South African Computer Journal
Number 10
September 1993

Suid-Afrikaanse Rekenaartydskrif
Nommer 10
September 1993

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R45,00	R15,00
Elsewhere:	$45,00	$15,00

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

Phone: +27 (11) 315-1619 Fax: +27 (11) 315-2276

Guest Contribution

Information Technology Research in the European Community

Pieter Kritzinger

Data Network Architectures Laboratory, Department of Computer Science, University of Cape Town Private Bag, RONDEBOSCH, 7700 South Africa. e-mail: psk@cs.uct.ac.za

Abstract

It has been said that one reason why the US and Japanese information technology industries are ahead of those in Europe is that, where these countries see opportunities, European industry and its customers see primarily risks. Recognising this as well as the strategic importance of information technology and integrated communications systems for the future economic development of Europe, the Community launched the first of several five year programmes in information technology and public communication systems in the mid-eighties. In this report we describe the main programme called ESPRIT, how it works and some of the results achieved to date.

Introduction

The European Community consists of 12 nations with a GNP of US$billion 4 862 (1988 figures). By the year 2000 the information technology and electronics sector of the European Community is likely to become the largest industry, representing some 300 billion ECUs (1 ECU = R3,82) or 6,7% of GDP. With the major impact these enabling technologies have on the competitiveness of the whole of a modern economy, Europe recognised very early that information technology is of crucial importance to the success of the planned, unified internal market and an essential factor in the Community's development strategy.

At the same time however, the positive balance of trade of the European Community in information technology amounting to some ECU 1,7 billion in 1975 was declining rapidly (it reached a deficit of almost ECU 22 billion at the end of 1988) and the Community decided something drastic had to be done. As a result it launched the First 5 year European Strategic Programme for Research and Development in Information Technology or ESPRIT I. It started on January 1st 1984 with a budget of 1,5 billion ECUs.

Table 1. The EC and its top 3 partners in numbers (1988)

	Population (millions)	GNP (US$ billion)	Per Capita GNP (US$)
West Germany	61,0	1 120,0	18 400
France	56,0	939,2	16 800
Italy	57,5	814,0	14 200
EC	**325,1**	**4 475,1**	**13 770**
USA	248,0	4 862,0	19 600

Source: US Department of State, Bureau of Public Affairs

The overall strategic goal of ESPRIT was to provide the European information technology industry with the technology base which it needs to become and stay competitive with the US and Japan in the 1990s. In addition to this primary objective, two secondary objectives were defined, namely:

- to promote cooperation in the information technology field between industries, universities and European research bodies on R&D projects up to pre-competitive level; i.e., prior to the development of commercial products, and
- to contribute to the development of international standards.

At about the same time the crucial importance of public digital telecommunications to the future social and economic infrastructure of Europe was recognised. Consequently a separate Research and Development programme in Advanced Communications Technologies in Europe (or RACE) was launched in 1985 with a budget of ECU 1,1 billion. The stated goal of this latter programme was

- to introduce Integrated Broadband Communication (IBC) into the European Community taking into account the evolving ISDN and national strategies while progressing towards Community-wide services by 1995.

Both these programmes have since progressed to second 5 year phases as, respectively, ESPRIT II with a budget of 3,2 billion ECUs, and RACE II with 1,039 billion ECU. In the meanwhile ESPRIT III is in its initial planning phases.

RACE is similar to ESPRIT in terms of its financing and organisation. Space does not allow us to detail all aspects of the programme in this report.

Strategic Themes

Although the ESPRIT programme broadly addresses the information technology and electronics industry, ESPRIT I had 5 major strategic themes.

1. *Microelectronics.* This field was perceived as the key strategic area for information technology R&D in the future.
2. *Software Technology.* The stated goal of this research area was to do what was necessary to put the software development process on a sound engineering footing. Sub-areas were defined to deal with formal methods, development tools, management aspects, quality measurement and the development environment.
3. *Advanced Information Processing.* This area covered knowledge-based systems, new computer architectures and speech- and image-processing.
4. *Office Systems.* When initially conceived in 1984, this application area was viewed as of strategic importance for the efficiency of business throughout the Community.
5. *Computer Integrated Manufacturing.* This area comprised the total range of computer integrated manufacturing activities, including: computer aided design (CAD), computer aided engineering (CAE), computer aided manufacturing (CAM), flexible machining and assembly systems, robotics, testing and quality control. The area was selected for its potential impact on the methods and economies of production, particularly in the information technology industries, and also for the manufacturing industry in general.

In addition, the Information Exchange System project was started with the twofold objective of

- providing communication services to ESPRIT participants, both industrial and academic; and
- encouraging the development and adoption of OSI standards.

It is indicative of the experience gained in ESPRIT I and technology developments since it was started, to note how the strategic fields chosen for ESPRIT II differ from those of ESPRIT I. R&D in ESPRIT II is carried out in the following four major areas:

1. *Microelectronics* was retained as the key strategic area for information technology R&D in the future.
2. *Information Processing Systems and Software.* The work in this field will provide the fundamental and generic technologies which will support the development of information technology products expected on the market in the next decade. Thereby ESPRIT II recognised that information and its efficient use is not only a means of administration and communication, but that it is part of an enterprise's competitive advantage.

