J M Bishop
20/3/90

South African Computer Journal
Number 1
January 1990

Suid-Afrikaanse Rekenaar-tydskrif
Nommer 1
Januarie 1990

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

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

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

Subscriptions

	Annual	Single copy
Southern Africa:	R32-00	R8-00
Elsewhere:	$32-00	$8-00

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

Editorial

At last the first edition of SACJ is available. I trust that readers will find it worth the waiting. There have been a number of teething problems in getting things together, the many details of which need not be spelt out here. One significant challenge was to cope with the consequences of the resignation of Quintin Gee, QI's highly competent production editor. He assisted in the initial phases of getting this publication together but had to resign for personal reasons. It is fitting to acknowledge here not only his initial advice and assistance in getting this first issue of SACJ off the ground, but also the many hours of work that he spent in previously producing QI.

Quintin's resignation meant that a new *modus operandi* for typesetting and printing had to be established. The exercise was not only time-consuming, but also has significant cost implications. Fortunately, the Unit for Software Engineering (USE) at Pretoria University has generously agreed to sponsor this first edition. On behalf of the South African computing community, I should like to thank them for their generosity. Now that they have made a first issue of SACJ possible, it is hoped to solicit the sponsorship of one of the larger computer companies for future editions.

It might be of interest to take readers on a walk through the new journal to highlight various aspects. To begin with, the cover design follows that of several journals whose titles have the format: *The South African Journal of Subject / Die Suid-Afrikaanse tydskrif vir Vakgebied* (where *Subject* and *Vakgebied* are appropriately instantiated). While colours vary, these journals generally have *Subject* and *Vakgebied* restated on the darker portion of the cover. SACJ's title was chosen in preference to a more descriptive but also more cumbersome title such as *The South African Journal of Computer Science and Information Systems*. The appearance of the words *Computer Science and Information Systems / Rekenaarwetenskap en Inligtingstelsels* on the cover are thus out of step with the original inspiration, but seem appropriate under the circumstances.

The inside cover is of interest for several reasons. Firstly, note that Peter Lay has kindly agreed to lighten my task by acting as an assistant editor. He will deal with matters relating to Information Systems. *Contributions in this area should henceforth please be sent directly to him*. Also note that an editorial board of distinguished persons has been assembled. I should like to once again thank board members for adding status to SACJ by agreeing to serve in this capacity. They will be consulted on matters of editorial policy whenever appropriate. Finally, the subscription costs have been increased to keep pace with production costs. This increase does not affect SAICS members, who will continue to receive the journal as one of the benefits of membership.

The guest editorial by Pieter Kritzinger makes for interesting reading. Several points of concern about computer-related research in South Africa are raised. I trust that the article will focus attention on these problems and stimulate a debate which will lead to eventual solutions. It is hoped to make guest editorials a regular feature of future SACJ issues.

Of the eight research papers offered in the journal, four have been gone through the normal channel of refereeing and revisions. The remainder were submitted to the Vth SA Computer Symposium and are published here by invitation. Each paper submitted to the chairman of the symposium's program committee was sent to three referees. A ranking scheme, reflecting an aggregate measure of referee evaluation, was used as a basis for deciding on papers to be presented. After further editorial evaluation, the authors of four of the five highest ranking papers were invited to submit their papers to SACJ. While it was not possible to contact the fifth author in time for this edition, but it may be possible to publish that paper, together with a selection of others from the symposium, in future SACJ editions.

In the section marked *Communications* various items of news arriving at the editor's desk have been published. It was particularly gratifying to receive book review submissions in response to a prior general appeal. There has also been an enthusiastic response from book publishers, who have sent in a number of books for review. Titles are listed in the *Communications* section. Please contact me if you are willing to review one (or more) of these. Naturally, reviews of other books of interest in your possession will also be welcomed.

The final point to highlight in this walk through the journal is the increase in page charges indicated on the back inside cover. These reflect the increased cost of production. Since research papers in SACJ qualify for state subsidy at academic institutions, the charges should not, in general, present major problems for authors. However, it is worth pointing out that the final format of papers submitted significantly impacts on both the financial and editorial load. Submissions in camera-ready format (or nearly so) result in both a cost savings and a speed up of turn-around time by several orders of magnitude. Since many readers may not be familiar with the printing process, it may be helpful to say something about it in order to substantiate this claim.

The printing process basically involves typesetting, shooting (or photographing), and then reproduction and binding. Apart from limiting the amount of material, the printer's client has very little control over the cost of shooting, reproduction and binding. On the other hand, anyone equipped with moderate text- or word processing facilities and a laser printer can go a long way (if not all the way) towards typesetting a paper. Even a partially typeset paper helps significantly, as I will explain below.

By typesetting I simply mean knocking the paper into the right shape and producing a laser printout. The printers regard this is a tedious, error-prone task, even if they start off with an ASCII file rather than a hardcopy of the paper. Consequently, they tend to handle large-scale typesetting by subcontracting the task. Moreover, while they may be willing to typeset uncomplicated text, they tend to balk at text containing specialized mathematical and other notation. However, they are quite skilful at cutting and pasting text, and at enlarging or reducing photographed or scanned diagrams. They are even willing to redraw sketches which are not too complicated.

As a result of the above, I have pressed several authors to do their own typesetting. In cases where it was problematic to produce double column format, a single column of appropriate width was requested. While this is a second-best option, it allows for cutting and pasting to be done by the printers. Some sketches have either been directly reduced from the author's original, while others have been redrawn by the printer. By way of exception, I have personally undertaken the typesetting of a few papers using WordPerfect. However, I would like to avoid this as far as possible in future, and consequently appeal to potential authors to make every effort to do their own typesetting.

From SACJ's point of view encouraging authors to do their own typesetting involves a compromise in that there will inevitably be slight variations in the print from one article to the next (as is in fact the case in this issue). If you are pedantically inclined, you might consider this to be a disaster. Personally, I regard it as a rather neat advertisement for the typesetting skills of SACJ contributors.

As an aside, since the handling of $\mathrm{T_EX}$ files was initially a problem for me, I was pleased to discover that Peter Wood and his colleagues at UCT have mastered the art of producing $\mathrm{T_EX}$ printout in the format now before you. Future authors who use $\mathrm{T_EX}$ should consult them on details.

As to the future, it is not possible at the this stage to commit to a fixed number of SACJ issues per year. The number of issues is constrained by finance, submissions of the right quality, and time available to the editorial staff (including our anonymous and unsung heroes - the referees). The ideal is to produce four issues per year, but this may not always be attainable.

