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.

As far as is possible, each module is self-contained and illustrated with relevant diagrams. It is easy to flick through the book and pick out relevant or interesting topics.

The book is divided into three parts. Part 1 is an overview of X, very much in the style of a sales presentation and tries (very effectively) to convince the reader that X is wonderful. The relationship between X and the various types of computer system it can be run on is discussed, and X is compared with other windowing systems. Part 2 describes the components that make up X, i.e. servers, clients, toolkits, window managers, 'look and feel' and inter-client communication. These are presented so as to describe their individual purposes within an X system, and detailed programming information is omitted. Part 3 is concerned with issues of interest to (say) a system administrator, such as security and performance benchmarking, together with customising applications and other miscellaneous topics.

The amount of detail presented is sufficient for the reader to understand the topics being discussed, and is not clouded by irrelevant technical information. It is a book which will clarify the concepts upon which X is built, but it is not a tutorial in X programming.

If you are a not a programmer, but need to use X, this book will tell you most of what you need to know. If you intend to write programs to run under X, *The Joy of X* will furnish you with the essential understanding of how X works which you will require.

MIKE JOY
University of Warwick

OLIVER JONES
Introduction to the X Window System. Prentice-Hall, 1989, £33.50, 521 pp., softbound, ISBN 0 13 499997 5

The X Window System is defined in terms of message passing between a *server* and one or more *client* programs, the messages being referred to as *events.* The server controls the graphics terminal on which the windows are displayed, and the clients are the applications programs which use (or manage) the display. The procedures contained in the library known as *Xlib* form the most basic building-blocks for creating an X application. Xlib is not user-friendly.

Many applications now are written using toolkits—high-level libraries which hide from the user the tedium of explicitly handling X events (and the other low-level features of X). An X toolkit stands in relation to Xlib in the same way that a high-level computer programming languages does to Assembler. Xlib is useful still, especially for system programmers, but is arguably not very relevant for general use of X.

This book has a potentially misleading title. It is not so much an introduction to the X Window System as to Xlib, and the concepts which are required to understand X are discussed in the context of Xlib. As an introduction to Xlib, however, I found this to be a very useful book. It is an easy reference text and it has deservedly earned itself a place on my bookshelf.

The chapters systematically cover the main topics, including Basic Concepts, Windows, Graphics, Text, Colour, Images, Mouse and Keyboard, and each chapter includes a clear short summary at the end. The text contains many program fragments which simplify greatly the somewhat laborious task of creating one's own X program and remembering everything that has to be done.

The book is now 5 years old; X11R4 has been superceded by X11R5 and X11R6, and some of the material is out-of-date. I hope that an updated version will soon be published.

If you intend to program in X using one of the many toolkits now available, and do not need to investigate X in more depth, then this book is probably unsuitable. If, however, you require the facilities provided by Xlib, it provides a very clear overview both of the Xlib library and of the concepts necessary to understand it.

MIKE JOY
University of Warwick

ROGER D. HERSCH (Ed.)
Visual and Technical Aspects of Type. Cambridge University Press, 1993, £35, 204 pp., hardbound, ISBN 0 521 44026 2

Reviewing an edited book with individual chapters written by different authors is always a complicated task. And when we have the authors coming from highly diverse disciplines like visual communication, typography, type design and computer science the styles of presentation, writing, illustrations and also the extent and depth of coverage are all bound to differ. On the other hand this is a beautifully produced book. The typeface, fontsize and layout are really pleasing to the eye.

The book is a collection of articles based on lectures given at the First European Summer School in Digital Typography, sponsored by the EEC COMETT DIDOT project and held in Lausanne in September 1991. The book has 11 chapters written by eight authors divided into three distinct parts.

The first part of the book consists of four chapters and goes by the title of 'Letterforms: The Basics'. This part basically tries to communicate visual aspects of type and includes articles on visual communications systems as well as the historical evolution of letterforms. Individually the chapters are good and have some excellent illustrations. Overall, however, it is not clear that 55 printed pages are necessary to communicate the few concepts that are being put across to the reader.

The second part of the book with five chapters by computer scientists has been entitled 'Digital Standards and Algorithms'. This part contains material on font