   As an aside, it is interesting to note that, of the 30 billion ECU expenditure on software and services in 1989, about 50% was provided by the manufacturing, banking and other financial services. This is expected to remain true through to 1994, when the market is expected to be worth 70 billion ECU. About one third of this market comprises customer services, consultancy, training and services while packaged software represents about 40% of the market. The latter component is expected to increase to 50% of the market by 1994 with services and training remaining constant at 30%.
3. *Advanced Business and Home Systems and Peripherals.* It is clear that information technology in the business environment is moving to advanced integrated systems capable of serving all the functions of the enterprise in an integrated multimedia environment. The priorities for work in the Community documents reflect these salient points.
4. *Computer Integrated Manufacturing.* The emphasis in this strategic area has not changed significantly from ESPRIT I to II.

In addition to the above, the *Open Microprocessor systems Initiative (OMI)* was started in ESPRIT II. The major motivating factor for the Community was the 82% dependence on non-European sources for microcomponents, representing 7 billion ECU in 1989 and which is expected to rise to 16 billion ECU by 1994.

Funding

ESPRIT is an industrial programme and it was not started for, or by, academics. The main driving force behind the ESPRIT I programme was industry, who first defined the research areas and then the goals and workplans. Industry was represented by the largest 12 information technology companies (known collectively as "The Twelve") in Europe.

ESPRIT R&D projects are implemented by shared-cost research and technological development contracts, with the Community financial participation normally not exceeding 50%. Universities and other research centres participating in shared-cost projects have the option of requesting, for each project, either 50% funding of total expenditure or 100% funding of the additional marginal costs. ESPRIT projects have a maximum duration of 5 years but should normally be shorter.

In the case of ESPRIT I, the Twelve received 50% of the ESPRIT budget and were involved in 70% of all projects. Small- to medium-sized enterprises (SMEs) participated in 65% of the projects and received 14% of the funding. The funding allocation by sector participating in ESPRIT I is illustrated by the chart in Figure 1.

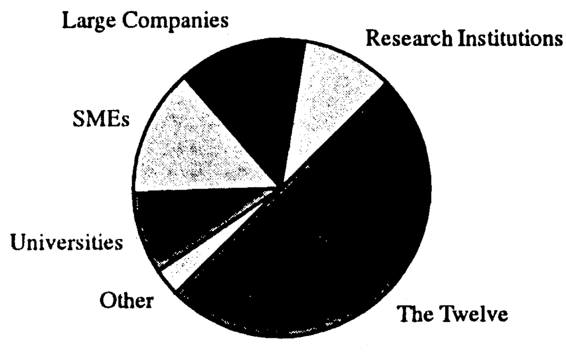

Figure 1. ESPRIT I funding allocation by participating sector

Basic Research

While ESPRIT I made no special provision for basic research, ESPRIT II includes a sub-programme, with a budget of 130 million ECU, aimed at developing new knowledge and expertise in the basic disciplines considered essential to secure the long-term future of information technology in European. Some 62 projects have been selected to carry out basic research in areas such as super-conductivity, optical and neural computers, speech and image processing and so on. In all, 211 university laboratories, 57 research bodies and 17 industrial companies are participating in these projects.

Apart from such projects, basic research activities also involve

- *Working Groups* which are concerted efforts to improve the systematic exchange of information and for which short scientific visits and workshop are funded, or
- *Networks of Excellence* which are composed of both academic and industrial teams geographically distributed throughout the Community. These are set up to provide a critical mass of complementary knowledge and expertise and to share limited and expensive resources. Funding for Networks of Excellence is restricted to the marginal costs of establishing the administrative and communications infrastructure necessary to carry out the coordination.

The evaluation criteria for basic research projects are less specific about the value for market exploitation of the expected results and more specific about conformity with the basic ESPRIT technical objectives, inter-disciplinary nature and scientific calibre of the partners.

Programme Management

Participation in the programme is solicited by a "call for proposals" made by a "consortium" comprising at least two participants or "partners" from different members within the Community [1] and usually no more than six – except for standards projects.

The proposals are then evaluated by external experts who take account of the following points in particular:

- The impact and potential for industrial exploitation of the expected results of the project.
- Eligibility of the partners.
- Technical merit of the proposal including a justification of the proposed theories and methods.
- Soundness of the proposal with regard to issues like the assessment of major technical risks and technological advances expected.
- All proposals are scrutinized for human and organisational factors to ensure that the results would be appropriate for the intended user base.
- Soundness of project plans with respect to the distribution of effort, clear and well defined roles for each partner, realistic timescales and the proposed management structure and methods of supervision.

Once a project has proceeded to contract signature[2] it is periodically subjected to four different audits until its completion:

- A *strategic audit* is carried out periodically to examine the evolution of the political, economic and social objectives in the light of world-wide strategic developments.
- An annual *technical audit* examines the progress of all projects which comprise the Programme. It is performed by a team of independent experts.
- A *programme management audit* evaluates overall management performance as well as individual project management and deliverables.
- The usual *financial audit* is done to ensure the correct use of public money.