In conclusion, if readers have as much fun in reading this first issue of SACJ as I have had in editing it, the hours spent on it will have been well worthwhile. Hopefully SACJ is destined not only to be a permanent feature of the Southern African computing scene, but also to significantly contribute to research in the region.

Derrick Kourie
Editor

This SACJ issue is sponsored by
The Unit for Software Engineering
(USE)
Department of Computer Science
Pretoria University

Funding Computer Science Research in South Africa

P S Kritzinger
Department of Computer Science, University of Cape Town, Rondebosch 7700

The word *research* has many connotations and is often abused. In everyday language a person does not simply *search for information in a library*, for example, but rather does *research*, thus pretentiously conferring an aura of intellectual activity on an effort which requires very little original thought.

Here I will interpret the term to mean work which generates results that gain international recognition. This implies that the work is published in good international journals or presented at international conferences. I believe this is the only valid index of the quality of research.

With very few exceptions, the computer industry in South Africa is a consumer of computer technology, rather than a developer. In contrast with, say, the chemical industry, there is therefore no tradition of research in computer science in the South African computer industry and computer science researchers therefore have, as virtually their only source of funding, the Foundation for Research Development (FRD) which has its origins in the CSIR.

The FRD was formed in April 1984 with the development and use of research expertise in the natural and applied sciences and engineering as its mission. This mission is primarily directed at the universities, museums and technicons with the ultimate aim of improving the life of all South Africans.

Although the FRD has several programmes, the two which are of main concern to computer scientists are the Core Programmes and the Special Programmes.

FRD Core Programmes foster the optimum development of a scientific and technological knowledge base by supporting individual self-initiated research. These programmes, started only about 4 years ago, have met with considerable acclaim, particularly in regard to the way in which research funding for a particular individual is decided. To qualify for support within a Core Programme, researchers must obtain a certain evaluation status within the FRD and funding is then linked directly and exponentially to the merit of the individual concerned, rather than being linked to the specific project proposed.

In the evaluation process, peer review is strongly emphasised. The researcher himself is expected to nominate referees, whose status and reports play a decisive role in the evaluation. As a result of this evaluation, an applicant is assigned a specific evaluation status category. There are currently 9 categories in all, but the ones of main interest are:

- **A** researchers who are without any doubt accepted by the international community as being amongst the leaders in their field (52);
- **B** researchers not in category A but who nevertheless enjoy considerable international recognition as independent researchers of high quality (182);
- **C** proven researchers who have maintained a constant high level of research productivity and whose work is regularly made known internationally, or proven researchers whose current research output is less but who are actively engaged in scholastic activity (433);
- **P** researchers younger than 35 years of age who have already obtained a doctoral degree and who have shown exceptional potential as researchers (10); and
- **Y** young researchers usually under 35 years of age, who are highly likely to achieve C status by the end of their support period (108).

The number of researchers in the various categories as of August 1989 has been indicated in parentheses above. Of these, only 7 persons are computer scientists: 1 in category B; 3 in category C; and 3 in category Y. Only 4 departments of computer science are involved.

The other main programmes of concern to computer persons are the Special Programmes which aim at developing research manpower in priority areas. After identification of an area that merits particular research development, given local expertise, a Special Programme is launched to address the problem in the national interest.

Although a manager of a Special Programme has to be an FRD evaluated researcher, the same need not be true for the other team members. Regular peer evaluation of researchers as well as evaluation of the progress and results of Special Programmes are considered essential. Special Programme awards will be made for the first time towards the end of 1989. It is therefore not yet known whether proposals already submitted for programmes in computer science have been successful.

It is clear that, in the context explained above, there is virtually no computer science research being done in South Africa - a scary thought which has considerable implications for this country! Why is this so? There are several reasons, but I would like to single out two in particular.

Qualified faculty and students is an abiding problem at the heart of computer science departments. Acquisition of new faculty members is an issue intimately linked to the number of graduate students successfully completing PhD degrees. This problem is by no means unique to South Africa. For instance, data gathered in North America indicates that in 1983 there were over 200 vacancies in the 91 departments that have doctoral programmes in computer science. At the same time, only approximately 250 PhD's were granted in North America - a figure that has remained relatively unchanged for the past several years. A large number of those graduates were attracted to industry and industrial research laboratories. Although I do not have solid data at my disposal, I would think that South Africa produces at most one PhD graduate in computer science per year. There are currently 20 departments of computer science at universities is South Africa. It will therefore take us 20 years to locally produce one new faculty member with a PhD in computer science for every university.

Contributing to the above problem is our current academic image. The graduate student usually sees concerned computer science faculty members as rather harried individuals, having large undergraduate classes, much committee and professional work, and labouring under an ill-fitting model (applicable to more established disciplines) for decisions on tenure, salary and promotion. Further, as undergraduates, many prospective graduate students were not engaged in research projects involving computer science faculty, and for that reason were not exposed to graduate students doing research, and rarely developed a camaraderie with any computer science professionals. At last count there were only 5 individuals in South Africa who completed their computer science doctorate at a university outside South Africa where they had the good fortune to work in an environment in which sufficient faculty and funds were available to create an ethos of research. It is difficult to convince students that their interests and goals can be served by a PhD in computer science or by an academic career.

The second problem, which is of greater concern to me since there is no immediate solution to it, has to do with the fact that senior persons who decide the fate and fortune of academic computer science departments are, in general, individuals whose professional careers started well before computing machines came into every day use - that is to say, in the years B.C. (Before Computers). These persons of influence do not always understand what "computers" are, and what their potential influence upon the workplace in particular and society in general are. As far as research (as opposed to teaching) is concerned, most of them understand that a medical school needs special and expensive equipment (not to mention, expensive faculty) and that engineers must have a workshop and special machinery to teach their students and conduct research. They understand that if one needs to build up a defense industry, it will cost billions of rands; but they are not so sure about computer science, even though many other countries have recognised it as of national strategic importance.

I believe that only time and dedication will lead to a solution of these seemingly insurmountable problems and allow computer scientists to take their rightful place in the research community in South Africa.

Bibliography

P J Denning, [1981], Eating our seed corn, *Communications of the A.C.M.*, **24**(6), pp.341 - 343.

J Tartar (Ed.), [1985], The 1984 Snowbird Report: Future issues in computer science, *Communications of the ACM*, **28**(5), pp490 - 493.

