ISSN 0254–2757

QI

QUÆSTIONES INFORMATICÆ

Volume 6 • Number 1

May 1988

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

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

QUÆSTIONES INFORMATICÆ

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

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

Wirth / Hoare / Dijkstra / Gries /
Bach

Kürchen

Kowalski

Guest Editorial.

Subscriptions

The annual subscription is

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

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

Lizette

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

Editorial

Volume six of QI heralds several changes. The most visible is the change in format. The black on red cover has been changed to a more readable blue on white, but we have retained the style of the old cover, for the sake of continuity. The papers are now set in a tighter format, using double columns, which will enable more papers to be published for the same cost.

For authors, the most significant change is that as from Volume 6 Number 2 (the next issue), a charge will be made for typesetting. The charge is quite modest – R20 per page – and will enable us to keep up the high standards that we have become used to with QI. It is worth recording that the alternative to this suggestion was that authors should present camera-ready typescript, as is done for *Quæstiones Mathematicæ*. Given that document preparation and electronic typesetting is one of the areas of computer science that we can feel proud of, it seemed right that our journal should use the most modern techniques available. Fortunately, the two controlling bodies, the CSSA and SAICS, eventually agreed to our proposal and the result is the professional journal you have in front of you now.

Supporters of QI may be interested in a few statistics that I compiled when I took over the editorship from Gerrit Wiechers in April this year. In the past two years (June 1985 to June 1988), 73 papers have been received. Of these 39 (53%) have appeared, 19 have been rejected or withdrawn (26%) and 15 (21%) are either with authors for changes or with referees. If we look at the complete picture for Volumes 4 and 5, we find the following:

Volume	Issues	Papers	Pages	Ave. pages per paper
5	3	27*	220	7.7
4	3	21	136	6.4

Although this issue contains one very long paper of 18 pages, the future policy of QI will be to restrict papers to 6 or 7 printed pages, and prospective authors are asked to bear this in mind when submitting papers.

For the future, we are hoping to move towards more special issues. Many of the papers being published at the moment were presented at the 4th SA Computer Symposium in 1987. Instead of continuing the policy of allowing such papers to be accepted by QI without further refereeing, we are hoping to negotiate with Conference organisers to produce special issues of QI. Thus the proceedings would *ab initio* be typeset by QI and all the papers would be in a single issue. Given the competitive charges of QI, there will be financial gains for both parties in such an arrangement.

As this is my first editorial, it is fitting that it should close with a tribute to the previous QI team. My predecessor as editor was Gerrit Wiechers. Gerrit took over the editorship in 1980 and served the journal well over the years. With his leadership, the number and quality of the papers increased to its present healthy state. I must also extend a big thank you to Conrad Mueller and the University of the Witwatersrand who pioneered desk top publishing of QI in August 1985, using the IBM mainframe and its laser writer. Without Conrad's diligence and the excellent facilities provided by the Wits Computer Centre and subsequently the Computer Science Department, QI would easily have degenerated into a second-rate magazine. Quintin Gee, also of the Wits Computer Science Department, has taken over from Conrad and has raised the production quality of QI to new heights, as this issue testifies.

I look forward to your help and support in the future. Long live QI!

Judy M Bishop
Editor
June 1988

A Structural Model of Information Systems Theory

J Mende
Department of Accounting University of the Witwatersrand, PO Box 1176 Johannesburg 2000

Abstract

The laws and techniques of mature subjects such as Physics and Electrical Engineering are logically connected to one another in the form of a deductive network There is a foundation of basic laws deriving from them are successive layers of logically consequent laws and techniques The field of study "Information Systems' (I S) shares a common feature with Physics and Electrical Engineering All three provide the knowledge necessary for designing Man s artefacts Consequently we can expect their logical structure to be similar That means I S techniques should be logically connected to I S laws and I S laws should be logically connected to laws of fundamental subjects such as Computer Science Psychology and Management

