

---

# **COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION**

---

---

**THE KLUWER INTERNATIONAL SERIES  
IN ENGINEERING AND COMPUTER SCIENCE**

---

# COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

*by*

**Richard Lai**

*La Trobe University  
Melbourne, Australia*

**Ajin Jirachiefpattana**

*Prince of Songkla University  
Songkhla, Thailand*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-7537-1 ISBN 978-1-4615-5549-0 (eBook)

DOI 10.1007/978-1-4615-5549-0

### **Library of Congress Cataloging-in-Publication Data**

A C.I.P. Catalogue record for this book is available  
from the Library of Congress.

---

**Copyright © 1998 by Springer Science+Business Media New York**

Originally published by Kluwer Academic Publishers 1998

Softcover reprint of the hardcover 1st edition 1998

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC

*Printed on acid-free paper.*

This book is dedicated by  
Richard Lai to his wife, Freda,  
and three daughters: Debbie,  
Jasmine and Amelia;

and by Ajin Jirachiefpattana to  
his wife, Waraporn, and his  
daughter, Nidjaree.

# Contents

List of Figures	xiii
List of Tables	xvii
Preface	xix
Acknowledgments	xxiii

## Part I Protocol Specification

1. COMMUNICATION PROTOCOL	5
1.1 Introduction	5
1.2 Technical Issues	6
1.3 Communication Protocols	6
1.4 The Need for Architecture	7
1.5 The OSI Reference Model	7
1.6 Concept of a Layered Architecture	8
1.6.1 The Need for Layering	8
1.6.2 Layering	9
1.6.3 Layer Operation	10
1.6.4 Introduction to the Specific Layer	11
1.7 Problems of Communication Protocols	13
1.8 ISO Layer Specification	14
1.8.1 Service Specification	14
1.8.2 Protocol Specification	16
1.8.3 The Procedure for Protocol Specification	17
1.9 The Sliding Window Protocol	17
1.9.1 Sequence Numbering	18
1.9.2 Timeout and Retransmission	18
1.9.3 Transmitter Behaviour	19

## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

1.9.4	Receiver Behaviour	19
1.10	Association Control Service Element	20
1.10.1	Association Establishment	21
1.10.2	Normal Release of an Association	22
1.10.3	Abnormal Release of an Association	22
	Chapter References	23
2.	FORMAL DESCRIPTION TECHNIQUES	27
2.1	Introduction	27
2.2	Formal Description Techniques	28
2.2.1	FDT Objectives	29
2.2.2	Tools for FDTs	30
2.3	Protocol Engineering	30
2.4	Protocol Development Methodology	32
2.5	FDT Types	33
2.5.1	Finite State Machine Model	33
2.5.2	Extended Finite State Machine Model	35
2.5.3	Other models	36
	Chapter References	37
3.	ESTELLE	39
3.1	Introduction	39
3.2	The Estelle Model	40
3.3	Modules	42
3.3.1	Interaction Points	43
3.3.2	Channels	44
3.4	Module Communications	44
3.4.1	Message Exchange	44
3.4.2	Restricted Sharing of Variables	45
3.5	Module Nesting and Attributes	45
3.6	Behaviour of an ESTELLE Specification	47
3.7	Specification Syntax	48
3.7.1	Communication Channels	48
3.7.2	Module Header Definition	49
3.7.3	Module Body Definition	50
3.8	Internal Behaviour of a Module	53
3.9	Estelle Statements	54
3.9.1	Init Statement	54
3.9.2	Connect and Attach Statements	54
3.9.3	Disconnect Statement	57
3.9.4	Detach Statement	58
3.9.5	Release and Terminate Statements	58

