

**South African
Computer
Journal
Number 20
December 1997**

**Suid-Afrikaanse
Rekenaar-
tydskrif
Nommer 20
Desember 1997**

**Computer Science
and
Information Systems**

**Rekenaarwetenskap
en
Inligtingstelsels**

**The South African
Computer Journal**

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

**Die Suid-Afrikaanse
Rekenaartydskrif**

*'n Amptelike publikasie van die Rekenaarvereniging
van Suid-Afrika en die Suid-Afrikaanse Instituut
vir Rekenaarwetenskaplikes*

Editor

Professor Derrick G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
dkourie@dos-1an.cs.up.ac.za

Subeditor: Information Systems

Prof Lucas Introna
Department of Informatics
University of Pretoria
Hatfield 0083
lintrona@econ.up.ac.za

Production Editor

Dr Riël Smit
Mosaic Software (Pty) Ltd
P.O.Box 23906
Claremont 7735
gds@mosaic.co.za

World-Wide Web: <http://www.mosaic.co.za/sacj/>

Editorial Board

Professor Judy M Bishop
University of Pretoria, South Africa
jbishop@cs.up.ac.za

Professor R Nigel Horspool
University of Victoria, Canada
nigelh@csr.csc.uvic.ca

Professor Richard J Boland
Case Western Reserve University, USA
boland@spider.cwrw.edu

Professor Fred H Lochovsky
University of Science and Technology, Hong Kong
fred@cs.ust.hk

Professor Ian Cloete
University of Stellenbosch, South Africa
ian@cs.sun.ac.za

Professor Kalle Lyytinen
University of Jyvaskyla, Finland
kalle@cs.jyu.fi

Professor Trevor D Crossman
University of Natal, South Africa
crossman@bis.und.ac.za

Doctor Jonathan Miller
University of Cape Town, South Africa
jmiller@gsb2.uct.ac.za

Professor Donald D Cowan
University of Waterloo, Canada
dcowan@csg.uwaterloo.ca

Professor Mary L Soffa
University of Pittsburgh, USA
soffa@cs.pitt.edu

Professor Jürg Gutknecht
ETH, Zürich, Switzerland
gutknecht@inf.ethz.ch

Professor Basie H von Solms
Rand Afrikaanse Universiteit, South Africa
basie@rkw.rau.ac.za

Subscriptions

	Annual	Single copy
Southern Africa:	R50,00	R25,00
Elsewhere:	\$30,00	\$15,00

An additional \$15 per year is charged for airmail outside Southern Africa

to be sent to:

*Computer Society of South Africa
Box 1714 Halfway House 1685*

Editorial Notes

For two reasons, this edition of SACJ is far later than it ought to have been. The first reason is that there have been some personnel changes in the editorial team. Lucas Introna, having continued for some time as IS editor after transferring to London, asked to be relieved of his duties. Niek du Plooy has kindly agreed to fulfill this role in a temporary capacity until a suitable replacement for Lucas can be found. Due to work pressure, Riël Smit has also withdrawn as production editor, and has been replaced by John Botha. SACJ owes the two retired members a huge debt of gratitude. During his period of tenure, Lucas did sterling work in setting and maintaining a solid standard for IS contributions. Riël put SACJ on a \LaTeX path, and has laboured diligently to produce an aesthetically pleasing product. Thanks are also due to Niek and John for their willingness to take over in their respective roles. Until further notice, IS contributors may forward their submissions directly to Niek at his address given on the front inside cover. I shall put successful authors in touch with John for further instructions regarding final preparation of their manuscripts.

The second reason for a delay in this edition has to do with authors who have not scrupulously followed guidelines for producing their final submissions. There have been a variety of problems ranging from missing citations and inappropriate production of figures to incompatible electronic file submissions. All of this, coupled with our new production editor (who—despite an extremely busy schedule—has valiantly climbed a steep \LaTeX learning curve) has resulted in an edition that should have been out to press several weeks earlier.

The editorial team will be giving attention to the general matter of format and submission procedures in future. SACJ's citation and reference methods are somewhat archaic and will probably be revised. All the necessary information will be provided on the new SACJ web site at www.cs.up.ac.za/sacj/. The site will also contain abstracts of articles in this and future editions.