D Gries, R Miller, R Ritchie and P Young, [1986], Imbalance between growth and funding in academic computing science: two trends colliding, *Communications of the A.C.M.*, **29**(9), pp.870 - 878.

J E Hopcroft and D B Krafft, [1987], Towards better computer science, *IEEE Spectrum*, pp.58 - 60.

Four Major Success Criteria for Information System Design

J Mende

Department of Accounting, University of the Witwatersrand, WITS 2050

Abstract

Both in the advancement of Information Systems theory and in practical development projects, significant errors of omission can occur unless one is aware of all aspects of the concept "information system success". At least sixty-five detailed aspects have been identified by previous authors. However one cannot bear them all in mind simultaneously. This paper reduces them to four major sucess criteria:

- *marginal effectiveness R, the extent to which the system's return of value on cost could be increased*
- *user effectiveness N, the extent to which the system's value could be increased without affecting its cost*
- *economic efficiency E, the extent to which the system's cost could be decreased without affecting its value*
- *technical efficiency T, the extent to which waste can be eliminated from the system's processes.*

Keywords: *information systems, sucess criteria, design objectives, effectiveness, efficiency.*
Computing Review Categories: *D.2.O, H.1.OI*

Received January 1989, Accepted May 1989

Introduction

Managers do not assign computer staff to develop information systems unless those systems are likely to benefit the organisation. Therefore the processes of logical and physical system design should be seen as "means to an end", namely to achieve as many improvements in business performance as possible. However logical and physical system designers cannot do that unless they know all the success criteria of the system to be developed. Consider a project to computerise an existing manual procedure. Suppose the system designers were only aware of one success criterion, for instance operating cost. Then they may only strive to achieve one kind of design objective, namely operating cost reduction. They might not even think of setting other objectives such as increased informational value, minimal development cost, etc. This implies that the benefits obtainable from a computer based information system depend on the extent of the designers' awareness of its success criteria. So designers need a comprehensive checklist of major success criteria in order to produce optimal systems.

Methodologists are in a similar position. Suppose that Yourdon and Constantine had established their *Structured Design* methodology [38] with only one success criterion in mind, for instance maintainability. Then users of that methodology would design systems that are maintainable but not necessarily reliable, usable or relevant. So methodologies which ignore major system success criteria will yield designs which are sub-optimal across the full spectrum of possible improvements. In general, this means that methodology authors should at least specify which criteria they are addressing, and should ideally devise methodologies that address *all* facets of system success. To know what is possible, they too need a comprehensive checklist of major success criteria.The present article reviews existing criteria, shows that they are inadequate for these purposes, and then develops a more appropriate checklist.

Existing Success Criteria

Many textbooks and articles on Information Systems mention success criteria. Sixty-five of these are cited below to give an indication of the wide variety available:

- system efficiency and effectiveness [27]
- economy, acceptability, accuracy, timeliness, security and robustness [36]
- portability, maintainability, data compatibility and flexibility [37]
- modifiability [31]
- development cost, operating costs, maintenance costs, capability, stability and adaptability [14]
- ease of use, reliability and generality [39]
- performance, throughput, response, changeability and control [11]
- technical efficiency, resource utilisation, output realisation and relevance [17]
- style complementarity [5]
- development efficiency and operational efficiency [12]
- sufficiency, cost-efficiency and understandability [10]
- utility, quality of information and quality of access [33]

- usability [28]
- hardware compatibility, vendor credibility, integrity, availability and expansion potential [35]
- data integration [6]
- adherence to schedule, integration and simplicity [13]
- support and utilisation [29]
- user satisfaction and currency of output [30]
- usability and functionality [32]
- profitability and contribution to user performance [9]
- learning time, user error rate and user confidence [19]
- reusability [34]-development time, completeness and documentation [26].

At first glance this set of criteria might seem to satisfy the need for a checklist. However, it has four serious drawbacks. First, Miller's Law of seven plus or minus two implies that a human designer cannot possibly bear all the criteria in mind at the same time [25]. Secondly, some criteria are synonymous with others, for example modifiability has the same meaning as changeability. Thirdly, some criteria are subsumed by others, for example both timeliness and accuracy are aspects of effectiveness. Finally, there is no way of knowing whether the list covers all the major aspects of information system success.

To overcome those drawbacks a concise but comprehensive set of criteria is proposed. This is derived from an extended form of the well-known Input--Process-Output model of an information system.

Extended I-P-O Model

Many textbooks, for instance Davis and Olson [4] and Kilgannon [15], introduce an information system basically as a set of Inputs, Processes and Outputs. An information system must produce outputs, otherwise it would be useless. In order to generate those outputs it must carry out transformation *processes*, because if the outputs were already available in required form there would be no need for the system. And the processes require *inputs* - data to be transformed, as well as hardware and software to effect the transformation.

The basic model usually needs to be extended for research in specific areas of Information Systems. For example, Davis and Olson [4, p315] mention that "for control purposes, a feedback loop is added to the basic model". For the purpose of identifying success criteria, environmental features will be added instead.

Like many other systems, an information system interacts with elements of its *environment*. In order to benefit the organisation, the system's outputs must be utilised by one or more users in its environment, e.g. employees, managers, shareholders, customers, suppliers, etc. And in order to receive inputs, there must be sources in the environment to provide them. So users and sources have been added to the basic I-P-O model, giving the extended model shown in fig. 1.

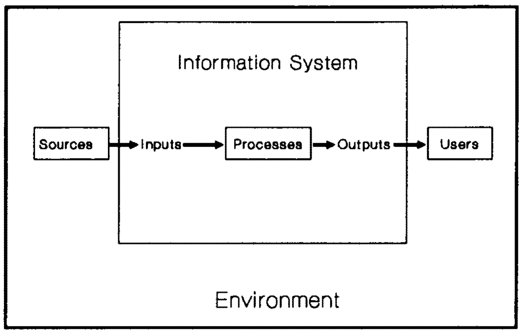

Fig. 1 Model of an information system

This model confirms King's [16] view of system design as a decision-making activity involving choices between alternatives. Users could receive many alternative combinations of data *outputs* with different contents, media and format. Each output combination can be produced with many alternative combinations of collection, storage and extraction processes. Each process combination can be implemented with many alternative *input* combinations involving different types of hardware, software and data. The designers' task is to select the best I-P-O alternative.