3.9.6	Output Statement	59
3.9.7	All, Forone and Exist Expressions	59
3.10	Protocol Specification Using Estelle	60
3.10.1	The Sliding Window Protocol	60
3.10.2	The ACSE Protocol	65
	Chapter References	80
4.	LOTOS	81
4.1	Introduction	81
4.2	Processes	82
4.3	Behaviour Expressions in Basic LOTOS	83
4.3.1	Process Termination	84
4.3.2	Two Basic Operators	85
4.3.3	Processes as Trees	85
4.3.4	Recursion	86
4.3.5	Nondeterminism and Internal Action	86
4.3.6	Sequential Composition	86
4.3.7	Parallelism	87
4.3.8	Disruption	89
4.3.9	Hiding	89
4.3.10	Restriction	89
4.4	LOTOS Data Types	90
4.4.1	Basic Concepts	90
4.4.2	Signature	90
4.4.3	Terms and Equations	91
4.4.4	Extensions and Combinations	91
4.4.5	Parameterization	93
4.4.6	Renaming	93
4.5	Structure of a LOTOS Specification	94
4.6	Full LOTOS	95
4.6.1	Structured Actions	95
4.6.2	Behaviour Expressions in Full LOTOS	97
4.7	Specification Styles	99
4.8	Protocol Specification Using LOTOS	101
4.8.1	The Sliding Window Protocol	101
4.8.2	The ACSE Protocol	105
	Chapter References	108
5.	SDL	111
5.1	Introduction	111
5.2	Overview of the Language	112
5.3	Basic Concepts	115
5.3.1	System Specification	115



## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

5.3.2	Block Specifications	117
5.3.3	Process Specifications	119
5.4	Structural Concepts	122
5.4.1	Partitioning	123
5.4.2	Refinement	123
5.5	Additional Concepts	123
5.5.1	Macros	123
5.5.2	Generic Systems	123
5.6	Data Concepts	125
5.6.1	Sorts	125
5.6.2	Operators, Literals and Terms	126
5.6.3	Equations	126
5.6.4	Generators	128
5.6.5	Inheritance	128
5.6.6	Constants and Sort Renaming	130
5.6.7	Records and Fields	130
5.6.8	Predefined Sorts	131
5.7	Protocol Specification Using SDL	131
5.7.1	The Sliding Window Protocol	131
5.7.2	The ACSE Protocol	132
	Chapter References	137

## Part II Protocol Verification

6.	PROTOCOL VERIFICATION	143
6.1	Introduction	143
6.2	Protocol Verification	144
6.3	Research on Protocol Verification	144
6.4	Verification Methodology	145
6.5	Protocol Properties	146
6.6	Petri Nets	147
6.6.1	Elements of Petri Nets	148
6.6.2	Reachability Graph	149
6.6.3	Some definitions	152
6.6.4	Limitations of Petri Nets	153
6.7	Reachability Analysis	153
6.7.1	Other Forms of RA	154
6.8	The State Space Explosion Problem	154
6.8.1	Techniques for Relieving the Problem	155
6.9	Other Verification Techniques	157
	Chapter References	159

7. A REVIEW ON ESTELLE VERIFICATION	165
7.1 Introduction	165
7.2 Principles of Estelle Verification	166
7.3 Existing Methods	169
7.3.1 Thalmann's Method	169
7.3.2 PRANAS-2	170
7.3.3 ESTIM	171
7.3.4 XESAR	173
7.3.5 VEDA	174
7.3.6 VESAR	175
7.3.7 EDB	177
7.3.8 Wu & Chanson's Method	178
7.3.9 EDS	179
7.3.10 Bojanova's Method	180
7.3.11 EVA	181
7.3.12 Dimitrov & Petkov's Method	182
7.3.13 PIPN	185
Chapter References	187
8. NPNS MODELLING ESTELLE	193
8.1 Introduction	193
8.2 Numerical Petri Nets (NPNs)	194
8.2.1 Extensions	194
8.2.2 Enabling and transition firing	195
8.2.3 An NPN specification language	195
8.3 Advantages of Employing NPNs to Model Estelle	197
8.4 A NPN-Based Model for Estelle	197
8.4.1 Tokens	197
8.4.2 NPN Notation for Estelle	204
8.5 Translation from Estelle into NPN	206
8.5.1 Translation of a module	206
8.5.2 Translation of the initialization part of a module	207
8.5.3 Translation of the transition part of a module	209
8.5.4 Translation of some Pascal statements in Estelle	210
8.5.5 Translation of Pascal data types and variables	211
8.5.6 Translation of Estelle specific statements	212
8.5.7 Transition enabling and firing	217
8.5.8 Net composition	218
8.5.9 Modelling dynamic operations of Estelle	219
8.6 An Example	225
8.7 Analysis of Generated NPNs	228
8.8 Merits and Limitations	229
Chapter References	230

## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

9. EVEN - A SOFTWARE ENVIRONMENT FOR ESTELLE VERIFICATION	235
9.1 Introduction	235
9.2 Overall Structure of EVEN	236
9.3 The Portable Estelle Translator (PET)	237
9.4 The NPN Generator	238
9.4.1 Design Issues	239
9.4.2 Class Design	239
9.4.3 Program Design	241
9.4.4 Files generated by the NPN generator	242
9.4.5 Invoking the NPN Generator	246
9.5 The NPN Verifier	246
9.5.1 An Overview of Prolog and MU-Prolog	246
9.5.2 Overall Structure of the NPN Verifier	247
9.5.3 PROTEAN-to-Prolog Translator (net2pl)	248
9.5.4 Reachability Graph Generator and Analysis Program	253
9.5.5 Invoking the NPN verifier	254
9.6 A Protocol Development Environment	254
Chapter References	255
10. A METHOD TO ADDRESS THE STATE SPACE EXPLOSION PROBLEM	259
10.1 Introduction	259
10.2 State Space Caching	260
10.3 Sleep Sets	263
10.4 Dependency Relations	265
10.5 Reducing interleavings caused by extra transitions	268
10.6 A Small Experiment	271
Chapter References	272
11. APPLICATIONS OF EVEN	275
11.1 Introduction	275
11.2 Verification Methodology	276
11.3 Sliding Window Protocol	277
11.3.1 NPNs for Estelle Specification of the Sliding Window Protocol	277
11.3.2 Verification of the Sliding Window Protocol Using EVEN	278
11.3.3 Performance Results	279
11.4 ACSE (Association Control Service Element)	281
11.4.1 NPNs for Estelle Specification of the ACSE Protocol	283
11.4.2 Verification of ACSE Using EVEN	287
11.4.3 Verification Results	287
11.4.4 Performance Results	292
11.5 Summary of the Verification Results	295
Chapter References	295
Index	297

## List of Figures

1.1	The OSI Reference Model	8
1.2	Layer concept of the OSI reference model	9
1.3	Service Primitives	10
1.4	Relationship between data units	11
1.5	A procedure for protocol specification	15
1.6	Data Structure for the Transmitter Window	18
1.7	Data Structure for the Receiver Window	20
1.8	A time-sequence diagram for the ACSE and Presentation Service Primitives	24
2.1	A Protocol Development Methodology	32
2.2	Finite State Machine	34
3.1	Hierarchical Representation of the Estelle Model	40
3.2	Embedded Representation of the Estelle Model	41
3.3	Graphical Representation of a Module	43
3.4	Module Attributes	46
3.5	Attach Operation	55
3.6	Connect Operation	56
3.7	End-Point to End-Point Communication	57
3.8	Architecture of the Sliding Window Protocol for Estelle Specification	61
3.9	Specification model for ACSE	65
4.1	Two interacting processes: P1 with gates a,b,c, and P2 with gates c, d	83
4.2	Two different processes with their behaviour represented in tree structure	85
4.3	An example of the use of parameterized data type construct	94
4.4	An action in full LOTOS	96

## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

4.5	The Architecture of the Sliding Window Protocol for LOTOS Specification	101
4.6	The overall structure of the Sliding Window Protocol LOTOS Specification	102
4.7	The Architecture for the ACSE Protocol LOTOS Specification	105
4.8	The overall structure of the ACSE Protocol LOTOS Specification	106
5.1	The structure of an SDL specification	113
5.2	Summary of SDL Graphical Symbols	114
5.3	An example of a system specification (a) in SDL/GR and (b) in SDL/PR	116
5.4	An example of a block specification (a) in SDL/GR and (b) in SDL/PR	118
5.5	An example of a process specification (a) in SDL/GR and (b) in SDL/PR	120
5.6	An example of a macro definition and a macro call (a) in SDL/GR and (b) in SDL/PR	124
5.7	An example of the use of system parameters	125
5.8	An example of the use of GENERATOR construct	129
5.9	SWP System in SDL/GR	132
5.10	(a) sender_entity block, (b) receiver_entity block and (c) medium in SDL/GR	133
5.11	ACPM System in SDL/GR	134
5.12	(a) Initiator_ACPM block, (b) Responder_ACPM block and (c) PS in SDL/GR	135
5.13	IAEST process behaviour in SDL/GR	136
5.14	RAEST process behaviour in SDL/GR	137
6.1	A Simple Petri Net	149
6.2	Reachability Set	151
6.3	Reachability Graph	152
7.1	Principles of Estelle Verification	166
7.2	Analysis system for Estelle specifications proposed by U. Thalmann	169
7.3	Functional architecture of the PRANAS-2 verification subsystem	170
7.4	The Estelle* verification toolset	172
7.5	Structure of XESAR	173
7.6	Overall structure of VEDA	175
7.7	Architecture of VESAR	176
7.8	Configuration of Estelle Compiler (EC) and Estelle Simulator/Debugger (EDB)	177
7.9	The components of the verification system of EDS	179