Results

A total of 227 projects were implemented during ESPRIT I. They involved 536 participating entities and some 3 000 full-time researchers.

- Of the 327 participating industrial companies, almost 45% were firms employing fewer than 500 people and 40% of those employed fewer than 50. SMEs were extremely active, being involved in more than half the projects and being responsible for more than 25% of the research work in 60% of the cases.
- Nearly 200 universities and research institutes participated in approximately 70% of the projects. In more than half the cases, these scientific institutions were responsible for at least 25% of the work.

Towards the end of ESPRIT I, nearly 165 projects had delivered concrete results. Of those, 75 had already helped to put specific products and services onto the market, while for another 60 projects, the research worked had resulted in the transfer of technology for uses not directly linked to the project itself.

One detailed example is work in the Information Processing Systems and Software sub-field which let to the definition of a reference model for CASE (computer-aided software engineering) tools that has been adopted by the European Computer Manufacturing Association (ECMA). This has let to requests from the US National Institute of Standards and Technology to collaborate on the ECMA model as the basis for their own work on a reference model. Details about this and all other European Community research projects can be obtained from CORDIS mentioned below.

ESPRIT I participants who were questioned about their perceived successes of the programme considered increased knowledge as the most important benefit (69%), followed by a belief that research goals more ambitious than would otherwise have been set, had been reached.

[1] Partners in ESPRIT from outside the community are not eligible for financial support. A programme called EUREKA fosters extra-European research.
[2] Only 20% of all proposed projects in the case of ESPRIT I.

There have been direct benefits in being able to cover a wider range of research topics quicker by sharing results with the project partners.

A significant number of responses claimed a contribution either to existing products (35%) or new products (45%). It was felt, however, that there needs to be a greater degree of concerted action by project teams and a sharper strategic focus on market opportunities while, simultaneously, basic research must continue and even be increased.

15% saw no direct benefit.

Apart from technological reasons, ESPRIT and RACE were started, in the first instance, as Community programmes to promote cooperation in the information technology field between industries, universities and European research bodies on R&D projects. The extent to which this was achieved is thus an important criterion for measuring its technological successes. In this respect it is a general consensus that ESPRIT has indeed achieved a profound change in attitude in the Community. Cooperative, precompetitive research and development is now a formula which is working effectively.

Summary

It is apparent from the many ESPRIT reports that some participants in ESPRIT I, particular those from the Twelve, were originally rather sceptical about the likely successes of the programme. No small reason for this was that they had no accord on the product priorities for the industry as a whole.

Five years ago, the largest European companies viewed one another much more as competitors than collaborators. Five years later, however, apart from the major technological progress, a major, if not *the* major achievement is that there now exists a spirit of pre-competitive cooperation in the Community to the common advantage of all.

ESPRIT has become symbolic of the technological awakening of a European Community wishing to ensure its freedom to make the technological choices necessary for its own future prosperity.

Further Information

The European Community has set up an on-line information service to give quick and easy access to information on European Community research programmes. The Community Research and Development Information Service (CORDIS) is at present offered free of charge and comprises eight data-bases.

More information and CORDIS registration forms can be obtained from

ECHO Customer Service
CORDIS Operations
BP 2373
L-1023 Luxembourg
Tel.: (+352)34 98 11 Fax.: (+352) 34 98 12 34

Pieter Kritzinger is professor and currently head of the Computer Science Department at the University of Cape Town. During 1992 he was on research and study leave at the University of Dortmund in Germany and was thus able to observe programmes like ESPRIT and RACE at close hand.

Editor's Notes

A number of the articles in this issue of the South African Computer journal are in the field of Information Systems. Research in this area is beginning to blossom in the country. There are probably many more researchers in Information Systems than there are in Computer Science. It is hoped that not only academics, but also professional practitioners will submit articles.

Research in this area normally falls into three main categories. The first of these is pure research. This is a difficult area. Few researchers make a contribution here, mostly because the theory progresses slowly. However, these articles are to be encouraged. The second category of research is the collection of information from a variety of people in the field by means of questionnaire or interview and the use of this data to formulate policy and trends. An important aspect of this research is in order to corroborate theories or to identify areas where new theories are needed, or old theories amended. This has proved to be a very fruitful area of research and many beneficial results have accrued from it. The third category of research is performing careful analysis on a specific Case Study. In this area the case under study will need to display something which is innovative, either in the system itself, or in the way it was implemented. The case will need to prove something new and important or to break grounds into areas which have not formally been addressed.

All three types of work is worthy of publication if the results that they deliver are of benefit to the community which they serve. All three will be considered for publication by this Journal.

The journal divides into two sections. The primary section is involved with research while there also is a section on viewpoints and communications. Articles submitted for the latter are not refereed, but can be included after study by the editors. On some occasions articles submitted for the research section have been found appropriate for this section. This policy also applies to articles in the Information Systems field.

John Shochot
Subeditor: Information Systems

This issue was produced with the kind support of Software Collage (Pty) Ltd.

