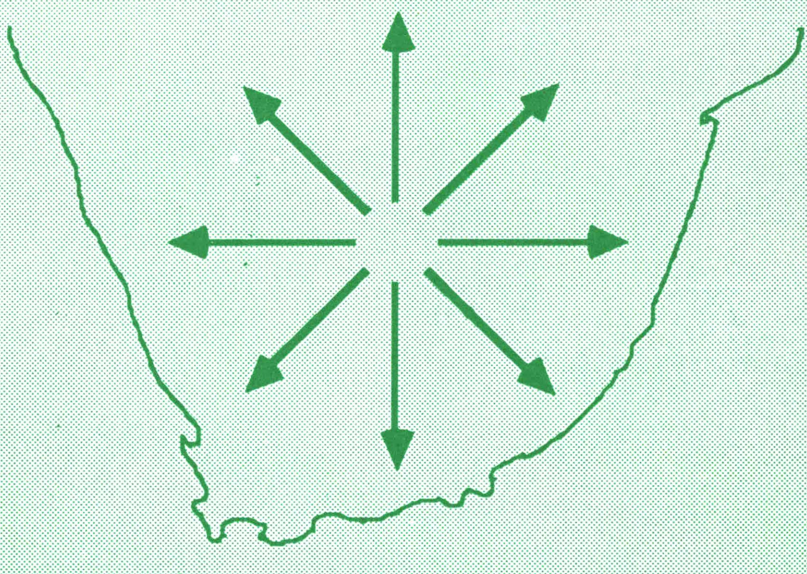
South African Computer Journal
Number 7
July 1992

Suid-Afrikaanse Rekenaar tydskrif
Nommer 7
Julie 1992

SPECIAL ISSUE



7th SA COMPUTER RESEARCH SYMPOSIUM

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:	R45,00	R15,00
Elsewhere	$45,00	$15,00

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

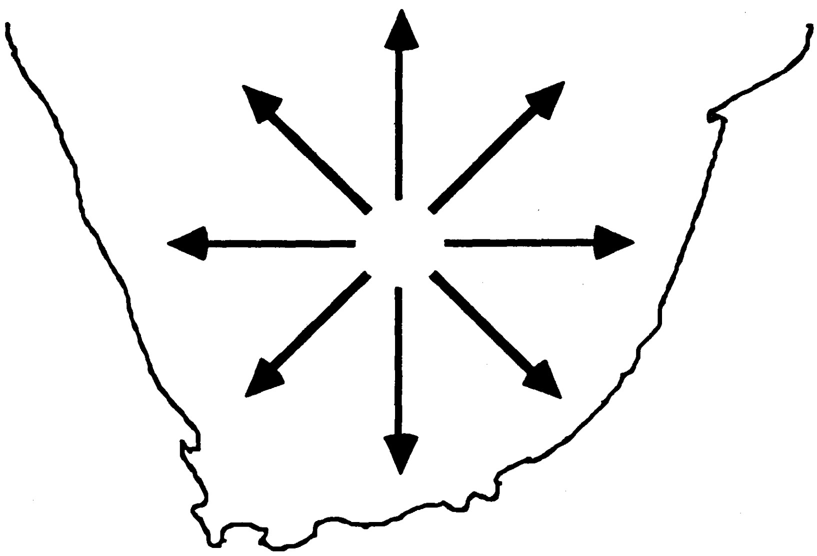
7th Southern African Computer Research Symposium

Karos Indaba Hotel, Johannesburg

1 July 1992

PROCEEDINGS

Guest Editor: Judy M Bishop



PERSETEL

Organised by the SA Institute of Computer Scientists
in association with the Computer Society of SA

Sponsored by Persetel and the FRD

SPECIAL ISSUE - 7th SA COMPUTER SYMPOSIUM

PREFACE

When the first SA Computer Symposium was held at the CSIR in the early eighties, it was unique. There was no other forum at the time for the presentation of research in computer science. In the intervening decade, conferences, symposia and workshops have sprung up in response to demand, and now there are several successful ventures, some into their third or fourth iteration. Each of these addresses a specific topic - for example, hypermedia, expert systems, parallel processing or formal aspects of computing - and attracts a specialised audience, well versed in the subject and eager to learn more. For the main part, the proceedings are informal, and certainly not archival.