These are times of conflicting stresses on both the academic and industrial IT communities. They are being felt somewhat more acutely in Southern Africa (and presumably in other developing countries) than in the developed world. Internationally there is tendency to cut back on state financing of universities and a seemingly insatiable demand for IT graduates. Many companies snap up new graduates at attractive salaries, positively discouraging full-time postgraduate studies. International recruitment agencies scour the South African scene for qualified candidates, luring some of our most promising young professionals out of the country. Job-hopping, a drift from academia to industry and from local industry to USA or

European industry seems to be the order of the day. Despite the availability of private colleges and institutes, virtual or otherwise, there is a rush of students to university and technikon IT departments, all hoping to get at the IT honey-pot. University administrations are struggling to correct the structural deficiencies of the past and to provide IT departments with sufficient resources to cope with demand. As editor of SACJ, I have no particular competence authoritatively to sum up or analyze these tendencies, but it does seem to me desirable that someone ought to do so. Bodies such as SAICSIT, the CSSA, university authorities, IT industry and state representatives ought actively to pursue joint strategies to ensure that our IT departments are properly resourced and that (non-Zuma) measures are taken to retain graduates in the country. It seems almost redundant to attempt to spell out the consequences of inactivity.

Derrick Kourie
EDITOR

Theory Meets Practice

Using Smith's Normalization in Complex Systems

AJ van der Merwe

WA Labuschagne *

* Department of Computer Science and Information Systems, University of South Africa, Pretoria,
vdmeraj@alpha.unisa.ac.za

Abstract

In 1985 Smith introduced a new way of deriving fully normalized tables. His method was based on the use of a dependency list and a dependency diagram to produce fifth normal form directly. We introduce the use of end-line, begin-line and in-line bubbles to simplify the diagram for large-scale applications.

Keywords: Normalization, database design, file design.

Computing Review Categories: H.2.1

Received: 28/8/1996, Accepted: 4/4/1997, Final version: 4/4/1997.

1 Introduction

Most people follow a procedural methodology when designing schemas for relational data base management systems (RDBMS). During the design phase an entity relationship (ER) model is used to represent the entities and relationships. The designer then transposes the entities' associated data fields into a set of candidate tables and, finally, all candidate tables are transformed step-by-step to fifth normal form by applying normalization guidelines [1]. The apparent simplicity of this ER-model is deceptive inasmuch as the model uses the semantics of entities and not the fields. The designer of the system must have a complete understanding of the normal-form guidelines and any real-world information applicable to the system when transposing the ER-model to candidate tables [2].

In 1985 Henry C Smith introduced a method to compose fully normalized tables from a rigorous dependency diagram. This method is based on field-related semantics. Because the semantic meanings of its dependency chains are specifically tailored to normal-form guidelines, the model directly and immediately synthesizes into fully normalized tables with completely specified primary keys (PK), foreign keys (FK), and join paths [4].

Since the dependency diagram plays a key role in Smith's method, it is essential that the diagram should be readable. Readability tends, of course, to diminish in proportion to the size and complexity of the application. In this paper we adapt Smith's method for application to complex large-scale systems.

2 Smith's diagrammatic conventions

The dependency diagram is developed from a dependency list in accordance with the following rules [3]:

- Field names occur once only.

- A single valued dependency (SVD) is diagrammed by drawing a single bubble around the originator field (the determinant) and using a single-headed arrow to the field dependent on it (called a singleton). See figure 1.
- Multi-valued dependencies (MVD) arise when a field A determines many values of field B. An MVD is diagrammed by drawing bubbles around field A and B, and using a double-headed arrow from determinant bubble A to end-multiple bubble B (Figure 2).
- Independent MVDs may exist between A and B and A and C (Figure 3).
- MVDs from A to B and from B to C form an MVD chain and are modelled as in Figure 4. A is the uplink-determinant bubble, B the determinant bubble and C the end-multiple bubble.
- When a field is a determinant of conflicting dependencies, a double bubble is used (Figure 5).
- $[n]$ at the bottom of fields indicates that these fields share the same domain values ($n = 1 \dots$).
- $\langle m \rangle$ at the bottom of a singleton field means that more singleton fields can be found at number m in the dependency list.

3 Smith's algorithm to create 5NF relations

1. Consider all the singleton chains. (A singleton chain is a series of interlinked bubbles ending in one or more singleton bubbles, i.e. bubbles with single-headed bubbles and no arrows pointing from them.) All the fields within all singleton bubbles linked to the same determinant bubble are the non-key attributes of the table. The field within that determinant bubble plus field(s) within all uplink-determinant bubbles form the table's primary key. Mark the line between the last determinant and all the singletons linked to it, as worked (\surd).

2. Consider all end-multiple chains. (These are chains ending in end-multiple bubbles, i.e. bubbles with double-headed arrows pointing to, and no arrows pointing from, them.) Work upwards from the end-multiple bubble: the fields within the end-multiple bubble, the determinant bubble and all uplink-determinant bubbles form the table's primary key. Mark the line between the last determinant and the end-multiple bubble linked to it, as worked (✓).

