

knowledge representation with objects and object behavior (i.e., methods). For “procedural programmers” who are not convinced of the power of this paradigm, the chapter is good reading. The treatment of object oriented databases was somewhat disappointing in light of the excellent coverage given beforehand. The chapter may require several readings for the novice but is well worth the effort.

The weakest section of the book was the coverage given to expert systems. The treatment was generally too abstract and too brief for a novice to comprehend easily. In addition, the cases used to illustrate both backward and forward chaining were too simplistic to give readers a sense of how expert systems might be of a more serious benefit. The discussion was quite similar to most you will find in introductory texts on expert systems. One exception was the discussion of knowledge representation. Generally, mapping from real-world objects to a computer representation of those objects is a difficult task in itself. Explaining to others how it is done is even that much more difficult. Their brief discussion of an “intuitive model of knowledge representation” makes for worthwhile reading.

Probably the most exciting and rewarding sections of the book were those which covered recent developments in hypertext, hypermedia, and text retrieval. Hypertext is the creation and representation of links between discrete pieces of data; hypermedia is the same kind of associative linking but between such non-textual elements as pictures, sound, and other media. As a technology, it is a way of presenting and using information that is unlike a book. Books represent a linear arrangement of information that provides the reader with a single path through a topic. Hypermedia, on the other hand, is non-linear and associative in nature; allowing the

user to navigate through various chunks of information at her discretion. The historical and technical discussions of hypertext and hypermedia systems are well done, and the authors also present the social and psychological implications of this technology. The latter discussion is enlightening and thought provoking. The section on text retrieval presents a fascinating account of recent developments in information technology. Text retrieval, like hypermedia, is the storage, retrieval, and management of information that is embedded within text (i.e., long documents). It is well written and gives the reader an idea of how tasks such as computerized library research will change over the next decade.

The first five sections of this book described the components that would be necessary for building intelligent databases: object oriented database systems, expert systems for deduction, and various forms of hypermedia. The concluding section of this book describes the integration of these technologies within an architecture called FORM. How integration between these facilities would be carried out is described in great detail, from a description of how objects are represented in FORM to the structure of the intelligent database engine within it. It is, in every sense of the word, a section that pulls together very well the ideas and concepts developed earlier. Overall, this is an excellent book about some very exciting technological developments and is highly recommended.

# SOFTWARE PROTOTYPING, FORMAL METHODS AND VDM

Sharam Hekmatpour and Darrel Ince, Addison-Wesley, 1988.

Reviewer: Lindsay Groves; Dept. of Computer Science; Victoria Univ. of Wellington; PO Box 600; Wellington, New Zealand; lindsay@comp.vuw.ac.nz.

This book is about rapid software prototyping based on formal methods. The first three chapters

comprise an introduction to rapid prototyping, beginning with an examination of the traditional software life-cycle and the problems which motivate the need for rapid prototyping, followed by a review of current approaches and techniques. The remaining seven chapters and the two appendices describe and illustrate a particular prototyping methodology and a prototyping environment, called EPROS, which supports this methodology.

This prototyping methodology uses a formal notation adapted from VDM for describing functional specifications and a form of state transition diagrams for describing user interfaces. A key component of EPROS is a wide-spectrum language, called EPROL, which includes the notation for functional specifications, notation for describing user interfaces (state transition diagrams, forms and menus), and imperative programming constructs similar to those in Pascal and C. The approach is illustrated by two case studies: a “cross-usage” system for recording which modules in a program use and are used by which modules (Chapter 6), and a library system (Chapter 10 and Appendix B).

The introduction to prototyping explains all the issues quite well, but does not present any new insights. The authors discuss different kinds of prototyping and make some comments about the relationship between prototyping and other parts of the software life-cycle. Some discussion of how prototyping might be incorporated in a more complete life-cycle model showing when prototyping is appropriate, as in Boehm’s Spiral Model [1], would help to give a more rounded view of where prototyping fits into the software development process.

