Quaestiones Informaticae

Vol. 3 No. 1 February, 1984

Quaestiones Informaticae

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

'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Editor: Prof G. Wiechers
Department of Computer Science and Information Systems
University of South Africa
P.O. Box 392, Pretoria 0001

Subscriptions

Annual subscriptions are as follows:

	SA	US	UK
Individuals	R6	$7	£3,0
Institutions	R12	$14	£6,0

Circulation manager
Mr E Anderssen
Department of Computer Science
Rand Afrikaanse Universiteit
P O Box 524
Johannesburg 2000
Tel.: (011) 726-5000

Quaestiones Informaticae is prepared for publication by Thomson Publications South Africa (Pty) Ltd for the Computer Society of South Africa and the South African Institute of Computer Scientists.

Documenting a Database System with a Database

D A Hunter
NRIO/CSIR Stellenbosch

Abstract
The South African Data Centre for Oceanography (SADCO) has implemented a multi disciplinary oceanographic database using DMS 170, a relational database management system Using a high level query language scientists from all over the country are able to access the database and browse the data freely In addition SADCO has other more specialized databases and also datasets as independent files, and in various other non computerized forms A vast number of programs and procedures to load and modify the database and produce plots or tables from the database have been developed For this reason a 'data dictionary system (the Data Catalogue) has been developed as a 'micro' database, using the Query Update language for the purpose of documenting the contents of the database, the resources of the data centre and the events that take place daily to modify resources The user can access this Data Catalogue using the same query language that is used to access the main database

Introduction

This paper is not intended to describe any new research on the subject of data dictionaries or databases but rather to show how in a given situation a large scientific database has been con structed with the essential aid of a documentation system im plemented successfully as a practical solution to overcome the limitations in software availability

The South African Data Centre for Oceanography (SADCO) performs the functions of archiving, exchange and dissemina tion of oceanographic data and information and to do this it has set up a large multidisciplinary oceanographic database and in addition a few smaller, more special purpose databases are being set up

Physical oceanographers, marine biologists, marine chemists and others throughout the country have contributed data to the centre, so that users can have access to a wide range of data, in sufficient quantity, to permit meaningful analyses Remote users may browse the database using a query language and also execute various specialised procedures to produce tables or plots The database is a large project, requiring of the order of two hundred and fifty data items, a large volume of data (over half a million relational records), a complex structure involving twen ty five files and a large taxonomic dictionary containing the full classification of all animals whose data is stored in the database An intense loading schedule is maintained to load new data, and verify data loaded New retrieval processes and various useful products are developed for use by the data centre and users of the database

The database, and all activities revolving around the storage, retrieval, modification and acquisition of data require a thorough documentation system so that the entire database does not stand or fall on the internal knowledge of one man, no longer with us, or the memories of others

Scientific data has the quality that it is mostly an approx imation to the real world and how good that approximation is, is difficult to ascertain at times It also has the quality that even when large volumes of data exist, scientists may not be very sure of the degree of usefulness of the data

The answer seems to be to give scientists as much data as may be of use to them verify the accuracy as far as possible and to allow them free access to the database so that they can try out ideas themselves with relatively little help and find patterns which would have been hardly possible without a large in tegrated on line store

The Data Centre

In addition to more than one database the data centre has also a number of independent data files which are either compatible with the current databases or have not yet been loaded into the database for various reasons These files are often the only means of answering certain requests for data, and must be kept on tape when not in use These altogether with the databases must be carefully documented and kept track of All data received, whether on cards, magnetic tape, satellite photographs or even analogue records from an X,Y recorder must be accessible by the data centre or users

Thus the data centre as a whole needs a comprehensive centralized system of documentation at a high level, in order that the manager may be able to keep track of what is available and where it may be found

The Database

The database has been implemented on the Cyber computer at the CSIR computer centre based in Pretoria, using DMS170, a relational inverted database management system Data is loaded into the database using either FORTRAN or COBOL database programs and queries are made almost solely by us ing the query language Query Update (QU), a very high-level language

To implement a system such as this, many relations are defin ed and for each discipline there are FORTRAN, COBOL and Query Update subschemas allowing interface to the database Also required for the system to be functional are

