LOTOSphere: Software Development with LOTOS

LOTOSphere: Software Development with LOTOS

Edited by

Tommaso Bolognesi
CNR, Pisa

Jeroen van de Lagemaat
University of Twente

and

Chris Vissers
University of Twente

SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

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

ISBN 978-1-4613-5924-1 ISBN 978-1-4615-2203-4 (eBook)
DOI 10.1007/978-1-4615-2203-4

Printed on acid-free paper

CONTENTS

PREFACE xiii

Part I INTRODUCTION AND OVERVIEW 1

1 LOTOSPHERE, AN ATTEMPT TOWARDS A DESIGN CULTURE
Chris A. Vissers, Luís Ferreira Pires, Jeroen van de Lagemaat 3
1 Introduction 3
2 Elements of a Design Culture 5
3 The Lotosphere Project 15
4 Lotosphere and the Elements of a Design Culture 16
5 Conclusions 25
REFERENCES 25

2 THE LOTOSPHERE DESIGN METHODOLOGY
Juan Quemada, Arturo Azcorra and Santiago Pavón 29
1 Introduction 29
2 Design by Stepwise Refinement using LOTOS 30
3 Formal Design 36
4 Implementation or Prototyping 45
5 Assessment 48
6 Example: The Two Key System 54
REFERENCES 58

3 DESIGN AND IMPLEMENTATION STRATEGIES
Jeroen Schot, Luís Ferreira Pires 59
1 Introduction 59
2 Structuring of the Implementation Phase 60
3 Correctness Preserving Transformations 63
4 Requirements for Implementation 66
5 Mapping onto Implementation Components 70
6 An Implementation Example 72
7 Conclusions 84

REFERENCES 84

4 GETTING TO USE THE LOTOSPHERE INTEGRATED TOOL ENVIRONMENT (LITE)

José A. Mañas 87

1 Introduction 87

2 A Sample Specification: The Case of NIM 89

3 Getting the Specification Right 91

4 Getting the Right Specification 92

5 Realizing the Specification 102

6 *lite* and Minilite Components 104

7 Conclusions 106

REFERENCES 107

5 LOTOS INDUSTRIAL APPLICATIONS

Edwin Wiedmer 109

1 Introduction 109

2 LOTOS Application: ISDN and Mini-Mail 110

3 LOTOS Application: OSI Transaction Processing 113

4 Experience with Methods and Tools 114

5 Some Industrial Users of LOTOS 117

6 Conclusion 119

REFERENCES 119

6 APPLYING LOTOS TO OSI APPLICATION LAYER PROTOCOLS

Robert E. Booth 121

1 Introduction 121

2 The OSI Context 123

3 OSI Perspectives on the Lotosphere Methodology 124

4 A Hybrid Development Trajectory 129

5 Achievements 131

6 Conclusions 132

REFERENCES 133

Part II SPECIFICATION AND TRANSFORMATION 135

7 LOTOS SPECIFICATION STYLE FOR OSI

Ken J. Turner, Marten van Sinderen 137

1 Introduction 137

2 Specification Elements for OSI Services 139

3 Specification Elements for OSI Protocols 149

4 Conclusion 158
REFERENCES 158

8 CORRECTNESS PRESERVING TRANSFORMATIONS FOR THE EARLY PHASES OF SOFTWARE DEVELOPMENT
Tommaso Bolognesi, David De Frutos, Rom Langerak, Diego Latella 161
1 Introduction 161
2 The running example: a Programmable Sound Sequencer 162
3 Gate splitting 164
4 Inverse expansion 168
5 Splitting processes 173
6 Regrouping parallel processes 175
7 Conclusions 179
REFERENCES 180

9 CORRECTNESS PRESERVING TRANSFORMATIONS FOR THE LATE PHASES OF SOFTWARE DEVELOPMENT
Alessandro Fantechi, B. Mekhanet, Elie Najm, P. Cunha, J. Queiroz 181
1 Introduction 181
2 Transformations applicable to the late phases of the design trajectory 182
3 Description of Museum Security System 188
4 A 3-way to 2-way rendez-vous transformation 189
5 LOTOS to LOTOMATON phase 193
6 Realisation phase 195
7 Discussion and Conclusion 198
REFERENCES 198

