FORTRAN suitable for writing production programs.

We have tried to keep the proposed FORTRAN extension as small as possible, and have excluded from consideration a number of other features which could be useful in data class definitions (Geschke and Mitchell, 1975; Geschke, Morris and Satterthwaite, 1977; Liskov *et al*, 1977; Shaw, Wulf and London, 1977; Wulf, London and Shaw, 1976). This makes mastery of Classy FORTRAN a quick and easy process. It also keeps the preprocessor fairly simple, making it straightforward to implement in a reliable manner. A user desiring more extensive features should probably consider using a language other than FORTRAN.

References

American National Standards Committee X3J3 (1976). Draft proposed ANS FORTRAN, *SIGPLAN Notices*, Vol. 11, No. 3, entire issue.

Dahl, O. -J., Dijkstra, E. W. and Hoare, C. A. R. (1972). *Structured Programming*, Academic Press, New York.

Dahl, O. -J., Myhrhang, B. and Nygaard, K. (1970). *SIMULA Common Base Language*, Norwegian Computing Center, Oslo.

Geschke, C. M. and Mitchell, J. G. (1975). On the problem of uniform references to data structures, *IEEE Trans. on Soft. Eng. SE*-1 Vol. 2, pp. 207–219.

Geschke, C. M., Morris, J. H. Jr and Satterthwaite, E. H. (1977). Early experience with Mesa, *CACM*, Vol. 20, No. 8, pp. 540–553.

Hoare, C. A. R. (1972). Proof of correctness of data representation, *Acta Inf.*, Vol. 1, pp. 271–281.

Knuth, D. E. (1968). *The Art of Computer Programming, Vol.* 1: *Fundamental Algorithms*, Addison-Wesley, Reading, Mass.

Liskov, B., Synder, A., Russell, A. and Shaffert, C. (1977). Abstraction mechanisms in CLU, *CACM*, Vol. 20, No. 8, pp. 564–576.

Liskov, B. and Zilles, S. (1974). Programming with abstract data types, Proc. ACM SIGPLAN Conf. on Very High Level Languages, *SIGPLAN Notices*, Vol. 9, No. 4, pp. 50–59.

Shaw, M., Wulf, W. A. and London, R. L. (1977). Abstraction and verification in Alphard: Defining and specifying iteration and generators, *CACM*, Vol. 20, No. 8, pp. 553–564.

Tennent, R. D. (1977). Language design methods based on semantic principles, *Acta Inf.*, Vol. 8, pp. 97–112.

Wegbreit, B. and Spitzen, J. M. (1976). Proving properties of complex data structures, *JACM*, Vol. 23, No. 2, pp. 389–398.

Wulf, W. A., London, R. L. and Shaw, M. (1976). Abstraction and verification in Alphard: Introduction to language and methodology, Tech. Report, Dept. of Comp. Sci., Carnegie-Mellon Univ., Pittsburgh, PA.

Book reviews

Communications Architecture for Distributed Systems by R. J. Cypser, 1978; 710 pages. (*Addison-Wesley*, £15·75)

Although distributed systems are currently a major topic of interest and activity in data processing, there are very few good books on the subject. My immediate reaction on receiving this book by Cypser was one of awe; a 711-page work on network architectures. But on second examination, the title says 'communications architecture' (singular) and is actually about IBM's SNA. This was not a disappointment. IBM did a lot of pioneering in this field and any insight into the design and development of this complex range of hardware and software products must be of value to the industry.

The book is 'official' in the sense that it is published as part of a series sponsored by IBM through an editorial board of staff members. This assures us of a certain level of authority and accuracy. But the 'wants' are not there; do not read the book if you want a balanced view. In spite of that Cypser is still essential reading for specialists in networking and distributed systems.

As one would expect, the book is well structured; starting with a good review of non-architected systems it goes on to deal with the trends in network structures, techniques for improving cost/performance on the communications channels, the influences towards greater distribution and nicely summarises the objectives of effective multiple computer networks. These early chapters lay the foundations for the main sections of the book which are concerned with the concepts of SNA, network services, dataflow and transmission control, operational control and advanced functions such as multi-domain networks. The style is as clear as it can be when encumbered by the unavoidable jargon and initials of such a complex system. However, the liberal use of examples and schematics helps a lot.

What is not clear from the book is the extent to which SNA has changed over the last five years. IBM's early announcements were uncompromising; VTAM would be the only access method, SDLC the only line protocol, all control would be centred on the host and so on. Today there are still three access methods (VTAM, TCAM and XTAM) which can be used in an SNA environment; products as recent as the 8100 have support for the BSC link control and the 8100 also has a networking capability not dependent upon a 370 host system.

This issue of central-host control is a major one and the book only touches on it in a final chapter on interfacing to new data networks. SNA and X25 do not go well together. For example, a schematic of 'one possible arrangement of an SNA path information unit within a packet, within a frame' shows no less than eight headers and trailers. Of course, X25 does not merely specify an interface but also a transmission technique and that technique conflicts strongly with SNA's centralised polling structure. The discussion of these issues makes fascinating reading.

This is not a management book, but is recommended for communications and distributed systems specialists.

David Hebditch (Otley)

Computer Security and Protection Structures by B. J. Walker and I. F Blake, 1977; 119 pages + indices. (*Academic Press*, £10·95)

This is a slim volume, containing a mere 119 pages, which surveys progress in this field to date under the three headings of Threats, which indicate the scope of the problems; Countermeasures, which indicate means for safeguarding various parts of a computer system; and Implemented Systems, which refer to the software and hardware aspects of machine architecture rather than any physical and administrative security techniques. The major part of the book, some 77 pages, deals with countermeasures. The style of presentation is to identify a number of issues in turn such as safeguarding the hardware, the operating system, the terminals, the communication system, etc. and then to discuss each issue briefly with reference to key papers identified in the Bibliography (185 papers). A large number of technical terms are used which are not fully explained, e.g. segmentation, capability addressing, virtual memory, active page registers, inter and intra domain calls, tagged architecture, descriptors, etc. and for the non expert or the student, except a computer science student, this is going to make progress difficult. The authors make no firm recommendations, accept that no system is inviolable and suggest that the best approach is to make the cost of the protection system match the value of the protected data. They conclude by stating that protection promises to be an active area of research for many years to come. I cannot strongly recommend the book since it lacks real substance of its own. However, its brevity means that it is readable and it provides a good set of pointers to anyone starting to take an interest in this area.

D. B. G. Edwards (Manchester)