(a) A means of checking that data corresponding to the child (or measured data) does not get stored before the parent data (or identification data) is stored — ie a set of constraints
(b) A means of ensuring that no data is loaded twice (data sets from two different contributors may have differing station numbers but be otherwise identical)
(c) A means of restricting scientists from having access to non released or restricted data belonging to someone else — security locks and procedures

Database Structure

Searching along a relation in DMS170 is best achieved when one of the search keys is a primary or alternate key of the parent file of a relation In this database most searches will encompass a relation with parent file the COMMON INFORMATION

file (COMINFO), since this file contains identification information such as date, position, station number The following disciplines are catered for PHYSICAL (including WEATHER), CHEMISTRY (including heavy metals for pollution studies) SEDIMENT TAXONOMY, SEDIMENT CHEMISTRY, TISSUE POLLUTION, PLANKTON TAXONOMY and PLANKTON CHEMISTRY Each discipline is usually associated with more than one file and there exists at least one relation for each discipline, the relation including the COM INFO file as parent and the files for that discipline There is always a COMINFO record corresponding to each record of data in any of the other data files and most often a COMINFO record points to records of many different disciplines (Figure 1)

The Biological Subsystem

There is also associated with the biological data (TISSUE, SEDIMENT and PLANKTON) a taxonomic dictionary file con taining the full taxonomic names of all animals of interest like ly to be found in the area of analysis This dictionary ensures that data is spelt correctly on input and also allows a code to be stored in the data file rather than the 120 character name

The taxonomic dictionary is itself fairly complicated and has a tree structure encompassing six levels, these being PHYLUM, CLASS, ORDER, FAMILY, GENUS and SPECIES Each animal is described by a set of six names, one for each level of classification and each of which is totally unique excepting for SPECIES Thus all data items except SPECIES and PHYLUM (because there are so few) are alternate keys in the dictionary enabling quick location of all animals of the same CLASS, ORDER, FAMILY or GENUS

In addition an alternate key has been defined called LOWID (LOWEST IDENTIFICATION) indicating the lowest level of classification Sometimes it may have been possible to classify an animal only down to FAMILY level (GENUS and SPECIES could not be identified), in this case the FAMILY would be LOWID On a display of what creatures were found the variable LOWID is usually used since it is much shorter than displaying the entire taxonomic name and gives the essential information about the animal concerned (Figure 2)

The taxonomic dictionary required a complex procedure to set it up and various QU reports were produced to give listings of the file in pure taxonomic order (ie pre order traversal of the tree) and also in alphabetical order of LOWID Documen tation informing the DBA of the existence of such procedures and reports and associated resources is essential

Taxonomic searching — relations

The database makes use of the fast search on the taxonomic dictionary by including a number of backward relations with TAXDIC as the parent file, the second file being either the Sedi ment, Plankton or Tissue taxonomic files, and then the last file being COMINFO file (See Figure 3)

This enables the user whose request centres on particular groups of animals to use, say, the FAMILY as a search key and restrict the search to all animals of that family A biologist may conceivably use the database to find the northernmost limits where any animals of a particular family have been detected

Thus it can be seen why so many relations exist in the database as such, and the necessity of users being able to find their way around the database along these relations

User Access of the Database

The database has been designed with a view to a number of remote users having read only access to a portion of the database This has been accomplished by giving each user his own set of Query Update subschemas, all security checks be ing controlled from the subschemas

Facilities for Users

It is necessary that each user has a rudimentary knowledge of Query Update if he wishes to browse the database but some users will simply access sessions or programs created by the data centre and they will need to know how to run these For the former type of user a detailed description of what is in the database (and what subschemas and relations are available) is

