

A Novel Development Methodology for Cooperative, Distributed Multi-agent Systems

Ebrahim Fahad Al-Hashel

Master of Science in Computer Science (University of Sheffield – UK)

FACULTY OF INFORMATION SCIENCES AND ENGINEERING

UNIVERSITY OF CANBERRA
AUSTRALIA

**This dissertation is submitted in fulfilment of the requirements for
the award of the degree doctor of philosophy**

November 2010

Abstract

The multi-agent systems (MaS) typology can be divided into “independent” and “cooperative” or closed and open respectively. Independent MaS embraces a set of agents linked together by predefined protocols that operate in a closed central control system. The closed system does not include or perform MaS dynamic behaviour; rather, it mainly performs agent team formation processes and promotes agents’ cooperation at runtime. This type of MaS is appropriate for application within fixed procedures that operate in one system boundary. In contrast, the cooperative MaS is an emergent system that has the potential to dynamically, at runtime, search in an open distributed computational environment and subsequently form a team of appropriate agents to achieve the defined goal. The agent cooperation behaviour is a key strength of MaS, which is characterised by agents’ autonomy.

This thesis investigates existing multi-agent system development methodologies: Prometheus, Gaia, MaSE, PASSI and Tropos. The results indicate that these methodologies are engineering an independent MaS focusing on the agent’s internal structure or system architecture through interaction protocols. However, the cooperative MaS development processes are minimally realised in these methodologies and the agent cooperation process is not implicitly addressed.

Further, the research aims to enhance MaS development methodology by proposing a novel development methodology for multi-agent systems (DMMAS) that can guide software practitioners in developing cooperative MaS with the ability to function in large-scale, open, distributed, incremental, heterogeneous systems. It is motivated by distributed architecture for problem solving in domains including military logistics, healthcare, transportation and travel agency systems. The research attempts to transition existing MaS from independent concepts to cooperative concepts.

To model agent autonomous behaviour, the research proposes a new organisational multi-agent systems architecture supported by an ontology-based search model and the agent cooperation, through dynamic team formation process is built on agent

adoptive strategy and *Share Plan* cooperation theory as an important characteristic of DMMAS.

The research has been conducted using design science in information system research method, and the case study research approach. For proof of the concept the research applied DMMAS development methodology on a real world case study “Travel Agency System (TAS)” which served as the motivating problem for the research work. The results are evaluated using a benchmark approach to compare DMMAS performance with the five existing MaS development methodologies.

This thesis makes four main contributions: first, it enhances the agent-based system by providing a new development methodology with an attempt to develop the multi-agent systems current state of the art from independent to cooperative. Secondly, the research presents new multi-agent systems architecture and a methodology on how to design and develop open distributed multi-agent systems. Thirdly, the research proposes how ontology analysis and design can be incorporated in software engineering practice. The research explains how ontology concepts, objects and relationships are identified to build the agent systems domain. Finally, the research introduces a new agent functionality ontology schema for a search to replace the agent name keyword based conventional search. The functionality based ontology approach utilises descriptor based semantics.

The proposed DMMAS design methodology is evaluated against software engineering principles and its strengths and inadequacies investigated. The research achievements are summarised and emerging research questions are outlined for future work.

Acknowledgement

To H.E. Major General Dr. Sheikh / *Mohammed bin Abdulla Al Khalifa*, Minister of State for Defence Affairs. You are an extra ordinary person; your leadership has guided me to achieve this work.

To my supervisor Professor / *Dharmendra Sharma*, thanks for being so kind and so supportive. His guidance is crucial to this thesis, and the will to drive the research forward.

To my brother *Yousif Al-Hashel*, thanks for the support and continues asking.

To my brother in life Brigadier General / *Ali bin Sager Al-Nuaimi*, I appreciate the efforts and the support you offered to me to make this work possible.

To *Jennifer Bradley*, thanks for the effort you contributed to edit this thesis.

To Dr. / *Masoud Mohammadian*, Mr. / *Robert Cox*, Dr. / *Bala Balachandran*, Dr. / *Wanli Ma*, thanks for the encouragements.

To my best friends in Australia *Sisira Adikari*, and *Patrick Collin*, stay in contact I enjoy yours friendships.