Relationships in the Model

The comparative success of two such alternatives depends on the relationships between the components of the extended I-P-O model. Combinatorial Theory [18] predicts that ten distinct binary relationships (R1, R2 .... R10) exist among the sources, inputs, processes, outputs and users in fig. 1. They are enumerated in table 1.

Higher order relationships may also occur between groups of three or four components in the extended I-P-O model. The properties of the higher order relationships comprise those of the component binary relationships, plus additional synergistic effects. As the component properties have already been identified in table 1 and interrelational synergies are beyond the scope of this paper, higher order relationships will not be considered. System success criteria will simply be derived from an analysis of the binary relationships.

R	Components	Characteristics
R1	Processes & inputs.	Alternative processes consume different quantities of hardware, software and data inputs to produce the same outputs.
R2	Processes & outputs.	Alternative processes produce different quantities of outputs from the same inputs.
R3	Sources & inputs.	Alternative inputs incur different charges from hardware suppliers, programmers and data entry personnel.
R4	Sources & processes.	Alternative processes require different inputs and therefore incur different source charges.
R5	Sources & outputs.	Alternative outputs require different processes and therefore incur different source charges.
R6	Users & outputs.	Alternative outputs support users to different extents in making management decisions and in performing operational tasks.
R7	Users & processes.	Alternative processes differ in the timing of outputs and therefore give different user support.
R8	Users & inputs.	Alternative inputs vary in reliability, user friendliness, etc. and therefore give different user support.
R9	Inputs & outputs.	The outputs produced by a system increase with inputs subject to the Law of Diminishing Returns.
R10	Sources & users.	The user support provided by a system increases the host organisation's profit by enabling users to generate revenue and save business costs. Source charges diminish the profit when purchase prices, rentals and salaries are paid.

Table 1. Binary relationship between components

Bases for Comparison

Analysis of relationships R1 and R2 suggests that I-P-O alternatives can be compared on the basis of *waste*. For example, in R1 a process wastes hardware input if temporally independent modules are included in the same program [23]. In R2 a process gives unnecessarily lengthy response times and therefore answers fewer enquiries per day if temporally independent enquiry modules are combined in the same program.

Relationships R3-5 suggest *source charge* as a second basis for comparing IPO alternatives. For example, in R3 an input combination which includes an available software package is usually cheaper than a combination which requires extensive in-house programming [35]. In R4 a process combination involving loosely coupled modules requires less programming effort - and therefore salary - than a tightly coupled combination [38]. In R5 simple outputs are usually easier to program than complex ones [13].

In relationships R6-8 *user support* emerges as another basis for comparison. For example, in R6 an output combination of summary report plus an online enquiry facility for details may be more helpful to a user than a single detailed report [1]. In R7 a process combination giving direct as opposed to sequential access to data files allows immediate answering of ad hoc enquiries. In R8 an input combination which includes flexible software is preferable to an input combination whose software cannot be adapted to changing user needs [28].

Finally in relationships R9 and R10 the concept of *return* emerges as a fourth basis for comparison. One I-P-O alternative is better than another if it gives much more output with only a little more input, particularly if that means much more user support for only a little extra source charge.

Quantification

The four bases can be quantified by considering potential improvements in an economically viable but imperfect I-P-O alternative [22]. The source obligations incurred during the system's lifetime have been discounted at the cost of capital, their total amounting to a cost c. Similarly the user support provided by the system during its lifetime has been estimated at a present value v by assessing the extent to which host organisation profit would decline in the absence of the system [20]. The resulting contribution to the host organisation's profit is: $p_0 = v - c > 0$.

Four distinct improvements are possible. Firstly, the designers may find an alternative combination of processes which wastes none of the existing inputs or outputs. Suppose the improved system gives more value v' with less cost c', resulting in a contribution: $p_1 = v' - c' > po$. Then the wastefulness of the system can be expressed as a fraction in the range 0 to 100%: $T = p_0 / p_1$.

Secondly, the system may be improved by finding the combination of inputs, processes and outputs which incurs least source charges to produce the same value v'. Suppose the improved system now incurs reduced cost c", so that the new contribution is: $p_2 = v' - c'' > p_1$. Then the system's success in relation to sources is represented by $E = p_1 / p_2$.

Thirdly, the system can be improved by finding the I-P-O alternative that yields maximum value with the same cost c". Suppose the improved system now gives increased value v", so that the new contribution is $p_3 = v'' - c'' > p_2$. Then the system's success in relation to users is $N = p_2 / p_3$.

Finally Appendix A suggests that the system may perhaps be improved still further by adding more inputs, processes and outputs. Suppose the optimal expansion alternative yields contribution p_4. Then the system's success in relation to sources and users is $R = p_3 / p_4$.

Success Criteria

From the definitions of T, E, N and R it follows that

$$S = P_0/P_4 = T * E * N * R.$$

The four factors on the right hand side are independent variables. For example, a system may be waste-free but may incur an exorbitant cost. Cost may be minimal yet users may not get the best value for money. Cost may be minimal and value may be maximal, but contribution could be increased by incorporating additional inputs, processes and outputs.

The factor S on the left hand side is a dependent variable. It is strongly affected even by small imperfections in the independent variables. For example if T, E, N and R are all 95% then S is 81%. If T, E, N and R are all 90% then S is a mere 66%.

So S represents the *overall success* of an information system while T, E, N, and R represent *specific* aspects of system success.

The definitions (or implicit meanings) of the existing criteria mentioned at the beginning of this article were compared with the definitions of the new criteria T, E, N and R. Table 2 shows that each existing criterion is either a synonym or a subsidiary criterion of one of the new criteria. Consequently T, E, N and R comprehend *all* existing criteria. For example, "resource utilisation" was defined by Kriebel and Raviv [17] as the extent to which available resources are underutilised - i.e. wasted - and is therefore listed as a sub-criterion of T. "Vendor credibility" was defined by Vaid-Raizada [35] as source ability to provide input at appointed dates, train operators, honour warranties, etc. - i.e. save implementation costs - and is therefore listed as a sub-criterion of E. "Portability" [37], the ability to transfer software from one computer to another, saves future conversion costs and is therefore listed as another E sub-criterion.

Applications

The following nomeclature is proposed:

- T = *technical efficiency*
- E = *economic efficiency*
- N = *user effectiveness*
- R = *marginal effectiveness.*

The names "technical efficiency" and "economic efficiency" were taken from Economics [3]. "User effectiveness" was drawn from Hamilton & Chervany [12]. "Marginal effectiveness" was suggested by a referee.