10 A CASE STUDY ON PROTOCOL DESIGN
Jean-Pierre Courtiat, Djamel-Eddine Saïdouni 201
1 Introduction 201
2 Specification of the user's requirements 202
3 The design trajectory 205
4 Assessment 213
5 Conclusion 216
REFERENCES 217

Part III ANALYSIS 219

11 EXECUTING LOTOS SPECIFICATIONS: THE SMILE TOOL

Henk Eertink 221

1 Introduction 221

2 Symbolic simulation 221

3 Symbolic simulation in SMILE 226

4 Functionality of SMILE 228

5 Advanced functionalities of SMILE 231

6 Implementation 233

7 Future developments and availability 233

REFERENCES 234

12 A PRAGMATIC APPROACH TO VERIFICATION, VALIDATION AND COMPILATION

Tomas de Miguel, Arturo Azcorra, Juan Quemada and José A. Mañas 235

1 Introduction 235

2 Validation 236

3 Verification 243

4 Compilation 247

5 Conclusions 252

REFERENCES 253

13 AN EXERCISE IN PROTOCOL VERIFICATION

Stefania Gnesi, Eric Madelaine, Gioia Ristori 255

1 Introduction 255

2 Behaviour Oriented Style versus Basic Lotos 257

3 Verification tools in lite 259

4 A Secure Datagram Protocol 262

5 Conclusion 277

REFERENCES 277

14 A TOOL FOR CHECKING ADT COMPLETENESS AND CONSISTENCY

Dietmar Wolz 281

1 Introduction 281

2 Algebraic Specifications with Conditional Equations 282

3 Criteria for the applicability of narrowing and rewriting 284

4 Example for data type checking 288

5 Implementation, limitations and performance 290

6 Relation to Persistency and Completion 291

7 Conclusion 292
REFERENCES 293

15 DERIVING TESTS FROM LOTOS SPECIFICATIONS
Clazien D. Wezeman 295
1 Introduction 295
2 Notation 296
3 The CO-OP method and LOTOS 297
4 Concise presentations of Compulsory and Options 301
5 Initial behaviour of a tester 303
6 Subsequent tester behaviour. 306
7 The alternative CO-OP method. 308
8 The full LOTOS CO-OP method 309
9 Conclusions 313
REFERENCES 314

Part IV IMPLEMENTATION 317

16 THE COLOS COMPILER
Ken Warkentyne and Eric Dubuis 319
1 Introduction 319
2 Functional Description 320
3 Restrictions and Annotations 321
4 Some Hints on Using COLOS 327
5 Future Work 329
REFERENCES 329
APPENDIX A The α Predicate 330
A.1 The β Predicate 330

17 TP PROTOCOL FROM SPECIFICATION TO IMPLEMENTATION
Ing Widya, Gert-Jan van der Heijden, François Juillot 333
1 Introduction 333
2 Transaction Processing 334
3 The Design Methodology 335
4 TP Design Trajectory 336
5 First Design Cycle 337
6 Second Design Cycle 345
7 Status of the Specifications 345
8 Experiences 346
9 Conclusions 347

REFERENCES 348

18 REALIZATION OF CCR IN C
Val Jones 349
1 Introduction 349
2 The Lotosphere Development Trajectory 350
3 Compilation using TOPO 351
4 Transformation to "high-level" C 355
5 Comparison of the two approaches 364
6 Conclusions 366
REFERENCES 367

19 ALTO: AN INTERACTIVE TRANSFORMATION TOOL FOR LOTOS AND LOTOMATON
Elie Najm, A. Serhrouchni, A. Lakas, Eric. Madelaine, Robert. de Simone 369
1 Introduction 369
2 The LOTOMATON Framework 370
3 The ALTO Tool 384
4 Conclusion 387
REFERENCES 388

Part V GRAPHICAL LOTOS 389

20 G-LOTOS: A GRAPHICAL LANGUAGE FOR CONCURRENT SYSTEMS
Tommaso Bolognesi, Elie Najm, Paul A.J. Tilanus 391
1 Introduction 391
2 Processes and actions 392
3 Behaviour diagrams 395
4 Data 411
5 Complete specification 414
6 A movie interpretation 424
7 LOTOS specification of the Museum Control System 428
8 Textual LOTOS to G-LOTOS conversion table 430
9 Conclusions 435
REFERENCES 436

21 GLOW 3.0 - A GRAPHICAL LOTOS BROWSER
Tommaso Bolognesi, Maurizio Caneve, Elena Salvatori 439
1 Introduction 439
2 The user's view 439
3 Tool architecture 446

