

*J. M. Bishop*  
18/7/84



# Quaestiones Informaticae

Vol. 3 No. 2

July, 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

## **Editorial Advisory Board**

**PROFESSOR D. W. BARRON**  
Department of Mathematics  
The University  
Southampton SO9 5NH  
England

**PROFESSOR K. GREGGOR**  
Computer Centre  
University of Port Elizabeth  
Port Elizabeth 6001  
South Africa

**PROFESSOR K. MACGREGOR**  
Department of Computer Science  
University of Cape Town  
Private Bag  
Rondebosch 7700  
South Africa

**PROFESSOR M. H. WILLIAMS**  
Department of Computer Science  
Herriot-Watt University  
Edinburgh  
Scotland

**MR P. P. ROETS**  
NRIMS  
CSIR  
P.O. Box 395  
Pretoria 0001  
South Africa

**PROFESSOR S. H. VON SOLMS**  
Department of Computer Science  
Rand Afrikaans University  
Auckland Park  
Johannesburg 2001  
South Africa

**DR H. MESSERSCHMIDT**  
IBM South Africa  
P.O. Box 1419  
Johannesburg 2000

**PROFESSOR P. C. PIROW**  
Graduate School of Business  
Administration  
University of the Witwatersrand  
P.O. Box 31170  
Braamfontein 2017  
South Africa

## **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.

## Editorial Note

Regrettably this is the last issue of *Questiones Informatica* to appear in its current format. There are a number of reasons for terminating the production of *QI* in printed form. Firstly, the cost of publication has tripled over the last couple of years. Secondly, the interest in the journal has dwindled to the point where one may question the need for a South African publication on the more academic aspects of computing: the readership has not expanded but, more seriously, despite many appeals no contributions are submitted. At present I have only one paper in the pipeline. It appears that the only supply of papers comes from the two-yearly Computer Science Symposium, and this is not sufficient to justify the expense of publishing *QI* in its present form.

Nevertheless, both the Computer Society and the Institute of Computer Scientists feel that a vehicle for publishing the results of research in the field of information technology is required for this country. We propose to continue *QI* in a format similar to that used by *Questions Informatica*. In other words we shall use a photo-reproduction process and a printed cover to continue the publication of *QI*, but at greatly reduced costs. This puts the onus of delivering reproducible copy of the authors, but relieves the printers from the problems experienced with unusual symbols, diagrams, computer printouts, etc.

The journal will continue to accept only papers which have been referred. In fact, we shall try to satisfy all the requirements for being regarded as a publication acceptable under the rules for obtaining university subsidies. It is also intended to guarantee quick publication, say not more than three months after acceptance of a paper. However, the journal can only continue to exist if it receives enough contributions of sufficiently high standard.

Again, I would like to appeal to all engaged in research and development in computer related areas, to consider publishing your results in *QI*.

Finally, it is my pleasure to thank Dick White, of Thomson Publications, for all he has done to produce this journal in its current form.

G WIECHERS,  
Editor

# Migrations: A Microcomputer-Based Generalized Information Retrieval System

José A. Pino

Division Ciencias de la Computación, Universidad de Chile, Santiago, Chile

## Abstract

This paper describes the design of MIGRATIONS, a software system under development at the University of Chile, whose purpose is to store, manage and retrieve unformatted text information.

Among the most important design objectives of the system are generality and simplicity. The first is achieved by allowing the user to describe the documents to be handled. The simplicity objective is attained by having a small set of self-explanatory menus and few concepts to be learned by the user.

## Introduction

Traditional Data Processing uses formatted fields for all the information it handles. This is not a serious restriction in many applications where the length of the data is fixed or variable within limited bounds. Other applications lend themselves to data coding. On the other hand, there are some applications where the basic data is written text, and the field length is very variable. Examples of data with these latter characteristics are: bibliographic references, contracts, letters, laws.

There exist several software systems to handle this type of data. Some, running on mainframes, provide service on just one type of applications, like the software for the Orbit and Dialog services (bibliographic references) [1] and the software for the Lexis service (litigation support information) [5]. Other software, implemented on mainframes as well, has a more general scope of application. Examples of such systems are STAIRS [4], BIRDS [7], and STATUS [8]. There are also some systems implemented on minicomputers, like DOMESTIC [6].