The four criteria may find both quantitative and qualitative applications in theoretical as well as practical contexts.

Their *quantitative* formulation provides a new perspective on the concept of productivity [7] which may be useful in advancing the Theory of Information Systems. In particular the definitions of p0 to p4 as net present values may allow us to transfer ideas from the Theory of Finance. Quantitative applications may also be possible in the practical construction of mathematical models for information system optimisation. For example, the simple multiplicative relationship between T, E, N, R and S might be used to extend the scope of the Kriebel-Raviv model [17].

Information value being difficult to quantify [1], the most significant applications will probably be *qualitative*. For instance, in practical system development projects the four criteria can be used to:

New criteria	Existing criteria: Synonyms	Sub-criteria of the new criteria
T	Performance, Operational efficiency, Technical efficiency.	Resource utilisation, Throughput, Output realisation, Response.
E	Economy, System efficiency.	Vendor credibility, Development cost, Development efficiency, Development time, Learning time, Confidence, Hardware compatibility, Portability, Maintenance costs, Data compatibility, Data integrity, Documentation, Reusability, Operating costs, User error rate, Maintainability, Changeability, Simplicity, Integration.
N	Utility, Acceptability, System effectiveness, Support, Usability, Contribution to user performance.	Functionality, Usability, Robustness, Ease of use, Adherence to schedule, Adaptability, Quality of access, Control, Security, Integrity, Capability, Reliability, Stability, Flexibility, Utilisation, Satisfaction, Quality of information, Accuracy, Timeliness, Currency, Relevance, Understandability, Style complementarity, Completeness, Generality, Availability.
R	Cost efficiency, Profitability.	Expansion potential, Sufficiency.

Table 2. New criteria subsume existing criteria

- identify all the weaknesses in an existing system
- set comprehensive objectives for the development of a new system
- select methodologies appropriate to the chosen objectives
- compare alternative system designs on multiple criteria.

Their role in objective setting is particularly important. As T, E, N and R comprehend *all* known criteria, they ensure that *all* potential design objectives will be considered. Without them the designer may by default emphasise criteria that are counter to the requirements of management.

The four criteria also serve as qualitative premises for the development of Information Systems Theory. T, E, and N have already been used to:

- identify problem areas for information systems planning [21]
- devise a classification scheme for design rules [24]
- resolve apparent contradictions between design rules [24]
- derive three new system design rules [23].

Many more applications are likely in the future, for instance:

- a hierarchy of system design objectives
- a procedure for prioritising design objectives
- a systematic method of applying formal design rules.

Conclusion

Information system design involves selecting the best system of alternative inputs, processes and outputs to suit a given environment of sources and users. In order to make the choice, designers need some means of comparing alternatives. An analysis of relationships between sources, inputs, processes, outputs and users suggests that alternatives may be compared on the basis of four factors: *waste, source charge, user support* and *return*.

Comparative success criteria can then be defined in terms of a system's contribution to host organisation profit when its inputs, processes and outputs are modified to optimise the four factors.

1 *Technical efficiency* is the relative contribution when physical waste is eliminated.
2. *Economic efficiency* is the relative contribution when inputs, processes and outputs are replaced with the cheapest alternatives capable of providing the same level of user support.
3. *User effectiveness* is the relative contribution when outputs, processes and inputs are replaced with the most valuable alternatives that incur the same source charges.
4. *Marginal effectiveness* is the relative contribution when inputs, processes and outputs are expanded to the point of optimal marginal return.

These criteria encapsulate all known aspects of system success in a concept set small enough to be borne in mind simultaneously. They have already proved useful in previous theoretical work on Information Systems, and are likely be useful in further theoretical work. They should also prove useful to the practising systems analyst/designer, particularly as design *objectives*.

References

[1] N Ahituv and S Neumann, [1986], *Principles of Information Systems for Management, 2nd ed*, Wm. C. Brown Publishers, Dubuque, Iowa, p23 and 54.

[2] D I Cleland and W R King, [1975], *Systems Analysis and Project Management*, McGraw-Hill Kogakusha, Tokyo, p60.

[3] C L Cole, [1973], *Microeconomics,* Harcourt Brace Jovanovich, New York, p120 and p180.

[4] G B Davis and M H Olson, [1984], *Management Information Systems,* McGraw-Hill, New York, p275 and p315.

[5] M De Waele, [1980], Style Complementarity as a Design Criterion for Decision Aids, *Omega,* **8**, 631-637.

[6] W J Doll and M V Ahmed, [1984], Objectives for Systems Planning, *Journal of Systems Management,* **35**(12), 26-31.

[7] S Eilon, [1982], Use and Misuse of Productivity Ratios, *Omega,* **10**, 575-580.

[8] P Ein-Dor and C R Jones, [1985], *Information Systems Management: Analytical Tools and Techniques,* Elsevier, New York, p57.

[9] P Ein-dor and E Segev, [1986], Attitudes, Association and Success of MIS: some empirical results in the context of a Business Game, *The Computer Journal*, **29**, 212-221.

[10] B J Epstein and W R King, [1982], An Experimental Study of the Value of Information, *Omega*, **10**, 249-258.

[11] F Gane and T Sarson, [1979], *Structured Systems Analysis*, Prentice-Hall, Englwood Cliffs, New Jersey, p177.

[12] S Hamilton & N L Chervany, [1981], Evaluating Information System Effectiveness - Part I: Comparing Evaluation Approaches, *MIS Quarterly*, **5**(3), 55-69.

[13] D R Jeffrey and M J Lawrence, [1984], *Systems Analysis and Design*, Prentice-Hall of Australia, Sydney, p116-120.

[14] H F Juergens, [1977], Attributes of Information System Development, *MIS Quarterly*, **1**(2), 31-41.

[15] P Kilgannon, [1980], *Business Data Processing and Systems Analysis*, Edward Arnold, London, p52.