4 Conclusions 450

REFERENCES 450

Part VI LOTOS ENHANCEMENTS 451

22 ENHANCEMENTS OF LOTOS

Ed Brinksma, George Leih 453

1 Introduction 453

2 Design Principles of Modular LOTOS 454

3 Data Revisions 456

4 Modularization Concepts 460

5 Discussion 464

REFERENCES 465

23 DATA SPECIFICATIONS IN MODULAR LOTOS

Rudolph Roth, Jan de Meer, Silke Storp 467

1 Introduction 467

2 Data Specification Case Study 467

3 Modular LOTOS Enhancements for Data Specifications 470

4 Conclusion 478

REFERENCES 479

INDEX 481

PREFACE

Introduction

Although LOTOS (Language of Temporal Ordering Specification) has become an international standard in 1989 (ISO standard IS8807), the application of preliminary versions of the language to communication services and protocols of the ISO-OSI family (Open Systems Interconnection) dates back to around 1984. Thus, when the 'Lotosphere' project started, in 1989, a five-year experience of writing LOTOS specifications of varying sizes was available, ranging from the usually small examples presented at the IFIP WG6.1 International Conferences on Protocol Specification, Testing, and Verification, to the complex specifications of OSI protocols and services produced within ISO Working Groups. The production of reference documents providing abstract and unambiguous descriptions of these protocol and service standards was indeed one of the primary motivations, in ISO, for the development of LOTOS. However, due to the experiences gained in those five years, with the language spreading outside the ISO community, it soon became clear that much higher advantages than the pure production of standard reference documents were to be expected from the investment in, and usage of, formal description techniques such as LOTOS. At the same time, it also became clear that these advantages could not be achieved at a low price, and that much work remained to be done beyond the formal definition of the language.

The Lotosphere project

The Lotosphere Project, partially funded by the Commission of the European Communities under the ESPRIT II Programme (Project N. 2304, 1989-92), has represented the most substantial international effort centered on the LOTOS specification language, with a budget of 11 MEcu, having involved 16 European partners for a total of about 96 man/years, including all the major European contributors to the design of the language and those that had been more active in the development of prototype LOTOS tools.

The main goal of Lotosphere was to pull LOTOS out of the ISO palace, and throw it into software engineering practice: LOTOS, with its associated theory, was to become a vehicle for efficient, yet formally based, industrial software specification, design, verification, implementation, and testing. More precisely, the objectives of the Lotosphere project were:

- defining a comprehensive LOTOS based design and implementation methodology,

- developing a coherent, integrated, open-ended toolset supporting this methodology,
- carrying out and assessing some realistic industrial product developments using the above methodology and toolset.

The strategy of forming the consortium was guided by the objective to achieve fast transfer from the academic to the industrial world. To this end, a combination of an industrial partner interested in the application of LOTOS, and an academic partner already knowledgeable in LOTOS, was formed in each participating country.

The five industrial partners in the consortium were: Alcatel Standard Electrica (E), Ascom Tech, British Telecommunications (UK), Océ Nederland (NL), and SYSECA Logiciel (F).

The six research centres in the consortium were: C.N.R.-CNUCE (I), Gezellschaft für Mathematik und Datenverarbeitung (D), C.N.R-Instituto Elaborazione Informazione (I), Institut National de Recherche en Informatique et en Automatique (F), Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS (F), PTT-Research Neher Laboratories (NL).

The four universities in the consortium were: Technische Universität Berlin (D), Universidad Politécnica de Madrid (E), University of Stirling (UK) and University of Twente (NL).

Lotosphere has terminated after three years of activity, leaving the clear impression to many of its participants that a train had been forced to stop shortly after it had reached its full speed. Furthermore, due to time constraints, the above mentioned main streams of activity, which would most naturally be conceived as sequential, had to be carried on by the corresponding working groups mainly in parallel, with more overhead and less effectiveness in the feedback loops linking 'providers' and 'consumers' of results. For example, industrial-level applications of elements of the methodology, and of their associated tools, could not be started before the industrial partners had reached sufficient expertise in the usage of the LOTOS language itself; we believe that, after three years, further feedback of consolidated experience from the industrial partners would have been increasingly beneficial for further tuning the methodology and adapting it to specific application areas and industrial needs.

