South African
Computer
Journal
Number 10
September 1993

Suid-Afrikaanse
Rekenaar-
tydskrif
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

Editor
Professor Derrick G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Email: dkourie@dos-lan.cs.up.ac.za

Subeditor: Information Systems
Prof John Shochot
University of the Witwatersrand
Private Bag 3
WITS 2050
Email: 035ebrs@witsvma.wits.ac.za

Production Editor
Dr Riël Smit
Software Collage (Pty) Ltd
P.O.Box 16650
Vlaeberg 8018
Email: gds@cs.uct.ac.za

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.

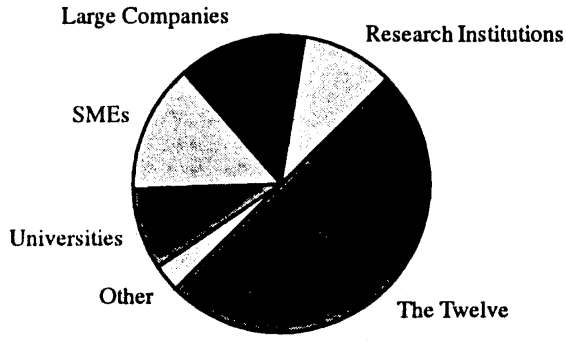

Figure 1. ESPRIT I funding allocation by participating sector

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.

Assessing the Usability of Computer Systems

T.C. Parker-Nance P.R. Warren
Department of Computer Science, University of Port Elizabeth, P.O. Box 1600, Port Elizabeth, 6000
csaprw@cs.upe.ac.za

Abstract

Usability problems with computer systems can be divided into three broad categories: problems due to the socio/political environment, problems due to the usefulness of systems, and problems due to the accessibility of utility. Any assessment of the usability of computer systems must span all three of these categories. This paper focuses on the problems associated with the accessibility of utility. These problems can be divided into three classes: firstly, problems due to the use of different systems when completing a single job; secondly, problems due to the way a user thinks about completing a job; and thirdly, problems arising from the interaction of the user and the system via the interface. Analysis techniques to assess the accessibility of utility in each of these areas are briefly outlined and applied to systems at Volkswagen.
Keywords: *User interface, Usability*
Computing Review Categories: *D.2.2, H.1.2*

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

1 Introduction

This paper assesses the usability of computer systems. Problems with usability are divided into three broad categories. The first category consists of problems associated with the socio/political environment in which the computer system is embedded. The second category consists of problems due the lack of useful functions in the system. Goodwin [9] and Bennett [4] have already argued that it is not sufficient for a system to be useful; the functions that comprise the usefulness must also be easily accessible. The third category thus consists of problems associated with the accessibility of the useful functions of the system. This paper focuses on the assessment of the usability of computer systems with respect to the last category, the accessibility of utility. A number of problems from a survey made at Volkswagen SA (VW) will be mentioned as examples.

This paper does not address the socio/political environment in which a system is used, however from examples in the literature [1, 2, 14], it is clear that the socio/political environment in which a system exists influences its success or failure.

The second category, the usefulness of a system, is also not addressed in this paper. Functionality is the set of functions provided by a system, in other words what the system can do. Not all of these functions may be usable for a specific task. Only those that provide a benefit to the user can be classed as useful. Utility is this set of useful functions [8]. The usefulness of the system is a measure of how well these functions match those needed by the user. Thus a useful system provides the user with utility, a set of useful functions, that can be used to do a job or task. Obviously a mandatory system will be used regardless of it's usefulness, however, where system use is discretionary, a system that is not useful will, almost by definition, not be used.