SACRS, though, is still unique, in that it deliberately covers a broad spectrum of research in computing, and in addition, seeks to provide a lasting record of the proceedings. To achieve the second aim, we negotiated with the SA Institute of Computer Scientists for the proceedings to form a special issue of the SA Computer Journal, and the copy you have in front of you is the result. The collaboration between the symposium committee and the journal's editorial board placed high standards on the refereeing and final presentation of the papers, to the symposium's benefit, while we were still able to maintain a fresh, audience-oriented approach to the selection of papers.

This is SACJ's first such special issue, and the largest issue (at 145 pages) to date. We hope that it is only the beginning of future such collaborations.

In all 29 papers were received, all were refereed twice, and 19 were chosen for presentation by the programme committee. All the papers were thoroughly revised by the authors on the basis of the referee's comments, and the committee's suggestions aimed at making the material more accessible to a broadly-based audience. Papers had to be new, and not to have been presented elsewhere, a requirement that is still unusual within the SA conference round.

A third goal of SACRS has been to invite keynote speakers, usually from overseas. This year, we are fortunate to present Dr Vinton Cerf, the father of the Internet and a world-renown expert on computer networks. Although his paper is not available for this special issue, it will appear later in SACJ. Through the good offices of Professor Chris Brink of UCT, we also have three other speakers from Germany, Canada and the US adding interest to the event, and two of their papers appear in this issue.

The programme committee originally devised a theme for the symposium - "Computing in the New South Africa". We received several queries as to the meaning of this theme, but unfortunately few papers that addressed it directly. One prospective author went as far as to enquire whether computer research would survive in the new South Africa. Another felt that his work was definitely not in the theme, as it was genuine, old world, basic, theoretical science! Neverthless, there are two papers that consider one of South Africa's key issues, that of language. Others look at the success we have achieved in applying technology to mining, and the future of low-cost operating systems. In all, the mix of papers represents a balance between the theoretical and the practical, the past and the future, all firmly based in the computing of the present.

Organising the symposium has involved the hard work of several people, and I would like to thank in particular

- Derrick Kourie, my co-organiser, and the editor of SACJ for his invaluable advice and hard work throughout the planning and implementation stages;
- Riël Smit, the production editor, for attaining such a high standard in such a short time for so many papers;
- Gerrit Prinsloo and the staff at the CSSA for their efficeint and quite delightfully unfussy organisation;
- Persetel for their very generous sponsorship of R25000, and Tim Schumann for taking a genuine interest in our events;
- the Foundation for Research Development for sponsoring Vint Cerf's visit;
- and finally the Department of Computer Science of the University of Pretoria for providing the ideal working conditions for undertaking ventures of this kind, and especially Roelf van den Heever for his unfailing encouragement and support.

Judy M Bishop
Organising Chairman, SACRS 1992
Guest Editor, SACJ Special Issue

Referees

The journal draws on a wide range of referees. The following were involved in the refereeing of the papers selected for this special issue. Their role in certifying the papers and their contribution to enhancing the quality of papers is sincerely appreciated.

John Barrow	UNISA
Ronnie Becker	University of Cape Town
Danie Behr	University of Pretoria
Sonia Berman	University of Cape Town
Liesbeth Botha	University of Pretoria
Theo Bothma	University of Pretoria
Chris Brink	University of Cape Town
Peter Clayton	Rhodes University
Ian Cloete	Stellenbosch University
Antony Cooper	CSIR
Elise Ehlers	Rand Afrikaanse University
Quintin Gee	DALSIG
Andy Gravell	University of Southampton
Wendy Hall	University of Southampton
David Harel	The Weizmann Institute of Science
Scott Hazelhurst	University of the Witwatersrand
Derrick Kourie	University of Pretoria
Willem Labuschagne	UNISA
Doug Laing	ISM
Dave Levy	University of Natal
Graham McLeod	University of Cape Town
Hans Messerschmidt	University of the Orange Free State
Deon Oosthuizen	University of Pretoria
Ben Oosthuysen	University of Pretoria
Neil Pendock	University of the Witwatersrand
D Petkov	University of Natal
Martin Rennhackkamp	Stellenbosch University
Cees Roon	University of Pretoria
Jan Roos	University of Pretoria
John Shochot	University of the Witwatersrand
Morris Sloman	Imperial College, London
Riel Smit	University of Cape Town
Pat Terry	Rhodes University
Walter van den Heever	University of Pretoria
Lynn van der Vegte	University of Pretoria
Herman Venter	University of Fort Hare
Herna Viktor	Stellenbosch University