Purpose of this book

The purpose of this book is to illustrate the variety of results that were achieved in Lotosphere with respect to the above, closely concatenated objectives. There is no doubt that Lotosphere made substantial contributions to the state of the art in LOTOS theory and practice in all three directions, and we believe that the material collected in this book reflects these achievements in a fairly balanced way. Besides the initial overviews, the reader can indeed find chapters dealing with methodological aspects, providing guidelines for specification, transformation, verification and implementation, other chapters dealing with tools, and some others describing industrial applications.

Structure and contents of this book

We must recall that Lotosphere has produced a large number of thick, sometimes multi-volume final deliverables, grouped according to the three main objectives of the project, some of which could be edited into self-standing books themselves. Furthermore, the LotoSphere library contains over 1500 technical reports. This book can obviously provide only a partial coverage of this material. In fact, rather than directly selecting from this huge documentation (which is largely available in electronic form), we have mainly collected here revisions of the papers that have been prepared by several authors explicitly for presenting the results of the project to the scientific and industrial communities. Early versions of most of these chapters have been presented at the Third LotoSphere Workshop & Seminar held at Pisa on September 14-16, 1992. While in principle the reader might have expected a three-part book precisely reflecting the three Lotosphere goals, we have preferred a presentation structure that perhaps suggests some fragmentation in the project results. We prefer this solution to giving an excessively optimistic impression of unity and coherence.

The results presented in this book

The making of a new object, such as a software system, is by definition a creative process, and there can not exist any methodology which automatically (or 'automagically'?) compensates for the lack of creativity and ingenuity of specifiers, designers and implementors.

Therefor the fully reliable, fully automated, highly efficient, LOTOS based software development methodology generally applicable to concurrent, distributed and reactive systems, can not be found in this book, nor in the Lotosphere documentation. Of course we are in good company, since this (fully naive) dream has not been realized for any other formal language.

In line with the actual meaning of word 'methodology' – a body of methods – what can be more realistically done, and has indeed been done in LotoSphere, is to put together a conceptual framework, that is, a number of inter-related conceptual tools, such as notions of specification styles, general patterns of design step, transformation and verification techniques, that support system developers by providing a rich panoply of useful conceptual tools along the specification-to-implementation trajectory. Such conceptual tools are less stringent than a precisely defined, fully deterministic recipe, and leave much room for creativity and ingenuity. Thus, the success in applying the Lotosphere methodology is still crucially dependent on the developers' skills in exploiting these conceptual tools and methods, and in combining them in effective ways.

We believe that the degree of freedom left by the methodology is essential, and have indeed experienced in the project, during the early industrial-level experiences with it, that a rigid, recipe-like formulation of the development steps (in particular, a precisely defined sequencing of specification styles) may, depending on the system under development, introduce inefficiency and even lead to unnecessary work.

The issue of the relation between the methodology and the type of system to be developed can not be mentioned only cursorily. Clearly there exists some tradeoff between the width of an application area and the effectiveness of a corresponding methodology: the narrower the application area, the highest the (potential) effectiveness of the methods. The application area primarily addressed by Lotosphere, that of communication systems, is relatively wide, and the systems found in it exhibite a rich variety of behavioural aspects concerned with concurrency and data manipulation. Correspondingly, the LotoSphere methodology is quite general, and is likely to be applicable to other areas as well. We believe that only by selecting a very specific and sufficiently narrow application area, could one attempt to devise a more stringent methodology where less freedom and more guidance is offered to system developers. In this respect, we look with much interest to some of the new and sufficiently specialized areas that seem to be emerging today for LOTOS application, such as digital logic or neural networks.

This book has at least one ideal predecessor, published in 1989, presenting the results of the SEDOS project ESPRIT I number 410, which was a large ESPRIT Project, partly devoted to LOTOS In many ways Lotosphere can be considered as a fruit originated from that initial 'seed'. For example, several tool developments in Lotosphere have built upon experiences conducted within SEDOS.

However, a substantial change of perspective has taken place in Lotosphere, which allows us to summarize the novelty offered by the present book with respect to its predecessor by a single keyword: transformation. In the SEDOS book a LOTOS specification is a static object, produced from scratch, perhaps in a burst of inspiration, much in the same way as a piece of art is created. In this LotoSphere book a LOTOS specification appears as dynamic object, which evolves along a trajectory of transformations and refinements for progressively achieving the requirements that originate the development. The evolution of this object can be driven and assisted by the conceptual tools offered by the methodology, and supported by their associated software tools. Thus, while the SEDOS book contained several examples of LOTOS specifications, we provide here several concrete examples of specification transformations, that represent in themselves a good basis for learning-by-example, and can be appreciated to a large extent even without going through the abstract descriptions of the various methods and tools.

