

ence proceedings is covered, but it benefits from being woven into a self-contained book covering both introductory and advanced topics. Concepts and practical algorithms are clearly introduced and described. On the whole the book is clearly written, well-organized and sourced.

If your professional interest is purely academic, and an introductory book on ILP is sought, then this would be a good choice. If however your interests are in the practical applications of this technique outside an academic setting, then some caution is advised. Despite the book's truly excellent coverage and honest evaluation of applied ILP techniques and extant systems, you may find, as I did, that one needs to read and re-read several sections in order to clearly understand the limitations and underlying assumptions when ILP techniques are used to deal with imperfect data. In this respect, the book will require some work despite its introductory nature and intended audience.

D. A. RANDELL
Birmingham University

W. BISCHOFBERGER and G. POMBERGER

Prototyping-oriented Software Development. Springer-Verlag, 1992, 72 DM, 215 pp., hardbound, ISBN 3 540 55448 3

This book consists of two parts. Part 1 describes four different paradigms for software development and the concepts and some of the existing tools behind two of these paradigms—namely 'prototyping-oriented development' and the 'exploratory programming model'. It concludes with a short chapter on some of the issues involved in prototyping. Part 2 describes a toolset called TOPOS, developed at the universities of Zurich and Linz, based on the concepts outlined in Part 1. In this section, exploratory programming is treated more as a component of the prototyping paradigm than as an independent approach to software development. User interface prototyping, system architecture prototyping and the use of simulation in prototyping are some of the concepts implemented in TOPOS. The book concludes with a note on the experience gained through TOPOS.

From the standpoint of a practising software engineer, I find it difficult to feel enthusiastic about the book. The approaches to prototyping-oriented development are considered in an application-independent framework. Consequently, there is little room for detailed discussion on the many specific problem areas that affect software developers. For software engineers who plan, design and implement systems, it would come as a surprise that the authors have not even made a reference to the many tools that are available under MS-Windows on the PC's for prototyping. For example, the development environment under Visual Basic provides for excellent prototyping and evolutionary development through its design-

time and run-time modes, 'object linking and embedding' and custom controls (which are Visual Basic objects with their own properties, events and methods). These are widely used by developers and are superior to tools like ET++ which the book describes.

The book talks about System Architecture prototyping and the use of simulation towards such efforts. Simulation is a good idea if it can be accomplished without too many tears. For large and complex systems, such prototyping itself becomes complex and expands to take up much of the resources. Project managers with schedules hanging over them often have valid reasons to see these efforts as taking the team away from the 'real problem' at hand. Additionally, a large class of problems come under the category of 'information systems'. The architecture for such systems is reasonably well understood by now and we have on-line transaction processors, relational models, report forms and the Windows' event-driven environment to provide a tested framework for such problems, leaving the developer to concentrate on prototyping the user interface and the application.

The book can be recommended for students who are interested in the software development process and in the concepts involved in prototyping-oriented software development. A section describing the kind of development environment that SMALLTALK provides helps in understanding the way TOPOS works. It would have been a good idea to give a demonstration diskette of TOPOS along with the book which the readers could install on their home or work computers to aid them in getting the best out of the book.

Except for a section on SMALLTALK, the tools described in the book are not in the mainstream experience of software professionals in industry. Professional software engineers involved in planning, designing and implementing systems would clearly note that the book is another symbol of the gulf that exists between them and the academic environment in so far as the perception of their problems is concerned.

V. RAGHUNATHAN
Freelance Consultant, Sydney, Australia

NIALl MANSFIELD

The Joy of X. Addison-Wesley, 1993, £21.95, 368 pp., softbound, ISBN 0 201 56512 9

The X Window System has long been a source of frustration to me. It is powerful, but very complex, and clarifying the relationships between the various GUIs, toolkits and libraries associated with X is not easy.

The Joy of X is a refreshing book which guides the reader gently through X, clearly explaining the concepts and utilities which form the X Window System. Niall Mansfield's style is very lucid; each section or subsection (referred to as a 'module') commences with a brief summary of its contents, and takes up a two-page spread.