[16] W R King, [1982], Alternative Designs in Information Systems Development, *MIS Quarterly*, **6**(4) 31-42

[17] C H Kriebel and A Raviv, [1980], An Economics Approach to Modeling the Productivity of Computer Systems, *Management Science*, **26**, 297-311.
[18] C L Liu, [1968], *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, p7.
[19] M P Martin, [1987], The Human Connection in Systems Design, *Journal of Systems Management*, **38**(7), 6-13.
[20] J Mende, [1984], A viability criterion for computer system development projects, *S. Afr. J. Bus. Mgmt.* **15**, 144-149.
[21] J Mende, [1986], Some recurrent problems of information systems planning, *Conference on Management & Economic Sciences*, Pretoria.
[22] J Mende, [1987], Three permanent objectives of information systems planning, *S. Afr. J. Bus. Mgmt*, **18**, 41-45.
[23] J Mende, [1987], Three packaging rules for information system design, *Quaestiones Informaticae*, **5**, 32-35.
[24] J Mende, [1988], A classification of partitioning rules for information system design, *Quaestiones Informaticae*, **6**, 63-66.
[25] G A Miller, [1956], The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information, Psychological Review, 63, 81-97.
[26] A R Montazemi, [1988], Factors Affecting Information Satisfaction in the Context of the Small Business Environment, *MIS Quarterly*, **12**, 239-256.
[27] E Myers, [1972], Effective vs. Efficient Computing, *Datamation*, **18**(6), 75-77.
[28] L Orman, [1983], Information Independent Evaluation of Information Systems, *Information & Management*, **6**, 309-316.
[29] B R Parker [1985], A Multiple Goal Methodology for Evaluating Management Information Systems, *Omega*, **13**, 313-330.
[30] L Raymond, [1985], Organisational Characteristics and MIS Success in the Context of Small Business, *MIS Quarterly*, **9**(1), 37-52.
[31] D T Ross, J B Goodenough, and C A Irvine, [1975], Software Engineering: Process, Principles, and Goals, *Computer*, **8**(5), 17-27.
[32] B Shakel, [1986], IBM makes usability as important as functionality, *The Computer Journal*, **29**, 475-476.
[33] E B Swanson, [1982], Measuring User Attitudes in MIS Research: a Review, *Omega*, **10**, 157-165.
[34] W Tracz, [1987], Reusability Comes of Age, *IEEE software*, **4**, 6-9.
[35] V K Vaid-Raizada, [1983], Incorporation of Intangibles in Computer Selection Decisions, *Journal of Systems Management*, **34**(11), 30-36.
[36] S J Waters, [1974], Methodology of computer systems design, *The Computer Journal*, **17**, 17-24.
[37] S J Waters, [1974], *Introduction to Systems Design*, NCC Publications, Manchester, Chapter 8.
[38] E Yourdon and L .Constantine, [1979], *Structured Design*, Prentice-Hall, Englewood Cliffs, New Jersey.
[39] ibid. pp8-13.

Appendix A

A given group of users can be served by many IPO alternatives with 100% T, E, and N. As information is merely one of many resources used by the host organisation, the "expansion path" [7] costs of these alternatives are subject to the Law of Diminishing Returns. So if value were plotted as a function of cost, the resulting curve f would have diminishing slope as in fig. 2.

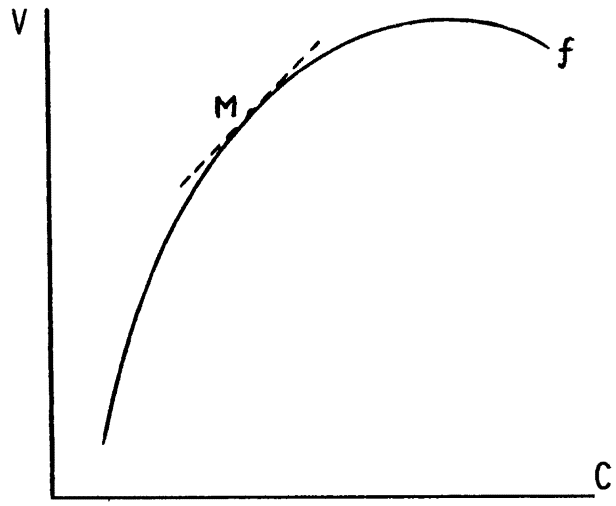

Fig. 2. Production function

All alternatives above the point M where the slope is 1 are uneconomic because the extra value is less than the extra cost. All points below M are sub-optimal because extra contribution is obtainable by increasing the cost. Therefore M represents the optimal alternative. Its contribution v-c is the maximum attainable.

However a typical organisation contains several user groups each requiring its own system. The available resources may not suffice to optimise all systems. Then the best choices for the various groups are at points M_1, M_2, etc. in fig. 3 where the slopes are equal and minimal.

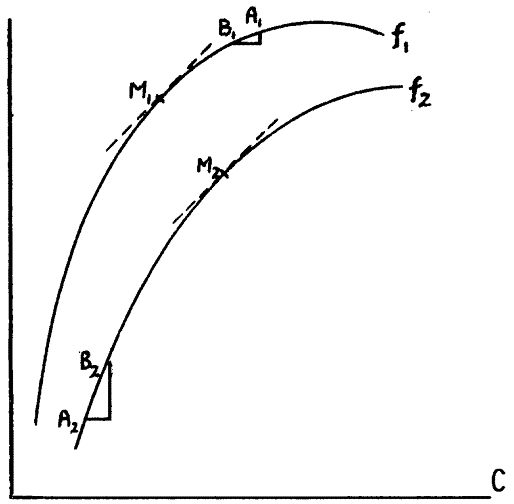

Fig. 3. Two systems

To prove this, suppose the host organisation has two user groups with production functions f_1 and f_2. Also suppose that resources suffice for alternatives A_1 and A_2 - the slope of f_1 at A_1 being less than the slope of f_2 at A_2. Then total value could be increased by replacing A_1 with B_1 which is 1 Rand cheaper, and replacing A_2 with B_2 which is 1 Rand more expensive. Similar value gains could be obtained by moving downwards on f_1 and upwards on f_2 to points M_1 and M_2 where the slopes are equal. The argument is extendable to organisations with more than two user groups.

Computers and the Law

Submitted by Antony Cooper
CSIR

The SA Law Commission has established a commission on "The Legal Protection of Information".

The commission is still in its preliminary stages and the assigned researcher, Mr Herman Smuts, is still preparing the working paper. He does not know when it will be finished, but once the working paper has been prepared, they will invite comments for about two years, before preparing the final report. I have contacted Mr Smuts, and he would be most grateful to receive input at this stage, especially regarding the terms of reference of the commission. His address is:

C/o SA Law Commission
Private Bag X668
PRETORIA
0001

In addition, there is an ad-hoc committee at the Registrar of Copyright investigating numerous copyright issues, including those relating to software and data. Mr Smuts' commission will be liaising with the ad-hoc committee.

