Anyway, from now on we must interpret 'specification' at the requirements level only.

The book is very p. c. (pedagogically correct). It has been laid out clearly. Each chapter has a helpful introduction, accurate summary, some exercises (mostly rather routine) and a brief list of references with further reading. Two contrasting styles of specification are treated, with two chapters devoted to comparative examples. There is an introductory chapter and a final 'drawing-together' chapter; and there is a glossary. Clearly the authors have gone to some trouble to make the book enjoyable to read and easy to understand.

Here is a description of the contents. The first chapter discusses what constitutes a specification (that is, a formal statement of functional properties). It is followed by an extremely informal introduction to those notions from set theory required later. Then follow several chapters introducing the model-based notation, VDM. A system is specified as an abstract data type, with states and operations on them. Both are described using set theory; VDM provides notations for structuring those descriptions. It is because the states of the data type are constructed in set theory that the method is said to be model based. The types of sequences and maps are described in detail and a chapter is devoted to a 'blocks world' case study.

An operation is specified in VDM by giving: its variables with their types; the precondition required for the operation; and its postcondition, or predicate relating the state before the operation to the state after. The final chapter on VDM describes a method for prototyping VDM specifications in Prolog. The treatment is discursive as it must be in the absence of criteria for a prolog program to meet a specification. But it indicates how to replace some non-executable structures like set, sequence and map with executable structures like lists; and how to execute the postcondition of an operation by regarding it as a prolog program. The restriction required of prolog that its predicates appear in horn-clause form is not discussed; nor is the treatment of negation in the specification (perhaps to be interpreted using prolog's 'closed world' assumption?). However the gross inefficiency of such prototypes is discussed, and the blocks-world example animated using the techniques described. The specification used for animation is the second of two given in the case-study chapter; as already mentioned the relationship between them is not analysed and so the reader must take on trust that the more concrete meets its more abstract specification.

Next the algebraic approach to specification is introduced. Algebraic specification concentrates on the operations of an abstract data type and describes their type and the relationship between them whilst abstracting the underlying state. The date types of queue and binary tree are used to exemplify the method. It is because of the way that an algebraic specification is interpreted that the approach is described as algebraic. The relevant theory is summarised fairly informally in a separate chapter. A simple neural net is specified as a case study in algebraic specification, and the result helpfully contrasted with a VDM specification. Prototyping algebraic specifications in OBJ3 is supported by brief discussions of prototyping using functional languages (too brief) or prolog.

The last chapter comprises 'background, comparison and summary' in which the choice of specification language is discussed and some history is given of the two styles of specification.

Because of its style the book seems more likely to appeal, as a text, to those studying an isolated course on formal specification than to those studying a more mathematically-based course. The authors have no doubt made the correct choice in adopting a discursive style; but with little extra effort the mathematical ambiguities and oversights could have been eliminated.

Strengths of this book are its educational style, its examples, its choice of two styles of formal notation and its comparison of them. Weaknesses are the very loose mathematical style, the scanty treatment of logic and dearth of laws for reasoning about formulae of predicate calculus, and the lack of challenging exercises. The reader should be aware of the authors' choice to consider specifications only at the level of requirements capture and the consequent decision to discuss prototyping of those specifications rather than refinement between levels of abstraction. (That choice is taken in the case of algebraic specification, in spite of their having surveyed the semantic framework required to discuss those matters.) In the case study of the neural net, the authors give an operational, relatively low-level specification (transitive closure, a way of giving a more abstract and less operational description in this and many similar circumstances, is not mentioned in the book). In the blocks-world case study, a specification and more operational design are presented but their relationship not discussed. Those seem to provide missed opportunities for discussing the range of levels of abstraction at which a specification might be pitched, with the chance to describe to the reader how systems are developed hierarchically using formal methods.

J.W. SANDERS
University of Oxford

GIL LERMAN and LARRY RUDOLPH

Lerman and Rudolph have written a readable survey of parallel machines covering the thirty years to 1992. However, this is not a book for the novice. It is assumed that the reader has a reasonable knowledge of the field of parallel processing and its concepts. That said, you don't have to be an expert to read this book.

This survey is itself both substantial and yet, inevitably, incomplete, something for which the authors...
apologise. However, Lerman and Rudolph appear to be presenting us with a sufficiently comprehensive survey to draw reliable conclusions.

The survey itself covers approximately 200 parallel machines which have been built or are about to be built, so excluding mere propositions and distributed systems. It was conducted through the examination of academic papers so is skewed towards novelty in the design of parallel machines and biased against most commercial machines.