## LIST OF FIGURES

7.10	An approach to Estelle verification proposed by I. Bojanova	180
7.11	Architecture of EVA	182
7.12	An example of transforming Estelle to Petri nets	183
7.13	Main schema of the approach proposed by Dimitrov and Petkov	184
7.14	An example of expressing a module header in PIPN	186
7.15	An example of expressing a transition in PIPN	186
7.16	Overall structure of the PIPN tool	187
8.1	Elements of a generic Numerical Petri Net	194
8.2	BNF syntax of the NPN specification language PROTEAN	196
8.3	Mapping of Estelle components to NPNs	198
8.4	An example of six token types on an NPN graph	199
8.5	BNF syntax of each type of token used in this model	201
8.6	A simple NPN graph for an Estelle transition	206
8.7	An NPN graph for an Estelle module	208
8.8	NPN modelling all-statement	214
8.9	NPN modelling for one-statement	215
8.10	NPN modelling exist-expression	216
8.11	An NPN graph for an Estelle transition	218
8.12	Integrating two NPN nets using input arcs and output arcs	219
8.13	An NPN model for module instance creation	220
8.14	NPNs modelling (a) a connection link and (b) an attachment link between two modules	221
8.15	An NPN model for the disconnect operation	223
8.16	An NPN model for module instance release	224
8.17	An Estelle specification of a simple data transfer protocol	226
8.18	An NPN model for the Estelle specification of a simple data transfer protocol	228
9.1	EVEN	237
9.2	The class structure used in the NPN generator	240
9.3	Class relationship diagram of <b>Place</b> , <b>Trans</b> , <b>Arc</b> and <b>Token</b> classes	240
9.4	Hierarchical structure of five main functions used in the NPN generator	241
9.5	Structure of the NPN verifier	248
9.6	Structure of a Rapid and Reliable Protocol Development Environment	257
10.1	Classical Exhaustive Depth-First Search vs. Stack Search	260
10.2	The reachability graph for the NPN shown in Figure 8.18	261
10.3	State space caching and sleep sets	264
10.4	An example of the dependence of three Estelle transitions in the same module	267

## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

10.5	An example of firing an extra transition immediately	268
10.6	State space caching and sleep sets with immediate extra transition firing	269
10.7	The reachability graph generated by Algorithm 4	270
11.1	An Estelle verification procedure	276
11.2	An NPN model for the Sliding Window protocol Estelle Specification	278
11.3	Performances of Algorithm 1 and Algorithm 4 for various states	281
11.4	The reachability graph for the Sliding Window Protocol with window size of 2	282
11.5	An NPN graph for the ACSE association establishment	284
11.6	An NPN graph for the ACSE normal release	285
11.7	An NPN graph for the ACSE abnormal release	286
11.8	A sequence of service primitives causing Deadlock 1	289
11.9	A sequence of service primitives causing Deadlock 2	290
11.10	A sequence of service primitives causing Deadlock 3	291
11.11	A sequence of service primitives causing Deadlock 4	292
11.12	Performances of Algorithm 1 and Algorithm 4 for various visited states for the ACSE protocol	294
11.13	The reachability graph and sleep sets for the abnormal release service with collision	296

## List of Tables

1.1	ACSE APDUs and mapping between application and presentation primitives	21
1.2	AARQ APDU fields and AARE APDU fields	22
1.3	RLRQ APDU fields, RLRE APDU fields and ABRT APDU fields	23
3.1	Module types and their interaction points in the SWPM Estelle specification	62
3.2	Module types and their interaction points in the ACSE Estelle specification	66
4.1	A list of basic LOTOS behaviour expressions	84
4.2	Types of interaction	97
6.1	Reachability Graph in Tabular Form	150
6.2	Details of the Markings for the Petri Net	150
6.3	Power of Protocol Verification Techniques	160
7.1	Methods for Verification and Validation of Estelle specifications	167
7.2	Protocols with their applied Estelle validation and verification methods	168
8.1	Mapping of Estelle components to NPNs	200
8.2	Estelle and NPN Data Types	211
8.3	A summary of the translation from Estelle to NPN	232
8.4	A summary of the translation from Estelle to NPN (cont.)	233
8.5	Input/Output Places, Input/Output Tokens and Transition Conditions of Figure 8.18	234
10.1	Initial marking for the NPN shown in Figure 8.18	262
10.2	A comparison between the performances of Algorithm 1, 2, 3 and 4	272
11.1	Initial marking of the NPN shown in Figure 11.2	279



## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

11.2	A comparison between the performances of Algorithm 1 and Algorithm 4 for the Sliding Window Protocol with six different window sizes	280
11.3	Sleep sets associated with each of the markings shown in Figure 11.4	283
11.4	A comparison between the performances of Algorithm 1 and Algorithm 4 for the seven cases of the ACSE protocol	293

## Preface

Communication protocols are rules whereby meaningful communication can be exchanged between different communicating entities. In general, they are complex and difficult to design and implement. Specifications of communication protocols written in a natural language (e.g. English) can be unclear or ambiguous, and may be subject to different interpretations. As a result, independent implementations of the same protocol may be incompatible. There is, therefore, a need for precise and unambiguous specification using some formal languages. In addition, the complexity of protocols makes it very hard to analyse in an informal way.

Many protocol implementations used in the field have almost suffered from failures, such as deadlocks. When the conditions in which the protocols work correctly have been changed, there has been no general method available for determining how they will work under the new conditions. It is necessary for protocol designers to have techniques and tools to detect errors in the early phase of design, because the later in the process that a fault is discovered, the greater is the cost of rectifying it.

Protocol verification is a process of checking whether the interactions of protocol entities, according to the protocol specification, do indeed satisfy certain properties or conditions which may be either general (e.g., absence of deadlock) or specific to the particular protocol system directly derived from the specification.

In the 80's, an ISO (International Organisation for Standardisation) working group began a programme of work to develop formal languages which were suitable for Open Systems Interconnection (OSI). This group called such languages Formal Description Techniques (FDTs). Some of the objectives of ISO in developing FDTs were: enabling unambiguous, clear and precise descriptions of OSI protocol standards to be written, and allowing such specifications to be

## COMMUNICATION PROTOCOL SPECIFICATION AND VERIFICATION

verified for correctness. There are two FDTs standardised by ISO - LOTOS and Estelle.

This book is written to address the two issues discussed above: the needs to specify a protocol using an FDT and to verify its correctness in order to uncover specification errors in the early stage of a protocol development process. The readership primarily consists of advanced undergraduate students, postgraduate students, communication software developers, telecommunication engineers, EDP managers, researchers and software engineers. It is intended as an advanced undergraduate or postgraduate text book, and a reference for communication protocol professionals.

### Contents

#### Part I: Protocol Specification

**Chapter 1** gives an introduction to communication protocols, the ISO reference model, layering concepts, and protocol specification. Then the specification of the Sliding Window and the ISO ACSE protocols are described.

**Chapter 2** describes the need for formally specifying communication protocols, gives an introduction to Formal Description Techniques and outlines the different types of FDTs.

**Chapter 3** gives a detailed description of Estelle and the uses of Estelle in specifying the Sliding Window and the ACSE protocols.

**Chapter 4** presents LOTOS and the uses of LOTOS in specifying the Sliding Window and the ACSE protocols.

**Chapter 5** describes SDL and the uses of SDL in specifying the Sliding Window and the ACSE protocols.

#### Part II: Protocol Verification

**Chapter 6** gives an introduction to protocol verification, the most commonly used techniques and the major difficulty - the state space explosion problem.

**Chapter 7** presents the principles of Estelle verification. This is followed by a survey of some of the Estelle verification methods.

**Chapter 8** describes an approach based on Numerical Petri Nets (NPNs) for modelling Estelle.

**Chapter 9** describes a software environment, EVEN (Estelle Verification Environment using NPNs), which facilitates the automatic verification of communication protocols specified in Estelle.

**Chapter 10** presents a method to address the well-known state space explosion problem and its algorithm which is implemented in EVEN.

## PREFACE

**Chapter 11** describes the results of using EVEN for the verifications of the Estelle specifications of the Sliding Window and the ISO ACSE protocols.

RICHARD LAI

AJIN JIRACHIEFPATTANA

## Acknowledgments

We would like to thank NIST of the US Department of Commerce for making Pet and Dingo and Telecom Australia for making PROTEAN available to us for the work on protocol verification, and the Australian Research Council for the financial support under the grant A49601203 for a part of the work on specification and implementation described in this book.