Expert Systems for Management Control: a Multi-Expert Architecture

Vevek Ram
Department of Computer Science, University of Natal, Pietermaritzburg

Abstract

The use of Expert Systems technology in management decision making domains is increasing rapidly as business environments worldwide grow more turbulent and as the cost of development tools decrease. Research effort in this field however, is concentrated largely on confined areas such as market analysis, financial diagnosis and production scheduling. The development of an Expert System to support a wider management area presents problems of both size and complexity since such a system would require a large monolithic knowledge base which would exhibit the associated problems of maintainability, consistency and reduction in inference speed.

This paper describes a blackboard based Multiexpert architecture that is capable of integrating the problem solving capabilities of a range of confined expert systems in order to provide problem solving support for a wide area such as management control at the strategic level. The system consists of several dedicated expert modules in the area of marketing, finance, production and so on as well as a control module that handles problem decomposition, task allocation and dynamic scheduling. A prototype version of such a system has been successfully implemented in Prolog.

Keywords: *Distributed Expert Systems, Blackboard Architecture, Management Control.*
Computing Review Categories: *I.2.1, J.1*

Received: August 1992, Accepted: January 1993, Final version: February 1993.

1 Introduction

Many organizations engage in Long Term or Strategic Planning in order to match their internal capabilities with the opportunities and threats that exist in their operating environments. Such a match is characterised by the commitment of the organization's resources to achieve a desired objective and is often referred to as the organization's Strategic Posture. Management Control is the process whereby the organization continually re-assesses the appropriateness of the match and re-aligns its Strategic Posture to accommodate changes in the environment. The rate at which the organization is able to respond to changes in the environment is known as the Strategic Response Rate.

The intensification of global competition has emphasised the importance of a rapid Strategic Response Rate as only those organizations that are able to seize environmental opportunities early, can compete effectively. Unfortunately, shifts in Strategic Posture involve the whole organization and the effects of it have to be assessed in various areas before changes can be implemented. This generates a lag in the response. For most organizations, posture shifts involve a reorganization of marketing, financial, production, research and human resources plans. Since these areas are separate in most organizations, the response lag can be attributed to the actual delay in assessment in each area and also the delay that can arise due to the communication and co-ordination between these departments.

Computer-based systems in the form of Decision Support Systems and Expert Systems have to a large extent provided assistance in reducing the problem in the individual areas. Descriptions of such systems have been developed by King and Dutta [13], King and Rodrigues [14], Klein and Newman [15], Bouwman [4], Smith *et al* [22], Cooper [8], Chandrasekaran and Ramesh [6, 7], Goul [10], Lee and Lee [16] and Biswas *et al* [3]. Although the problem of communication and co-ordination can be solved by developing a single system that is representative of the collective activities of the various organizational areas, such a system would require a large and complex knowledge base. Large knowledge bases exhibit problems of maintainability and consistency [18]. Also, there is a considerable reduction in inference speed and efficiency as the size of a knowledge base increases.

An alternative is to build a system that can integrate the functions of various individual systems by enabling them to co-operate to solve a common problem while at the same time retaining their individual status. This approach is used extensively in the area of Distributed Artificial Intelligence and many useful techniques have been developed as a result. In such systems, often referred to as Distributed Problem Solving networks, individual knowledge-based systems communicate and cooperate in order to solve a problem that requires the combined expertise of all of them. Various types of distributed networks have been developed depending on the nature of the control of the individual knowledge bases and the communication between them. Two of these types that are readily applicable are the blackboard architecture [9, 17, 11] and the centralised multiagent framework [5]. The individual knowledge bases that interact in these systems are also referred to as knowledge sources or intelligent agents.

The blackboard architecture is based on a shared global data structure called the blackboard. The blackboard is divided into levels of varying abstraction depending on the application. Independent knowledge bases may read from and write to one or more levels of the blackboard. Multiagent frameworks use a single knowledge source or a group of knowledge sources to form a coherent plan for solving a multiagent problem. Dependencies and potential conflicts among the agents are identified in advance. In centralised multiagent frameworks, one agent acts as the controller and coordinator for the whole network of agents. A combination of the centralised multiagent framework and the blackboard architecture facilitates the integration of discrete knowledge sources and enables the power of their collective knowledge to be used as a single large knowledge base without the associated problems. The remainder of this paper describes the architecture and operation of such a distributed system for use in the management control area. Construction details for some of the more important aspects of the system is also included.

2 Architecture

The distributed management control system consists of a control module, a scanning module and several functional modules as shown in Figure 1. A brief description of the knowledge-based modules and a discussion of their major roles in the distributed network follows.

The Control Module

The control module acts as the strategic management expert and also as the manager of the network. As the strategic management expert, it controls the direction and format of the network problem solving process. It contains knowledge about the strategic management process and it also contains meta-knowledge, which is knowledge about how the rest of the system's knowledge is distributed throughout the network. This meta-knowledge allows the control module to decide that interest rates concern the financial expert, product cost concerns the production expert and so on. As the network controller, the control module controls the execution of individual modules as well as the management of the status of the blackboard.

The Scanning Module.

The scanning sub-system acts as the machine interface between the network and the organization. The scanning module monitors a set of strategic factors and reports all variances to the control module via the blackboard. It performs a simple but nevertheless important role in the network. The module is non-intelligent in that it reports all variances. The control module decides on the severity of an occurred variance by examining the extent of the variance against predetermined limits. This is explained below.