2 The Accessibility of Utility

The existence of utility in a system is essential, but not enough. System utility must also be easily accessible. One of the problems which could prevent this from happening is that a system is designed within a designer's, rather than user's frame of reference [17]. In other words, design may be determined by the constraints placed on the designer, rather than the needs of the user. A rather simple example of the influence of constraints on design is bath water control [17]. A user wants to control the rate of total water flow into the bath, and the temperature of the water in the bath. The designer's problem is that there is a hot water pipe and a cold water pipe. The easiest solution for the designer is to place a volume control on each pipe (two separate taps), and leave the temperature control up to the user (change temperature by changing the rate of flow of the hot and cold water). In this design, to change the volume but keep the temperature constant requires the user to play around with both taps separately until the combined flow is about right, and then make slight alterations to each flow to get the temperature exactly right. A second design is to have one control for volume and another separate control for temperature.

In both designs utility exists because users can control the total volume and temperature of the water in the bath, but in the first design the control is indirect. In the second design the control is direct as changing the volume means using only one control. From this simple example it can be seen that the problem with the first design is that the control is indirect, in other words the utility provided is difficult to access.

The Retail Enquiry system at VW is another example of how the designer's frame of reference influences the design of a system negatively. This system provides a pseudo query language which allows users to generate their own

Table 1. Task-System diagram for the Company Vehicle System

Tasks	Systems
Applications	Manual
Distribution	Manual from production list
Find oldest car in fleet	Manual from vehicle history
Order cars	Company vehicle
Licence cars	Manual
Service cars	Manual
Issue cars	Company Vehicle
Fringe benefits	Finance
Fixed Assets	Finance
Fines	Company Vehicle
History	Company Vehicle

queries. The designer's frame of reference is a database consisting of a number of rows and columns. To query the database, users place the selection criteria for the columns into fields on the screen. These fields are arranged vertically over a number of screens. The user has no control over the format of the output. Output is a file of those rows that satisfy the selection criteria. The user browses this file to get the information needed. Design is determined by the designer's constraints, namely a flat file. The user is presented with a file to browse that is larger than the screen. The user has in effect a window into the file. Consider the case where the user needs information from columns 1, 4, 14, and 16. Because the information extends over more than one screen, the user has to jump back and forth between these screens to access it, which can be time consuming. For single queries users often look up the information from a hard-copy (printout) of the database. This is because the system is difficult to use.

From these, and other examples cited by Goodwin [9], it is clear that the usability of a system is influenced by the accessibility of functionality. From a survey of users at Volkswagen, problems with the accessibility of utility have been divided into three areas [18]. Firstly, problems due to the use of different systems when completing a single job; secondly, problems due to the way a user thinks about completing a job; and thirdly, problems arising from the interaction of the user and the system via the interface. Thus, based on these three areas of problems identified at Volkswagen, the assessment of the accessibility of utility has been divided accordingly into the following analyses: *Integration Analysis*, *Goal Analysis* and *Interaction Analysis*. Analysis techniques for each of these will be outlined and applied to systems at VW. Full details can be found in [18].

Integration Analysis

Integration Analysis shows how a user has to switch between systems in order to complete a job. Some users at VW had problems because they had to make such switches. A *Task-System diagram* is proposed to show how users move between systems when completing a job. The diagram maps user tasks onto computer systems and is constructed as follows: firstly, the tasks comprising the user's job are listed under the *Task* heading. Then, for each task, the system used to complete the task is listed under the *System* heading. A mapping of tasks onto systems is thus obtained. If different systems are needed to complete the job, and switching between these systems is difficult, then usability problems can be expected [3, 6].

Integration Analysis of the Company Vehicle system is used as an example. The system is used to track the fleet of cars leased to employees of VW. The user of the system receives written applications from employees for new vehicles. A list of vehicles that are available for the lease fleet is needed, this is extracted from the vehicle production list by the user. The applicant who has had a car for the longest period of time is the most eligible for a new vehicle. A prioritized list of applicants is generated by the user using the vehicle history listing. From these lists vehicles are allocated to employees. Vehicles must also be sent for servicing and must be licensed. The lease car fleet is an asset, the finance division needs to be informed of any changes to the fleet. Lease cars are a fringe benefit, so employees' tax deductions must be updated. Table 1 shows how the tasks of the user's job map onto systems. From the table it is clear that the user has to switch between a number of systems in order to complete a single job. These switches between systems can be time consuming. Certain portions of the job that are manual at present could be computerised, to make the user's job less tedious. The user also has to indirectly maintain other systems. To improve the usability of the system a mechanism is needed to facilitate system switching, for example a window manager, or by improving system integration.