***Keywords** information systems metatheory research*
***Computing Review Category** H O*

Received December 1987, Accepted April 1988

Islands of Knowledge

Since their invention some forty years ago, electronic computers have spread rapidly into many spheres of human activity Coupled with rapid advances in the technology itself, the enormous rate of computerisation has deluged the Information Systems researcher with urgent problems of practical systems development and management As a result, the academic field of study "Information Systems" (I S) has evolved rapidly, but not altogether soundly

In particular, many I S textbooks present their subject matter as independent "islands of knowledge" – much like the "islands of mechanisation" [1] prevalent in the business firms of the nineteen-sixties For instance, Gane & Sarson's "Structured Systems Analysis" [5] starts with several law-like statements in chapter 1 Yet there is no reference to those laws in the subsequent chapters on the techniques of data-flow diagramming, process-logic analysis, etc A similar omission occurs in Jackson's "Principles of Program Design" [6] The book describes a 'basic design technique' of data structure analysis, program structure formation and task allocation Yet this technique is almost entirely unsupported Jackson makes no attempt at logical derivation from underlying laws, indeed he does not even state such laws

Further examples of logical omission can also be found in Martin's "Strategic Data Planning Methodologies" [9], Lundberg's "Information Systems Development" [8], Weinberg's "Structured Analysis" [13] and G B Davis's "Strategies for information requirements determination" [2]

In order to identify and avoid such shortcomings, researchers should consider what kinds of knowledge the I S practitioner needs, and how different types of knowledge are interconnected A previous paper in Quæstiones Informaticæ – "Laws and Techniques of Information Systems" [11] – introduced a general classification of knowledge types The present paper now extends that classification, establishes a general model of utilitarian knowledge, and derives a specific model of I S knowledge

Laws and Techniques

The academic field of study "Information Systems" is concerned with a particular kind of man-made **artefact** [3] computer based systems which produce information Information systems are similar in purpose to many other devices that people have developed over the ages, from simple hoes and clay tablets through ploughs and ledgers, to tractors and accounting machines All such inventions were motivated by the same objective, namely increased human **productivity** The prospect of ever greater output per man per hour constantly lures us on to seek more powerful artefacts However, the designs of our artefacts are critically dependent upon the knowledge base provided by subjects such as Physics, Electrical Engineering, Chemistry, Chemical Engineering, Biology, Agriculture – and **Information Systems** The previous paper showed that this dependency relationship strongly influences the desired structure of those subjects Firstly, the dependency demands that the knowledge base should include two distinct types of ideas laws and techniques

The process of designing any artefact involves a series of decisions If those decisions violate the characteristic properties of the artefact's components, then the artefact will not work as intended, and the design process will therefore be ineffective So effectiveness demands that design decision-making include inferences from **laws** – statements which describe the attributes of the various entities that are combined to form the artefact [11] For example, in deciding the appropriate curvature of a microscope's lenses, the optical designer needs to make deductions from Snell's Law of diffraction

> "The sine of the angle of refraction bears a constant ratio to the angle of incidence" [7]

In designing Man's various artefacts , some of those decisions are made over and over again For efficiency of the design process, such recurring decisions need to be supported by **techniques** – statements which prescribe the steps a designer should take to reach a conclusion quickly For example, designers of optical instruments often face decisions on the spacing between lenses These decisions can be made more rapidly if one applies standard "graphical ray tracing" techniques [7] instead of Snell's Law

So for efficiency and effectiveness, the Design Process in general requires laws which describe the **operands** of design, and techniques which prescribe feasible sequences of decision-making **operations** The operations of an effective technique must be consistent with the characteristics of the operands This means that techniques should be logically connected to underlying laws [11] For example, the optical ray tracing techniques mentioned above are logical consequences of Snell's Law Similarly electrical engineering techniques of electric circuit analysis were derived form underlying laws of electricity, chemical engineering techniques of mass and energy transfer follow from basic laws of mass and energy conservation, and so on

