LNAI 3904

Matteo Baldoni
Ulle Endriss
Andrea Omicini
Paolo Torroni (Eds.)

Declarative Agent Languages and Technologies III

Third International Workshop, DALT 2005
Utrecht, The Netherlands, July 2005
Selected and Revised Papers

Springer

Matteo Baldoni   Ulle Endriss
Andrea Omicini   Paolo Torroni (Eds.)

Declarative Agent Languages and Technologies III

Third International Workshop, DALT 2005
Utrecht, The Netherlands, July 25, 2005
Selected and Revised Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Matteo Baldoni
Università di Torino, Dipartimento di Informatica
via Pessinetto 12, 10149 Torino, Italy
E-mail: baldoni@di.unito.it

Ulle Endriss
University of Amsterdam, Institute for Logic, Language and Computation
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
E-mail: ulle@illc.uva.nl

Andrea Omicini
Università di Bologna, DEIS, Dipartimento di Elettronica, Informatica e Sistemistica
Alma Mater Studiorum, Via Venezia 52, 47023 Cesena, Italy
E-mail: andrea.omicini@unibo.it

Paolo Torroni
Università di Bologna, DEIS
Alma Mater Studiorum, Viale Risorgimento 2, 40136 Bologna, Italy
E-mail: paolo.torroni@unibo.it

Library of Congress Control Number: 2006922191

CR Subject Classification (1998): I.2.11, C.2.4, D.2.4, D.2, D.3, F.3.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-33106-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33106-3 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11691792 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3904

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

The workshop on Declarative Agent Languages and Technologies is a well-established venue for researchers interested in sharing their experiences in the areas of declarative and formal aspects of agents and multi-agent systems, and in engineering and technology. Today it is still a challenge to develop technologies that can satisfy the requirements of complex agent systems. The design and development of multi-agent systems still calls for models and technologies that ensure predictability, enable feature discovery, allow for the verification of properties, and guarantee flexibility. Declarative approaches are potentially a valuable means for satisfying the needs of multi-agent system developers and for specifying multi-agent systems.

DALT 2005, the third edition of the workshop, was held in Utrecht, The Netherlands, in July 2005, in conjunction with AAMAS 2005, the Fourth International Joint Conference on Agents and Multiagent Systems. Over 30 persons attended the workshop confirming the success of the previous editions in Melbourne 2003 (LNAI 2990) and New York 2004 (LNAI 3476). The workshop series is a forum of discussion aimed both at supporting the transfer of declarative paradigms and techniques into the broader community of agent researchers and practitioners, and at bringing the issues of designing real-world and complex agent systems to the attention of researchers working on declarative programming and technologies.

A twofold process led to this volume. On the one hand, the best papers presented at the workshop were selected after a further, meticulous reviewing process. On the other hand, an open call was issued for contributions that were not submitted to the original workshop call for papers, that resulted in a few other papers added, chosen through a very strict reviewing process. As a result, this volume contains 14 papers and it is organized in four parts corresponding to the main topics of DALT: *agent programming and beliefs*, *architectures and logic programming*, *knowledge representation and reasoning*, and *coordination and model checking*. Each paper was reviewed by at least three members of the Programme Committee in order to supply the authors with rich feedback that could stimulate the research.

Part I - Agent Programming and Beliefs

The first part of this volume contains three papers. The first work, "Beliefs in Agent Implementation", by Winkelhagen, Dastani, and Broersen, extends the language 3APL with beliefs represented as explicit modal operators. A proof procedure is also presented which is shown to be sound. The second work, "Modelling Uncertainty in Agent Programming", by Kwisthout and Dastani, tackles the uncertainty of agent beliefs modelling it by means of Dempster-Shafer theory, reporting complexity results. The last work, "Complete Axiomatizations of Finite Syntactic Epistemic States", by Ågotnes and Walicki, discusses a formal model of knowledge as explicitly computed sets of formulae, extending the epistemic language with an operator which expresses what an agent knows *at most.*

Part II - Architectures and Logic Programming

The second part contains four papers. The first, "An Architecture for Rational Agents", by Lloyd and Sears, proposes an agent architecture in which agents have belief bases that are theories in a multi-modal, higher-order logic. Machine learning techniques are used to update the belief base. The second paper, "LAIMA: A Multi-Agent Platform Using Ordered Choice Logic Programming", by De Vos, Crick, Padget, Brain, Cliffe, and Needham, introduces a deductive reasoning multi-agent platform based on an extension of answer set programming. Agents are represented as ordered choice logic programs. The third work, "A Distributed Architecture for Norm-Aware Agent Societies", by García-Camino, Rodríguez-Aguilar, Sierra, and Vasconcelos, describes a distributed architecture that accounts for a "social layer" in which rules are used for representing normative positions. Last but not least, "About Declarative Semantics of Logic-Based Agent Languages", by Costantini and Tocchio, provides a declarative semantics to logic-based agent-oriented languages, focussing on DALI as a case of study and paying particular attention to communication among agents.