This paper describes the design of MIGRATIONS, a generalised software system under development at the University of Chile, whose purpose is to store, manage and retrieve text data using a microcomputer.

first handles the administration of the data bases: creation, update, removal, and password setting. The second module interfaces with the end user: it performs searches, it displays documents or parts thereof, portions of the index, the queries already asked, etc. Only one of these subsystems may be active at any time.

To access a protected data base through one of these modules, the user must issue a password. The password for MIGRATIONS-ADM may be different from the one for MIGRATIONS-USM, thereby allowing access only for update or retrieval.

Once access is granted, the user is presented with a menu displaying all the available options. To choose one of them, the user has just to type one of the associated codes. Then, a specific menu for the chosen option is displayed.

In this new menu, the user is prompted to choose an option or enter some data (depending on the menu); one of the options is, naturally, to return to the main menu.

Only these two levels of menus are built into the system. Therefore, the design prevents the user from getting lost about

## General description

The need for a generalised information retrieval system implemented on a microcomputer seems clear: many applications cannot justify installation of a mainframe (or minicomputer). Moreover, many offices already have microcomputers for applications such as word processing or spread-sheet analysis. Finally, the user wants personal control of his data, to avoid DP personnel bureaucracy, etc.

In order to be widely applicable, the system must have a high degree of generality and simplicity. Generality is required since documents related to different applications may have very different structures. This objective is attained by accepting a user-defined structure of the documents, much like the Schema definition of a Data Base management System [2].

Simplicity is needed to make the system usable by personnel with little or no training in computers, as it is very desirable in cases like an administrative office, where a secretary may use the system to handle letters and contracts, for example. This objective is partly achieved by designing a set of largely self-explanatory menus to use the system. Consistency is also important in this respect (for instance, assigning the same key to mean "exit from this menu" in all menus).

Other important system objectives are portability (since there are many models of microcomputers in the market), efficiency (both in terms of time to perform operations and in terms of the relatively scarce storage space) and reliability.

The system has been designed as partitioned in two subsystems: MIGRATIONS-ADM and MIGRATIONS-USM. The

```
MIGRATIONS          CREATE DATA BASE

DATA BASE NAME: contracts
STOPWORDS FILE NAME: commwords

FIELD
NUMBER      FIELD NAME  TYPE
001         title      text
002         date-signed date
003         date-effect date
004         expir-date  date
005         amount     number
006         contractor-1 text
007         contractor-2 text
008         referee     text
009         matter     text
010         type       text
011         clauses    text
012         

PRESS RETURN KEY TO EXIT
```

FIGURE 1. Using the "CREATE" menu to describe a data base structure.

where the "return" option leads while working with a given menu.

In both MIGRATIONS-ADM and MIGRATIONS-USM there are "HELP" options, which provide explanations of the system, relevant options and terminology.

### Document definition

As part of MIGRATIONS-ADM, the "CREATE" option allows the user to define the document structure of a data base.

The basic information unit of a MIGRATIONS data base is the document. A document is a composite of one or more fields. Examples of fields are: a title, a date, the text of a letter, etc.

There are three types of fields: text, dates, and numbers. Text fields may contain words, numeric symbols and punctuation signs. Date fields contain specifications of calendar dates (examples: December 6, 1948; Oct. 8, '62; 9/27/82). Number fields may only contain digits, dots and decimal comma. Figure 1 depicts a sample screen showing the definition of a contracts data base (lowercase information has been entered by the user).

The stopwords file contains words the user does not want to be included in the data base index, as is usual in bibliographic information retrieval systems [3].

Once the data base is defined, the documents may be loaded using the "UPDATE" option of the menu. The structure of the data base cannot be changed unless the data base is empty.

### Information retrieval

Searches to the data base are performed using the "FIND" option of the menu. A successful search generates a set of documents which may be examined via the "BROWSE" option of the menu.

Under the "FIND" mode, the user is prompted to enter a logical expression defining the query. The system answers back with the number of documents satisfying the given expression (Fig. No 2).

The simplest queries may have this form:

"xxx"

which means that "xxx" is to be searched in all text fields in all documents. At least one of the words in the "xxx" string must be in the data base index in order to generate a non-empty set.