Goal Analysis

The user has a job to complete. This job can be broken up into a number of tasks. Integration Analysis looks at how these tasks are spread across systems, whereas Goal Analysis looks at the structure of individual tasks. The user has the problem of completing a task. Goal Analysis shows how the user thinks about this problem and how the system forces the user to solve this problem. Card, Moran and Newell [7] state:

> A person attempts to achieve his goals by doing those things the task itself requires. Much of the complexity of human behaviour derives not from the complexity of the human himself (he is simply trying to achieve his goals), but from the complexity of the task environment in which the goal-seeking is taking place. It follows that, to understand and predict the course of human behaviour, one should analyze a task to discover the paths of rational behaviour.

So in order to explain a user's behaviour, the task involved must be analyzed. The Rationality Principle [7] states that humans behave in a goal-oriented way, and within their limited perceptual and information-processing abilities, they attempt to adapt to the task environment to attain their goals. Here computer systems provide the task environment.

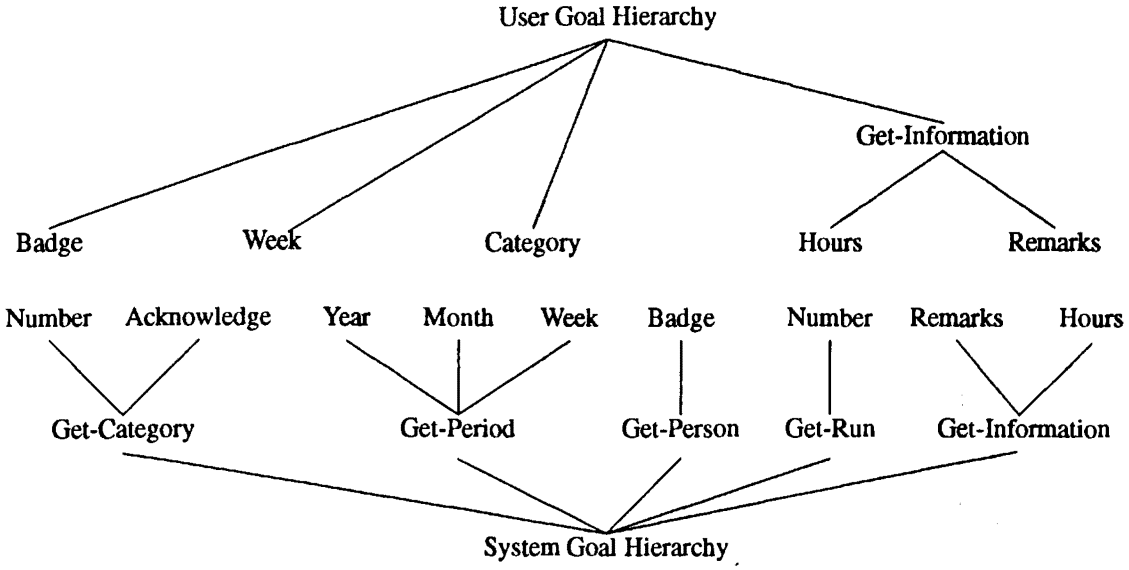

Figure 1. User and System Goal Hierarchy diagrams for Promis

To determine how a computer system influences the cognitive processes of the user, a model for user behaviour is needed. Card *et al.* [7] propose GOMS (Goals, Operators, Methods and Selection rules) to model user behaviour. A user's job consists of a set of tasks. Users tend to break a large task into a series of sub-tasks. The lowest level of sub-tasks are classed as unit-tasks. Within unit-tasks behaviour is highly integrated and between unit-tasks dependencies are minimal. For each task, the user has a set of goals that need to be fulfilled to complete it. A goal is a symbolic structure that defines a state of affairs to be achieved and determines a set of possible methods by which it may be accomplished. A method describes a procedure for accomplishing a goal. Operators are elementary perceptual, motor, or cognitive acts. The execution of an operator is necessary to change any aspect of the user's mental state or to affect the task environment. When an attempt is made to achieve a goal there may be more that one method available to the user to accomplish the goal. Method selection is handled by the selection rules.