Natural and Artificial Laws

Secondly, the dependency between Knowledge and Design demands two distinct types of laws – natural, and artificial Modern artefacts represent design "hierarchies" They consist of artificial components, which in turn are composed of natural components For example, an artesian well consists of a pump, pipes and an electric motor the motor in turn consists of copper wire and an iron frame Natural components such as copper and iron are described by "Laws of Nature" – laws that reflect their **inherent** properties On the other hand, artificial components such as motors have **contrived** properties which transcend those of their natural constituents These additional properties are described by "Laws of the Artificial" [11] For example, the natural metallic conductors in a motor are subject to Ohm's Law – a law of Nature On the other hand, an artesian well is described by overall performance formulae which are Laws of the Artificial

Again, these two types of knowledge should be connected The contrived properties of an artificial entity are functions of the inherent properties of its natural constituents That means artificial laws should be logically related to natural laws For example, the artificial laws of electric motors follow logically from natural laws of resistance and induction, the artificial laws of capacitors follow from the natural law of electrostatic force, and so on

General and Specific Laws

Thirdly, the Design-Knowledge dependency also requires a distinction – not mentioned in the previous paper – between "general" and "specific" laws Design components have two kinds of properties – general and specific – and those need to be described by corresponding types of laws For example, very many mechanical design components involve motion at sub-light velocities The set of all "sub-light motions" is described by Newton's **general** Laws of Motion However, that set contains smaller subsets of specific motions, such as rotation, oscillation and orbital motion [4] These subsets are described by **specific** laws – such as Kepler's Laws of orbital motion

If a design decision involves an operand which is a member of a specific sub-class within a general class of design components, the decision can be made more efficiently using the specific rather than the general law For example, in many orbital calculations it is easier to use Kepler's Laws rather than Newton's Laws Similarly in optical design involving "thick" lenses, the Gaussian Formulae [7] are more convenient than Snell's Law So the design process requires both general and specific laws

General and specific laws are necessarily related Suppose a large class of design operands are described by a general law Then the same law must also apply to every sub-class That means the specific law of the sub-class must be consistent with the general law So it should be possible to deduce the specific law from the general law using the specific properties as "boundary conditions" For example, Kepler's Laws are deducible from Newton's Laws, and the Gaussian Formulae can be deduced from Snell's Law

Knowledge For Design

The foregoing analysis suggests that the knowledge required by the designers of artefacts can be represented by Fig 1 At the base of the network there are general Laws of Nature Deriving from that foundation we have specific Laws of Nature, then

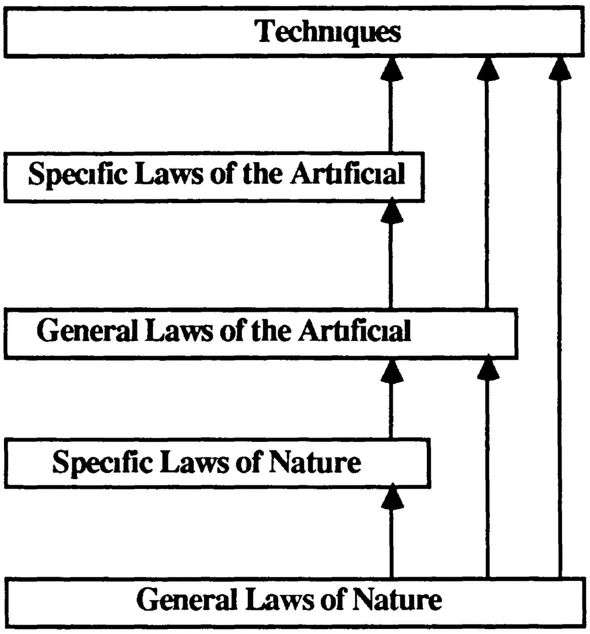

Figure 1 Network of Design Knowledge

general as well as specific Laws of the Artificial, and finally techniques

The Subject Information Systems

The network includes laws and techniques belonging to many different fields of study Physics, Mechanical and Electrical Engineering provide laws and techniques for the design of mechanical and electrical devices Chemistry and Chemical Engineering supply laws and techniques for the design of chemical processes Biology and Agriculture furnish laws and techniques for the design of farming processes