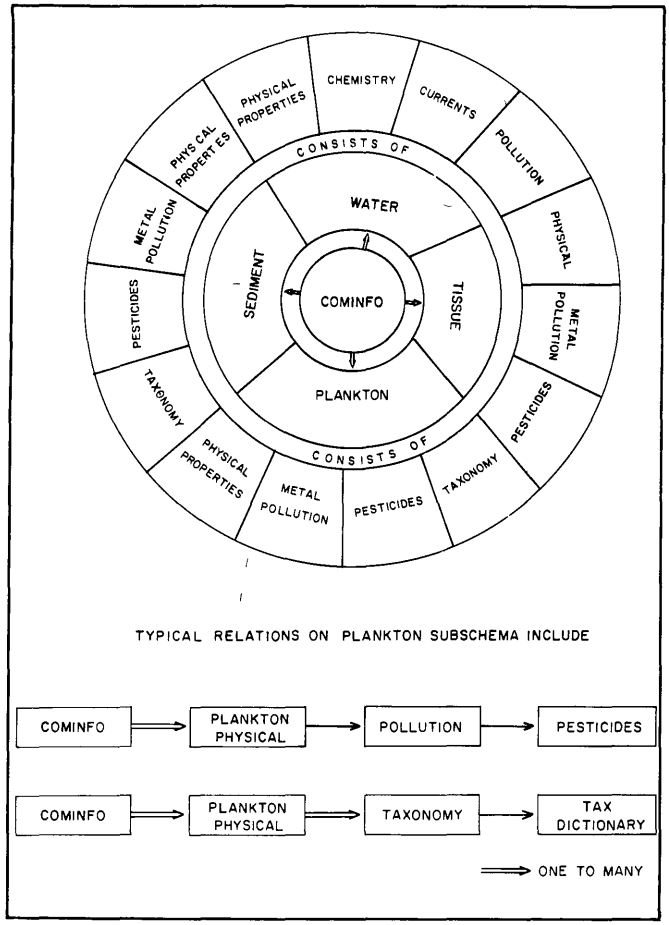

FIGURE 1 SADCO oceanographic database structure

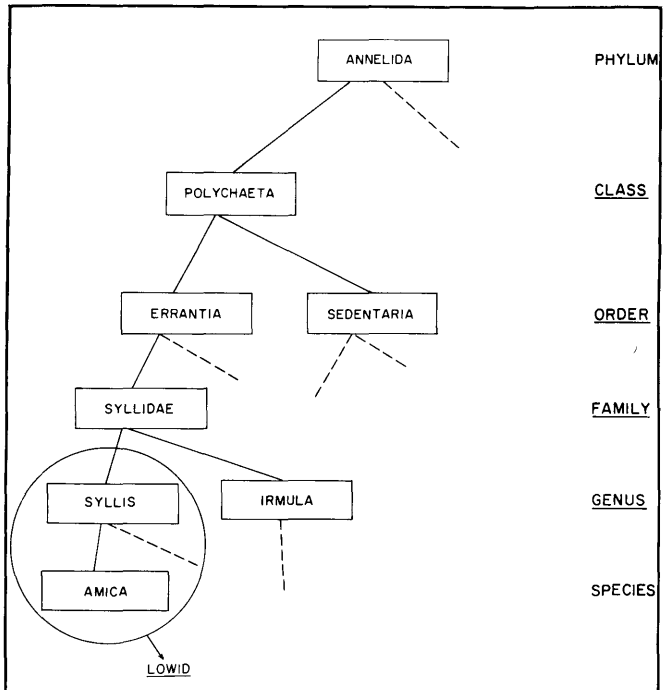

FIGURE 2 The taxonomic dictionary structure

For a given path lowid contains the name of the leaf node of the tree except in the case of species where genus and species becomes the lowid for that tree path In effect the tree has been expanded in the data file to allow each node name except at species level to ap pear as a unique lowid (Each path in the tree becomes a record in the database file) each record contains

Phylum class order family genus species lowid

(Field to the left of lowid may be blank but not phylum)

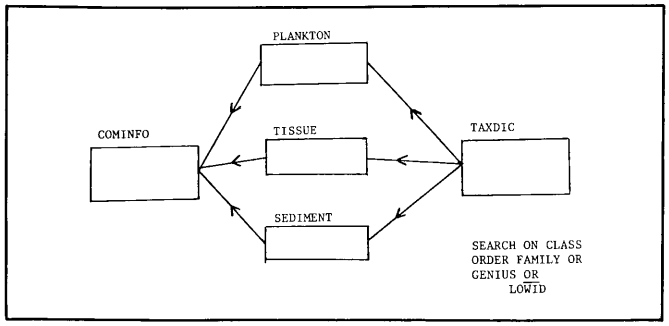

FIGURE 3 Backward relations from the taxonomic dictionary

needed together with simple methods of accessing these descrip tions, for the latter type of user there is a need to provide a menu to the products that are available to run on the database and perhaps how to use them