Part III - Knowledge Representation and Reasoning

This part consists of five papers. The first paper, "Goal Decomposition Tree: An Agent Model to Generate a Validated Agent Behaviour", by Simon, Mermet, and Fournier, presents the Goal Decomposition Tree agent model, which allows both the specification and validation of agent behavior. The second paper, "Resource-Bounded Belief Revision and Contraction", by Alechina, Jago, and Logan, is set in the context of the AGM postulates and presents a linear time belief contraction operation that satisfies all but one of these postulates for contraction. The third paper, "Agent-Oriented Programming with Underlying Ontological Reasoning", by Moreira, Vieira, Bordini, and Hübner, defines a version of the BDI agent-oriented programming language AgentSpeak which is based on description logic. The authors use as a running example the well-known smart meeting-room scenario. The fourth paper, "Dynagent: An Incremental Forward-Chaining HTN Planning Agent in Dynamic Domains", by Hayashi, Tokura, Hasegawa, and Ozaki, presents an agent algorithm that integrates forward-chaining HTN planning, execution, belief updates, and plan modifications. By this approach agents are enabled to deal with dynamic worlds. The fifth paper, "A Combination of Explicit and Deductive Knowledge with Branching Time: Completeness and Decidability Results", by Lomuscio and Woźna, introduces a combination of Computational Tree Logic and an epistemic logic, which encompasses an epistemic operator to represent explicit knowledge. The properties of the obtained logic, such as decidability, are presented.

Part IV - Coordination and Model Checking

The last part of the volume contains two papers. "An Intensional Programming Approach to Multi-agent Coordination in a Distributed Network of Agents", by Wan and Alagar, presents an extension of Lucx and discusses the Intensional Programming Paradigm, with the aim of providing a programming model for coordinated problem solving in a multi-agent system. The last work in this collection, "A Tableau Method for Verifying Dialogue Game Protocols for Agent Communication", by Bentahar, Moulin, and Meyer, proposes a tableau-based model checking technique for verifying dialogue game protocols, defined using a social commitment-based framework for agent communication called Commitment and Argument Network.

DALT is now looking forward to its fourth meeting, which will take place in May 2006 in Hakodate, Japan, again as an AAMAS workshop, and will be chaired by Matteo Baldoni and Ulle Endriss. Besides the traditional DALT topics, the next edition will pay particular attention to the impact of the development of declarative approaches to application areas such as the *semantic web*, *web services*, *security*, and *electronic contracting*.

January 2006

Matteo Baldoni
Ulle Endriss
Andrea Omicini
Paolo Torroni

Organization

Workshop Organizers

Matteo Baldoni	University of Turin, Italy
Ulle Endriss	University of Amsterdam, The Netherlands
Andrea Omicini	University of Bologna a Cesena, Italy
Paolo Torroni	University of Bologna, Italy

Programme Committee

Natasha Alechina	University of Nottingham, UK
Rafael Bordini	University of Durham, UK
Brahim Chaib-draa	Laval University, Canada
Alessandro Cimatti	ITC-IRST, Trento, Italy
Keith Clark	Imperial College London, UK
Marco Colombetti	Politecnico di Milano, Italy
Stefania Costantini	University of L'Aquila, Italy
Mehdi Dastani	Utrecht University, The Netherlands
Jüergen Dix	University of Clausthal, Germany
Boi Faltings	EPFL, Switzerland
Michael Fisher	University of Liverpool, UK
Wiebe van der Hoek	University of Liverpool, UK
Michael N. Huhns	University of South Carolina, USA
Catholijn Jonker	Radboud University Nijmegen, The Netherlands
Peep Küngas	NUST, Trondheim, Norway
Yves Lespérance	York University, Toronto, Canada
Brian Logan	University of Nottingham, UK
Alessio Lomuscio	University College London, UK
Viviana Mascardi	University of Genova, Italy
John-Jules Ch. Meyer	Utrecht University, The Netherlands
Eric Monfroy	University of Nantes, France
Sascha Ossowski	Universidad Rey Juan Carlos, Madrid, Spain
Julian Padget	University of Bath, UK
Lin Padgham	RMIT University, Melbourne, Australia
Wojciech Penczek	Polish Academy of Sciences, Warsaw, Poland
Luís Moniz Pereira	Universidade Nova de Lisboa, Portugal
Enrico Pontelli	New Mexico State University, USA
Juan Rodriguez-Aguilar	Spanish Research Council, Barcelona, Spain
Luciano Serafini	ITC-IRST, Trento, Italy
Marek Sergot	Imperial College London, UK