So, a task is broken up into a number of unit-tasks by the user. A unit-task results in a goal, which can be subdivided into sub-goals. Attempting to achieve goals eventually causes operators to be executed. It is during the execution of operators that interactions with the physical world take place.

Goal Hierarchy diagrams are proposed to show how a system influences the user's thinking about solving the problem of completing the task. In other words, these diagrams show how the user's thinking is influenced by the computer system. A *User Goal Hierarchy diagram* is a representation of the sequence of the goals fulfilled by a user. The lowest level of the diagram is the sequence of the sub-goals fulfilled to complete the task. Higher level goals are simply groupings of related sub-goals. A *System Goal Hierarchy diagram* is similar to the User Goal Hierarchy diagram, except that in this diagram the lowest level is the goal sequence that the system forces onto the user. By comparing the User and System Goal Hierarchy diagrams, mismatches of the form of additional goals (redundancies) or different goal sequences can be seen.

Goal Hierarchy diagrams are based on a proposal by Kieras and Polson [12] of a device-task mapping, that is the relationship between the characteristics of the user's task and the device (computer system) being used. Their work is based on GOMS. Kieras and Polson derive the user's goal structure (a characterisation or representation of the task) from the production system formalised for the specific computer system. The notion of goal structures is used to obtain Goal Hierarchy diagrams.

A Goal Analysis of the VW time-reporting system Promis provides a good example of a user/system goal mismatch. The user and system goal hierarchies can be seen in Figure 1. At the lowest level the System Goal Hierarchy has nine goals, while the User Goal Hierarchy has only five goals. Not only are there extra goals, but the sequence of goals differ. The user has to remember this alternate goal structure because it is different to the way he would normally solve the problem. The user needs only the week number to identify which period is being reported, but the system forces the user to enter the year, month and week number. Two of these subgoals are redundant, the sub-goals Get-Run-Number and Acknowledge-Category are a result of design constraints, and serve no purpose to the user. Clearly mismatches between the User and System Goal Hierarchy diagrams exist.

In summary, Goal Analysis shows how closely a system conforms to the way the user thinks about the job or task. Any mismatches between how the user thinks about a task, and how the system forces the user to do the task, can be observed by comparing the User and System Goal Hierarchy diagrams.

Interaction Analysis

As far as the user is concerned, a system can be viewed as being in one of a number of different states. To achieve a