Example: Dependency List

1. For each student a Student Number (Student#), Name, Address and Telephone must be stored.
2. For each Lecturer a number, Name and Office Number (Office#) must be stored.
3. Each Subject is identified by an Subject_Code and described by a name (Sub_Name).
4. Each Lecturer can have more than one student and each student can take more than one subject.

5NF relations

Student(Student#, Name, Address, Telephone)

Lecturer(Lecturer#, Name, Office#)

Subject(Subject_Code, Sub_Name)

Class(Student#,Lecturer#, Subject_Code)

4 Large-scale applications

When one applies Smith's method on big complicated systems, the readability of the dependency diagram can be adversely affected in two principal ways, namely by lines that cross one another and by a proliferation of double-bubbles. We will use the MetAIS (Meteorological Air Information System) system developed by Netsys International to illustrate the problems.

The MetAIS system was developed for a customer that provides flight briefing information to aviators. Information is gathered from satellite and terrestrial links and is then processed and presented in an easily accessible format to the users. A typical use would be for a pilot to ask what the flight information will be from point A to point B. The system must then gather all the information applicable to the route and draw the pilot's attention to operational as well as meteorological problem areas.

5 Representing multiple dependencies

The dependency list for the MetAIS system consisted of about 70 different dependencies of importance. To represent all the different dependencies explicitly in one diagram, and still keep it readable, was impossible. We used the rule: *< m > at the bottom of a singleton field means that more singleton fields can be found at number m in the dependency list to simplify the diagram.* Although this helped, the readability of the diagram remained affected by the crossings of lines from one side of the diagram to another. Figure 7 illustrates the manner in which lines may be forced to cross one another.

An end-line bubble (ELB) and begin-line-bubble (BLB) were created to eliminate the crossing of lines. Suppose the following is the k'th dependency in some dependency list related to Figure 7:

- k. For Case A every value of field C will uniquely determine field D.

Our problem is that field C is positioned at the bottom lefthand corner and field D at the top right-hand corner of the diagram. Linking the two requires crossing various other lines. By making use of an end-line bubble (Figure 8) from C to link it to its begin-line bubble counterpart (Figure 9), we simplify the diagram (Figure 10). The number written in each ELB and BLB is just the relevant dependency list number.

6 Double-bubble representation

Double-bubbles are used to represent conflicting dependencies (Figure 5). The following is an extract from the MetAIS dependency list.

29. Each NSWC has a TTAAii with more than one YYGGgg. For each YYGGgg more than one CCCC and BBB exist and the combination of the four has one ValidTo and one set of Data.
30. Each SWC has a TTAAii with more than one YYGGgg. For each YYGGgg more than one CCCC and BBB exist and the combination of the four has one ValidTo and one set of Data.
36. A low level forecast has a combination of TTAAii, CCCC, BBB, YYGGgg that is unique and identifies a single ValidTo and Data field.
44. For each WC a TTAAii with more than one YYGGgg is stored. For each YYGGgg more than one CCC and BBB exist and the combination of the four has one Flight_Level, ValidTo and one set of Data.

The diagram for the four dependencies in the dependency list is given by Figure 11.

The fields TTAAii, YYGGgg, CCCC, BBB, ValidTo and Data are repeated for four different conflicting dependencies: a repetitive dependency of degree 4. In general the double-bubble representation of repetitive dependencies of degree greater than 2 impedes the comprehension of the diagram. We propose the insertion of in-link bubbles as an alternative way to represent repetitive dependencies.

Consider the example in figure 12 with the following dependency list:

1. For Case 1 S# can have more than one D# and for D#,S# there exists only one O#.
2. For Case 2 S# can have more than one D# and for D#,S# there exists only one O#.
3. For Case 3 S# can have more than one D# and for D#,S# there exists only one O#.
4. For Case 4 S# can have more than one D#.

An in-link bubble (ILB) is created between two fields to show the degree of the link. The value 3 in the in-link bubble between D# and O# means that, irrespective of any links to D# or O# and irrespective of any links that exist from D# and O#, this specific link must be used in exactly

three iterations of the normalizing algorithm. Each iteration reduces the values inside the in-link bubbles by one. After the first iteration on the example in figure 12, we are left with the diagram in figure 13 and the relation CASE1 (S#, D#, O#) is derived.

After the second and third iteration, the following relations are derived: CASE2 (S#, D#, O#) and CASE3 (S#, D#, O#). We are left with the diagram in figure 14.