Another possibility is to specify searches in just one field:  
"xxx" IN field-name

or in several fields:

"xxx" IN field-1, field-2, . . . , field-n

MIGRATIONS	FIND MODE CONTRACTS
12 :	"smith, parker, ltd." in contractor-2 27 HITS
13 :	. 12 and date-signed > "Jan. 1, 1980" 22 HITS
14 :	13 or (12 and amount > "100 000") 24 HITS
15 :	14 butnot "mills country" in contractor-1 24 HITS
16 :	"bridge construction" in matter 236 HITS
17 :	"damages" in clauses 693 HITS
18 :	<input type="checkbox"/>
ENTER QUERY OR JUST PRESS RETURN KEY TO EXIT	

FIGURE 2. Using "FIND" to search queries. Last six queries are always shown.

meaning a document is to be included in the set if it contains "xxx" in any of the specified fields.

Logical operators (AND, OR, BUTNOT) may be used to specify more complicated queries, for example:

"fiction" in title butnot "bradbury" in author

It is also possible to combine queries already done, identifying them by their system generated query numbers. For instance, "fiction" in title and 8

will construct a set consisting of documents which both have the word "fiction" in the title field and belong to the set specified by query number 8.

Numeric and date fields are used with relational operators. The general format is:

field-name rel-op "value"

where "value" is a number or a date and rel-op is greater than (>), greater than or equal to (>=), less than (<), less than or equal to (<=), equal to (=) or unequal to (<>).

Relations using numeric or date fields may be logically combined with sub-expressions involving other relations, text fields and/or query numbers.

To display the documents contained in a set, the user has to choose the BROWSE option in the main menu. The user is then prompted to enter a query number and then he may list the fields he wants to be displayed (the default is all the fields). Documents are then shown on the screen, with the last line indicating the options the user has when the screen is full: continue displaying, skip to next document, and exit to main menu (the second choice is desirable for the case of very long documents).

The user may also wish to print the contents of a selected set. To this end, he has the "PRINT" option of the main menu. In the relevant menu he may specify several special features (like page headings, field name printing, limit for the number of documents) or just use the default settings.

The "INDEX" option of the main menu allows the end user to see a portion of the data base index. In the corresponding menu the user is prompted to enter a word. Then the system displays 15 index words alphabetically close to the one provided by the user, along with the number of documents in which they are contained.

The "QUERIES" option of the main menu is provided to display the defining expression of the queries already done by the system.

The "DATA BASE DEFINITION" option displays the structure of the data base under retrieval.

The "SWITCH" option allows the user to consult another data base. After providing the corresponding password (if the data base is protected), the user is asked whether the queries done during the previous session are to be kept or destroyed. Therefore, it is possible to work intermittently in two or more data bases.

Finally, MIGRATIONS-USM has the "OUTPUT" option which is intended to ease the update process. A selected set of documents (only specified fields if desired) is stored in a file in the same format required by the update program.

Thus, if a user wants to change part of a field x of document y, for instance, he has to define a set (query number) containing document y; then using "OUTPUT", build a file containing x; then use a text editor to modify the file, and finally, change the data base selecting the "UPDATE" option of the MIGRATIONS-ADM menu.

A file made by "OUTPUT" may also be used, after some editing, as input to a word processor.

### Data Base Maintenance

The "UPDATE" option of the MIGRATIONS-ADM menu is intended to add, delete and change documents. The user is prompted to give a file name where update commands are mixed with the new information, if needed (see Fig. 3 for an example of an update file). The user is then told about each command from the file being executed. At the end, a summary, giving

statistics on the update process, is provided. A message file, containing the erroneous commands which were not executed is also produced. If a document is referenced by a command containing an error, it will not be modified by any subsequent command in the same update batch.

The MIGRATIONS-ADM menu also offers password modification, data base removal, display of the data base structure, besides the already mentioned "HELP" and "CREATE" options.

### Concluding remarks

Generality of the system is achieved by letting the user define the document structure of each data base.

Simplicity of use is attained by self-explanatory menus. The novice user will probably only access MIGRATIONS-USM. Introducing him to the simplest forms of "FIND" and to the mechanics of returning to the main menu will suffice for getting started. Updating processes may be a little harder to grasp.

The other objectives of the system are mainly implementation dependent, and the external design does not conflict with them.

The ubiquitousness of microcomputers, added to the relatively high generality and simplicity of this software system should

```
@ADD
*TITLE
MAULLIN BRIDGE CONSTRUCTION
*AMOUNT
485,000
@DELETE 458
@CHANGE 396
*EXPIR-DATE
DEC. 31, 1984
@CHANGE 1011
*CONTRACTOR-1
LLANQUIHUE COUNTY
```

FIGURE 3. An Update Commands File.

make it usable in a wide variety of environments.