Using Information Systems Methodology to Design an Instructional System

B C O'Donovan
Department of Accounting, University of the Witwatersrand, P O Wits, 2050

Abstract

Designers of both computerized information systems and instructional systems are faced with a need to find more effective methods of identifying and supplying the information that the users of these systems require. Both systems are comparable in their processes and development cycles. The level of structure and acceptance of information systems methodologies suggest that a transfer of selected information systems methodologies to instructional design would be of benefit.

A simplified form of entity relationship diagram is found to be particularly appropriate to the logical design of an instructional system. For illustration, an entity relationship diagram is drawn up for a simple course and once decomposed and set up in canonical form is shown to be a useful tool for setting objectives, establishing modules and the learning sequence and listing topics. These modules are sufficiently defined and their objectives sufficiently clarified to be developed independently.

The successful use of entity relationship diagrams in education emphasises the importance of using conceptual modelling techniques in I.S.

Keywords: *Modelling, entity-relationship, education*
Computing Review Categories: *H.2.1, K.3.2*

Received March 1992, Accepted April 1992

1 Introduction

When computer systems began to proliferate, Drucker [7, p. 12] referred to the problem of converting the cornucopia of data caused by advancing technology to information. Anyone responsible for the design of instructional and information systems will recognize that this problem still prevails.

Despite a growing stockpile of educational material, the education system is accused of producing an increasingly inferior product [11, p. 32]. In the information systems (I.S.) field, Shemer [18] comments that much of the maintenance cost of computer systems can be attributed to the failure to identify the real needs at the design stage. In the design of both these systems, the fundamental problem is the clear identification of information needs.

Both systems have been described as combinations of human and machine components designed to produce information/knowledge [19, p. 38] [17, p. 12]. In the computer field, the cycle of collecting, storing and processing data into information is well recognized [19]. An educational course has a cycle of inputs (course content), processes (instructional methods) and outputs (learning as measured by course objectives) [15, pp. 37–41].

I.S. developers have defined a systems development life cycle (SDLC) which proceeds from the general level of defining the problem to the detailed level of systems design [20]. Several cycles of course development have been described e.g. [15, 17]. These cycles, allowing for semantic differences, are comparable in sequence and nature of the steps to that of the SDLC.

Due to the similarity of information and instructional systems and their development cycles, there should be much to be gained from a transfer between the design methodologies of these two systems particularly as there are similar problems in identifying information requirements.

It is contended that the intensive effort expended over the years to develop I.S. design methodologies has produced some which can be usefully applied to instructional course development. This paper proposes to identify some of those methodologies and to show the benefits of their application to the design of instructional systems.

2 Why Use I.S. Methodology?

The structured approach to computer systems analysis and design has been advocated and promoted since the late seventies and is used by the majority of companies with a high level of satisfaction [12]. Properly implemented, structured SDLC methodologies improve the efficiency, communication, control, role definition, documentation and consistency within systems [20].

The use of structured methodologies in the SDLC can be contrasted with the many models (set of methods) of instructional design. Andrews and Goodson [1] not only mention a "bewildering array" of models but also comment on the inadequacy of the description of some models which precludes their use. Despite this, they point out the importance of models of instructional design and the need for a systematic approach in education.

The successful transfer of I.S. design methodologies to instructional design should assist educationists in defining and achieving teaching objectives.

3 Modelling the System

The concept of logical design is considered by some to be the most significant contribution of the structured approach [19, p. 201]. There is little emphasis on the separation of logical and physical design of instructional systems with the result that many courses are poorly structured.

Logical design uses sophisticated modelling techniques to define the requirements of the system before addressing how these will be met. The technique of modelling has been used since the seventies to enable analysts to clearly present their ideas and understanding of the situation [9]. A model is a graphic analog of the physical properties of the actual or potential system being analysed and serves as a starting point for identifying possible solutions. There are two widely used information system analysis modelling techniques, the process oriented perspective (e.g. data flow diagrams) and the data oriented perspective (e.g. entity relationship diagrams) [13].