The Functional Knowledge modules

There are currently four functional modules in the prototype network. Each module contains conceptual knowledge of the domain area in general and also of specific policies of the organization in that area. The domain areas are Marketing, Finance, Production and Organizational. Each module has sufficient domain and control knowledge in order to function as a stand alone expert or knowledge-based system. Additional modules for other functional areas such as Research and Development or Distribution can be added when required.

The Organizational and Environmental Database

This is the collective data repository that holds normal organizational data from accounting, marketing, human resource and manufacturing subsystems as well as external data regarding interest rates, inflation, competitor information, consumer demand, political and technological trends. The scanning module obtains strategic organizational data by interrogating the files of the appropriate subsystem and computing the factors that are monitored by the control system.

3 Operational Overview

Each knowledge source is responsible for maintaining a set of strategic variables in its own domain. Variables are categorised as either internal or external depending on whether the entity that a variable relates to is changed from within or outside of the organisation's boundary. As long as the values of these variables remain within predefined limits, there is a balance between the organisational ability, environmental pressure and a chosen strategy. As an example, if increasing market share was one objective in the chosen strategy, then a market share growth rate would be determined on the basis of internal competence and environmental pressure. When the monitored growth rate falls below this level, there is an imbalance which must be analyzed and corrected by the distributed control system. This continuous monitoring and correction process is the essence of management control and helps to optimize organizational performance. The idea of maintaining a balance as a form of control is derived from Ansoff's market dynamics model [1, 2].

The values for the internal variables are held in the organisational database which is constantly updated through the organisation's information system. External variables are updated through manual input on a regular basis. All variables are monitored by the scanning subsystem. When the value of a variable changes, the scanning subsystem communicates this change to the control module. The control module decides on the degree of severity of the variance (and others which may occur simultaneously), assigns priorities and then decides on which modules need to be called in order to resolve the problem. It then posts a request with parameters describing the nature of the variance on the blackboard and activates the appropriate expert module or knowledge source. The individual knowledge-base module assesses the impact of the change in relation to the present strategic posture and communicates the result back to the control module via the blackboard. If the result concerns other knowledge-base modules, these are then activated by the control module. The process continues until a final result is obtained that is consistent with all the ex-

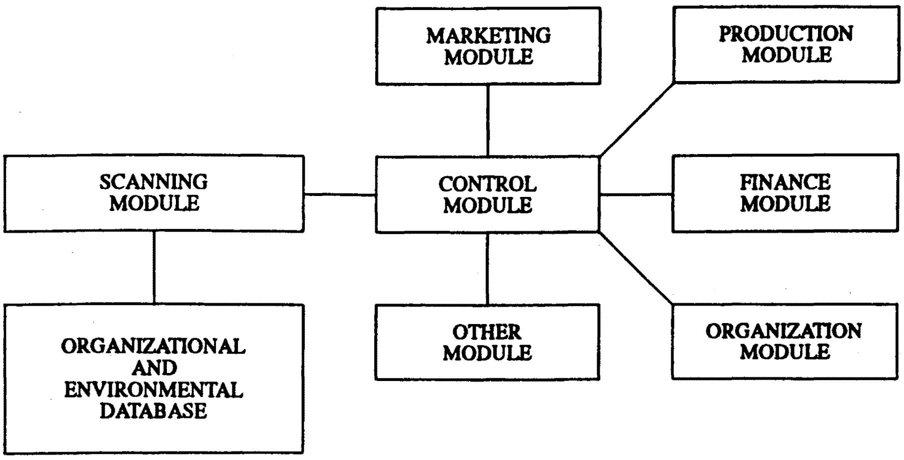

Figure 1. The distributed system architecture.

perts individual results. If two or more experts put forward recommendations that are conflicting, the control module can resolve the conflict by choosing the recommendation with the highest utility value or by modifying and reposting variables on the blackboard so that the individual experts reassess their respective recommendations and in so doing resolve the conflict themselves after a number of cycles.

Since the system is centred around a blackboard, the control of processing is asynchronous and opportunistic. It is asynchronous since individual knowledge-base modules post hypotheses on the blackboard in no particular order and it is opportunistic since the individual knowledge-base modules use data that they recognise on the blackboard as opportunities to initiate problem solving [12]. This system however, is slightly different from other blackboard systems in that opportunistic reasoning can only occur once the control module has initiated at least one knowledge-base module. If the module that is called first is capable of solving the problem at hand completely, then no opportunistic reasoning takes place.

4 Communication and Control

The System Blackboard

The blackboard does not exist as a physical entity in the system but rather as a communication mechanism through which the knowledge sources communicate both their requests and their findings. All individual knowledge source activities are initiated from the blackboard and all conclusions or results from knowledge sources are directed to the blackboard. In this application, the blackboard is in the control of the strategy formulation and control knowledge source.

