  - an appropriate keyword list
  - a list of Computing Review Categories.
- Tables should be typed on separate sheets of A4 paper, and should be numbered and titled.
- Figures should also be supplied on separate sheets of A4 paper, and should be identified on the back in pencil with the author's name and the figure number. Original line drawings, and not photocopies, should be submitted.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin. Distinguish clearly between:
  - upper and lower case letters
  - the letter O and figure zero
  - the letter I and the number one
  - the letter K and kappa.

References should be listed after the text in alphabetical order of the (first) author's surname, cited in the text in square brackets. References should take the following form:

[1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.

[2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm. ACM*, 9 (3), 366-371.

Manuscripts accepted for publication should comply with the above guidelines, but may be in one of the following three formats:

- typewritten and suitable for scanning
- provided as an ASCII file on diskette
- camera-ready.

Authors wishing to provide camera-ready copy may obtain a page specification from the production editor.

Charges

A page charge, scaled to reflect production costs, will be levied on papers accepted for publication. The costs per final page are as follows:

Typed format, not camera-ready R60
Disk in ASCII format R40
Camera-ready format R20

These charges may be waived upon request of the author, and at the editor's discretion.

Proofs

Proofs of accepted papers will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Copyright

Copyright in published papers will be vested in the publisher.

Letters and Communications

Letters to the editor will be welcomed and will provide a forum for discussion on topical issues. They should be signed, and should be limited to about 500 words.

Communications reflecting minor research contributions will be considered for publication in a separate section of the journal. Such communications will, however, not be regarded as a fully-fledged publication for FRD subsidy purposes.

Book Reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertising

Placement of advertisements at R1000 per full page per issue and R500 per half page per issue will be considered. Enquiries should be directed to the production editor.