Dedication

To my family:

My wife: Anissa Al-Rowaie

My sons: Ahmed and Sufyan

Yours truly, love and patience, yours constant encouragement and support provided a solid result. Being away from you was a tough time, all what I have done was for a better life.

To my parents:

Fahad Al-Hashel and Amina Al-Thawadi

Table of Contents

Chapter 1 Introduction.....	1
1.1 Background and Motivation	1
1.2 Why a New Methodology	3
1.3 Research Aims and Objectives	5
1.4 Research Questions	6
1.5 Hypothesis.....	6
1.6 Research Methodology	7
1.7 Thesis Outline	12
Chapter 2 Agents and Software Engineering: A Review	14
2.1 Introduction.....	14
2.2 Agents 14	
2.2.1 Agent Typology	16
2.2.2 Agent versus Object.....	17
2.2.3 Agent Architecture.....	21
2.3 Multi-agent Systems	25
2.4 Agent Cooperation Concept and Theories.....	27
2.5 Other Collaboration Models	32
2.6 Team Formation Process.....	33
2.7 Agent Negotiation.....	35
2.8 Agent Coordination.....	38
2.9 Ontologies	40
2.10 Ontologies in Software Engineering.....	41
2.11 Software Engineering Development Methodology	44
2.11.1 Waterfall Approach.....	45
2.11.2 Incremental Approach.....	46
2.11.3 Formal Specification Method	47
2.12 Existing Agent-based Development Methodologies	49
2.12.1 Agent-UML.....	51
2.12.2 Prometheus.....	51
2.12.3 PASSI.....	54
2.12.4 Gaia	55
2.12.5 MaSE.....	57

2.12.6	Inadequacies in the Existing Development Methodologies.....	60
2.13	Agent Communication Languages.....	62
2.13.1	Knowledge and Query Manipulation Language (KQML).....	63
2.13.2	FIPA Agent-Communication Language (FIPA-ACL).....	65
2.13.3	Knowledge Interchange Format Language.....	66
2.14	Agent Software Development Platform.....	67
2.14.1	Java Agent Development Environment (JADE).....	68
2.14.2	JACK.....	69
2.14.3	Aglet.....	70
2.14.4	Cougaar.....	71
2.15	Summary.....	73
Chapter 3 A Review of Existing Agent-based Software Engineering		
	Methodologies.....	75
3.1	Introduction.....	75
3.2	Travel Agency System: Motivating Problem	75
3.3	Prometheus, MaSE and Gaia	76
3.3.1	Selection of Prometheus, MaSE, and Gaia	76
3.3.2	Prometheus.....	77
3.3.3	MaSE.....	81
3.3.4	Gaia.....	84
3.4	Comparison of Results.....	88
3.5	Inadequacies in Existing Development Methodologies.....	90
3.6	Summary.....	92
Chapter 4 DMMAS: A Novel Development Methodology for Multi-agent		
	Systems.....	94
4.1	Introduction.....	94
4.2	DMMAS Criteria	94
4.3	DMMAS Applications Domains.....	96
4.4	DMMAS Architecture	98
4.5	System Requirements.....	103
4.5.1	System Requirements: High-level	104
4.5.2	System Requirement: Goals Analysis.....	105
4.6	System Specification.....	109
4.6.1	DMMAS Cooperation Architecture.....	110

4.6.2	Identify the System Organisation Structure.....	113
4.6.3	Ontology Design Consideration.....	118
4.6.4	Ontological Analysis.....	119
4.7	System Architecture Design	128
4.8	Detailed Design.....	145
4.9	Summary.....	158
Chapter 5 Travel Agency System: A Case Study		160
5.1	Introduction.....	160
5.2	Travel Agency System.....	160
5.3	TAS Requirement Analysis	165
5.4	TAS Specification Analysis Phase.....	170
5.5	TAS Architecture Design.....	186
5.6	TAS Detailed Design	199
5.7	Summary.....	216
Chapter 6 Evaluation and Assessment.....		217
6.1	Introduction.....	217
6.2	The Evaluation Framework.....	217
6.2.1	Process Related Criteria.....	218
6.2.2	Technique-Related Criteria.....	221
6.2.3	Model-Related Criteria	224
6.2.4	Supportive-Feature Criteria	226
6.3	Comparison of Concepts.....	228
6.4	DMMAS Strengths and Limitations	231
6.5	Summary.....	235
Chapter 7 Conclusion and Future Work		237
7.1	Main Contributions	239
7.2	Research Questions.....	240
7.3	Limitations	242
7.4	Future Work.....	243
 Appendix		
Appendix A: XML and XML-Schema Selected Code for Goal and Skill-agent....		266
Appendix B: Ontology Selected Code for Skill-agent Functionality.....		269