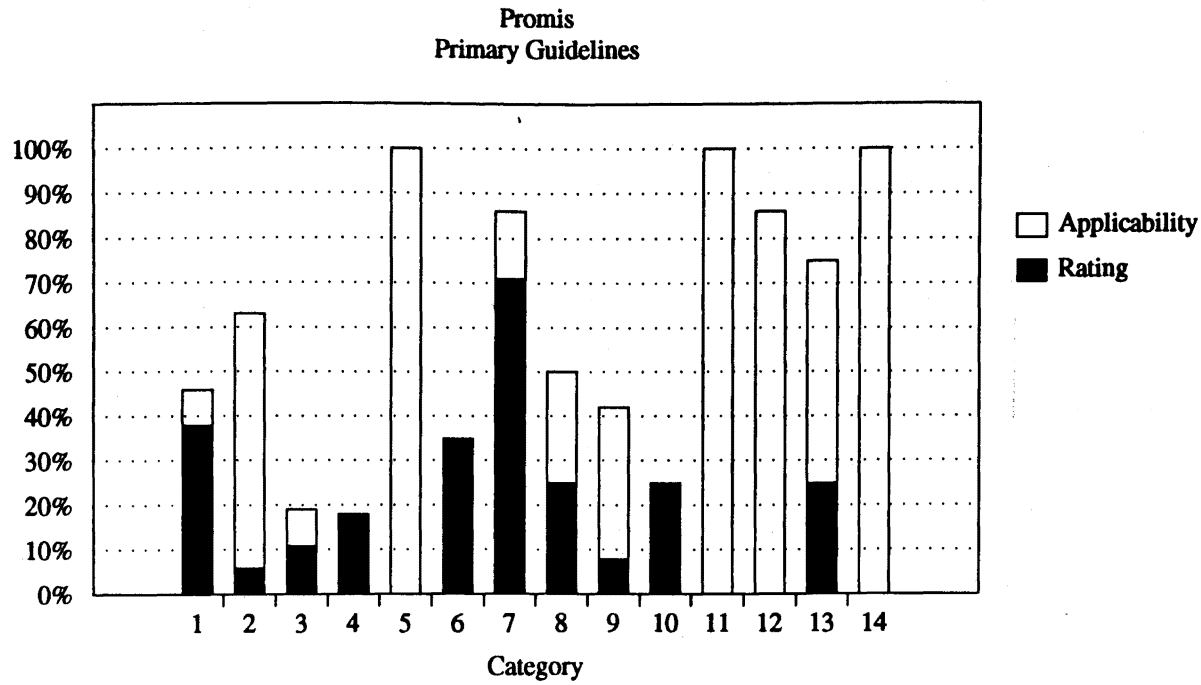

Figure 2. Rating as percentage of Applicability

goal the user, via the interface, manipulates the system to pass through the appropriate intermediate states. The 'ease of use' of the system can be seem as the quality of the interaction between the user and the system when manipulating the system to pass through these states. Many guidelines exist in the literature on how to design easy to use systems. Example guidelines are: 'enumerate items with numbers not letters', 'make the main menu easily accessible', and 'use fixed function keys for common functions' [5]; 'be consistent', and 'provide feedback' [16].

From attempting to determine how well these guidelines are applied in systems at VW, two levels of guidelines were observed, namely, high and low-level guidelines. The high-level guidelines were found to be too general. Examples like 'Be consistent' [16], and 'early focus on the user' [11, 10] are too general to show specific usability faults. Methods based on these guidelines were found to be easy to apply, but the results could not pin-point specific usability problems, such as error messages not providing error correction suggestions. The low-level guidelines were found to be too specific and too numerous. Brown [5] cites 300 guidelines, Smith and Mosier [19] cite 680 guidelines. Examples like 'use a 2 to 5 Hz blink rate', 'display warnings in yellow', and 'restrict saturated blue to background use' [5] are too specific. Using these guidelines is complex and time consuming because of their sheer number, and their specific nature.

The guidelines proposed by Marshall, Nelson and Gardiner [15] provide the compromise between being too specific or too general. The guidelines are derived from research on high-level cognitive processes and thus have a strong cognitive science base. The 162 low-level guidelines are divided into 14 high-level categories. Thus, at the high-level the categories can be used to obtain an overall feel for the usability of a system, but at the low-level the specific guidelines can be used to show individual usability problems. The 14 categories form a framework of concepts for human-computer interaction design. The categories are summarised in Appendix A.

The quality of interaction as a result of using a system is evaluated with these guidelines, because of the compromise between high and low-level guidelines they provide. To do so for a specific system, each guideline is classed as being not applicable, satisfied, or not satisfied in the system. Thus, the 162 guidelines are divided into 14 categories, each category C_i has n_i guidelines. For each category C_i, S_i is the number of guidelines in C_i that are satisfied, D_i is the number of guidelines in C_i that are not satisfied, and P_i is the number of guidelines in C_i that are applicable. From this the applicability and rating of each category is calculated. The applicability A_i is the percentage of the guidelines in C_i that are applicable. The rating R_i is the percentage applicable guidelines in C_i that are satisfied.

$$P_i = S_i + D_i$$
$$A_i = 100 \times \frac{P_i}{n_i}$$
$$R_i = 100 \times \frac{S_i}{P_i}$$

