

# Contents

---

<b>1 Introduction</b> .....	<b>1</b>
<b>1.1 Overview</b> .....	<b>2</b>
<b>1.2 Communication Systems</b> .....	<b>5</b>
1.2.1 Structuring Principles .....	6
1.2.2 Architectural Concepts .....	12
<b>1.3 Protocol Engineering</b> .....	<b>14</b>
1.3.1 Life-cycle .....	15
1.3.2 Tool Support .....	17

---

<b>2 Protocol Specification with Formal Description Techniques</b> .....	<b>21</b>
<b>2.1 Informal Specification</b> .....	<b>22</b>
<b>2.2 Formal Description Techniques (FDTs)</b> .....	<b>29</b>
2.2.1 Estelle and SDL .....	31
2.2.2 Numerical Petri Nets .....	44
2.2.3 LOTOS .....	49
2.2.4 Z .....	57
<b>2.3 Comparative Assessment</b> .....	<b>62</b>

---

<b>3 Protocol Specification with Communicating Rule Systems</b> .....	<b>65</b>
<b>3.1 Extended Abstract Syntax Notation One (X-ASN.1)</b> .....	<b>66</b>
3.1.1 Type Definition and Operations .....	68
3.1.2 Type Equivalence .....	82
3.1.3 Encoding Rules .....	83
<b>3.2 Communicating Rule Systems (CRS)</b> .....	<b>86</b>
3.2.1 Rule Systems .....	88
3.2.2 Gates .....	93
3.2.3 Structuring Mechanisms .....	96
3.2.4 Abstract Data Types .....	101
3.2.5 Assessment .....	103

---

<b>3.3 Operational Semantics of CRS</b> .....	105
3.3.1 Labelled Transition Systems (LTSs) .....	105
3.3.2 Basic and Communicating Rule Systems .....	106
3.3.3 Interleaving Semantics .....	117

---

<b>4 Protocol Engineering with Formal Description Techniques</b> .....	125
<b>4.1 Executable FDTs</b> .....	125
4.1.1 Estelle, SDL, and PASS .....	126
4.1.2 Petri Nets .....	129
4.1.3 LOTOS .....	130
4.1.4 Logic interpreters .....	132
<b>4.2 Protocol Engineering Environments</b> .....	136
4.2.1 Validation environments .....	136
4.2.2 Implementation environments .....	140
4.2.3 Testing environments .....	142
4.2.4 Integrated environments .....	146

---

<b>5 Protocol Engineering with Communicating Rule Systems</b> .....	149
<b>5.1 Executable CRS</b> .....	149
5.1.1 Implementation Concepts .....	150
5.1.2 Runtime Environment .....	168
5.1.3 Compilation .....	176
<b>5.2 CRS Integrated Tools Environment</b> .....	196
5.2.1 Architecture .....	197
5.2.2 Kernel .....	200
5.2.3 Shells .....	204
5.2.4 Selected Application Tools .....	211

---

<b>6 Conclusions</b> .....	223
<b>6.1 Summary of Results</b> .....	223
<b>6.2 Implementation Status and Early Experiences</b> .....	225
<b>6.3 Prospects and Future Work</b> .....	227
<b>References</b> .....	229
<b>Index</b> .....	245

