Books

Distributed Computing: A Practical Synthesis, Amjad Umar, Prentice-Hall, Englewood Cliffs, New Jersey, 1993. 744 pp. (ISBN 0-13-036252-2).

This is a deeply serious book, written with perspective, mastery of technical detail, and narrative care. It provides a uniquely satisfying combination of tutorial and professional material in its descriptions of the technologies (and politics) of distributed computing.

The author provides a reference model at the beginning of the book that serves as a useful intellectual structure for the material. He worries about definitions in a careful and precise way, and discusses the motives and goals of various system approaches to provide a useful viewpoint within which to process later detail.

After the excellent conceptual treatment of the topic area, the book builds upward in the usual way, from network architectures through higher level abstractions. It integrates standards issues with technology issues in a smooth and understandable manner. The treatment of basic concepts of networking and network architecture is complete and clear. Particularly useful is the treatment of systems management technologies including SNMP and CMIP protocols. So often that particular topic is muddled and confused in claim and opinion. Not so here.

At higher levels of abstraction, e.g., "client-server," the discussion continues its consistent clarity and detail. This reviewer particularly appreciates the author's comments on the strengths and weaknesses of the remote procedure call and other aspects of "client-server." One incompleteness here, however, and one of the very few of the book, is that client/server application development tools associated with the database management technologies are not discussed in any of the three sections (Client/Server, Distributed Data, or Application Development) where the discussion would be appropriate.

The chapter on operating systems is particularly elegant. It introduces the goals and issues and provides important examples of university solutions (Amoeba, Mach, Project Athena**) that promise to have a maximum impact on commercial technologies in UNIX**, NetWare**, Microsoft Windows NT**, and other environments. Perhaps not enough attention is given to how UNIX or NetWare or NT address certain interoperability issues in particular, but enough is provided so that an easy understanding of those systems can be achieved if necessary.

The discussion of data management issues shows the author's strength in that area. It is remarkably distilled and reduced to its essential elements. It is elegantly integrated with transaction management issues, themselves discussed with good perspective and detail.

Throughout the book one feels the professionalism as well as the scholarship of the author. The references are voluminous, and their quality splendid—almost entirely "primary sources," articles by the development or academic community directly involved in the technologies. In addition to the seriousness of the work, the book also demonstrates great judgment. Vendor-promulgated architectures are kept in their proper proportion, receiving a handful of pages that position them in the greater contexts of the key *de facto* and *de jure* standards that influence them.

Certainly, any reviewer would have a helpful comment or so. For a book of this encyclopedic completeness, one wishes the author could have

given a little more space to IPX** and SPX** details and some more words on some of the other LAN protocols, as one wishes for some more words on client/server tools associated with database management systems. But these are carpings only and do not change the fact that this book may be the best in its area.

The author is especially good at balancing professional and academic viewpoints. So many professional books are extensions of the vendors' marketing material. So many academic books do not distinguish between (for example) file systems that have impacted the industry and file systems that are graduate projects known only to the author-professor and four graduate students.

The solid basis of the book proofs it against time. Its recent publication, 1993, helps it resist an aura of "*deja vu*," but its perspective and level of detail suggest that this will be a useful book for some years, the basic principles and technologies it discusses will endure shifts of focus for a time uncommon in this kind of encyclopedic approach to the vast topic of distributed computing.

This could be the only book one needs. However, the reader must be serious and somewhat sophisticated. The excellent tutorials in the appendices should be looked at before the main body of the text is read. For the motivated reader the effort is worthwhile. For the academic, the book contains case studies and other highly useful material to support academic course work at the junior-to-early-graduate levels. In the ranking of the Michelin Guides—"3 Stars—Worth the Trip"—the best.

Harold Lorin
The Manticore Consultancy
and Senior Adjunct Professor,
Hofstra University
New York

**Trademark or registered trademark of Massachusetts Institute of Technology, X/Open Company, Ltd., Novell, Inc., or Microsoft Corporation.