Appendix C: Goal Execution Plan Database and SQL Statements.....	270
Appendix D: AUML Interaction Diagrams and Notations	272
Appendix E: A List of Publications from Thesis Research	275
Bibliography	276

List of Figures

Figure 1.1: Information system research framework (Hevner et al., 2004).	8
Figure 1.2: Research methodology diagram in context of Henver's model.	10
Figure 1.3: The generate/test cycle (Simon, 1996).	11
Figure 2.1: Part view of an agent typology (Nwana, 1996).	16
Figure 2.2: Two aspect of autonomy (Odell, 2002).	19
Figure 2.3: The BDI architecture.	22
Figure 2.4: Horizontal layer.	23
Figure 2.5: One-pass and two-pass layer.	23
Figure 2.6: Subsumption architecture for robot navigation (Rodney, 1986).	24
Figure 2.7: Cooperation typology (Doran et al., 1996b).	27
Figure 2.8: Waterfall approach lifecycle.	45
Figure 2.9: Evolutionary delivery process.	46
Figure 2.10: Categorisation of AOSE methodologies.	50
Figure 2.11: Prometheus architecture.	52
Figure 2.12: The models and phases of the PASSI methodology.	54
Figure 2.13: Model of the Gaia methodology (Hederson-Sellers and Giorgini, 2005).	56
Figure 2.14: MaSE overview diagram.	58
Figure 2.15: The three layers of the KQML communication language.	64
Figure 2.16: JADE architecture (Bellifemine et al., 2007b).	68
Figure 2.17: Relationship between Prometheus construct and JACK abstract.	70
Figure 2.18: BlackboardService API Internal Concepts (BBN, 2004b).	71
Figure 3.1: Prometheus functionality diagram for TAS.	77
Figure 3.2: Prometheus data coupling.	78
Figure 3.3: Prometheus capability diagram.	79
Figure 3.4: Prometheus interaction diagram.	80
Figure 3.5: TAS role diagram produced during the analysis process.	82
Figure 3.6: Agent class, agent information, and their conversation diagram.	83
Figure 3.7: TAS organisations structure diagram.	85
Figure 3.8: TAS Role schema for BookRoom.	86
Figure 3.9: TAS book a room protocol.	87

Figure 4.1: DMMAS problem characteristics.	98
Figure 4.2: DMMAS Architecture.	99
Figure 4.3: DMMAS architecture in context of components.	100
Figure 4.4: DMMAS development phases and system components.	102
Figure 4.5: System context diagram including stakeholder and system goals.	105
Figure 4.6: Why- Refinement extend the main goal assignment.	106
Figure 4.7: How-refinement extends the main goal object.	107
Figure 4.8: Goal descriptor template.	109
Figure 4.9: DMMAS cooperation architecture.	111
Figure 4.10: System basic organisation structure example.	114
Figure 4.11: <i>Professional-agent</i> goal execution plan.	115
Figure 4.12: Alternative Plan Descriptor.	116
Figure 4.13: Initial system organisational structure example.	116
Figure 4.14: data-used and data-produced model example.	117
Figure 4.15: Domain ontology scope descriptor.	121
Figure 4.16: Domain ontology high-level model.	123
Figure 4.17: Domain and scope ontology descriptor.	124
Figure 4.18: Ontology domain objects diagram.	126
Figure 4.19: <i>Skill-agent</i> transition status.	127
Figure 4.20: <i>Skill-agent</i> architecture example.	130
Figure 4.21: <i>Professional-agents</i> architecture including search component.	131
Figure 4.22: DMMAS ontology notations for design phase.	133
Figure 4.23: Domain ontology class diagram.	135
Figure 4.24: Execution Plan example for Achieve PhD goal focus on “Enrol in PhD course” <i>skill-agent</i>	137
Figure 4.25: Goal execution plan diagram.	138
Figure 4.26: <i>Skill-agent</i> functionalities structure.	139
Figure 4.27: Information and Identification components example for <i>skill-agent</i> "PhDEnrolment".	140
Figure 4.28: <i>Skill-agent</i> functionality architecture.	140
Figure 4.29. Example for <i>skill-agent</i> functionality ontology.	143
Figure 4.30: <i>Skill-agent</i> Search model structure.	145
Figure 4.31: Diagram illustrating notation used in the DMMAS detailed design phase.	146