The review of prototyping techniques is divided into discussion of techniques for functional prototyping and techniques for user interface prototyping. In each case the various techniques are described quite briefly, presenting enough information to give the reader a

glimpse of each approach, but certainly not enough to enable the reader to evaluate the different approaches or judge their merits in relation to the techniques incorporated in the EPROS methodology. There is a table at the end of Chapter 3 listing domain of application and advantages and disadvantages of each technique, but otherwise there is no evaluation or comparison of the techniques discussed.

The description of EPROS begins with a general description of the EPROS system (Chapter 4). I found this to contain too few details to be very informative, though it made a lot more sense when I reread it after reading the sections in Chapters 6 and 10 (showing EPROS being used to test the specifications for the case studies) and glancing over Appendix A (a Reference Manual for EPROS).

Chapter 5 introduces the notation for expressing functional specifications. The main emphasis in functional specifications is on defining abstract data types using the abstract model approach. Data structures used in the abstract model are defined using predefined types and structuring mechanisms such as sets, lists, trees, and mappings. Operations are defined in terms of pre- and post-conditions. Although the Preface says that a knowledge of VDM is assumed, this description is self-contained, albeit rather terse, and would be quite straight forward reading for anyone with a basic knowledge of logic and discrete mathematics. Lack of previous exposure to this type of notation and style of specification would, however, make the examples and case studies heavy reading.

Chapter 7 introduces the imperative notation in EPROL that is used for implementation, which is described as being a mixture of Pascal and C. In fact, there is much more than this to the language, as it includes all of the EPROL mechanisms for

defining data types, including lists as built-in data types, and polymorphism. Chapter 8 describes the use of state transition diagrams to specify dialogues and the EPROL notation for dialogues, menus, and forms. This is generally well explained, though I felt the connection between menus (and forms) and the description of a dialogue using state transition diagrams needed some clarification. A reader who was interested in user interface prototyping, but not in functional prototyping, should be able to read this chapter after looking over the notation introduced in Chapters 5 and 7.

Chapter 9 introduces structures called "clusters" and the associated notion of "meta-abstraction." A *cluster* is a form of procedural abstraction providing great flexibility in defining the syntax used to invoke the abstraction and the information that can be passed to it. Clusters are illustrated by defining abstractions to create menus and dialogue boxes. This is the most novel material in the book, and to a researcher possibly the most interesting. I would have appreciated seeing further examples to illustrate the power and scope of the idea.

The two case studies are carefully chosen and positioned to illustrate the techniques that have been introduced. They are about the right size to illustrate the techniques in non-trivial examples, without introducing too much complex detail. The first case study, in Chapter 6, illustrates the techniques for prototyping from functional specifications. After a description of the problem, an initial specification is presented and verified (showing that the operators preserve the data type invariant), then the specification is tested using EPROS. A first refinement of the prototype is presented, in which a more explicit data structure is used. The new data structure is shown to provide an adequate

representation, and one of the operators is verified.

The second case study is introduced in Chapter 10, where the problem is described and an initial functional specification is given, tested, and verified. Appendix B gives the state transition diagrams for the user interface and a listing of the final prototype (after three refinement stages), complete with user interface, in which all data is represented using explicit structures (no sets or mappings), and all operators are implemented using the imperative constructs of EPROL.

The Preface states: "This book is intended for four classes of readers: researchers in software engineering, developers who use formal methods in software development, industrial staff who are looking for viable prototyping techniques, and university lecturers who are interested in using a software tool in their formal methods and prototyping courses." This is a very broad audience, and it is doubtful whether any book would adequately meet the needs of all four classes. My impression is that the first part of this book would be a good introduction to software prototyping for someone with little background in software engineering, though such a reader would probably get lost in the formal details in the rest of the book. A reader who is better acquainted with the problems and techniques of software engineering and is interested in learning about a particular prototyping system would find little of interest in the first few chapters, but would find the rest of the book interesting, though somewhat lacking in detail. I believe a practitioner or instructor would want to see more examples of the specification methodology in use, while the researcher would want to see more details about the prototyping system.