Process models show the flow of data into, within and out of systems. These are best used when there is a regular and predictable pattern of data flows which can be represented in such a way that duplications are minimized. A data model identifies the things of importance in the system (known as entities or objects), the properties of these entities (attributes) and how they are related to one another.

The choice of use in instructional design is clarified by two aspects of education. These are the nature of data flows and the importance of relationships. In education the flows of data (i.e. the student–teacher interaction) are not regular repetitive flows as in information systems but differ according to the subject under discussion. Any attempt to model these flows is likely to result in excessive complication, absence of essential aspects or relationships insufficiently dealt with [15, pp. 134–135]. Thus process type modelling such as data flow diagrams is not suitable for modelling instructional design.

The establishment of relationships between concepts and their use in teaching is important to course design and student understanding [10]. The required model is one which looks at the concepts of a course, their attributes and the inter concept relationships. This is referred to by Donald [6] as the knowledge structure or core content of a course. Data modelling techniques in the form of entity relationship diagrams (ERD's) satisfy this requirement.

There are several widely used methods of producing ERD's. This study uses a simplified version of Barker's [3] system as it emphasises the defining of relationships and has rigorous rules designed to avoid redundancy and omissions.

The first step in producing an ERD is to define the entities, which are the significant 'things' about which information needs to be known or held. In an educational ERD, entities will be concepts, referred to by Duchastel and Steve [8] as specific chunks of knowledge or by Donald [6] as discrete entities or clusters of information. These are drawn as rectangles with rounded corners.

The second step is to identify attributes. Barker [3] defines attributes as "any detail that serves to qualify, identify, classify, quantify or express the state of an entity" (pp. 5–6). These are listed in each rectangle.

The final step is establishing the nature of the relationships and naming them. This follows the suggestion of Lawless [10] that instructional models, which set up a network of concepts, lack a further analysis of the relationships between the concepts.

An ERD can be illustrated (figure 1) by using the example of diet and nutrition course given by Rowntree [17, pp. 129–131].

There are often many concepts in a course, for example, Donald [6] identified between 33 to 170 concepts in the sixteen courses she examined. A complete ERD for a course may thus be complex. To assist in this, an overview diagram is drawn. This overview diagram is a simplified ERD derived by identifying and mapping only the significant entities, attributes and relationships which express the essence of the course. This is achieved by using summary entities (super-types) which contain further entities (sub-types). Most sub-types are not shown at the overview level [3, p. 11–3]. This overview diagram is then decomposed into sub-diagrams which reintroduce the subtypes, entities, attributes and relationships omitted at the overview level. Decomposition is a major tool of analysis and important to the understanding of the complexity of a system [5, p. 14].

4 Hierarchical Structures

Most of the hierarchical structures used in the design of systems are tree-like structures which do not sufficiently model the complexity of systems [5].

Reti [14] refers to the prerequisite relationships in the learning hierarchies of Gagné as being concerned preeminently with subordinate and superordinate relationships and not with relationships between coordinate elements. He suggests that learning should not be based on a purely hierarchical pattern as there are relationships within as well as between modules.

The solution to this requirement is to draw a set of ERD decomposition diagrams in canonical form. A canonical form of hierarchy is a combination of class structure, which focuses on intracomponent relationships, and object structure which focuses on the intercomponent relationships at each level [5, pp. 12–13].

Figure 2 has been drawn to illustrate the set of decomposed ERD's which would be drawn for the example diet and nutrition course.

The class hierarchy (shown on the left) is based on Gagné's three learning levels of defined concept, rule and higher order rule [2, p. 84], and the interrelationships are shown by the respective concepts at each level as defined by their attributes and the relationships between the concepts.

At the 'defined concept' intellectual skill level, the concepts of 'nutrient', 'effect of nutrients' and 'food group' are dealt with as entities with their sub entities and attributes.