Note that the link between D# and O# is marked off as worked (\checkmark). A link is marked 'worked' when it was used in an iteration and no longer needs to be taken into account in the normalization process. Only the repetitive dependency between S# and D# (of degree 1) is of any further importance and the relation CASE4(S#, O#) is derived according to the rules.

To make provision for in-link bubbles a small modification must be made to the algorithm deriving the normalized tables.

1. Consider all the singleton chains. All the fields within all singleton bubbles linked to the same determinant bubble are the non-key attributes of the table. The field within that determinant bubble plus fields within all uplink-determinant bubbles form the table's primary key. If an in-link bubble exists between any two fields, reduce the degree by one in the in-link bubble. In case of a degree of 0, mark the line as worked (\checkmark).
2. For an end-multiple chain, work upwards from the end-multiple bubble: the fields within the end-multiple bubble, the determinant bubble and all fields in the uplink-determinant bubbles form the primary key. If an in-link bubble exists between any two fields, reduce the number by one in the in-link bubble. In case of a degree 0, mark the line as worked (\checkmark).

It should be borne in mind that any link to a singleton or end multiple without an in-link bubble is taken to be of degree 1, i.e. is used in just one iteration. The simplified diagram for the MetAIS system can be seen in figure 15.

The relations derived for the system are the following:

NSWC(TTAAii,YYGGgg,CCCC,BBB,ValidTo,Data)
 SWC(TTAAii,YYGGgg,CCCC,BBB,ValidTo,Data)
 LLF(TTAAii,YYGGgg,CCCC,BBB,ValidTo,Data)
 WC(TTAAii,YYGGgg,CCCC,BBB,ValidTo,Data,Flight_Level)

7 Conclusion

Many systems are designed over a long period, and new programmers may be added to the team at a late phase of the development. It can be difficult for these programmers to understand the data processes and how the data interlink. The use of diagrams for the system designer may sometimes entail tedious effort, but as a documentation tool for the user and programmers added to teams the advantages are worth the effort, provided only that the diagrams remain readable. By employing end-line and begin-line bubbles the crossing of lines can be eliminated, and by inserting in-line bubbles the representation of repetitive dependencies can be simplified, thus improving the readability

of the diagrams involved in large-scale applications.

References

1. E Codd. 'A relational model of data for large shared databanks'. *Communications of the ACM*, 13(6), (June 1970).
2. C Date. *An Introduction to Database Systems - Volume I*. Addison Wesley, Reading Mass, 4th edition, 1986.
3. H Smith. 'Database design: Composing fully normalised tables from a rigorous dependency diagram'. *Communications of the ACM*, 28(8):826-838, (August 1985).
4. H Smith. 'Dependency modelling for relational database design'. *Database Programming and Design*, pp. 39-49, (September 1990).