It deserves mention that a great number of sessions and reports have already been created using the Query Update language, these proliferate easily due to the relative ease of creating these entities, thus there is a great need to keep a record of what sessions and reports are available

Maintenance and loading

The data centre staff up to now have been pre dominantly loading data part of the process being writing lengthy load programs

Each batch of data in a different format may require a com pletely new load program In many instances it is easier to refor mat the data into an existing format for which a load program already exists than to write a new load program but this re quires pre processing of data prior to loading

The database also needs to be edited regularly and for this a number of edit programs have been developed with standard input From our own experience and that of other big users of DMS170 the update facility of the Query Update language has been thought unsafe to use on large files

There are also various procedures that have been developed to restructure database files, compile schema and subschemas

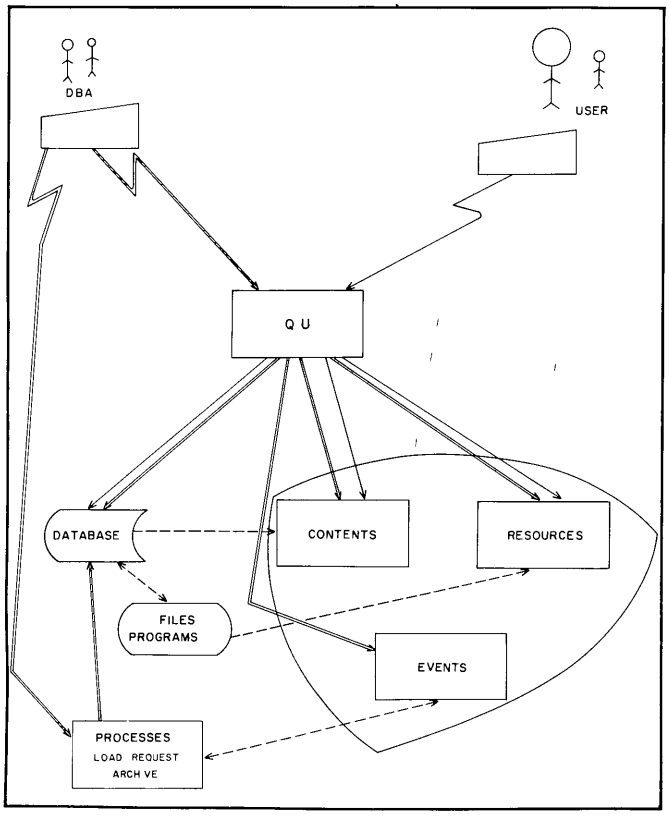

FIGURE 4 The use of the data catalogue

and perform various other file housekeeping tasks, modifying the taxonomony dictionary and adding new names is another difficult task

Thus much time is spent in simply loading a batch of data checking the data by means of listings obtained via a QU report run on the database editing the database, archiving the database prior to a load or restoring the database after an erroneous load All of these activities constitute important events and a log of these events is imperative In addition a list of requests answered, of data received from users, and of tapes mailed among other things needs to be kept

The Data Catalogue

The need for a data dictionary system

The need for a thorough centralized, high level, documenta- tion system is obvious bearing in mind the vast quantity of resources proliferating from within the data centre and in par ticular from the various databases, and also the need for users to find out easily what is available in the database, and to find out what standard products can be provided It can be seen that unless such questions as

what programs are available?
what 'canned' sessions can be run to access biological data?
what are these programs/sessions called?
where can I find them?
how can I use them?
when was the database updated with biological data? by whom?