Figure 4.32: <i>Skill-agent</i> detailed design diagram.	149
Figure 4.33: <i>Professional-agents</i> detailed design diagram.....	151
Figure 4.34: XML Schema for system goals.	153
Figure 4.35: Goal execution plan detailed design.....	154
Figure 4.36: Screenshot for multi-agent systems components database (Achieving PhD example).....	155
Figure 4.37: Example for SQL statement to retrieve user goal execution table.....	156
Figure 4.38: AUML interaction diagram for <i>skill-agent "Enrolment"</i> example.....	157
Figure 5.1: TAS main components.....	161
Figure 5.2: Numbers of skill-agents team in TAS.....	165
Figure 5.3: High level TAS over view including system external entities.....	166
Figure 5.4: TAS context diagram.	167
Figure 5.5: TAS goals diagrams.....	167
Figure 5.6: Goal descriptor for Flight Reservation.	168
Figure 5.7: Goal descriptor for Hotel Booking.	169
Figure 5.8: Goal descriptor for Car Rental.....	170
Figure 5.9: TAS Professional-agent.	171
Figure 5.10: TAS skill-agents.....	171
Figure 5.11: TAS skill-agent team graph.....	172
Figure 5.12: TAS goal execution plan descriptor.....	173
Figure 5.13: TAS alternative plan descriptor.....	174
Figure 5.14: TAS organisation structure.....	175
Figure 5.15: Flight Reservation skill-agent data model.....	176
Figure 5.16: Hotel Booking skill-agent data model.....	177
Figure 5.17: Car Rental skill-agent data model.	178
Figure 5.18: TAS ontology scope.	179
Figure 5.19: Input-status and Output-status within each TAS skill-agent.....	180
Figure 5.20: Domain and scope ontology descriptor for TAS.....	181
Figure 5.21: Flight Reservation skill-agent domain objects diagram.....	182
Figure 5.22: HotelBooking skill-agent domain object diagram.....	183
Figure 5.23: Car Rental skill-agents domain object diagram.	184
Figure 5.24: Flight Reservation skill-agent transition status.	185
Figure 5.25: Hotel Booking skill-agent transition status.	185
Figure 5.26: Car Rental skill-agent transition status.....	186

Figure 5.27: Flight Reservation skill-agent state diagram.	187
Figure 5.28: Hotel Booking skill-agent state diagram.	188
Figure 5.29: Car Rental skill-agent state diagram.	189
Figure 5.30: TAS professional-agent process diagram for flight reservation example.	190
Figure 5.31: TAS skill-agents functionality identification and information components.	192
Figure 5.32: Flight Reservation skill-agent functionality component.	194
Figure 5.33: Hotel Booking skill-agent functionality description component.	195
Figure 5.34: Car Rental skill-agent functionality component.	195
Figure 5.35: Flight Reservation skill-agent functionality ontology classes diagram.	196
Figure 5.36: Hotel Booking skill-agent functionality ontology classes diagram.	197
Figure 5.37: Car Rental skill-agent functionality structure.	198
Figure 5.38: Flight reservation skill-agent detailed design.	200
Figure 5.39: Hotel booking skill-agent detailed design.	201
Figure 5.40: Car rental skill-agent detailed design.	202
Figure 5.41: Professional agent detailed design diagram.	203
Figure 5.42: TAS Goals execution plan detailed design.	204
Figure 5.43: XML Schema for flight goal.	205
Figure 5.44: XML Schema for Hotel goal.	206
Figure 5.45: XML schema for Car goal.	207
Figure 5.46: Skill-agent and transition status database relationship.	210
Figure 5.47: TAS goals, priority, sequence, and team database.	211
Figure 5.48: Interaction diagram for flight reservation skill-agent.	212
Figure 5.49: Interaction diagram for hotel reservation skill-agent.	213
Figure 5.50: Interaction diagram for car rental skill-agent.	213
Figure 5.51: Interaction diagram for TAS professional-agent and functionality search.	214
Figure 5.52: XML selected code for TAS Skill-agent functionality.	215
Figure 6.1: Structure of the evaluation framework.	218
Figure 6.2: Comparison between DMMAS and rival development methodologies on software life cycle completeness.	232
Figure 6.3: Graph classification of development methodologies.	234

