Contents

1. Introduction 1

Part I. Positioning

2. Multi-Agent Systems 9
2.1 Introduction 9
2.2 What Is an Autonomous Agent? 9
2.2.1 Definitions 10
2.2.2 Autonomy and Embodiment 11
2.2.3 Generic Agent Architectures 12
2.3 Characteristics of MASs 14
2.4 Modeling MASs 15
2.4.1 Objective Coordination 17
2.4.2 Subjective Coordination 19
2.4.3 Emergence 21
2.5 Our Target Class of MASs 21
2.5.1 A Generic Model for an Autonomous Agents' System . 22
2.5.2 A Typical Application: *Gathering Agents* 23
2.6 Implementing MAS Applications 24
2.6.1 Languages for MAS Applications 24
2.6.2 Methodologies for MAS Applications 28
2.6.3 Using Coordination Models and Languages for Designing and Implementing MASs 29

3. Coordination Models and Languages 33
3.1 What Is Coordination? 33
3.2 What Are Coordination Models and Languages? 33
3.2.1 Motivation 33
3.2.2 Key Elements 34
3.3 Data-Driven Coordination Models 35
3.3.1 Linda 36
3.3.2 Linda-Based Models 38
3.3.3 Models Based on Multiset Rewriting 39

3.4 Process-Oriented Coordination Models 41
3.4.1 IWIM 42
3.4.2 Other Approaches 44
3.5 Hybrid Coordination Models 45
3.6 Prerequisites for a Coordination Model and Language 48

Part II. ECM **and Its Instances**

4. The ECM **Coordination Model** 53
4.1 Introduction 53
4.2 Blop 55
4.3 Process 55
4.4 Ports and Connections 55
4.4.1 Port Features 55
4.4.2 Connections 56
4.5 Port Matching 56
4.6 Events 58
4.7 ECM Instances 58

5. The STL **Coordination Language** 61
5.1 Introduction 61
5.2 Blops 63
5.3 Processes 63
5.4 Ports and Connections 64
5.5 Port Matching 66
5.6 Events 66

6. The STL++ **Coordination Language** 69
6.1 Introduction 69
6.1.1 Design Decisions 69
6.1.2 An Overview 71
6.2 Blops 73
6.3 Processes 74
6.4 Ports and Connections 75
6.4.1 Port Features 75
6.4.2 Creation and Destruction of Ports 77
6.4.3 Basic Port Types and Their Connections 78
6.5 Port Matching 80
6.6 Events 81
6.7 A Tutorial Example 82
6.7.1 The Restaurant of Dining Philosophers 83
6.7.2 General Description of the Implementation 83
6.7.3 The Restaurant Blop and the Waiter Agent 85
6.7.4 The Philosophers 85

6.8 Implementation of a Prototype . . . 88
6.8.1 Introduction . . . 88
6.8.2 PT-PVM . . . 88
6.8.3 Concurrency and Object-Orientation Integration . . . 90
6.8.4 Blops and Agents . . . 90
6.8.5 Ports and Port Managers . . . 91
6.8.6 Matching . . . 91
6.8.7 Connection Setup . . . 93
6.8.8 Events . . . 93
6.9 Discussion . . . 93
6.9.1 STL++ as a Coordination Language . . . 94
6.9.2 STL++ for MASs . . . 95

7. The AGENT&CO Coordination Language . . . 97
7.1 Introduction . . . 97
7.2 Blops . . . 97
7.3 Agents . . . 98
7.4 Ports and Connections . . . 99
7.5 Matching . . . 99
7.6 Events . . . 100
7.7 Implementation . . . 100

Part III. Case Studies in STL++

8. Collective Robotics Simulation . . . 105
8.1 Introduction . . . 105
8.2 Global Structure . . . 105
8.3 `init` Agent . . . 107
8.4 Sub-environment Blops . . . 107
8.5 `initSimRobot`Agent, `NewSimRobot_Evt` Event . . . 109
8.6 `SimRobot` Agent . . . 110
8.7 `subEnv` Agent . . . 111
8.8 `taxi` Agent . . . 112

9. Trading System Simulation . . . 115
9.1 Introduction . . . 115
9.2 The *TradeWorld* Blop . . . 116
9.3 The Brokers and the Broker Assistants . . . 116
9.4 The Trade Manager . . . 117
9.5 Transactions . . . 118

10. Conclusion . . . 121

A. Core STL++ Interfaces 125
A.1 World Class 125
A.2 Blop Class 125
A.3 Agent Class 126
A.4 Port Template Classes 126
A.4.1 KK-Stream Ports 127
A.4.2 S-Stream Ports 127
A.4.3 Blackboard Ports 128
A.4.4 Group Ports 129
A.5 Event Class 129
A.6 Condition Classes 129
A.7 Macros and Miscellaneous Functions 130

B. STL Code Example 133

C. Linda, Gamma and Manifold Code Examples 135

Bibliography 137