REVOKE

A NO LONGER HAS UNIVERSAL SCOPE

LIST OF NAMES( )

REVERSE

4. Conclusion

Critical considerations for designing system conventions consistent with the semantics of non-procedural languages are:

1. To specify clearly the overall input/output behaviour of the system and the scope of definitions.
2. To specify that function definitions remain static across every ⟨*i*-expression⟩. This is particularly important for dealing with the functions for providing system state information (3.4.6)
3. To control output layout by constructing values which represent the desired output, e.g. for devices such as printers, this involves using control characters for new line, new page, etc.
4. To handle interactions with a running program by allowing some functions to be 'implemented' by the user at run time.
5. To treat error messages as a separate data type having no special status and error handling by 'system implemented' function checks.

We do not envisage the provision of these facilities in a multi user system to be very different from that in a single user system. Essentially each user would be provided with a system as described herein with inter user communication being provided by a function USER (name, message) where the value of this function is the response by the named user to the message sent to him/her/it.

6. Acknowledgements

The authors wish to thank D. R. Brough, K. Clark, J. Darlington, D. J. Howarth and R. A. Kowalski for their comments on the draft of this paper. Special thanks to Mrs M. Moriarty for her speedy and efficient typing of this work and to Kathy Jones for typing the revisions.

References

BURSTALL, R. and GOGUEN, J. A. (1977). Putting Theories Together to make Specifications. Invited paper at 5th Int. Jt. Conf. on A.I. Boston, August 1977.

GOGUEN, J. A. (1977). Abstract Errors for Abstract Data Types, *Proc. Conf. on Formal Description of Programming Concepts*, (Ed. Jack Dennis) North-Holland.

HARRISON, P. G. (1974). Alternatives to the Assignment Statement, M.Sc. Thesis, Dept. of Computing and Control, Imperial College, University of London.

HARRISON, P. G., and PRITCHARD, A. (1980). Efficient Storage Management for Functional Languages, Dept of Computing and Control, Imperial College, University of London Research report.

HASKELL, R. (1973). Recursion, Stacks and Go-to's, Research report No. 73/23, Dept. of Computing and Control, Imperial College, University of London.

HASKELL, R. (1975). Efficient Implementation of a Class of Recursively Defined Functions, *The Computer Journal*, Vol. 18 No. 1, pp. 23 to 29.

HASKELL, R. and HARRISON, P. G. (1977). System Conventions for Non Procedural Languages, Research report No. 77/21, Department of Computing and Control, Imperial College.

INGLIS, J. (1976). Structured Programming and Input Statements, *The Computer Journal*, Vol. 19 No. 2, pp. 188-189.

MCCARTHY, J., ABRAHAMS, P. W., EDWARDS, D. J., HART, T. P. and LEVIN, M. I. (1965). *Lisp* 1.5 *Programmers Manual*, second edition MIT Press.

MORRIS, D., FRANK, G. R. and THEAKER, C. J. (1977). Machine Independent Operating Systems, presented at IFIP Congress 1977, Toronto, Canada.

MICHIE, D. (1967). Memo functions: A language facility with rote learning properties, Research Memorandum MIP-R29, Department of Machine Intelligence and Perception, University of Edinburgh.

ROUSSEL, P. (1975). *PROLOG Manuel de reference et d'utilisation*, Groupe d'intelligence artificielle, Universite d'Aix-Marseille II.

SCHWARTZ, J. S. (1977). Using Annotations to make Recursion Equations behave, Extended Abstract, POPL conference.

VUILLEMIN, J. (1973). Proof Techniques for Recursive Programs, Ph.D thesis, Computing Science Department, Stanford University.

Book reviews

Data Communications and Teleprocessing Systems by T. Horsley, 1979; 262 pages. (*Prentice-Hall*, £14·25)

Writing in a clear, concise and entertaining style, the author manages to pack a surprising amount of information into this rather slim volume, which is based on his series of training courses.

The simple, and often amusing, diagrams effectively illustrate basic ideas making this book a valuable starting point for beginners. It should also appeal to specialists looking for a straightforward explanation of salient principles.

The subjects covered include basic theory, system building blocks, error control and network protocols and line control procedures. There are also chapters on public communications facilities and system planning considerations.

The treatment is not deep, but is, nevertheless, adequate and the book is therefore recommended for consideration by anyone concerned with learning about the fundamental technical aspects of data communications.

D. L. A. BARBER (Teddington)

A Technique for Software and Systems Design by R. J. Lano, 1979; 112 pages. (*North-Holland*, $29.25)

From the author's introduction: 'This paper presents a detailed discussion of the N^2 chart and its application to interface and relational analysis . . . The N^2 chart is a visual aid which can be effectively employed to communicate functional or physical interface or relationship information to a large group and/or mixed discipline audience in a very short time period. The N^2 chart additionally provides the user with an effective tool for the definition, tabulation, design and analysis of these interfaces. In this book, the N^2 chart is explained largely through examples rather than text.'

So it continues for 112 pages of self-congratulatory jargon. The essential idea is trivial enough to be stated in half of page 3. The relation between this system and Pert charts and Gantt charts is sketched. Feedback cycles are discussed. The price tag seems massively exorbitant, but is small compared to the price industry pays to send analysts to hear about such things.

J. M. BRADY (Colchester)