Practical Software Metrics for Project Management and Process Improvement, Robert B. Grady, Prentice Hall, Englewood Cliffs, New Jersey, 1992. 270 pp. (ISBN 0-13-720384-5).

History proves that the production of high quality software is difficult. It requires a considerable understanding of the processes used and the work produced at each stage of the product life cycle. Metrics, properly used, can assist in improving software quality. The problem for many is the selection and use of a reasonable set of metrics. Robert Grady presents many excellent insights on a set of practical and useful measurements that can be obtained at a reasonable cost. He details an actual metrics program developed at Hewlett-Packard (HP) Co. that spans the entire product life cycle. The author provides a number of stories related to the development and implementation of a company-wide measurements program. This greatly enhances the book's readability.

This latest work by Grady provides an update on the use of metrics as developed in his previous published effort, *Software Metrics: Establishing a Company-wide Program*, which he coauthored with Deborah Caswell. In *Practical Software Metrics for Project Management and Process Improvement*, the author discusses in greater detail the insights gained, the lessons learned, and the process improvements realized from implementing a comprehensive software metrics program.

The text is divided into two parts. Part I addresses the tactical application of software metrics. It starts by offering a number of "rules of thumb" for measuring the development life cycle, including project definition, design, implementation, test, and maintenance. While the numbers provided may not be directly applicable to the reader's specific project or enterprise, the author provides actual data that can be used as benchmarks, if little or no reliable historical data are available. The author includes numerous charts, graphs, and tables providing examples of the metrics implemented. He also discusses data collection tools, data interpretation, and how to present the results. He provides sufficient detail to give the reader a good understanding of their use and assistance in determining their relevance to other projects.

The remainder of Part I details the metrics utilized by HP to evaluate customer satisfaction, effi-

ciency, schedules, quality, and test coverage. The author addresses the following in some detail:

- The Goal/Question/Metric paradigm used to determine appropriate metrics for a project or function
- The use of the Quality Functional Deployment method used to determine customer requirements and needs
- The use of the Functionality, Usability, Reliability, Performance, and Supportability quality model along with customer surveys and interviews to determine customer satisfaction

The book provides useful information on the application of design and code complexity analyzers, test tools, and design and code inspections.

In Part II the author explores the strategic application of software metrics. The general topics discussed in Chapters 9 and 10 concern the use and misuse of metrics. He provides guidelines for selecting appropriate metrics and determining whether they should be used by the project team, project manager, or by the functional manager.

While the collection and analysis of defect data provides a number of benefits to both project engineers and project managers, the most beneficial use of these data is in determining the root cause of defects. The author discusses a defect analysis model, giving examples of the types and categories of data collected along with a method for dissecting and analyzing the results. A major stumbling block to change is the perceived additional cost required. The author provides a process for comparing the additional costs to the potential benefits received. Grady includes models for cost and benefit analysis as well as sanity checks to evaluate design and code inspections, structured analysis and design, and complexity analysis.

A number of strategies are discussed for selecting the classes of problems that should be dealt with first. Some alternatives that are discussed include selecting the largest class, the class easiest to fix, the class most likely to succeed, or some combination of classes. While no one strategy is best for all cases or situations, the discussion presents a variety of approaches to consider and the rationale for each.

In closing, the author provides a set of graphs that can be used to give a comprehensive picture of an enterprise's software development processes, product quality, and customer satisfaction. Grady also includes a discussion of the benefits gained so far from HP's metrics program, what could have been done differently, and the direction metrics may follow in the future.

The author has provided a very comprehensive bibliography. The bibliography contains 396 entries on a wide range of software topics for those who want to pursue specific areas further. As an added bonus for 45 entries, he provides a brief abstract with comments on the usefulness and relevance of the material.

This work is a very complete and frank discussion of an actual software metrics system. It is also an excellent tutorial.

Michael J. Flaherty
Jordan and Jordan
New York

Note—The books reviewed are those the Editor thinks might be of interest to our readers. The reviews express the opinions of the reviewers.