can be answered the whole system will deteriorate rapidly

SADCO's solution

When the situation became urgent in 1981 the answer seem ed to be to use a good data dictionary system but the Com puter Centre could not justify acquiring such a system In addition the emphasis in SADCO's case was on a documentation system for general systems operation rather than a very power ful system for internal database documentation and automatic generation of database modules It has also been said that although modern data dictionaries have developed rapidly in the past few years they are still not easy to query[1]

Thus SADCO decided to develop its own data dictionary It was intended at the outset that the system should be very sim ple, would document anything even remotely connected to the database, including manual operations and non computer ac tivities and also allow for expansion of entities, entity attributes and files Because of this automation of input or automatic generation of database modules was not a major concern

Since the database construction team was very small and since the database could not progress further until the 'data dic tionary' system was developed, the homemade system was designed with speed of implementation in mind and with a view to a semi automatic facility being introduced later

Looking at the software tools available we had no hesitation in using the Query Update language to implement the 'data dic tionary' completely since then no tedious 3rd level language pro gramming would be necessary

The system, called the *data catalogue* is a micro 'database' in its own right, which ensures that expansion of the system is straightforward and more important searching of the data catalogue is possible using the simple Query Update (QU) com- mands From the database staff point of view it means exper tise in Query Update is used for database retrieval, to update (and query) all other QU databases and to keep up to date and use the data catalogue

But of particular note is the fact that the user having learnt Query Update to access the database can have on line query access to the data catalogue without learning anything new

In addition the powerful QU report writer is available to generate standard reports from the catalogue (See Figure 4)

The Three Sections

There are three main sections to the data catalogue, the CON TENTS, the RESOURCES and the EVENTS section Since QU itself supports relational database these sections can be linked

where it is useful to do so and new relations formed where necessary The reason for keeping these three sections separate is the obvious differences in the type of information needed for each section

The CONTENTS section refers to the database contents itself, is of most use to users and contains detailed information on files records and fields in the database In a scientific database the units in which the data is stored is absolutely vital to the user — eg Nitrate NO3 is it stored as microgram atom per litre or microgram per litre? Since the names of items were limited to seven characters, (initially because we thought FORTRAN access would be the most popular and also to make typing quicker) the detailed description of the name is also given This section is expected to be expanded to include check limits on allowable values for a given variable

The user may do the following

```
IF YRMON = 197806 AND ARENAM = $DURBAN
BAY$ DISPLAY DATUM POSITION DEPTH TEMP
NO3 NH3
```

and receive results, but looking at the data he wants to be sure of the units, so he moves to the CONTENTS subschema with

```
INVOKE CONTENTS LIB CATLOG (UN = X101)
```

and asks

```
IF ITEM NAME = $NO3$ OR $NH3$ DISPLAY ITEM
NAME ITEM UNITS
```

The CONTENTS section is soon to be attended to with a view to automatic generation of the skeleton ITEM record directly from the database schema, obviously UNITS and ITEM DESCRIPTION are not derivable from the schema

Such a modification is simple to implement

The RESOURCES section

This section includes fields such as user-number, resource type source (ie where to find the resource), the date of creation, the library if applicable and a detailed description of the resource In addition, up to nine keywords can be entered for each resource to enable searching on categories of resources The Identifier, which is the primary key of the RESOURCES file, consists of the Institute and the resource name This was made necessary since resource name may not always be unique eg the name FIZCHEM will always be given to the subschema associated with the Physical and Chemistry files regardless of institute (for ease of user manual construction) but different FIZCHEM subschemas may differ according to different database procedures invoked, different items used etc

The resources also include a resource type 'terminal procedure' which is a sequence of steps on a terminal, requiring human intervention, to perform a certain task There are such procedures typically to create, modify or compile a resource The procedure to set up the taxonomic dictionary involves creating the LOWID and expanding the taxonomic dictionary for ease of searching using the QU report-writer, removing duplicate names by means of a temporary QU'database' and then extracting the final records for this database before loading

When the description of how to use any procedure is short the description is then found in the RESOURCE TEXT and no external documentation is needed

Information on resources is entered into the data catalogue at their creation time

EVENTS Section

This section contains a journal of each operation affecting the main database such as load, archival among others as well as all important events or activities within the data centre It provides a means of tracing in detail the history of a resource and gives the DBA up to date knowledge of the status of the database at any time (past or present)