At the level of 'rule', the entity of 'nutrient' is integrated with those of 'effects of nutrient' and 'food group' taking into account their respective relationships. After

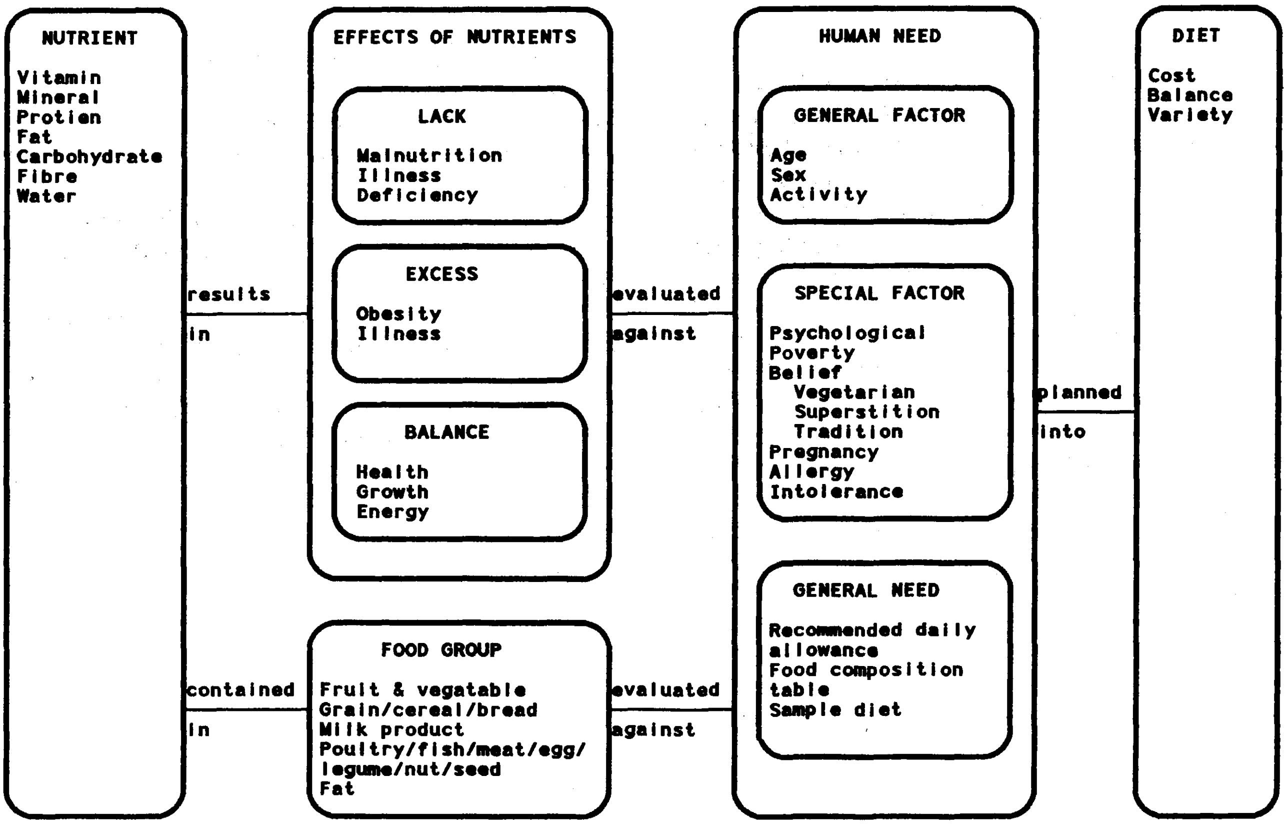


Figure 1. An entity relationship diagram for a diet and nutrition course

dealing with the entity of 'human need' and its sub entities of 'general factor', 'special factor' and 'general need' at the 'rule' level, integration takes place at the higher order rule level taking into account all the entities, attributes and relationships.

This method caters for relationships within as well as between levels and is of benefit to the overall structuring of the course particularly with the identification of modules. Benefits could also be gained by conveying the structure to both students and teachers [10, 14].

5 Developing Objectives, Modules, Course Sequence And Topics

The computer system designer's concern is the specifications given to the programmer for implementation [19]. These have two major elements: modules and the programming for each module. The more thorough and complete these are, the less clarification the programmer needs and the smaller the likelihood of problems [19, p. 653]. In instructional design, detailed specification of each module helps to avoid later problems. This applies particularly where implementation is left to less experienced teachers [16].