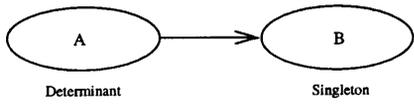


Figure 1. Single valued dependency

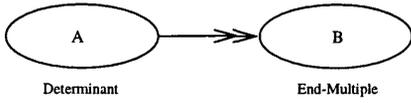


Figure 2. Multi-valued dependency

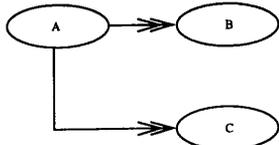


Figure 3. Independent MVDs

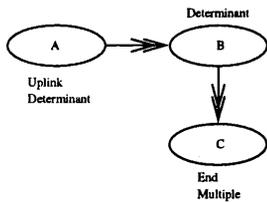


Figure 4. An interdependent MVD

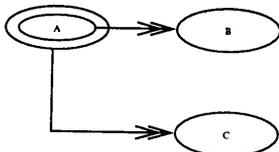


Figure 5. Conflicting dependencies

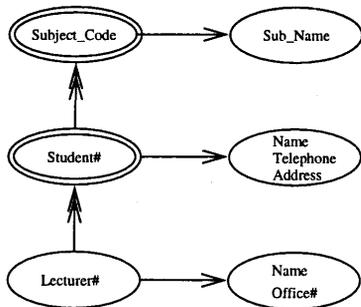


Figure 6. Dependency diagram depicting a student system

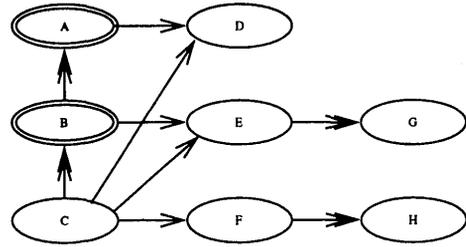


Figure 7. Lines crossing to represent different relationships

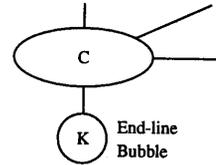


Figure 8. End-line bubble

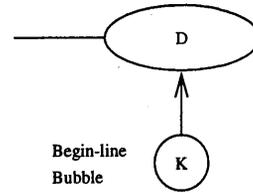


Figure 9. Begin-line Bubble

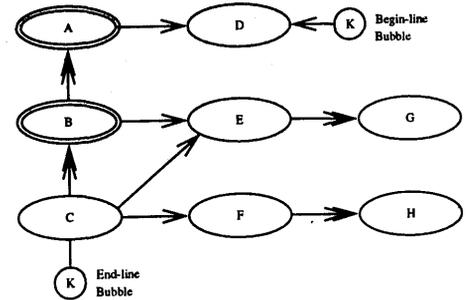


Figure 10. ELB and BLB dependencies

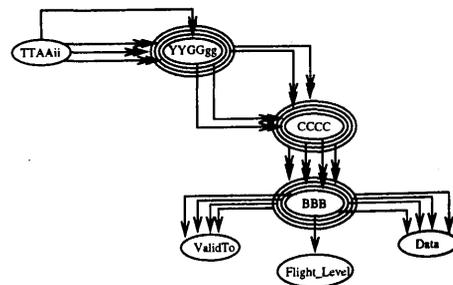


Figure 11. Double-bubbles arising from the MetAIS system

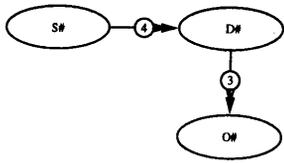


Figure 12. Conflicting dependencies

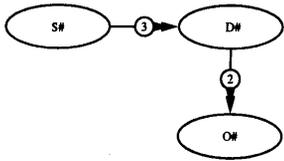


Figure 13. Diagram after 1 iteration

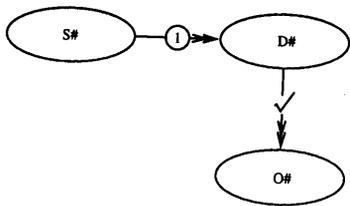


Figure 14. Example after third iteration

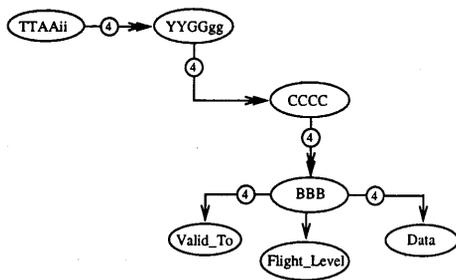


Figure 15. MetAIS system simplified using ILBs

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 notes. 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 1½ 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.
- 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 [1–3]. 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. As an ASCII file accompanied by a hard-copy showing formatting intentions:
 - Tables and figures should be original line drawings/printouts, (not photocopies) 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.

Contact the production editor for markup instructions.

3. In exceptional cases camera-ready format may be accepted – a detailed page specification is available from the production editor;

Authors of accepted papers will be required to sign a copy-right transfer form.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for \LaTeX or camera-ready contributions that require no further attention. The maximum is R120-00 per page (charges include VAT).

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

Proofs

Proofs of accepted papers in category 2 above may 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. Camera-ready submissions will only be accepted if they are in strict accordance with the detailed guidelines.

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.

Contents

Editorial	
DG Kourie	1
Research Contributions	
The Abstraction-First Approach to Encouraging Reuse	
P Machanick	2
Secure Mobile Nodes in Federated Databases	
MS Olivier	11
Word Prediction Strategies in Program Editing Environments	
I Sanders and C Tsai	18
A Computerised-consultation Service for the Computerisation of the Very Small Small-business Enterprise	
CW Rensleigh and MS Olivier	25
Some Typical Phases of a Business Transformation Project: The First Steps Toward a Methodology?	
D Remenyi	36
<hr/>	
Technical Reports	
Theory Meets Practice: Using Smith's Normalization in Complex Systems	
AJ van der Merwe and WA Labuschagne	44
Applying Software Engineering Methods to Instructional Systems Development	
P Kotzé and R de Villiers	49
<hr/>	
Communications and Viewpoints	
Mobile Agents at ISADS 97	
I Vosloo	A57
The Recovery Problem in Multidatabase Systems—Characteristics and Solutions	
K Renaud and Paula Kotzé	A62