This lack of detail is my main criticism of the book. I finished almost every chapter feeling I wanted to know more (though

that may be a good thing!). In particular, I wanted to know how EPROS executes its functional specifications and just what kinds of specifications it can execute. There is a brief note on page 81 to the effect that "certain styles of VDM predicates, while expressible in EPROL, are not executable," but no explanation of what these are, nor what EPROL does with specifications containing them. I also wanted to know what facilities EPROL provided to support the process of refining prototypes: are there tools to assist in constructing the refined specifications or to manage the different versions of the specifications that result? The body of the book is only 137 pages (222 including references, appendices, and index), so concern about size should not have prevented more detail being provided.

These criticisms aside, I enjoyed reading the book and will certainly refer to it in the future. It is generally well written and I found very few errors. The bibliography is extensive, containing around 270 entries, though none is dated later than 1986. I am eager to try using EPROS and will consider using it in teaching. Although the book says nothing about the availability of EPROS, I contacted Sharam Hekmatpour and discovered that EPROS (which is implemented in LISP and runs under Unix) is available free of charge, and can be obtained by anonymous FTP from the University of Melbourne (munari.oz.au).

[1] Barry W. Boehm, "A Spiral Model of Software Development and Enhancement", *Computer*, May 1988, pp 61-72.

## COMPUTERS IN THE HUMAN CONTEXT

Tom Forester (Ed.), The MIT Press, 1989, 548pp.

Reviewed by: Lynne Alexander, Consultant; Interfaces Xcetera; Houston, Texas.

This collection of 43 articles presents a variety of opinions on

how information technology (IT) affects humankind. The editor chose articles representing contrasting opinions on many themes and issues. The articles offer historical and philosophical perspectives on IT. Most offer or imply suggestions on directions for the future use and development of IT. Most of the articles are reprints from other publications.

The book is well organized. It is divided into four parts:

- 1) Computers and Society,
- 2) Computers and People,
- 3) Computers and Organizations, and
- 4) Computers and the Future.

Each part is divided into three or four chapters, and each chapter contains two or three articles. The editor's introduction gives an excellent preview of the assembled articles, how they relate to one another, and their purpose in the book. Each article is abstracted with a paragraph written by the editor detailing the article's particular purpose in the book. Each part ends with a section on "Selected Further Reading."

The first part consists of three chapters on IT in general (not only computers). The first two chapters ("IT as Revolution" and "IT as Evolution") debate how advances in information technology have affected society from a historical perspective. The last chapter ("The Future with IT") contains articles that suggest how the growth of IT will affect the historical perspective on our society and how IT might change society in the future.

The second part deals with how computers are used by people. Chapter 4 ("Minds and Machines: the AI Debate") contains three viewpoints on the potential uses and abuses of artificial intelligence. "Machines and Users," Chapter 5, introduces human factors issues in computer design. This chapter includes a well written article ("Designing the

User Interface") by Ben Shneiderman, summarizing some of the ideas put forth by his most recent book. The next chapter, "IT in the Home," is a debate on whether Home Informatics (new technologies to bring information to the home) will eventually create the "electronic cottage" (people working and obtaining information at home). Chapter 7 is a discussion of the role computers play in the classroom ("IT in Schools").

Part three of this book presents various authors' disillusionment with the role computers have played in the office, factory, commerce, and management. The articles describe difficulties with past and current computer solutions to problems, the issues these solutions present to industry, and potential remedies to the problems. One particularly interesting article in this part gives a quantitative assessment of the negative role computerization has played in the banking business. An emphasis of this part is the importance of maintaining the role of humans in industry, i.e. that computers have not replaced humans effectively.

The last part consists of some entertaining articles on IT issues we will probably be dealing with in the future. Included is an informative article on intellectual property ownership. Equally interesting is an article on the people who commit computer crimes and what to do to prevent this kind of crime. This part considers ethical, political, economic, social, and global issues.

In general, I found this book to be an interesting read. It provides me, as a technical person, with information to formulate my own ideas on moral issues that I face in my profession. The author certainly did a lot of work to accumulate many well written articles on the subject of information technology and how it affects humans. However, the majority of the articles seemed to be somewhat pessimistic and foreboding about the effects of