I feel that SAICS has an obligation to submit evidence to the commission, and I would appreciate it if you would circulate the members of the Council of SAICS, and perhaps the general membership as well, to solicit ideas concerning SAICS's input.

I shall prepare something for the commission, either in my personal capacity, or in my professional capacity here at CSIR. I would be willing to assist in the preparation of any evidence SAICS might submit.

4th National MSc/Phd Computer Science Conference

Report by Danie Behr
University of Pretoria

This conference was held from 7th to 10th September 1989 at the Cathedral Peak Hotel in the Drakensberg. The conference was attended by 61 postgraduate students from 11 South African universities. Most were engaged in MSc studies, although 5 Phd students also attended. These numbers are encouraging for the South African computer science community. This type of conference is rather unique in that it affords students the opportunity of sharing their research, and getting to know other researchers in the country. The number of Afrikaans and English speaking students attending the conference were roughly equal. Presentations were made in the language preferred by the student. Invitations were sent to all universities with computer science departments. The conference was organized by the students themselves.

Some of the more popular research topics that were presented included expert systems, data communications, computer security, graphics, software engineering, user interfaces and data bases. The main sponsor for this year's conference was the Division for Microelectronic Systems and Communication Technology of the CSIR. The conference was opened with an interesting talk on the myths and motivations of post graduate studies by Prof DG Kourie, acting head of the Computer Science Department at Pretoria University.

The next conference will be presented by the University of Port Elizabeth. People requiring further information about the next conference should contact Andre Calitz, Charmaine du Plessis or Jean Greyling of the Department of Computer Science at UPE.

A list of authors and papers presented at the symposium follows:

S Crosby, University of Stellenbosch
Performance Analysis of Wide Area Computer Communication Networks

A B Joubert, PU for CHE Vaal Triangle Campus
Image Processing Libraries

A Calitz, University of Port Elizabeth
An Expert System Toolbox to assist in the classification of objects

L von Backström, University of Pretoria
Integrated Network Management

R Foss, Rhodes University
The Rhodes Computer Music Network

A McGee, University of Natal
On Fixpoints and Nondeterminism in the Sigma-Lambda Calculus

P G Mulder, Randse Afrikaanse Universiteit
A Formal Language and Automota approach to Data Communications

A Tew, Randse Afrikaanse Universiteit
Drie dimensionele grafiek grammatikas

T C Parker-Nance, University of Port Elizabeth
Human-Computer Interaction: What Determines Computer Acceptance

E Coetzee, PU vir CHO Vaaldriehoekkampus
Opsporing van rande in syferbeelde dmv verskerping en drempelbepaling
D A Sewry, Rhodes University
Visual Programming
A Cooper, University of Pretoria
Improvements to the National Exchange Standard
E S Badier, University of Port Elizabeth
A Computer Assisted Diagnostic System (CADS)
C du Plessis, Universiteit van Port Elizabeth
Persoonsidentifikasie dmv naampassing in 'n genealogiese databasis
J Greeff, University of Stellenbosch
The Entity-Relationship Model and its Implementation
D A de Waal, PU vir CHO
Flat Concurrent Prolog (FCP) en Flat Guarded Horn Clauses (FGHC): 'n Vergelyking
E Naude, UNISA
Interne metodes in Liniêre Programmering
A Deacon, University of Stellenbosch
Global consistency in non-locking DDBMS
A Wilks, Rhodes University
The Synchronisation and Remote Configuration of the Resources in a Computer Music Network
J Greyling, University of Port Elizabeth
The design of a User Interface with special reference to an Interactive Molecular Modelling Program
L Drevin, PU vir CHO
Rekenaarsekuriteit: Verskillende vlakke van kontrole
Dieter C Barnard, University of Stellenbosch
The design and implementation of a modest, interactive proof checker
R A Schmidt, University of Cape Town
Knowledge Representation Systems and the Algebra of Relations
J Hartman, Randse Afrikaanse Universiteit
Die Gebruik van Objek-georiënteerde Programmering in die Moderne Sneltrein Omgewing
S Lawrie, Rhodes University
The Design and Implementation of a System for the Interactive Control of a MIDI-based Studio
E Mulder, Rand Afrikaans University
A Formalisation of Object-Oriented Principles
C J Tolmie, UOVS
Die Ontwikkeling van 'n Ekspertrekenaarstelsel vir die beoordeling van die resultate van die Technicon H1-Bloedselanaliseerder
R Breedt, University of Pretoria
Realism with Ray Tracing
J van Jaarsveld, University of Pretoria
Developing Medical Expert Systems: A knowledge acquisition perspective
W Appel, University of Pretoria
TCP/IP Implementation on Ethernet
E Goedeke, University of Natal
Eggspert's Control Structure
M Harmse, University of Stellenbosch
Modelling of I/O Subsystems
H L Viktor, University of Stellenbosch
A Quantitative Model for Comparing Recovery Techniques in a Distributed Database
M Olivier, Randse Afrikaanse Universiteit
Rekenaarvirusse in Suid-Afrika

Book Reviews

An Introduction to Functional Programming Through Lambda Calculus
by Greg Michaelson, Addison-Wesley, 1988.
Reviewer: Dr. E P Wentworth, Rhodes University

Recently we have seen a number of excellent *second generation* texts on Functional Programming. Michaelson's text assumes some previous programming experience with imperative languages, and presents the functional approach as an alternative paradigm. He begins with a very accessible exposition of the Lambda Calculus, and carefully develops this foundation to encompass the important aspects and paradigms of functional programming. The programming notation is language-independent, although the last chapters are devoted to a brief look at two specific languages, Standard ML and Lisp. The examples and exercises are mainly utility in nature, e.g. "insert a sublist after the first occurrence of another sublist in a list", and can generally be solved in a couple of lines. Answers to the exercises are provided in an appendix.

The approach is slanted towards developing a solid base for understanding functional languages and computing. In this respect the book achieves a good balance between the theoretical underpinnings and their practical application. On the practical side, however, I found the lack of more substantial examples and exercises disappointing. Most programming texts tackle a set of 'standard' problems which are well-understood in the academic community and provide an informal benchmark for comparisons. Since the book is targeted for those already versed in imperative languages and standard algorithms, one might expect the examples to clearly demonstrate the elegance and power of the *problem-oriented* functional approach in these areas. Having laid an excellent foundation I was left with the feeling that the book failed to capitalize and deliver the cherry on the top.