Thus two issues exist for each category, the applicability and the rating (see Figure 2). A category with a low applicability means that the system is not using all the features available in that category, this may lead to usability problems. For example, category 6 (selecting terms, wording and objects) has a number of guidelines involving the use of icons, and guideline 107: 'The chances of fatal consequences through selecting an icon should be minimised.' The systems at VW do not use icons, thus this category has a low applicability. So the applicability of a category shows how many of the usability features in the category

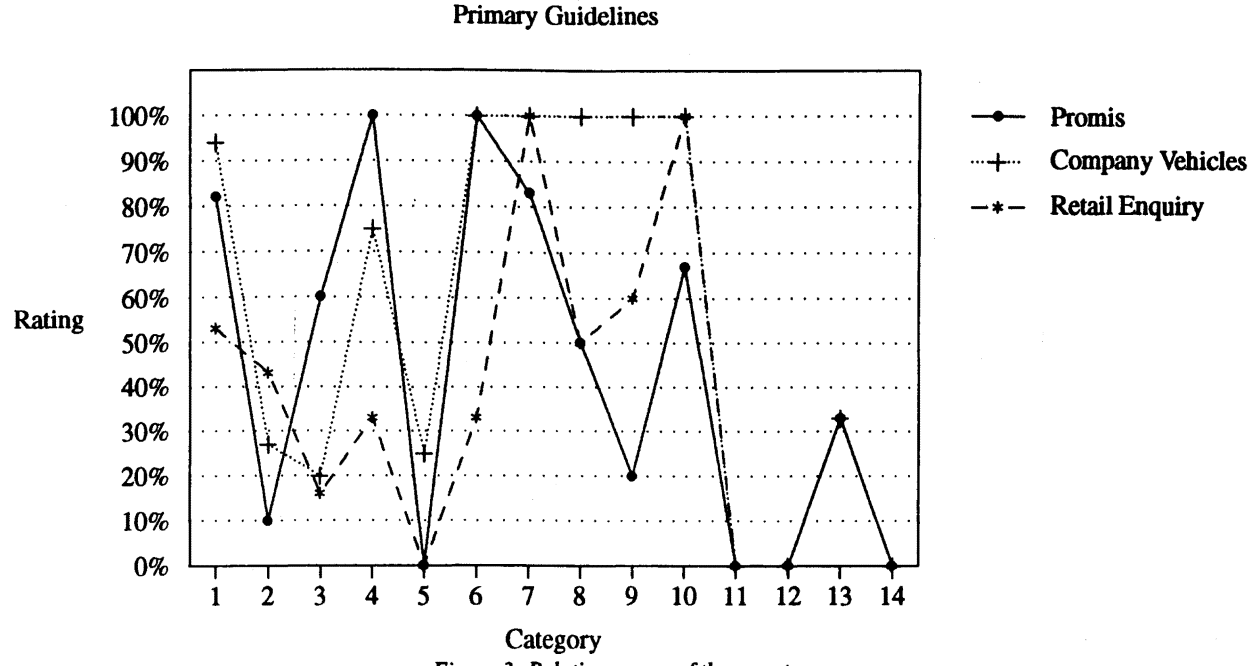

Figure 3. Relative scores of three systems

the system is using. Preferably the applicability of all categories should be high, as this implies that the system is using all the usability features available.

A category with a low rating means that very few of the applicable guidelines are satisfied, thus the system has usability problems in that specific category. For example, category 2 (analogy and metaphor) has an applicability of 63%. In other words, 63% of the guidelines were applicable to the system. Of the 63% of applicable guidelines, only 10% were satisfied. Thus very few of the design guidelines for usability were satisfied resulting in usability problems. So the rating shows how well a system is satisfying the applicable guidelines. Preferably the rating of all categories should be high, as this implies that the system satisfies the applicable guidelines and should thus be easy to use.

So using this diagram enables one to identify specific areas in the system with usability problems. These diagrams can also be used to show problems across systems. Figure 3 shows a comparison of the quality of interaction between three systems. It is clear that for categories 11, 12, 13 and 14 all three systems have the same rating. This is due to the computing environment at VW. Category 11 (navigation) will always rate badly because the mainframe environment at Volkswagen does not lend itself to easy navigation as switching from one system to another can be time consuming. Thus, design flaws across systems can be seen from this diagram.