How this book is organised.

In Part 1 - Introduction and Overviews - Chapter 1 provides an overview of the objectives and results of Lotosphere, and relates them with the general needs of a design culture in software companies, Chapters 2 and 3 describe various aspects of the Lotosphere design and implementation methodology, Chapter 4 gives a quick introduction to the whole tool environment (LITE), and Chapters 5 and 6 offer an overview of some industrial applications carried on during the project.

In Part 2 - Specification and Transformation - Chapter 7 introduces a discipline for building LOTOS specifications of communication systems of the OSI family, Chapters

8 and 9 introduce techniques for transforming generic specifications, and Chapter 10 illustrates in detail some transformations of a realistic OSI-like communication service.

In Part 3 - Analysis - Chapter 11 describes SMILE, a tool offered by LITE for the analysis by execution of LOTOS specifications, Chapter 12 illustrates LOLA and TOPO which are tools that support key phases of the software life-cycle , Chapter 13 illustrates some verification approaches and tools, by applying them to the LOTOS specification of a datagram protocol, Chapter 14 introduces a tool of the LITE environment that supports its users in analysing LOTOS specifications of abstract data types, and Chapter 15 introduces a tool for deriving tests from LOTOS specifications.

In Part 4 - Implementation - Chapter 16 introduces one of the LOTOS compilers included in LITE, Chapters 17 and 18 describe the experiences of implementing, respectively, the OSI-TP and OSI-CCR protocols, starting from their LOTOS specifications and using LITE tools, and Chapter 19 introduces a model and technique for transforming LOTOS specifications into implementation-oriented specifications, and an associated tool.

In Part 5 - Graphical LOTOS - we have included for completeness, as Chapter 20, a paper that has already appeared in Computer Networks and ISDN Systems, Vol. 26 (1994), which introduces the graphical syntax of the language but may serve also as a general introduction to LOTOS for the novice reader. Chapter 21 describes the G-LOTOS browser of LITE.

In Part 6 - Lotos Enhancements - Chapters 22 and 23 describe the main enhancements envisaged for LOTOS within LotoSphere.

Besides being grouped into six parts, the chapters of this book are interconnected in several useful ways. For instance some tools and transformations are discussed, from different viewpoints, in different chapters. This may offer alternative reading policies, and can be traced by consulting the index at the end of the book.

It is fair to say that the pre-requisites for reading the book vary across its chapters. If we exclude Part 1, a good knowledge of LOTOS is required for appreciating all the other chapters; the introduction to (LOTOS and) G-LOTOS can compensate only in part for the lack of experience with the language. More generally, knowledge of the basic concepts in process algebra, such as structural operational semantics and behavioural equivalences, is certainly beneficial for fully appreciating the proposed techniques and tools. The introduction to these concepts, discussed in hundreds of papers that have appeared in the last 15 years, is out of the scope of this book. Nevertheless we believe that, in the absence of these theoretical bases, the reader can still follow most of what is presented about the methodology and the tool functionalities. There has been an effort, in Lotosphere, for selecting and exploiting those ingredients of the LOTOS theory that best lend themselves to intuitive understanding and practical usage. An example is the use made of the notion of 'testing equivalence'.

Finally a few words about the references found at the end of each chapter. We have avoided pointers to Lotosphere technical reports, and have avoided duplication by collecting the most general and useful references at the end of this Preface. They are grouped into three lists: LOTOS language, Lotosphere deliverables and other related publications.

References on LOTOS language

[G01] T. Bolognesi, E. Brinksma, 'Introduction to the ISO Specification Language LOTOS', Computer Networks and ISDN Systems 14 (1987), pp. 25-59.

[G02] C. A. Vissers, G. Scollo, M. van Sinderen, 'Architecture and Specification Style in Formal Descriptions of Distributed Systems', in: S. Aggarwal and K. Sabnani (eds.), Protocol Specification, Testing, and Verification, VIII, North-Holland, 1988, pp. 189-204.

[G03] E. Brinksma (ed.), ISO - Information Processing Systems - LOTOS - A Formal Description Technique Based on th Temporal Ordering of Observational Behaviour, IS8807, 1989.