Francesca Toni	Imperial College London, UK
Wamberto Vasconcelos	University of Aberdeen, UK
Michael Winikoff	RMIT University, Melbourne, Australia
Franco Zambonelli	University of Modena and Reggio Emilia, Italy

Additional Reviewers

Alessandro Artale
Cristina Baroglio
Valentina Gliozzi
Alberto Martelli
Eric Pacuit
Viviana Patti
Gian Luca Pozzato
Sebastian Sardina

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3904: M. Baldoni, U. Endriss, A. Omicini, P. Torroni (Eds.), Declarative Agent Languages and Technologies III. XII, 245 pages. 2006.

Vol. 3899: S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. XIV, 216 pages. 2006.

Vol. 3890: S.G. Thompson, R. Ghanea-Hercock (Eds.), Defence Applications of Multi-Agent Systems. XII, 141 pages. 2006.

Vol. 3885: V. Torra, Y. Narukawa, A. Valls, J. Domingo-Ferrer (Eds.), Modeling Decisions for Artificial Intelligence. XII, 374 pages. 2006.

Vol. 3881: S. Gibet, N. Courty, J.-F. Kamp (Eds.), Gesture in Human-Computer Interaction and Simulation. XIII, 344 pages. 2006.

Vol. 3874: R. Missaoui, J. Schmidt (Eds.), Formal Concept Analysis. X, 309 pages. 2006.

Vol. 3873: L. Maicher, J. Park (Eds.), Charting the Topic Maps Research and Applications Landscape. VIII, 281 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006.

Vol. 3862: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 267 pages. 2006.

Vol. 3849: I. Bloch, A. Petrosino, A.G.B. Tettamanzi (Eds.), Fuzzy Logic and Applications. XIV, 438 pages. 2006.

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006.

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005.

Vol. 3830: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems II. VIII, 291 pages. 2006.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Villar, J. Roure, V. Espinosa-Duro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2006.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005.

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005.

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T.R. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005.

Vol. 3763: H. Hong, D. Wang (Eds.), Automated Deduction in Geometry. X, 213 pages. 2006.

Vol. 3755: G.J. Williams, S.J. Simoff (Eds.), Data Mining. XI, 331 pages. 2006.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005.

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005.

Vol. 3721: A.M. Jorge, L. Torgo, P.B. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005.

Vol. 3720: J. Gama, R. Camacho, P.B. Brazdil, A.M. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005.

Vol. 3690: M. Pěchouček, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005.

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005.

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005.

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005.

Vol. 3673: S. Bandini, S. Manzoni (Eds.), AI*IA 2005: Advances in Artificial Intelligence. XIV, 614 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005.

Vol. 3661: T. Panayiotopoulos, J. Gratch, R.S. Aylett, D. Ballin, P. Olivier, T. Rist (Eds.), Intelligent Virtual Agents. XIII, 506 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005.

Vol. 3651: R. Dale, K.-F. Wong, J. Su, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2005. XXI, 1031 pages. 2005.

Vol. 3642: D. Ślęzak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005.

Vol. 3641: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005.

Vol. 3635: J.R. Winkler, M. Niranjan, N.D. Lawrence (Eds.), Deterministic and Statistical Methods in Machine Learning. VIII, 341 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005.

Vol. 3630: M.S. Capcarrère, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Ávila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3601: G. Moro, S. Bergamaschi, K. Aberer (Eds.), Agents and Peer-to-Peer Computing. XII, 245 pages. 2005.

Vol. 3600: F. Wiedijk (Ed.), The Seventeen Provers of the World. XVI, 159 pages. 2006.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E.T. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A.K. Dey, B. Kokinov, D.B. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3550: T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch, M.N. Huhns (Eds.), Multiagent System Technologies. XI, 246 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrović (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.-B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V.A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E.P. Stabler, J.V. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3490: L. Bolc, Z. Michalewicz, T. Nishida (Eds.), Intelligent Media Technology for Communicative Intelligence. X, 259 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J.A. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J.A. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Table of Contents

Coordination and Model Checking

Beliefs in Agent Implementation

Laurens Winkelhagen, Mehdi Dastani, and Jan Broersen

Institute of Information and Computing Sciences,
Utrecht University, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands
{lwinkelh, mehdi, broersen}@cs.uu.nl

Abstract. This paper extends a programming language for implementing cognitive agents with the capability to explicitly represent beliefs and reason about them. In this programming language, the beliefs of agents are implemented by modal logic programs, where beliefs are represented by explicit modal operators. A distinction is made between a belief base language that can be used to represent an agent's beliefs, and a belief query language that can be used to express queries to the agent's belief base. We adopt and modify a proof procedure that decides if a belief query formula is derivable from the belief base of an agent. We show that the presented proof procedure is sound.