In summary, for single systems, the analysis shows specific weak usability areas. Across systems, the analysis shows general usability flaws. Although no absolute measure of the quality of interaction between a user and a system is obtained, the analysis can be used to compare the quality of interaction of different systems.

3 Conclusions

A framework for the assessment of the usability of systems has been proposed. The assessment of the usability of computer systems is divided into the socio/political environment, the usefulness of systems, and the accessibility of utility. Albrecht [1] and Marcus [14] cite examples of the socio/political aspects. Without metrics from the social sciences no quantitative assessment of the socio/political environment can be made. However, system designers must be aware that unfavourable socio/political factors can result in the system failing, regardless of how well the system is designed. As far as the usefulness of systems is concerned, a system that is not useful is unlikely to be used. By applying the principles of Software Engineering, system designers can provide useful systems because the User Requirement Specification should address and the users' needs and thus the Functional Specification should ensure that the system has the useful functions required [13].

Based on users problems found at Volkswagen, the analysis of the accessibility of utility has been divided into three areas: Integration Analysis, Goal Analysis and Interaction Analysis. In Integration Analysis the Task-System diagram shows how the user does the tasks comprising a job via systems. From the diagram a designer can get insights into how users use these different systems to complete a job. With this knowledge the usability of the system can be improved by better integration between the systems. In Goal Analysis the User and System Goal Hierarchy diagrams show where users are forced into different goal sequences by an application. The User Goal Hierarchy diagram can possibly be used as the basis of interface design for a system. This should minimise mismatches between the eventual User and System Goal Hierarchy diagrams. In Interaction Analysis the diagrams for the analysis of the quality of interaction show firstly how well an application

implements the usability features available, and secondly, how many of these usability features are actually being utilised. These diagrams can be used to improve the quality of interaction with a system by raising the rating and applicability of the categories. They can also show specific usability problems within a system, or computing environment. Finally, these analysis techniques have been applied to systems at Volkswagen and have successfully shown usability problems in VW applications. Verification of the correctness of the techniques is required. Metrics for the socio/political environment and usefulness are needed, as well as an indication of the relative importance of the 14 categories of guidelines for different types of users. Using these a weighted index of usability should be possible to obtain, and thus an effective analytical tool.

References

1. G Albrecht. 'Defusing technological change in juvenile courts'. *Sociology of Work and Occupations*, **6**:259–283, (1979).
2. R Baeker and W A S Buxton. *Readings in Human-Computer Interaction*. Morgan Kaufmann, California, 1987.
3. L Bannon, A Cypher, S Greenspan, and M L Monty. 'Evaluation and analysis of user's activity organization'. In *Proceedings of CHI '83*, pp. 54–57, New York, (1983). ACM Press.
4. J L Bennett. 'Incorporating usability into system design: The opportunity for interactive computer graphics'. In *Proceedings of the International Conference on Cybernetics and Society*, pp. 1119–1124, New York, (1978). IEEE.
5. C M Brown. *Human-Computer Interface Design Guidelines*. Ablex, Norwood, New Jersey, 1988.
6. S K Card and A Henderson. 'A multiple, virtual-workspace interface to support user task'. In *Proceedings of CHI + GI '87*, pp. 53–59, New York, (1987). ACM press.
7. S K Card, T P Moran, and A Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.
8. K D Eason. 'Towards the experimental study of usability'. *Behaviour and Information Technology*, **3**:133–143, (1984).
9. N C Goodwin. 'Functionality and usability'. *Comm. ACM*, **30**:229–233, (1987).
10. J D Gould. 'How to design usable systems'. In M Helander, ed., *Handbook of Human-Computer Interaction*, pp. 757–789. North-Holland, Amsterdam, (1988).
11. J D Gould and C Lewis. 'Designing for usability: Key principles and what designers think'. *Comm. ACM*, **28**:300–311, (1985).
12. D Kieras and P G Polson. 'An approach to the formal analysis of user complexity'. *International Journal of Man-Machine Studies*, **22**:365–394, (1985).
13. A Macro and J Buxton. *The Craft of Software Engineering*. Addison-Wesley, Workingham, England, 1987.
14. M L Marcus. 'Power, politics, and mis implementation'. *Comm. ACM*, **26**:430–444, (1983).
15. C Marshall, C Nelson, and M M Gardiner. *Design Guidelines*. John Wiley and Sons, 1987.
16. R Molich and J Nielsen. 'Improving a human-computer dialogue'. *Comm. ACM*, **33**:338–348, (1990).
17. D A Norman. *Cognitive Engineering*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.
18. T C Parker-Nance. *Human-Computer Interaction*. University of Port Elizabeth, Port Elizabeth, RSA, 1991.
19. S L Smith and J N Mosier. *Design Guidelines for User-System Interface Software (Technical Report ESD-TR-84-190)*. MITRE Corporation, Bedford, 1984.