[G04] P. van Eijk, C. A. Vissers, M. Diaz (eds.), The Formal Description Technique LOTOS - Results of the ESPRIT/SEDOS Project, North-Holland, 1989.

[G05] A.J.Tocher, 'LOTOS and the Formal Specification of Communication Standards: An Example', in: P.N.Scharbach (ed.). Formal Methods: Theory and Practice, BSP Professional Books, 1989.

[G06] L.Logrippo, M.Faci, M.Haj-Hussein, 'An Introduction to LOTOS: Learning by Examples', Computer Networks and ISDN Systems 23 (1992), pp. 325-342.

[G07] K. J. Turner (ed.), Using Formal Description Techniques - An Introduction to ESTELLE, LOTOS and SDL, Wiley, 1993.

[G08] L. Ferreira Pires and C. A. Vissers. Overview of the Lotosphere design methodology. In Commission of the European Communities, DG XIII: Telecommunications, Information Industries and Innovation, editor, *ESPRIT'90 – Conference Proceedings*, pages 371–387. Kluwer Academic Publishers, 1990.

[G09] ISO TC 97. *Information Processing Systems – Open Systems Interconnection – Basic Reference Model – IS 7498.* ISO, 1982.

[G10] E. Brinksma, G. Scollo, and C. Steenbergen. LOTOS Specifications, their Implementation and their Tests. In *Sixth International Workshop on Protocol Specification, Testing and Verification*, Montreal, June 1986.

[G11] P. van Eijk, H. Kremer, and M. van Sinderen. On the use of specification styles for automated protocol implementation from LOTOS to C. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification Testing and Verification, X*, pages 157–168. North-Holland, 1990.

[G12] P. van Eijk and J. Schot. An exercise in protocol synthesis. In K. Parker and G. Rose, editors, *Formal Description Techniques, IV*, pages 117–131. North-Holland, 1992.

[G13] C. A. Vissers, G. Scollo, M. van Sinderen, and E. Brinksma. Specification styles in distributed systems design and verification. *Theoretical Computer Science 89*, pages 179–206, 1991.

[G14] ISO. (1989) *Information Processing Systems – Open Systems Interconnection – Formal Description in LOTOS of the Connection-Oriented Session Service*, ISO/IEC TR 9571, ISO, Geneva.

[G15] ISO. (1989) *Information Processing Systems – Open Systems Interconnection – Formal Description in LOTOS of the Connection-Oriented Session Protocol*, ISO/IEC TR 9572, ISO, Geneva.

[G16] ISO. (1992) *Information Processing Systems – Open Systems Interconnection – Formal Description in LOTOS of the Connection-Oriented Transport Service*, ISO/IEC TR 10023, ISO, Geneva.

[G17] ISO. (1992) *Information Processing Systems – Open Systems Interconnection – Formal Description in LOTOS of the Connection-Oriented Transport Protocol*, ISO/IEC TR 10024, ISO, Geneva.

[G18] ISO/IEC: *Information Processing Systems – Open Systems Interconnection – Guidelines for the Application of Estelle, LOTOS and SDL*, ISO/IEC TR 10167, International Organisation for Standardisation, Geneva, 1991.

Lotosphere deliverables

[D01] Task 1.1/3.3. *The Lotosphere Design Methodology: Basic Concepts.* ESPRIT II Lotosphere Project, March 1992. Lo/WP1/T1.1/N0045/V04, Final Deliverable.

[D02] Task 1.1/3.3. *The Lotosphere Design Methodology: Guidelines.* ESPRIT II Lotosphere Project, March 1992. Lo/WP1/T1.1/N0044/V04, Final Deliverable.

[D03] Task 1.1/3.3. ESPRIT II Lotosphere Project, March 1992. Lo/WP1/T1.1/N0046/-V03, Final Deliverable.

[D04] Task 1.1/3.3. *The Lotosphere Design Methodology: Experience in Industrial Environment.* ESPRIT II Lotosphere Project, March 1992. Lo/WP1/T1.1/N0047/V04, Final Deliverable.

[D05] Task 1.2. *Catalogue of LOTOS Correctness Preserving Transformations.* ESPRIT II Lotosphere Project, April 1992. Lo/WP1/T1.2/N0045/V03, Final Deliverable.