The book is highly recommended as one of the new breed of Computer Science books which gives substantial attention to the fundamentals of the subject without becoming bogged down in over-rigorous formality.

Artificial Intelligence and the Design of Expert Systems
by George F Luger & William A Stubblefield, The Benjamin/Cummings Publishing Co., 1989.
Artificial Intelligence: A Knowledge-based Approach
by Morris W Firebaugh, PWS-Kent Publishing Co., 1989.
Reviewer: Prof G D Oosthuizen, University of Pretoria

One of the primary goals of an Honours course is to introduce students to a field in such a way that they arrive at enough insight into relevant issues to enable them to conduct further research on their own. To this end a text book which is used ought to reflect the current view of the field. Because of the rapid expansion of the field of Artificial Intelligence (AI), we have now finally outgrown the era dominated by the books by Winston and Charniak and McDermott. In the past five to ten years much new work has been done, and new insights have been gained. Introducing AI, therefore, requires a marked shift from the previous emphasis on a few historical systems embodying a number of famous methods, to a more generic approach - an approach which highlights those fundamental representation and search models that span all the different application areas and strategies of problem solving. Of course, since AI still does not have a well developed theory, references to seminal systems continues to fulfil an important role.

Both of the above books are good text books, characterised by a balanced coverage of Prolog and Lisp. They also reflect and consolidate much of the work of the past few years done in areas such as knowledge representation, machine learning, the work done under the heading of Expert Systems and even the recent work on neural networks. But the most important feature that they share is the accurate and up to date overall picture of the subject provided; the broad framework for the understanding of AI that is created without neglecting work of historical importance. There are still references to these works, but they are placed in perspective in relation to new developments.

The book of Luger & Stubblefield (L&S) is more language oriented than Firebaugh's book. A characteristic of L&S is that AI approaches to representation are related to the Object Oriented approach. Whereas L&S includes chapters on advanced AI programming techniques in Prolog and Lisp, it does not address pattern recognition, computer vision and robotics. (Firebaugh has chapters on each of these themes.) These omissions are understandable, since AI has diversified so much recently that it is difficult to cover all applications in one book.

If I had to select one of the books, it would be L&S. Although L&S gives poor coverage of Machine Learning, the book's overall presentation is very good. In particular, the chapters are well-organised, and the overall approach to AI - starting with the core aspects of *representation* and *search*, followed by chapters on AI languages - is coherent. The authors also make very good use of graphical representations and illustrations to convey ideas.

Books Received

The following books have been sent to SACJ. Anyone willing to review a book should contact the editor. The book will be sent to him for review, and may be kept provided that a review is received.

- D Bustard, J Elder & J Welsh, [1988], *Concurrent Program Structures*, Prentice-Hall Inc., Englewood Cliffs.
- R Cafolla & A D Kauffman, [1988], *Turbo Prolog Step by Step*, Merrill Publishing Company, Columbus, Ohio.
- S Hekmatpour, [1988], *Introduction to LISP and Symbol Manipulation*, Prentice-Hall Inc., Englewood Cliffs.
- K L Clark & F G McCabe, [1984], *micro-PROLOG: Programming in Logic*, Prentice-Hall Inc., Englewood Cliffs.
- D Crookes, [1988], *Introduction to Programming in Prolog*, Prentice-Hall Inc., Englewood Cliffs.
- M J C Gordon, [1988], *Programming Language Theory and its Implementation*, Prentice-Hall Inc., Englewood Cliffs.
- J G Hughes, [1988], *Database Technology : A software engineering approach*, Prentice-Hall Inc., Englewood Cliffs.
- R Milner, [1989], *Communication and Concurrency*, Prentice-Hall Inc., Englewood Cliffs.
- T J Myers, [1988], *Equations, Models and Programs*, Prentice-Hall, Inc., Englewood Cliffs.
- N C Rowe, [1988], *Artificial Intelligence through Prolog*, Prentice-Hall Inc., Englewood Cliffs.
- D A Protopapas, [1988], *Microcomputer Hardware Design*, Prentice-Hall Inc., Englewood Cliffs.
- H Eisner, [1988], *Computer-aided Systems Engineering*, Prentice-Hall Inc., Englewood Cliffs.
- S H Unger, [1989], *The essence of logic circuits*, Prentice-Hall Inc., Englewood Cliffs.
- R J Young, [1989], *Practical Prolog*, Van Nostrand Reinhold, New York.

NOTES FOR CONTRIBUTORS

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as a Communications or Viewpoints. While English is the preferred language of the journal papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted **in triplicate** to the editor.

Form of Manuscript

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

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be on separate sheets of A4 paper, and should be numbered and titled. Figures should be submitted as original line drawings, and not photocopies.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin. Distinguish clearly between such cases as:
  - upper and lower case letters;
  - the letter O and zero;
  - the letter I and the number one; and
  - the letter K and kappa.
- References should be listed at the end of the text **in alphabetic order** of the (first) author's surname, and should be cited in the text in square brackets. References should thus take the following form:

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

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

  [3] S Ginsburg, [1966], *Mathematical theory of context free languages,* McGraw Hill, New York.

Manuscripts *accepted* for publication should comply with the above guidelines, and may provided in one of the following three formats:

- in a **typed form** (i.e. suitable for scanning);
- as an **ASCII file** on diskette; or
- **in camera-ready** format.

A page specification is available on request from the editor, for authors wishing to provide camera-ready copies.

Charges

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

Typed format:	R80-00
ASCII format:	R60-00
Camera-ready format :	R20-00

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

Proofs

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

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

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

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal

Number 1, January 1990
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 1, Januarie 1990
ISSN 1015-7999

Contents

EDITORIAL 1

GUEST EDITORIAL

Funding Computer Science Research in South Africa
P S Kritzinger 3

RESEARCH ARTICLES

The NRDNIX Distributed Database Management System
M Rennhackkamp 5

Finding Regular Paths in Acyclic Graphs
P Wood 11

Programming Using Induction
C Meuller 19

A Design Environment for Semantic Data Models
J Kambanis 24

The Use of a Lattice for Fast Pattern Matching
G D Oosthuizen 31

Image Reconstruction via the Hartley Transform
H C Murrel and D Carson 36

Four Major Success Criteria for Information System Design
J Mende 43

A Multi-criteria Partitioning Technique for Information System Design
J Mende 50

COMMUNICATIONS

Computers and the Law 60
4th National MSc/PhD Computer Science Conference 60
Book Review 61
Books received 62