While IS design methodology does not assist directly in setting objectives, there is an indirect method from the ERD models. Working from the bottom of the hierarchical structure, the learning at each successive level would be: discriminating between the attributes of each entity, combining these attributes into entities as concepts and identifying and applying the relationships between the concepts, taking into account the class levels. The use of a taxonomy such as those of Gagné [2] or Bloom [4] tends to produce objectives which are more cognitive as one proceeds up the hierarchy [8]. At the same time, the nature of the relationships between entities will move from concept formation or knowledge towards the formation of higher order rules or evaluation. For example, in the diet and nutrition course the objective at the lowest level will be to describe the nutrients found in food while at the highest level it would be to discuss the formulation of diets taking into account human needs and cost, balance and variety.

A distinction is made between the process of specifying objectives for procedural knowledge and for representational knowledge [8]. If the knowledge is mainly procedural, the entities will be derived from the student's roles and the knowledge, skills and attitudes needed to fulfil those roles. If the knowledge is representational, the entities will be clusters of information identified as having a relationship with other clusters in the body of knowledge to be imparted.

The decomposed ERD's lend themselves to modularization as each super type tends to form a module. The learning sequence can also be set up using the ERD. The lower the item on the hierarchical diagram, the earlier it should be dealt with. Further, the direction of the relationship indicates which entity at the same level should be dealt with first. If there is more than one entity at each level, the set of entities has an internal relationship and must be dealt with before moving to a higher level.