# Figures

1. Communication system interconnecting open systems	6
2. Communication system as a collection of subsystems	7
3. Decomposition of communication services	8
4. Layered communication system as defined by the OSI Basic Reference Model	9
5. An (N)-layer in a layered communication system	9
6. Mapping between data units in adjacent layers	11
7. Formal development process	16
8. X.25 protocols	23
9. Use of X.25 to provide the OSI connection-oriented network service	24
10. Network service primitives	24
11. X.25 packets	24
12. X.25 PLP CALL REQUEST and DATA packet formats	26
13. X.25 PLP call setup state diagram	28
14. Time sequence diagram for call setup in the X.25 PLP	28
15. X.25 DTE in Estelle	35
16. Estelle module header and channel definitions for X.25 PLP	35
17. Creating an X_25_DTE in Estelle	36
18. Estelle ("PASCAL") type definitions for network SPs and X.25 PLP packets	36
19. Estelle module body for X_25_DTE (in part)	37
20. X.25 DTE in SDL	40
21. SDL process for X_25_DTE	41
22. SDL type definitions for network SPs and X.25 PLP packets	42
23. High-level Petri net graphical representation	45
24. Modelling different communication modes with Petri nets	47
25. X.25 DTE as Numerical Petri Net	48
26. Representation of agents in a process algebra	50
27. Syntax of behavior expressions in Basic LOTOS	52
28. X.25 DTE in LOTOS	54
29. LOTOS process for X_25_DTE	54
30. LOTOS (ACT ONE) type definitions for network SPs and X.25 PLP packets	55
31. X.25 DTE in Z	59
32. Z operation schema for X_25_DTE	59

---

33. Z type definitions for network SPs and X.25 PLP packets	61
34. Abstract, local and transfer syntax in communication systems	67
35. ASN.1 module definition frame example	69
36. Simple type examples	72
37. Type examples constructed with SEQUENCE	75
38. Type examples constructed with SET	76
39. Type example constructed with SEQUENCE OF	77
40. Type example constructed with SET OF	78
41. Type examples constructed with CHOICE	80
42. Subtype examples	81
43. Basic encoding rules	84
44. Encoding of the identifier field	85
45. CRS network service configuration	87
46. CRS specification of the network service configuration	87
47. Syntax of rule system specifications	89
48. CRS specification of X.25 DTE	89
49. Syntax of rules	91
50. CRS specification of X.25 PLP call setup initiation	91
51. CRS specification of X.25 PLP duplex and expedited data transfer	93
52. Syntax of event expressions	94
53. Syntax of gate specifications	94
54. CRS specification of X.25 PLP gates	95
55. CRS specification of X.25 window mechanism	97
56. Syntax of contexts	98
57. Syntax of views	98
58. Context and view structure of the X.25 PLP specification in CRS	99
59. Contexts and views example for X.25 PLP data transfer specified with CRS	100
60. Syntax of ADT specifications	102
61. Syntax of abstract operations	102
62. CRS specification of ADT DataQueue	103
63. Example of LOTOS semantics expressed as "PROLOG" rules	131
64. Example control clause in a logic program	135
65. Distributed test configuration	146
66. CRS implementation concepts	151
67. X-ASN.1 implementation concepts	154
68. Structure blocks	155

---

69. Structure tree for the X.25 CALL REQUEST packet type	156
70. Structure trees	157
71. Data blocks	157
72. Data tree for an X.25 CALL REQUEST packet	158
73. Data trees	159
74. Rule sets of the CRS inference machine	160
75. WES set of the CRS inference machine	163
76. Run set of the CRS inference machine	163
77. Protocol automaton of the synchronous CRS gate machine	165
78. Protocol automaton of the asynchronous CRS gate machine	168
79. Architecture of the local CRS runtime environment	171
80. Rule system control block	173
81. Rule control block	173
82. Gate control block	175
83. Event control block	175
84. Compiling CRS specifications into executable form	177
85. Example of X-ASN.1 extended "C++" code	178
86. "C++" classes for X-ASN.1 expressions and variables	179
87. Example gate header file	181
88. Example gate source file	182
89. Example rule system header file	184
90. Example rule system source file	185
91. Dependency graph example for conjunctions	190
92. Dependency graph for disjunction	191
93. Example rule functions	192
94. Example ADT header file	194
95. Example ADT source file	195
96. Example configuration source file	196
97. Integrated tools environment	199
98. CRS and X-ASN.1 based kernel	201
99. Rule-based shell	205
100. State-based shell	207
101. Event-based shell	209
102. Behavior simulator	213
103. Test case development tool	216
104. Test case validation tool	220