List of Tables

Table 2.1: Illustration of cooperation in AOSE.....	61
Table 2.2: FIPA ACL Message Parameters.....	66
Table 3.1: Development phase details assessment.....	88
Table 3.2: Presents the measure of an agent concept that each methodology supports.	88
Table 3.3: Shows the scale of the modelling criteria within each methodology.....	89
Table 3.4: Compares the properties of the methodologies.....	89
Table 3.5: Illustrates the available activities in each development phase.....	89
Table 3.6: Types of system domain for each methodology.....	90
Table 3.7: Software development tools support.....	90
Table 3.8: Inadequacies in methodologies inadequacies with respect to the TAS case study.....	92
Table 4.1: DMMAS analysis phase outputs.....	128
Table 4.2: DMMAS semiformal set.....	136
Table 4.3: Gaia semiformal set (Wooldridge et al., 2000).....	136
Table 4.4: Detailed design phase components.....	158
Table 5.1: TAS facilities.....	163
Table 6.1: Comparison of Process-related criteria.....	220
Table 6.2: Evaluation of DMMAS Steps and Techniques.....	222
Table 6.3: Evaluation of DMMAS steps and Techniques.....	223
Table 6.4: Comparison regarding steps and usability of techniques.....	225
Table 6.5: Comparison regarding model related criteria.....	227
Table 6.6: Comparison regarding concepts.....	229
Table 6.7: Comparison regarding supportive related criteria.....	230
Table 6.8: Measurement scale for assessing multi-agent systems type.....	233

List of Acronyms

The following acronyms and abbreviations of standards phrases are used throughout the thesis:

ARPA	Advance Research Project Agency
AbSE	Agent-based Software Engineering
AbS	Agent-based System
AOSE	Agent-Oriented Software Engineering
AOS	Agent-Oriented Software
AI	Artificial Intelligence
BDI	Belief, Desire, and Intention agent architecture
DARPA	Defence Advance Research Projects Agency
DFI	Design Fabricator Interpreter
DMMAS	Development Methodology for Multi-agent Systems
DAI	Distributed Artificial Intelligence
DPS	Distributed Problem Solving
DPS	Distributed Problem Solving
ERD	Entity Relationship Diagram
EVO	Evolutionary Delivery
SXML	Extensible Markup Language Schema
XML	Extensible Markup Language
EP	Extreme Programming
FIPA	Foundation for intelligent physical agents (FIPA)
HERM	High Entity Relationship Diagram
IS	Information System
JVM	Java Virtual Machine
JAD	Joint Application Development
JAD	Joint Application Development
KIF	Knowledge Interchange Formalism
KQML	Knowledge Query Manipulation Language
KSE	Knowledge Sharing Effort
KSE	Knowledge Sharing Effort
MaS	Multi-agent Systems

NII	National Information Infrastructure
OMT	Object Modelling Technique
OOSE	Object-oriented Software Engineering
OWL	Ontology Web Language
PTA	Planned Team Activity
PASSI	Process for Agent Societies Specification and Implementation
Pa	Professional-agent
PDT	Prometheus Development Tools
RAD	Rapid Application Development
RAD	Rapid Application Development
RDFS	Resources Definition Framework Schema
RDF	Resources Definition Framework
SP	Shared Plans Theory
Sa	Skill-agent
SDM	Software Development Methodology
SAT	Speech Act Theory
SQL	Structured Query Language
SDLC	System Development Life Cycle
TFP	Team Formation Process
TAS	Travel Agency System
DOD	United States Department of Defence
W3C	World Wide Web