For each entity, the attributes will identify the main

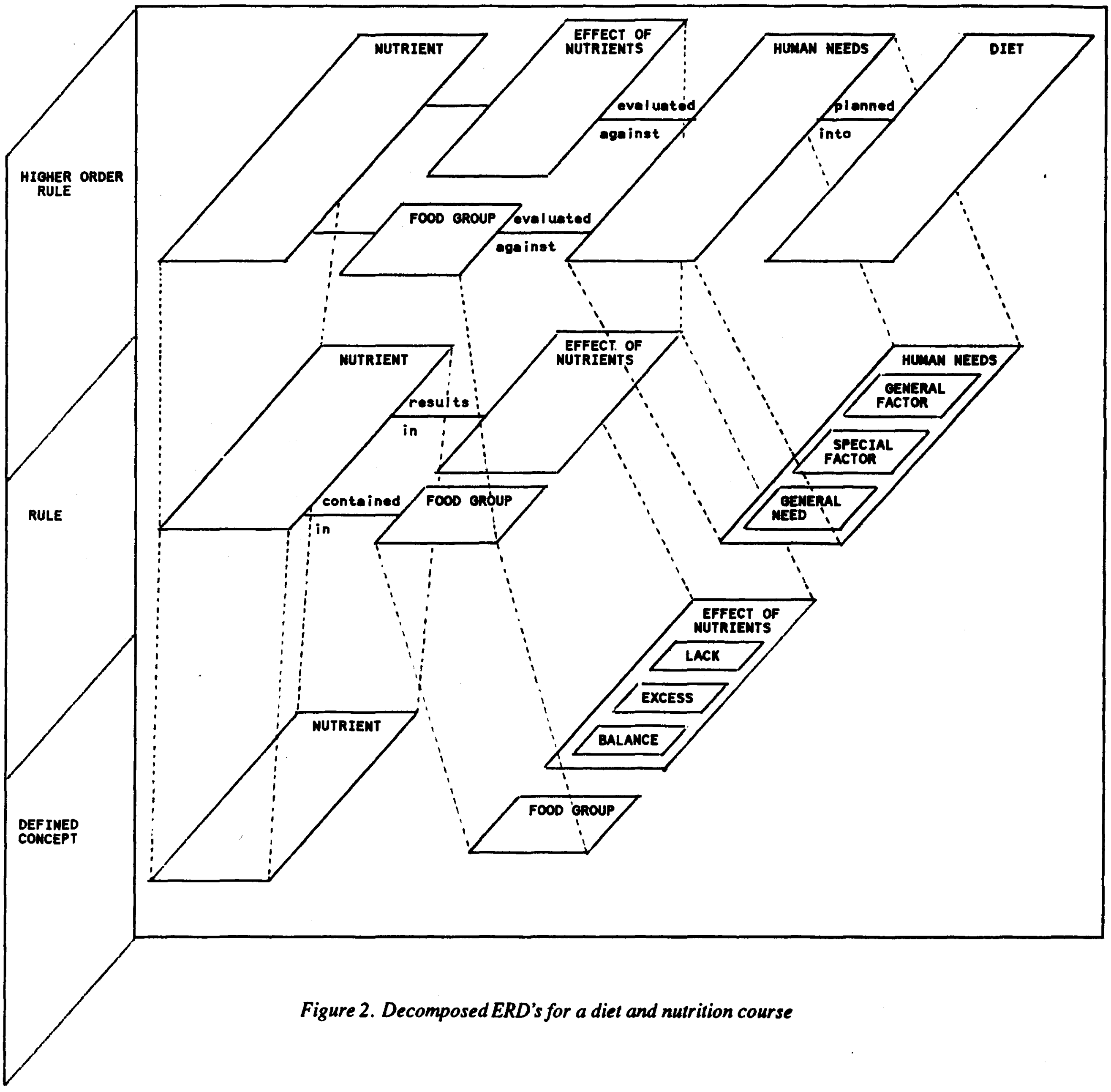


Figure 2. Decomposed ERD's for a diet and nutrition course

topics to be covered. It will usually be necessary to break down the attributes into sub headings and even to draw further ERD's for modules. This facilitates the physical design and implementation of conventional or computer based education courses by providing a clear and understandable model of the structure and sequence.

6 Conclusion

Designers of both information and instructional systems are faced with the problem of finding more effective methods for identifying information needs. As both systems are comparable in process and development, it was contended that transfer between the respective development methodologies is possible.

The use of logical design to break away from existing systems requires a modelling technique and a simplified form of the ERD was found to be appropriate to the design of instructional systems. The problem of hierarchical patterns not taking account of inter and intra module relationships was solved by setting up decomposition diagrams in canonical form.

The well accepted method of ERD modelling in information systems design caters effectively for many elements of instructional design. Reti [14] sought a paradigm for stating objectives "that provides for a progressive overview of the interrelationships between coordinate elements as they develop" [p. 33]. The decomposition of an ERD solves this by deriving objectives, learning sequence and topics at each learning level and identifying modules which can be developed independently.

The successful use of ERDs in education highlights the value of this methodology in the I.S. field. This easily learnt methodology is valuable for modelling concepts and organizing knowledge. This is particularly relevant to modelling database systems and to representing knowledge in artificial intelligence.

References

1. D H Andrews and L A Goodson. 'A comparative analysis of models of instructional design'. *Journal of Instructional Development*, 3(4):2–16, (Summer 1980).
2. D T Aronson and L J Briggs. 'Contributions of Gagné and Briggs to a prescriptive model of instruction'. In C M Reigeluth, ed., *Instructional-design Theories and Models: An Overview of their Current Status*. Lawrence Erlbaum Associates, London, (1983).
3. R Barker. *Case*Method Entity Relationship Modelling*. Addison-Wesley, Wokingham, 1990.
4. B S Bloom, ed. *Taxonomy of Educational Objectives: Cognitive Domain*. David McKay, New York, 1956.
5. G Booch. *Object Oriented Design: With Applications*. Benjamin/Cummings, California, 1991.
6. J G Donald. 'Knowledge structures: Methods for exploring course content'. *Journal of Higher Education*, **54**(1):31–41, (1983).
7. P F Drucker. *Technology, Management and Society*. Pan Books, London, 1972 edition, 1958.
8. P Duchastel and M Steve. 'Instructional goals: Determination and specification'. *British Journal of Educational Technology*, **2**(14):137–142, (May 1983).
9. C Gane and T Sarson. *Structured Systems Analysis: Tools and Techniques*. Prentice-Hall, Englewood Cliffs, New Jersey, 1979.
10. C J Lawless. 'Information processing: A model for educational technology'. In *Aspects of Educational Technology XIII*, pp. 333–342. Nichols Publishing Co., New York, (1979).
11. J Naisbitt. *Megatrends*. London, Futura (1984) edition, 1982.
12. C R Necco, C L Gordon, and N W Tsai. 'Systems analysis and design: Current practices'. *MIS Quarterly*, pp. 461–475, (December 1987).
13. T W Olle, J Hagelstein, I G Macdonald, C Rolland, H G Sol, F J M van Assche, and A A Verrijn-Stuart. *Information Systems Methodologies*. Addison-Wesley, Wokingham, England, 1988.
14. P G Reti. 'Integrative structural objectives'. *British Journal of Educational Technology*, **1**(9):27–35, (January 1978).
15. A J Romiszowski. *Designing Instructional Systems, Decision Making in Course Planning and Curriculum Design*. Kogan Page, London, 1981.
16. G D Roopchand and D Moss. 'A systematic approach to the design of secondary school lessons in Guyana'. *British Journal of Educational Technology*, **19**(1), (1988).
17. D Rowntree. *Educational Technology in Curriculum Development*. Harper and Row, London, 2nd edition, 1982.
18. I Shemer. 'Systems analysis: A systematic analysis of a conceptual model'. *Communications of the ACM*, **30**(6):506–512, (June 1987).
19. J L Whitten, L D Bentley, and V M Barlow. *Systems Analysis and Design Methods*. Irwin, Boston, 2nd edition, 1989.
20. R K Wysocki and J Young. *Information Systems: Management Principles in Action*. John Wiley and Sons, New York, 1990.