Each machine surveyed was classified according to a comprehensive classification scheme (rather than a simple taxonomy) covering such aspects as: control, data exchange, interconnection, memory, etc. However, external factors were also considered in the classification, namely: application, period of construction and type of constructing organisation.

Rough statistical correlation analyses were performed attempting to relate all the different values for the areas of classification for the surveyed machines to one another. Particular correlations were evident, such as SIMD machines being most likely to be used in image processing or that machines for numerical applications are most likely to use shared memory; and these are all listed in the book.

Again, the authors are apologetic, expressing their pessimism regarding their classification scheme, somewhat over-modestly. Consequently, the reader is further advised to treat the conclusions of the correlation with a certain degree of scepticism.

Given their analysis of past trends in parallel machines, the authors add their own prophecies to the classification scheme used in the survey. Each machine surveyed was classified according to a certain degree of scepticism.

Finally, this is a well written book aimed at a reader familiar with the area of parallel processing, the various others regarding how they foresee parallel machines developing in the near future. They predict larger parallel machines in many application areas to satisfy growing computational requirements. The machines will include specialised hardware for certain functions, with mixes of archetypal control mechanism, also combinations of message passing and shared memory, with processors forming clusters in the interconnection network.

Rough statistical correlation analyses were performed attempting to relate all the different values for the areas of classification for the surveyed machines to one another. Particular correlations were evident, such as SIMD machines being most likely to be used in image processing or that machines for numerical applications are most likely to use shared memory; and these are all listed in the book.

Again, the authors are apologetic, expressing their pessimism regarding their classification scheme, somewhat over-modestly. Consequently, the reader is further advised to treat the conclusions of the correlation with a certain degree of scepticism.

Given their analysis of past trends in parallel machines, the authors add their own prophecies to the various others regarding how they foresee parallel machines developing in the near future. They predict larger parallel machines in many application areas to satisfy growing computational requirements. The machines will include specialised hardware for certain functions, with mixes of archetypal control mechanism, also combinations of message passing and shared memory, with processors forming clusters in the interconnection network.

Finally, this is a well written book aimed at a reader familiar with the area of parallel processing, such as a postgraduate or researcher seeking a substantial survey and analysis of parallel machines over the past thirty years. The main text is supplemented by a substantial bibliography, and an appendix listing each machine surveyed with its particular characteristics according to the classification scheme used in the survey.

D. K. G. Campbell
Exeter University

ROY MACLEAN, SUSAN STEPNEY, SIMON SMITH, NICK TORDOFF, DAVID GRADWELL, TIM HOVERD, SIMON KATZ

Primarily aimed at the computer science professional, this book attempts an in-depth explanation of certain systems analysis methodologies, concentrating heavily on ORCA (Object-Oriented Requirements Capture & Analysis).

Few of the concepts encountered in this text are likely to be especially new or difficult to a practising computing professional, since much of the work seems dedicated to the formalisation of common sense. Nevertheless, there is much to be gathered from this material, which manages to avoid being too theoretical with at least half the book filled with case study notes and worked examples, because the techniques outlined there-in potentially provide a useful 'consultants' checklist'. In many places, however, the reader may feel that their time is being wasted by an unnecessarily in-depth study of the situation under consideration and, further, one may consider that, while it is useful to have a formal method to rely on in certain circumstances, ORCA itself is presented with too much scope for informal to be an aid in highly complicated systems.

The book constitutes a very gentle introduction the operational research angle of systems analysis, spending ample time on each concept and new set of notation. As is customary in the computer science field, each new concept is accompanied by its own set of notation, which is tackled well by using the appendices as references for the 'languages' used.

Personally, I find it difficult to believe that this book will be of genuine benefit to a practising systems analyst, since the concepts are presented in too academic a fashion, although the work may be of use to an I.T. manager wishing to sharpen up skills. I honestly feel that this book has a better place as a university business studies course text book than as a professional reference, purely because of its content. The book itself introduces the concepts in an intelligent, formal and well organised manner and is an excellent introduction to methods of systems analysis, which in no way attempts to replace experience and, sadly, I feel is supplanted by it.

Michael Ashton
Warwick University

ERNST WALLMOLLER
Software Quality Assurance—A practical approach.

There can be little doubt concerning the importance of this topic. Certainly this is an area of particular interest to myself and hopefully to every modern software manufacturer. This book is part of the 'B.C.S. Practitioner' series and, as such, is aimed at the computing professional, with which objective the author copes masterfully by producing a highly modularised work, containing only short and relevant paragraphs.