### Acknowledgements

Some ideas of this design were previously tried in BIRDS [7], a system designed by a team including Alfredo Piquer, Patricio Poblete and the author. Mario Jofré, José Piquer and Iván Tabkha contributed to the design of MIGRATIONS.

### References

- [1] CHRISTIAN, R., The Electronic Library: Bibliographic Data Bases, 1978-79 (Knowledge Industry Publications, Inc., White Plains, N.Y., 1978).
- [2] DATE, C., An introduction to Database Systems (Addison-Wesley, Reading, Mass, 1975).
- [3] HEAPS, H.S., Information Retrieval: Computational and Theoretical Aspects (Academic Press, New York, 1978).
- [4] IBM CORP., Storage and Information Retrieval System Data Language/I (STAIRS-DL/I), General Information Manual (IBM Corp., Pub. No GH12-5118, 1977).
- [5] MEAD DATA CENTRAL, INC., LEXIS, A Primer (Mead Data Central, Inc., New York, 1975).
- [6] OMER, Y. et al., DOMESTIC, A minicomputer Based Information Storage and retrieval System, Jrnl. of Information Sc., Vol. 3, No 2 (1981), 59-74.
- [7] PINO, J. et al., BIRDS: Bibliographic Information Retrieval and Dissemination system, General Discription, Div. Ciencias Comput., U. de Chile (1980). BIRDS is available from Burroughs Corp. under the name TEXTRIEVE.
- [8] TESKEY, F.N.: STATUS and Integrated Information System, Jrnl. of Documentation, Vol. 36, No 1 (1980), 33-39.

This work was partially supported by a grant from Sonda Ltda. and grant No. I-1687-8312 from the Depto. de Desarrollo de la Investigación of the University of Chile.



## New reading

***Mathematical Foundations of Programming* by Frank S Beckman. Published by Addison-Wesley, Reading, Massachusetts, 1980. 443 pages, exercises, chapter references, index.**

THE phrase 'mathematical maturity' is used to describe a way of thinking that is required not only of computer scientists but also of programmers, not only of systems analysts, but also of managers of programming projects and thus also of dp managers. Very few people would like to attempt a definition of this desirable quality, but one may indicate a relatively painless way of acquiring it: Peruse the book under review.

In its breadth of coverage (with one omission noted below), in the topics dealt with, in the way all topics are related to actual or potential computing practice, the book contributes to the acquisition of the vitally important quality of mathematical maturity and also a quality that can only be termed algorithmic maturity. In other words, this book contributes to the appreciation of what can but also of what cannot be computed, and the various ways in which a computation may be characterised, realised, criticised, and analysed.

The author covers the area variously known as 'theory of computation' or 'meta-theory of computing', in a descriptive, almost intuitive, but mathematically sound fashion. The topics covered include: The concept 'effective'; functions and sets; recursive functions; computability and its limitations; automata and languages; and computational complexity.

In every chapter the author introduces the main topics by examples (so necessary for the development of the 'feel' for a subject) and then progresses through mathematical descriptions to applications. In the process many sidelines are touched upon which in textbooks are mentioned but then tantalisingly ignored. This is thus a book not only to be read but to be dipped into.

The practitioner in computing will find the book worth reading, not only to acquire the aforementioned maturity, but also to gain a background knowledge of the many developments in progress all over the world which go by the name 'fifth generation'. It is rather unfortunate, however, that the predicate logic receives but scant attention in the book since the Japanese fifth-generation R and D is based on Prolog-programming in logic. The research thrust in America, by the MCC, is to be based on a Lisp foundation and various aspects of this functional style are covered in the book.

For students taking a formal course at honours level in one or more topics covered by the author, the book should be recommended reading in order to put their field of study in perspective. For honours students not taking such topics, the book should be required reading.

STEF W POSTMA

# 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 I, 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

Syllabi for Computer Science as a Scientific Discipline.....	3
Stef W Postma	
On a Generalisation of Cayley Diagrams.....	7
W A Labuschagne and H O van Rooyen	
Migrations: A Microcomputer - Based Generalized Information Retrieval System*.....	11
José A Pino	
Software Configuration Management — A practical approach....	15
L S du Preez	
An Adaptive Response Algorithm.....	21
Peter C Pirow	
Specification and Performance Prediction of Fourth Generation Language Run Units*.....	23
S Wulf	
Developing an Intelligent Editor for Microcomputers*.....	25
T S McDermott	
New reading. A book review.....	28
Stef W Postma	

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