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as 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 wide margins and $1\frac{1}{2}$ or double spacing.
- 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 numbered and titled. Figures should be submitted as original line drawings/printouts, and not photocopies.
- 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 [1, 2, 3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) LaTeX file(s), either on a diskette, or via e-mail/ftp – a LaTeX style file is available from the production editor;
2. In camera-ready format – a detailed page specification is available from the production editor;
3. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

   Further instructions on how to reduce page charges can be obtained from the production editor.
4. In a typed form, suitable for scanning.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R20-00 per final page for LaTeX or camera-ready contributions and the maximum is R100-00 per page for contributions in typed format.

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

Proofs

Proofs of accepted papers in categories 3 and 4 above 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 less than 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.

References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, 9:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages.* McGraw Hill, New York, 1966.

South African Computer Journal

Number 7, July 1992
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 7, Julie 1992
ISSN 1015-7999

Contents

Forward i
Referees ii

Machine Translation from African Languages to English
DG Kourie, WJ vd Heever and GD Oosthuizen 1

Automatically Linking Words and Concepts in an Afrikaans Dictionary
PZ Theron and I Cloete 9

A Lattice-Theoretic Model for Relational Database Security
A Melton and S Shenoi 15

Network Partitions in Distributed Databases
HL Viktor and MH Rennhackkamp 22

A Model for Object-Oriented Databases
MM Brand and PT Wood 27

Quantifier Elimination in Second Order Predicate Logic
D Gabbay and HJ Ohlbach 35

Animating Neural Network Training
E van der Poel and I Cloete 44

HiLOG - a Higher Order Logic Programming Language
RA Paterson-Jones and PT Wood 53

ESML - A Validation Language for Concurrent Systems
PJA de Villiers and WC Visser 59

Semantic Constructs for a Persistent Programming Language
SB Sparg and S Berman 65

The Multiserver Station with Dynamic Concurrency Constraints
CF Kriel and AE Krzesinski 75

Mechanizing Execution Sequence Semantics in HOL
G Tredoux 81

Statenets - an Alternative Modelling Mechanism for Performance Analysis
L Lewis 87

From Batch to Distributed Image Processing: Remote Sensing for Mineral Exploration 1972-1992
N Pendock 95

Galileo: Experimenting with Graphical User Interfaces
R Apteker and JM Bishop 99

Placing Processes in a Transputer-based Linda Programming Environment
PG Clayton, FK de-Heer-Menlah, EP Wentworth 109

Accessing Subroutine Libraries on a Network
PH Greenwood and PH Nash 117

A Multi-Tasking Operating System Above MS-DOS
R Foss, GM Rehmet and RC Watkins 122

Using Information Systems Methodology to Design an Instructional System
BC O'Donovan 126

Managing Methods Creatively
G McLeod 131

A General Building Block for Distributed System Management
P Putter and JD Roos 141