The system blackboard is divided into three main areas as shown in Figure 2. These three areas are used for static knowledge, dynamic knowledge and control knowledge respectively. Static knowledge is the domain specific knowledge that is relevant to the problem and normally remains relatively stable during the solution process. In the system blackboard, the static knowledge area holds the collection of organizational data that is scanned by the scanning module. Dynamic knowledge is knowledge that is generated during the execution of the system. It consists typically of new facts, hypotheses and suggestions that are made by the knowledge sources. Control knowledge is knowledge about the current state of the network itself and also of the status of the problem solving. In the system blackboard, the control knowledge is made up of a set of requests which form a dynamic queue. The requests are either from the control module to a functional module or vice-versa. The control module extracts from this request list a single request which it then converts into a call to an individual module. The static and dynamic areas also form levels of abstraction that distinguish between factual knowledge at the lower level and conceptual knowledge at the higher. The responsibility of keeping the blackboard "clean", that is, erasing old or unwanted entries or archiving previous entries rests with the control module. This is an essential activity since the blackboard tends to become cluttered after a reasonable amount of network activity and this can lead to a degradation of the network efficiency.

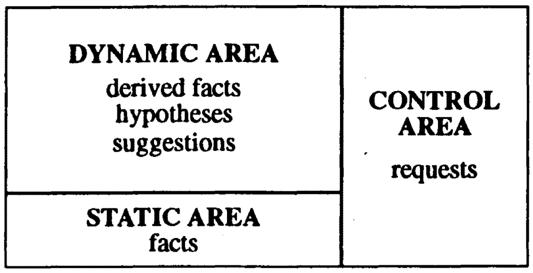

Figure 2. The system blackboard

Scheduling and Control

Network control can be achieved by selecting an individual knowledge source and calling on it to execute inside a problem solving cycle, or it can be achieved by placing knowledge on the blackboard that will cause a knowledge source to execute on its own. The distributed system uses the control knowledge source as the network controller and therefore makes use of the former method. The network as a whole makes use of three control mechanisms:

Goal-driven control which is the control exerted on the network to attain a network-wide or global system goal;

Request-driven control which is the control exerted on the network by inter knowledge-source requests, and

Event-driven control which is the control exerted on the network due to the occurrence of certain events.

The goal-driven function of control is the classical strategic management function of monitoring and controlling of strategic and functional plans. The goal of the network is to ensure that the implemented strategy adheres to certain performance limits that were used in the formulation of the strategy. Variances that exist obviously affect the strategic posture as a whole and must be accommodated at the strategic level. Variances are translated into network action by the event-driven control mechanism. A variance is regarded by the scanning subsystem as a strategic event that triggers the network into a resolution process.

The event-driven control function is to alert the control module into initiating the network. Individual knowledge sources would then attempt to reduce the variance or the effects of it and failing this, to reevaluate the strategy.

The request-driven mechanism allows the control module as well as other modules to alter the direction of the problem solving process in a dynamic way by posting requests on the blackboard. These requests may be for further information or for initiating the activation of other modules. There may be many such requests on the blackboard at any one time and in a serial network, that is a network in which the knowledge sources cannot be executed in parallel, these requests need to be serviced in some sort of order. This is accomplished by establishing a schedule of ready-to-be-called modules. This schedule must be flexible enough to be dynamically modified since the execution of one module may cause others that were ready to execute to be no longer required or, the execution of one module may cause others that were not executable, ready to execute. Also, there may be more than one consecutive request for the same module, and the schedule must allow a single activation call with all the requests consolidated rather than allow more than one consecutive call to the same module.

5 Implementation

The prototype version of the distributed management control system has been implemented in Prolog on a microcomputer. As the exact constructional details are beyond the scope of this paper, only the more unusual aspects of communication, task decomposition and dynamic scheduling are described. Also, as each individual knowledge module is a conventional production rule and frame based Expert System, these will not be examined. Further detail can be found in Ram [20, 19].

Communication

The mechanism used to control and facilitate the flow of information in the network is the blackboard. The blackboard is a communication and storage mechanism which is accessible by all the modules and which is divided into different levels. Prolog has an internal database which can be modified during execution and can also be stored and retrieved. This forms a convenient implementation of the blackboard. The different levels can be distinguished by using a separate predicate for each level. At the static level which holds organizational data for example, the predicate has the form:

Factval(FactorName, value)

An example of such a clause is *Factval("InterestRate", "LOW")*.

The control level of the blackboard is used to hold control information for the scheduling mechanism. This information is in the form of request clauses that are inserted into the blackboard at the control level by those functional modules that require assistance or additional information. Requests are held in the following format:

Request(CallMod, DestMod, Factor, Action, Ref, Status)

CallMod identifies the module issuing the request, *DestMod* is the module to which the request is directed, *Factor* and *Action* are as before, *Ref* is a request reference number and *Status* indicates the status of the request. A status value of *"U"* denotes unresolved and a status value of *"OK"* or *"NotOK"* denotes a resolved request. When the control module assembles requests into a queue, it examines the status value in each request and ignores requests that have already been resolved. An example of a request is

Request("MARK",,"PRICE","DEC",1,"U")

This is a request from the marketing module requesting that the control module investigate the possibility of a reduction in product price. Requests in the individual modules are typically invoked by rules which test for the existence of required data. The *DestMod* slot is left blank since the individual modules do not have knowledge of each others expertise. The control module, through its decomposition procedure, decides on the module to which it can best delegate the resolution of the request and fills the *DestMod* slot before the delegated module is called. If the request can only be resolved by more than one module, the control module issues as many requests as the decomposition procedure generates.