Then there is also the subject Information Systems Its function is to support the design of informational artefacts It ought to provide laws and techniques to support obvious design activities such as configuration and network design, program and physical system design, logical system design (analysis), DP-organisation and human-interface design Furthermore, it should provide laws and techniques for information systems planning, project management and other decision-making activities whose names do not contain the word "design" The reason is that

> "Everyone designs who devises courses of action aimed at changing existing situations into preferred ones" [12]

Therefore virtually the entire subject matter of Information systems is part of the Knowledge Base for design This implies that I S laws and techniques should reflect the logical network structure of Fig 1

Knowledge About Information Systems

Thus our subject-matter should include

- laws and techniques to support decisions involving **artificial** entities such as hardware, software, systems, projects etc
- laws and techniques to support decisions involving **natural** entities such as the human element in systems and projects

For effective and efficient decision-making, the techniques should be logically connected to laws, artificial laws should be logically connected to natural laws, and specific laws should be connected to general laws

Most general laws of human behaviour, management, and computers properly fall within the ambits of other subjects – Psychology, Management Theory, Computer Science, etc Accordingly, many I S laws ought to be specific instances of general laws established in other subjects Therefore one would expect the subject to be structured like a tree I S laws should be logically derived from external laws, like a tree stem connected to ground water by its **roots** I S techniques should be logically derived from I S laws, like a tree's foliage connected to the stem by branches (Fig 2)

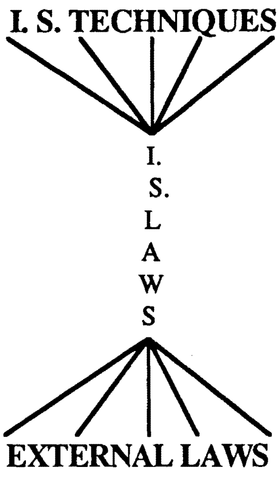

Figure 2 Tree of I.S. Knowledge

Structural Evidence

Empirical evidence of this structure can be found in two I.S. publications. Firstly, Yourdon & Constantine's "Structured Design" [14] reflects the anticipated root and stem structure. It refers to an external law, namely Miller's psychological law:

> "People can mentally ... deal with ... only about 7 ... concepts at a time" (p69).

From Miller's general natural law the book derives a specific natural law:

> "we can win if we can divide any task into independent sub-tasks" (p70).

Then from that law the book logically derives a general Law of the Artificial:

> "total systems cost will be strongly influenced by the degree of coupling between modules" (p85).

This leads on to a specific Law of the Artificial:

> "data-coupled systems have lower coupling than control-coupled systems" (p.86).

Secondly, the author's own paper, "A priority criterion for serial computer system development projects" [10] provides evidence of the expected branch structure. It presents a three-step technique for determining the relative priority of systems development projects: a) estimate parameters, b) calculate priorities, and c) rank projects in declining priority sequence. This technique has been derived mathematically from four artificial laws:

1. a system's contributions decline with age
2. a distant future contribution is less valuable than an equal contribution received in the near future
3. organisational growth increases a system's contribution
4. organisational learning enhances the system's contribution

These two publications suggest that Information systems laws and techniques can conform to the predicted tree structure. However, as indicated at the beginning, many of our publications lack the expected logical connections.

Practical Implications

The prevalence of insular techniques suggests that the subject Information systems is deficient in laws. This implies that design decisions are being made in practice with inadequate formal knowledge of the properties of design components, and are therefore likely to be ineffective. As a result, practitioners are obliged to adopt a trial and error approach.

Insular techniques are also easy to mis-apply. If a decision maker is unaware of a technique's implicit assumptions, he cannot recognise its limitations, and will use it blindly. Then if he applies the technique in a situation where the design components do not actually comply with the underlying laws, the resulting decision will be ineffective. So a costly practical learning process is necessary before one can identify situations where the technique is or isn't applicable. This may explain many an employer's preference for experience before qualifications when hiring computer personnel.

Furthermore, an insular technique is in danger of unwarranted condemnation. In the absence of explicit assumptions and logical derivation the reader may well suppose that it is intended to be universally applicable. Then if the reader has experience of a situation in which it fails, that single exception may totally invalidate the technique from the reader's point of view, even though it might be very useful in many other situation.

