

Contents

Volume I

Title	i
Contents	ii
Acknowledgements	vii
Declaration	viii
Abstract	ix
Chapter 1: Parallel programming	1
1 Introduction	2
1.1 Thesis outline	2
1.2 Programming is difficult	4
1.3 Parallel programming is very difficult	6
1.4 Procedural programming	9
1.4.1 von Neumann languages	9
1.4.2 Object-oriented languages	12
1.4.3 Constraint-based languages	14
1.4.4 Parallelism in procedural programs	15
1.5 Nonprocedural programming	15
1.5.1 Functional languages	16
1.5.2 Dataflow languages	18
1.5.3 Logic languages	19
1.5.4 Parallelism in nonprocedural programming	21
Chapter 2: Definitive programming and the abstract definitive machine	22
2 Definitive programming and the abstract definitive machine	23
2.1 What is definitive programming ?	23
2.1.1 Systems of definitions can model state-transition systems	23
2.1.2 Definitive notations	27
2.1.3 Definitive programming	28
2.2 The abstract definitive machine	31
2.2.1 The program store, action store and definition store	31
2.2.2 Operation of the adm	34
2.3 Programming the adm	34
2.3.1 Parameters	35
2.3.2 Output	37
2.3.3 Input	38
2.3.4 Singular states and invalid transitions	39
Chapter 3: Implementation issues	41
3 Implementation issues	42
3.1 Handling singular states and invalid transitions in am	42
3.2 Output in am	47
3.3 Input in am	48
3.4 How are definitions stored ?	49
3.5 The rand() function	52
3.6 What does the execution cycle involve ?	53
3.7 Example programs for am	54
Chapter 4: Blocks example	57
4 Blocks example	58
4.1 Specification of the problem	58

4.2	Components of the model	58
4.3	Program development	59
4.4	Proof of correctness	66
4.5	Limitations of the adm	70
Chapter 5: Modelling and simulating concurrent systems		72
5	Modelling and simulating concurrent systems	73
5.1	What is LSD ?	74
5.1.1	Variables	75
5.1.2	Protocols	77
5.2	Behaviour described by an LSD specification	78
5.3	Synchronisation	80
5.3.1	Synchronisation by perception	80
5.3.2	Synchronisation by assumption	82
5.4	LSD specifications simulated on the adm	83
5.4.1	The transformation	85
5.4.2	More elaborate transformation techniques	93
5.5	What does an agent describe ?	96
Chapter 6: Telephone example		99
6	Telephone example	100
6.1	Telephone specification	101
6.2	Synchronisation issues	102
6.3	Transformation of the telephone system	104
6.3.1	The <code>user()</code> entity	104
6.3.2	The <code>telephone()</code> entity	107
6.3.3	The <code>dial()</code> entity	107
6.3.4	The <code>connect()</code> entity	108
6.3.5	The <code>exchange()</code> and <code>environment()</code> entities	109
6.3.6	Instantiations at compile time	109
Chapter 7: Jugs example		111
7	Jugs example	112
7.1	LSD specification	113
7.2	Adm specification	118
Chapter 8: Appraisal and comparison		121
8	Appraisal and comparison	122
8.1	Use of definitions in the adm	122
8.2	Appraisal of definitive programming	123
8.2.1	Interaction	123
8.2.2	Invariants	125
8.2.3	Parallelism	128
8.3	Comparison with procedural and nonprocedural programming	129
8.4	Extensions to the adm	131
8.4.1	Windows	132
8.4.2	Superuser	132
8.4.3	Asynchronous mode of execution	135
8.5	Use of definitions in LSD	137
8.6	Unresolved issues in LSD	137
8.7	Appraisal of the modelling and simulation method	140
8.7.1	Simulation of an LSD specification	140
8.7.2	Analysing the results of simulation	141
8.7.3	Interaction with the simulation	142

8.7.4 Animation of the simulation 143

8.8 A modelling and simulation tool 144

Chapter 9: Conclusion 150

9 Conclusion 151

9.1 Summary 151

9.2 Further research 153

References 155

Volume II

Appendices	163
Appendix 1: Describing concurrent behaviour	164
Appendix 2: Reactive systems	171
Appendix 3: User documentation	176
Appendix 4: Maintenance documentation	182
4 Maintenance documentation	183
4.1 Program execution	183
4.2 Input syntax	183
4.3 Constants	186
4.4 Abstract data types	187
4.5 Structures	188
4.6 Modules	191
4.7 Procedure index	192
4.8 Procedure descriptions	195
Appendix 5: Blocks testing	212
5 Blocks implementation	213
5.1 DoNaLD interface	216
Appendix 6: Door controller testing	220
6.1 The door control program	221
6.2 Execution of the first door control program	222
6.3 Analysis of results	227
6.4 Execution of the second door control program	228
Appendix 7: Telephone testing	231
7.1 The transformed telephone program	232
7.2 Execution of the program	238
7.3 More chance of replacing the receiver	241
7.4 Answering more often than initiating or terminating a call	243
7.5 More calling than terminating of calls	245
7.6 Removal of the slowing down of the user	247
7.7 Allowing the dialler to time out	248
7.8 Dialling quicker	249
7.9 Interactive use of the telephone simulation	250
Appendix 8: Jugs testing	261
8.1 Output of messages	262

8.2 Execution of message program 264

8.3 Output of pictures 266

8.4 Execution of pictures program 267

8.5 Allowing input 270

8.6 Execution of input program 271

8.7 Allowing concurrent activity 275

8.8 Execution of concurrent activity program 276

Appendix 9: Source code 278

Appendix 10: Glossary of terms 322