Problem Decomposition

When an individual functional module encounters a subproblem during its problem solving activity that is outside its domain of expertise, it would issue a request to the control module for assistance. It is the function of the control module to redirect these requests to the appropriate modules. A major problem for the control module in the execution of this function is "knowing" which module to call for a given request. A simple and effective way to overcome this problem is to maintain a list that links all the relevant organizational data items with the modules responsible for them. Such a list represents meta-level knowledge since it represents knowledge about the use of the distributed expertise in the most efficient way. When a request that can be resolved by a single module is received, the control module need only scan the list in order to identify the module best suited to resolving the request. A problem arises when a request is received that cannot be resolved by one module alone. Such a request has to be decomposed into subrequests that can be resolved by individual modules. This decomposition process can be implemented by organizing the decomposition relationships into a taxonomy of meta-knowledge frames as follows.

MFrame(Problem, PRef, DecompList, Dmodule)

where:

MFrame is a label distinguishing the Meta-Knowledge frames from other frames in the program;

Problem is the label identifying the problem that this instance of the frame is representing;

PRef acts as a reference number for the problem represented by this frame and is used to establish priorities in the problem solving process;

DecompList is a list of all the subproblems that *Problem* can be decomposed into;

Dmod is the Domain module responsible for solving *Problem* and is only present in a frame if the DecompList contains a single element, or if it contains more then one element, then all these elements are the responsibility of the same domain module.

Consider an example that was used during the testing of the system. Values in the database were manipulated to simulate a drop in market share. The scanning module detects this and passes it on to the control module. One option for strategic realignment is to restore the situation by stimulating primary market demand which expands the total market or by stimulating selective demand which increases market share within the existing market. A marketing action plan of reducing product price or increasing advertising can achieve both these. Since product price is outside the domain of the marketing module, it will request the control module to investigate the feasibility of price reduction. The control module has to refer this request to the appropriate module or modules and makes use of the meta-knowledge frame taxonomy search to decide which module or modules are appropriate. The search begins by finding a frame which has price as the label in the problem slot.

MFrame("Price", 1, ["Cost", "Margin"],)

The *PRef* slot is arbitrarily set to 1 and the domain slot is empty since *DecompList* contains more than one element. This frame represents the decomposition of the price problem into the two subproblems of cost and margin. The search then continues by finding a frame for each of the elements in the *DecomList*. These are found as

MFrame("Cost", 1, ["ProdCost", "Ohead"],)
MFrame("Margin", 1, ["Margin"], FIN)

The first frame further decomposes the cost problem into the two subproblems of production cost and production overhead. The second frame asserts that margin cannot be decomposed further and that it is the responsibility of the *FIN* or Financial module. The control module continues the search by finding frames with *ProdCost* and *Ohead* as labels in the problem slot. This produces the following frames

MFrame("Prodcost", 1, ["ProdCost"], PROD)
MFrame("OHead", 1, ["OHead"], FIN)

Since both these frames contain only one element in their respective *DecomLists*, the search terminates and the control module posts a request to the *PROD* or production module to investigate the reduction in product cost. The production module contains rules that relate product cost to raw material and labour costs and so is able to function independently in solving this subproblem. The control module also posts a request to the *FIN* or financial module to investigate the possibility of a reduction in profit margin and production overhead. Both the financial and the production modules communicate the results of their investigations to the control module. Both the requests derived from the decomposition have the same *PRef* number as the original request and the scheduling mechanism uses this number to keep them in the same logical group.

Control of Dynamic Scheduling

Dynamic Scheduling is accomplished by establishing and managing a queue of ready-to-be-called modules. The queue is represented by a prolog list and is constructed by examining all the requests held in the control level of the blackboard. The scheduling procedure terminates when the queue is empty which occurs when there are no unresolved requests on the blackboard. Once a queue has been constructed, the control module calls the functional module represented by the first entry in the queue. When the call terminates, that is, when the functional module has completed its task, the control module then reconstructs the queue and the process is repeated. Reconstructing the queue each time a functional module call is terminated, ensures that the scheduling mechanism makes use of the most current problem solving knowledge available. This is necessary since at any stage, a called module may issue a request and suspend its problem solving activity until the request is resolved. The module chosen for investigating this new request must be inserted at the head of the queue and called. On its termination, the original module which is waiting for the response is called and continues its task.

6 Testing and Results

The system was tested on a number of scenarios that were adapted from various cases in strategic management. In each scenario, strategies were developed based on the information given in each case. Strategic variables that had to be monitored in order to ensure successful implementation of the strategy were chosen. Limits of acceptable movement of each strategic variable were set and loaded into the scanning module. Data that contributes to the value of each variable was then manipulated to investigate the activity of the network. Since the strategic variables used in this system is monitored in the same way as the Critical Success Factors of Rockart [21], one of the test cases was developed on the basis of the critical success factors described in his article. The "Microwave Associates" case describes market share as a critical success factor and this was used as a strategic variable. The results of changing this variable is described in the implementation section above.