[D06] Task 1.3. *A Theorethical and Methodological Framework to Conformance Testing.* ESPRIT II Lotosphere Project, November 1990. Lo/WP1/T1.3/N0046/V05, Final Deliverable.

[D07] Task 1.4. *Task 1.4 Deliverable on Language Enhancements.* ESPRIT II Lotosphere Project, April 1992. Lo/WP1/T1.4/N0016/V03.

[D08] Task 1.4. *G-LOTOS: DAM1 Draft Addendum 1 to LOTOS.* ESPRIT II Lotosphere Project, March 1992. Lo/WP1/T1.4/N0022, Final Deliverable.

[D09] Lite Metatools evaluation Lo/WP2/N0009/V01

[D10] WP2. *MiniLite – The Components.* ESPRIT II Lotosphere Project, April 1992. Lo/WP2/N0042/V08, Final Deliverable.

[D11] WP2. *URD: User Requirements Document.* ESPRIT II Lotosphere Project, November 1990. Lo/WP2/N0013/V04, Final Deliverable.

[D12] WP2. *ADD: Architectural Design Document.* ESPRIT II Lotosphere Project, March 1992. Lo/WP2/N0025/V06, Final Deliverable.

[D13] WP2. *Test Plan.* ESPRIT II Lotosphere Project, April 1992.

[D14] WP2. *Lite User Manual.* ESPRIT II Lotosphere Project, March 1992. Lo/WP2/-N0034/V08, Final Deliverable.

[D15] Lite Guided Tour Lo/WP2/T2.2/CPR/N0023/V01

[D16] FC2 Common Format Lo/WP2/T2.1/INRIA/N0026/V02

[D17] WP2. *The Lite Common Representation.* ESPRIT II Lotosphere Project, April 1992. Lo/WP2/T2.1/UT/N0009/V10, Final Deliverable.

[D18] Common Report Abstract Syntax Lo/WP2/T2 1/INRIA/N0018/V06

[D19] The Unix structure of Lite-v2.0 Lo/WP2/T2 2/ASCOM/N0018/V01

[D20] How to Build Minilite Lo/WP2/T2 2/ASCOM/N0021/V01

[D21] WP2. *Lite User Interface (LUIS): Detailed Design Document.* ESPRIT II Lotosphere Project, September 1991. Lo/WP2/T2.2/UPM/N0016/V03, Final Deliverable.

[D22] Basic LOTOS Algebra Lo/WP2/T2 1/INRIA/N0014/V01

[D23] Algorithm LOTOS to Automata and Ports Lo/WP2/T2 2/ASCOM/N0003/V01

[D24] The "Generalized Regrouping Parallel Transformation" Tool: Detailed Design Document Lo/WP2/T2 2/CPR/N0021/V01

[D25] The "Regrouping Parallel Processes Transformation" Tool: Functionality, Decription and Implementation Lo/WP2/T2 2/CPR/N0005/V02

[D26] The "Bipartition of Functionality Transformation" Tool: Detailed Design Document Lo/WP2/T2 2/CPR/N0017/V02

[D27] The "Bipartition of Functionality Transformation" Tool: Functionality, Decription and Implementation Lo/WP2/T2 2/CPR/N0012/V03

[D28] Batch Transformation Tool Detailed Design Document Lo/WP2/T2 2/INRIA/-N0008/V02

[D29] The Auto front-end: theoretical basis Lo/WP2/T2 2/INRIA/N0001/V01

[D30] A Verification Environment for LOTOS Lo/WP2/T2 2/CPR/N0011/V01

[D31] Verification Widget Detailed Design Document Lo/WP2/T2 2/INRIA/N0013/V01

[D32] Specification and Verification of a Sliding Window Protocol in LOTOS Lo/WP2/-T2 1/INRIA/N0024/V01

[D33] EFSM to FC2 Detailed Design Document Lo/WP2/T2 2/INRIA/N0011/V01

[D34] AST2CR Detailed Design Document Lo/WP2/T2 2/UPM/N0014/V03

[D35] G-LOTOS-Browser Detailed Design Document Lo/WP2/T2 2/CPR/N0022/V01

[D36] Temporal Logic checker Detailed Design Document Lo/WP2/T2 2/CPR/N0019/-V02

[D37] Theory of Conditional Completion Lo/WP2/T2 2/RNL/N0009/V01

[D38] Completion Tool Detailed Design Document Lo/WP2/T2 2/GMD/N0005/V01

[D39] Browser Detailed Design Document Lo/WP2/T2 2/RNL/N0019/V02

[D40] Gate-sortlist report generator Detailed Design Document Lo/WP2/T2 2/RNL/-N0020/V02