Anything that happens of relevance can be recorded in the EVENTS section, even the sending off of a tape to a user in response to a request for data, thus it is not possible to automate fully data entry into this section

The EVENTS can be used in producing monthly reports of what work has been done, by whom and even to remind oneself of something that must be done in the future (these events are given the event type 'TO DO')

A queue of requests not answered could also be obtained or a list of loads to the database over the last six months for a particular institute may be useful

The date and time is the primary key to this file

The data catalogue requires manual entry by database staff but it is a very useful safeguard against tasks being performed thoughtlessly, without planning, and the discipline imposed of keeping resources and events up to date is certainly worth the effort — it forces a programmer to know what he is currently doing

In fact the whole system revolves around the data catalogue with all activities and resources located from this one central computerized source Any important activity can be located and any entity can be traced from the data catalogue It can even be used to give new staff a good idea of what processes occur in the data centre and what has been produced in the way of software during the years

Conclusion

The data catalogue was initially set up as an emergency solution to a documentation problem

It has the following advantages

1) It is a database and this can be expanded easily and queried on line for information by the data centre
2) It can be quickly accessed by users who need to know what is in the database and what resources are available that are of use to them
3) It is a simple, inexpensive system to set up, maintain and use since it uses the Query Update package which is extensively used for other purposes in the data centre and by users
4) It is an effective management tool for controlling the efficient operation not only of the database but of much of the data processing for the entire data centre

The system has proved invaluable over the past two years and although certain useful features have not yet been implemented in full, the framework has been built for the necessary expansion

References

[1] Allen F W Loomis M E S and Manning M V The Integrated Dictionary/Directory System ACM Computing surveys vol 14 no 2 June 1982 p245 286

[2] Curtice R M and Diekmann E M A survey of Data Dictionaries DATAMATION vol 27 no 3 March 1981 p135 158

Notes for Contributors

The purpose of this Journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles, 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: Prof. G. Wiechers at:

Department of Computer Science
University of South Africa
P.O. Box 392
Pretoria 0001
South Africa

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins. The original ribbon copy of the typed manuscript should be submitted. Authors should write concisely.

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

Tables and figures

Illustrations and tables should not be included in the text, although the author should indicate the desired location of each in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Illustrations should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the Author's name and figure number. Original line drawings (not photoprints) should be submitted and should include all relevant details. Drawings, etc., should be submitted and should include all relevant details. Drawings, etc., should be about twice the final size required and lettering must be clear and "open" and sufficiently large to permit the necessary reduction of size in block-making.

Where photographs are submitted, glossy bromide prints are required. If words or numbers are to appear on a photograph, two prints should be sent, the lettering being clearly indicated on one print only. Computer programs or output should be given on clear original printouts and preferably not on lined paper so that they can be reproduced photographically.

Figure legends should be typed on a separate sheet and placed at the end of the manuscript.

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 l, the number one and prime; between K and kappa.

References

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

1. ASHCROFT, E. and MANNA, Z. (1972). The Translation of 'GOTO' Programs to 'WHILE' Programs, in *Proceedings of IFIP Congress 71,* North-Holland, Amsterdam, 250-255.
2. BÖHM, C. and JACOPINI, G. (1966). Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, *Comm. ACM,* **9**, 366-371.
3. GINSBURG, S. (1966). *Mathematical Theory of context-free Languages,* McGraw Hill, New York.

Proofs and reprints

Galley proofs will be sent to the author to ensure that the papers have been correctly set up in type and not for the addition of new material or amendment of texts. Excessive alterations may have to be disallowed or the cost charged against the author. Corrected galley proofs, together with the original typescript, must be returned to the editor within three days to minimize the risk of the author's contribution having to be held over to a later issue.

Fifty reprints of each article will be supplied free of charge. Additional copies may be purchased on a reprint order form which will accompany the proofs.

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.

Hierdie notas is ook in Afrikaans verkrygbaar.

Quaestiones Informaticae

Contents/Inhoud

*Presented at the Third South African Computer Symposium held on 14th to 16th September, 1983.