The individual modules in the prototype system contain very little domain knowledge and so the system as a whole was not expected to generate results of any great strategic significance. The network activity however, that was initiated by the various scenarios showed that the individual knowledge-base modules can cooperate to solve common problems. Extending the individual knowledge bases does add complexity to the interactions between them but it is far easier to manage and maintain many small knowledge bases than a single large knowledge base. Each small knowledge-base module has its own inferencing mechanism and the interactions are known by the control module. In a single large knowledge base, the interactions between various domains have to be embedded in the inferencing mechanism itself and any extensions to the knowledge must also be reflected in it. This makes it extremely difficult to maintain consistency. Individual knowledge-base modules also have the added advantage that they can be used as stand-alone experts in their respective domains and they can be fired concurrently in a parallel implementation of the system.

7 Conclusion

A description of a distributed knowledge-based management control system has been presented. A prototype version which has been implemented in Prolog, has generated very favourable results in an important area. It is hoped that the success on this limited scale will encourage further research in other areas. It is considered that the most important aspect of this work is the illustration that complex areas requiring knowledge-based support can be structured into relatively self-contained knowledge modules which can then be integrated into a system which, while addressing the original problem, is easier to build, debug and maintain.

References

1. H I Ansoff. *Strategic Management*. Macmillan, London, 1979.
2. H I Ansoff. *The New Corporate Strategy*. John Wiley, New York, 1988.
3. G Biswas, M Oliff, and A Sen. 'An expert decision support system for production control'. *Decision Support Systems*, **4**:235–248, (1988).
4. M J Bouwman. 'Human diagnostic reasoning by computer: An illustration from financial analysis'. *Management Science*, **29**(6):653–672, (1983).
5. S Cammarata *et al.* 'Strategies of cooperation in distributed problem solving'. In *Proceedings of the IJCAI*, (1983).
6. G Chandrasekar and R Ramesh. 'Microcomputer based multiple criteria decision support system for strategic planning'. *Information and Management*, **12**:163–172, (1987).
7. G Chandrasekar and R Ramesh. 'An integrated framework for decision support in corporate planning'. *Decision Support Systems*, **4**:365–375, (1988).
8. P Cooper. 'Expert systems in management science'. In T Bertold, ed., *Expert Systems and Knowledge Engineering*, pp. 61–71. Elsevier, Holland, (1986).
9. L Erman, F Hayes-Roth, V Lesser, and D Reddy. 'The hearsay ii speech understanding system: Integrating knowledge to resolve uncertainty'. *Computing Surveys*, **12**(2):213–253, (1980).
10. M Goul. 'On building expert systems for strategic planners: A knowledge engineers experience'. *Information and Management*, **12**:131–141, (1987).
11. B Hayes-Roth. 'A blackboard architecture for control'. In A H Bond and L Gasser, eds., *Readings in Distributed Artificial Intelligence*, pp. 505–540. Morgan Kaufman, California, (1988).
12. B Hayes-Roth and F Hayes-Roth. 'Cognitive processes in planning'. Technical Report R2366ONR, Rand Corporation, California, (1978).
13. W R King and B P Dutta. 'A competitive scenario modelling system'. *Management Science*, **26**:261–273, (1980).
14. W R King and J I Rodriquez. 'Competitive information systems'. *Long Range Planning*, **10**:59–64, (1977).
15. H Klein and W Newman. 'How to use spire: A systematic procedure for identifying relevent environments for strategic planning'. *Journal of Business Strategy*, pp. 32–45, (Summer 1980).
16. J K Lee and H G Lee. 'Interaction of strategic planning and short-term planning: An intelligent DSS by the post-model analysis approach'. *Decision Support Systems*, **3**:141–154, (1987).
17. H P Nii. 'Blackboard systems, the blackboard model of problem solving and the evolution of blackboard architectures'. *AI Magazine*, **7**(3), (1986).
18. D S Prerau, A S Gunderson, R E Reinke, and M R Adler. 'Maintainability techniques in developing large expert systems'. *IEEE Expert*, pp. 71–79, (June 1990).
19. V Ram. *A Distributed Knowledge-Based Support System for Strategic Management*. PhD thesis, University of Natal, Pietermaritzburg,RSA, 1990.
20. V Ram. 'A system of cooperating experts for strategic planning support in business'. In A Mallouppas, ed., *Management Technology: Control Tools for the 90's*. Peter Perigrinus Ltd, London, (1990).
21. J F Rockart. 'Chief executives define their own data needs'. *Harvard Business Review*, pp. 81–93, (Mar–Apr 1979).
22. L D Smith, J Blodgett, M Janson, and B V. 'Decision support for marketing research and corporate planning'. *Information and Management*, **8**:133–145, (1985).

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R45,00	R15,00
Elsewhere:	$45,00	$15,00

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

Phone: +27 (11) 315-1619 Fax: +27 (11) 315-2276

South African Computer Journal
Number 10, September 1993
ISSN 1015-7999

Suid-Afrikaanse Rekenaartydskrif
Nommer 10, September 1993
ISSN 1015-7999

Contents