[D41] Data-type report generator Detailed Design Document Lo/WP2/T2 2/TUB/-N0014/V01

[D42] CrossReferences report generator Detailed Design Document Lo/WP2/T2 2/-UPM/N0028/V01

[D43] Process Dependency Report Generator Detailed Design Document Lo/WP2/T2 2/INRIA/N0012/V01

[D44] Data type Evaluation Detailed Design Document Lo/WP2/T2 2/TUB/N0007/-V02

[D45] Narrowing Introduction Lo/WP2/T2 2/TUB/N0003/V01

[D46] Narrowing Detailed Design Document Lo/WP2/T2 2/TUB/N0010/V01

[D47] The LOTOS editor Crie: User Manual Lo/WP2/T2 2/UT/N0020/V01

[D48] Simulator Detailed Design Document Lo/WP2/T2 2/UT/N0010/V01

[D49] Symbolic Execution Lo/WP2/T2 2/UT/N0014/V01

[D50] Toolsupport for Implementation Lo/WP2/T2 2/UPM/N0022/V02

[D51] Behaviour Compiler performance: Synthetic Specifications Lo/WP2/T2 2/UPM/-N0025/V01

[D52] First Performance Results for the Lotosphere compiler Lo/WP2/T2 2/ASCOM/-N0022/V01

[D53] Test tool Detailed Design Document Lo/WP2/T2 2/UPM/N0027/V01

[D54] WP2. *DDD: COOPER.* ESPRIT II Lotosphere Project, April 1992. Lo/WP2/-T2.2/UT/N0022, Final Deliverable.

[D55] Testing ADT specs Lo/WP2/T2 2/UPM/N0024/V01

[D56] Task 3.1. *LOTOS Industrial Applications: Preface to the OSI Deliverables.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.1/N0083/V02, Final Deliverable.

[D57] Task 3.1. *LOTOS Industrial Applications: The OSI Association Control Protocol.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.1/BT/N0042/V02, Final Deliverable.

[D58] Task 3.1. *LOTOS Industrial Applications: Specification and Implementation of TP Protocol.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.1/N0077/V04, Final Deliverable.

[D59] Task 3.1. *LOTOS Industrial Applications: Implementation-oriented (C-oriented) Specification of the OSI CCR Protocol.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.1/UST/N0035/V04, Final Deliverable.

[D60] Task 3.3. *LOTOS Industrial Applications: Report on C realization of CCR.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.3/UST/N0038/V04, Final Deliverable.

[D61] Task 3.2. *LOTOS Industrial Applications: Mini-Mail.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.2/N0068/V02, Final Deliverable.

[D62] Task 3.2. *LOTOS Industrial Applications: ISDN Layer 3.* ESPRIT II Lotosphere Project, March 1992. Lo/WP3/T3.2/N0073/V01, Final Deliverable.

Other sources of information

Smile, the simulator for LOTOS, can be obtained from ftp.cs.utwente.nl directory pub/-src/lotos-tools. On this server postscript versions of most deliverables can be found together with the full text of LOTOS specifications referred to in this book in directory pub/doc/lotos/lotosphere.

lotos@dit.upm.es is a public forum to discuss topics related to LOTOS. Subscribe by sending a request to listserv@dit.upm.es

A LOTOS newsletter is published by Prof. Ken Turner. kjt@compsci.stirling.ac.uk

Acknowledgements

The editors would like to express their high respect and appreciation for all experts that have worked in the project which are too numerous to mention, and all those that struggled with LaTeX to produce documents, deliverables for it and this book. In particular we thank the individual Work Package and Task Managers, whose competence and support has been instrumental for the success of the project.

The authors would like to thank all experts with who they collaborated for their contributions, support and debates which all took place in a pleasant and productive Lotosphere.

The project as a whole would like to express its gratitude to the CEC project officers and reviewers, for their critical, but always constructive, guidance of the project.

We wish to thank Elsevier Science B V for having kindly granted permission to include in this book the paper "G-LOTOS: a graphical language for concurrent systems" (Chapter 20), that has already appeared in Computer Networks and ISDN Systems, Vol 26 (1994), pp 1101-1127