Conclusions

The tree model abstracts the distilled experience of researchers in Physics, Engineering, etc and transfers it to the field of Information Systems. We can benefit from that experience in three ways. Firstly, the model induces a healthy scepticism of techniques which are unsupported by logical derivation from underlying laws. Conversely, it lets us appreciate the value of those few publications which introduce laws and techniques with page after page of abstract reasoning before proceeding to concrete applications.

Secondly, it suggest ways of identifying and correcting insular techniques in our existing literature. Such techniques can be identified simply by checking their logical connections. They can then be corrected by isolating the implicit assumptions on which they are based, and developing a connecting chain of reasoning.

Finally, the model provides guidelines for future research. It recommends that we deduce I.S. techniques from underlying laws. Similarly, it suggests that we deduce I.S. laws from root laws in Psychology, Management, Computer Science, etc. Above all, it urges us to pay more attention to the **theoretical development** of our subject.

References

[1] S. Blumenthal, [1969], *Management Information Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 2.

[2] G.B. Davis, [1982], Strategies for information requirements determination, *IBM Systems Journal*, **21** (1), 5-30.

[3] P. Drucker, [1981], Work and Tools, *Ann. History of Computing*, **4** (4), 342-347.

[4] G. Fowles, [1962], *Analytical Mechanics*, Holt, Rinehart & Winston, New York.

[5] C. Gane and T. Sarson, [1979], *Structured Systems Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.

[6] M.A. Jackson, [1975], *Principles of Program Design*, Academic Press, London.

[7] F. Jenkins and H. White, [1957], *Fundamentals of Optics*, McGraw-Hill, New York, 34-36, 46-48, 63-68, 119-128.

[8] M. Lundberg, G. Goldkuhl and A. Nilsson, [1981], *Information Systems Development*, Prentice-Hall, Englewood Cliffs, New Jersey.

[9] J. Martin, [1982], *Strategic Data Planning Methodologies*, Prentice-Hall, Englewood Cliffs, New Jersey.

[10] J. Mende, [1985], A priority criterion for serial computer system development projects, *S.Afr.J. Bus. Mgmt.* **15** (3), 55-60.

[11] J. Mende, [1986], Laws and Techniques of Information Systems, *Quæstiones Informaticæ*, **4** (3), 1-6.

[12] H.A. Simon, [1969], *The Sciences of the Artificial*, M.I.T. Press, Cambridge, Massachusetts, 55.

[13] V. Weinberg, [1980], *Structured Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey.

[14] E. Yourdon and L. Constantine, [1979], *Structured Design*, Prentice-Hall, Englewood Cliffs, New Jersey.

NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles and exploratory articles of general interest to readers of the journal. The preferred languages of the journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to:

Professor J M Bishop
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins.

The first page should include the article title (which should be brief), the author's name and affiliation and address. Each paper must be accompanied by an abstract less than 200 words which will be printed at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review Categories.

Manuscripts may be provided on disc using any Apple Macintosh package or in ASCII format.

For authors wishing to provide camera-ready copy, a page specification is freely available on request from the Editor.

Tables and figures

Tables and figures should not be included in the text, although tables and figures should be referred to in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Figures should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the authors name and figure number. Original line drawings (not photocopies) should be submitted and should include all the relevant details. Photographs as illustrations should be avoided if possible. If this cannot be avoided, glossy bromide prints are required.

Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters; between the letter O and zero; between the letter I, the number one and prime; between K and kappa.

References

References should be listed at the end of the manuscript in alphabetic order of the author's name, and cited in the text in square brackets. Journal references should be arranged thus:

[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.

Proofs

Proofs will be sent to the author to ensure that the papers have been correctly typeset and *not* for the addition of new material or major amendment to the texts. Excessive alterations may be disallowed. Corrected proofs must be returned to the production manager within three days to minimise the risk of the author's contribution having to be held over to a later issue.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion of recent problems.