Acknowledgements: This research was funded by grants from Volkswagen SA and the FRD. Thanks also to Lester Cowley and Janet Wesson for their comments on earlier drafts.

Appendix A HCI Design Categories

The 14 high-level categories proposed by Marshall, Nelson and Gardiner [15] are:

1. **The Design of Procedures and Tasks**. This category is concerned with the compatibility of the procedures involved in achieving a given task and the known cognitive characteristics of the user. Issues that fall under this category are task sequences, redundant information, informative information, labels, and the consequences of familiarity with the information provided.
2. **Analogy and Metaphor**. Issues in this category are using familiar procedures, training, metaphor mixing and matching, metaphor mismatches, confirmatory input, undoing and redoing tasks, and flexibility in input.
3. **Training and Practice**. Issues in this category are user models, practise, training modes, online help, feedback, and defaults.
4. **Task-User Match**. The structure and arrangement of tasks and procedures should be compatible with the known user cognitive capabilities. The guidelines in this category concern the cognitive characteristics which, if incorporated into system design, will help ensure that the user is not required to perform beyond his or her limitations. Issues in this category are memory demands, concentration and information groupings.
5. **Feedback**. Two types of feedback exist. The first is required feedback, which is needed during task performance, the second is confirmatory feedback which is needed on task or action completion only.
6. **Selecting Terms, Wording and Objects**. This category concerns the rules for the physical presentation of information. Issues in this category are language used in the dialogue, sentence construction, icons and diagrams.

7. **Consistency.** This category is concerned with the importance of consistency and the possible trade-offs that can be made.
8. **Screen Design.** This category is concerned with the way information is presented: the spacial layout and properties, the information organization and format, the shape(s) and continuity of presentation.
9. **Organisation.** This category concerns the structure, functionality, status and consistency of a system. Issues in this category are information grouping, categorisation, and memory demands.
10. **Multimodal and Multimedia Interaction.** This category concerns making effective use of the different modes of communication.
11. **Navigation.** This category concerns ways of making navigation within a system easy.
12. **Adaptation.** Issues in this category are different levels of verbosity in dialogues, partial input, and different levels of use for different users.
13. **Error Management.** Two approaches exist when dealing with errors. The first to prevent or reduce errors, and the second to recover from the errors which inevitably occur. This category concerns both these approaches.
14. **Locus of Control.** The locus of control can reside anywhere from total system control to total user control. This category is concerned with the position of the locus of control.

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

Editor
Professor Derrick G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Email: dkourie@dos-lan.cs.up.ac.za

Subeditor: Information Systems
Prof John Shochot
University of the Witwatersrand
Private Bag 3
WITS 2050
Email: 035ebrs@witsvma.wits.ac.za

Production Editor
Dr Riël Smit
Software Collage (Pty) Ltd
P.O.Box 16650
Vlaeberg 8018
Email: gds@cs.uct.ac.za

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
Rekenaar-
tydskrif

Nommer 10, September 1993
ISSN 1015-7999

Contents