PREFACE

Protocols are sets of rules that govern the interaction of concurrent processes in distributed systems. Protocol design is therefore closely related to a number of established fields, such as operating systems, computer networks, data transmission, and data communications. It is rarely singled out and studied as a discipline in its own right. Designing a logically consistent protocol that can be proven correct, however, is a challenging and often frustrating task. It can already be hard to convince ourselves of the validity of a sequentially executed program. In distributed systems we must reason about concurrently executed, interacting programs.

Books about distributed systems, computer networks, or data communications often do no better than describe a set of standard solutions that have been accepted as correct by, for instance, large international organizations. They do not tell us why the solutions work, what problems they solve, or what pitfalls they avoid.

This text is intended as a guide to protocol design and analysis, rather than as a guide to standards and formats. It discusses design issues instead of applications. Two issues, therefore, are beyond the scope of this text: network control (including routing, addressing, and congestion control) and implementation. There is, however, no shortage of texts on both topics. The design problem is addressed here as a fundamental and challenging issue, rather than as an irritating practical obstacle to the development of reliable communication systems. The aim of the book is to make you familiar with all the issues of protocol validation and protocol design.

The first part of the book covers the basics. Chapter 1 gives a flavor of the types of problems that are discussed. Chapter 2 deals with protocol structure and general design issues. Chapters 3 and 4 discuss the basics of error control and flow control.

The next four chapters cover formal protocol modeling and specification techniques, beginning in Chapters 5 and 6 with the introduction of the concept of a protocol validation model, that serves as an abstraction of a design and a prototype of its implementation. In Chapter 5 a terse new language called PROMELA is introduced for the description of protocol validation models, and in Chapter 6 it is extended for the specification of protocol correctness requirements. In Chapter 7 we use PROMELA to discuss a number of standard design problems in the development of a sample file transfer protocol. Part II closes with a discussion, in Chapter 8, of the extended finite state machine, a basic notion in many formal modeling techniques.

The third part of the book focuses on protocol synthesis, testing, and validation techniques that can be used to battle a protocol's complexity. Both the capabilities and the limitations of the formal design techniques are covered.

The fourth and last part of the book gives a detailed description of the design of two protocol design tools based on PROMELA: an interpreter and an automated validator. Based on these tools, an implementation generator is simple to add. Source code for the tools is provided in Appendices D and E. The source is also available in electronic form. Ordering information can be found in Appendix E.

LECTURE PLAN

The core of this book is contained in Chapters 2, 5, 6, 7, and 11. These chapters explore a design discipline that is supported by the tools discussed in Chapters 12 to 14. The remaining text is meant to make the book relatively self-contained. Chapter 3 on error control, Chapter 4 on flow control, and Chapter 8 on finite state machines give background information that should be part of the working knowledge of every protocol designer. Chapters 9 and 10 bring the reader up-to-date with the latest techniques in closely related fields of protocol engineering.

For a one-semester course in protocol design the following sequence of chapters and appendices is suggested: 1, 2, A, 3, B, 4, 5 & C, 6-14. A shorter course, for instance embedded in full semester course on operating systems or data networks, would consist of Chapters 1, 2, 5, 6, 7-11, 14. The software discussed in the book can be used for class projects in the design and validation of sample protocols. Suggestions for exercises are included throughout the text.

ACKNOWLEDGMENTS

Many people have helped in the preparation of this book. Friendly readers who worked their way through earlier drafts include Jon Bentley, Geoffrey Brown, Tom Cargill, John Chaves, Mohamed Gouda, Paul Haahr, Brian Kernighan, David Lee, Doug McIlroy, Sally McKee, Norman Ramsey, Howard Trickey, and Colin West.

The validation software was developed over many years. Crucial help in the derivation of the basic algorithm of supertrace was given by Doug McIlroy, Rob Pike, Jim Reeds, and Ken Thompson. Costas Courcoubetis and Mihalis Yannakakis extended the software with algorithms for analyzing liveness properties.

I am also grateful to Greg Chesson, Tony Dabhura, Sandy Fraser, Joop Goudsblom, Andrew Hume, Mike Lesk, Don Mitchell, Beate Oestreicher, John Peterson, Björn Pehrson, Dave Presotto, S. Purushothaman, Krishan Sabnani, Ravi Sethi, and M. Sullivan for references and valuable suggestions.

Gerard J. Holzmann