1 Introduction

This paper presents an extension of the agent programming language 3APL [1]. This programming language provides data structures such as beliefs, goals, plans and reasoning rules, as well as programming constructs to manipulate these data structures. Examples of such constructs are updating the beliefs, planning a goal and executing a plan. In multi-agent settings, agents are assumed to have the ability to communicate. Several specifications have been proposed to facilitate agent communication, amongst these, the FIPA[1] standards are important to 3APL. According to FIPA, agents can communicate by sending each other messages that contain, amongst other things, a communicative act and the message content. The communicative acts specified by FIPA [2] require that a message with a certain performative can only be sent if certain belief formulae, the preconditions of the act, hold. This necessitates capabilities in the agent programming language to implement agents that can reason with beliefs. These beliefs can be about the beliefs of the agent that wants to send the message, or about the beliefs of the receiver of the message. For example, the INFORM act specifies that the sender believes that receiver has no beliefs about the message content.

The programming language 3APL implements agent belief in terms of a set of Horn clause formulae and uses a Prolog-engine to verify if the agent has a certain belief. However, 3APL lacks (1) the possibility to represent beliefs of agents about their own beliefs and the beliefs of other agents. It also lacks (2) the possibility to reason with these beliefs.

[1] The Foundation for Intelligent Physical Agents: http://www.fipa.org/

M. Baldoni et al. (Eds.): DALT 2005, LNAI 3904, pp. 1–16, 2006.

In this paper we show how we can extend the language of 3APL and the underlying Prolog mechanism in order to implement these two features. We take existing work [3, 4] on adding modal reasoning capabilities to logic programming, and investigate how to use this in combination with 3APL. We extend the programming language with an explicit modal operator of belief and provide a proof method that allows 3APL agents to function correctly with this modal operator. We show that this proof method is sound.

In sections 2 and 3 we give a quick introduction to 3APL and modal logic programming respectively. The bulk of the paper is section 4 in which we combine one approach to modal logic programming with the 3APL programming language, first discussing the syntactical changes, then the semantical interpretations and finally giving a soundness proof for these semantics. Section 5 is the conclusion, in which we will also point out some areas for further research.

2 3APL

Like other agent programming languages, 3APL provides data structures and programming constructs to manipulate them. Since 3APL is designed to implement cognitive agents, its data structures represent cognitive concepts such as beliefs, goals, plans, and reasoning rules. These data structures can be modified by programming constructs, also called deliberation operations, such as selecting a goal, applying a planning rule to it, or executing a plan. These operations constitute the deliberation process of individual agents which can be viewed as the agent interpreter. The formal syntax and semantics of 3APL are given in [1]. In this section, we will explain the ingredients of this programming language and give the formal definition of only those ingredients that are relevant for the research problem of this paper, i.e. the belief of 3APL agents.

The beliefs and goals are logical formulae representing the current and desirable state of the world, respectively. The goals of the agents are represented as logical formulae which are conjunction of atomic ground[2] formulae. The beliefs of 3APL agents can be specified by formulae in the following belief base language:

Definition 1. *(base language and belief base language) Let $\mathcal{V}ar$, $\mathcal{F}unc$ and $\mathcal{P}red$ be the sets of domain variables, functions and predicates, respectively. Let $Term$ be the set of terms constructed from variables and functions in usual way. The base language $\mathcal{L}$ is defined as the set of atomic formulae built on terms $Term$ and predicates $\mathcal{P}red$ in the usual way. Let $\psi \in \mathcal{L}$ be ground (atomic) formulae of the base language and let $\phi, \phi_1, \ldots, \phi_n \in \mathcal{L}$. The belief base language $\mathcal{L}_{BB}$, which represents the beliefs of agents, is defined as follows.*

$$\psi,\ \forall_{x_1,\ldots,x_n}(\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi)\ \in \mathcal{L}_{BB}$$

where $\forall_{x_1,\ldots,x_n}(\varphi)$ denotes the universal closure of the formula φ for every variable $x_1, \ldots, x_n$ occurring in φ.

[2] Throughout this paper we will use such standard terminology. In case of doubt, we will use the same terminology as used in [1].

In order to reach its goals, a 3APL agent adopts plans. A plan is built from basic elements that can be composed by sequence operators, if-then-else constructs, and while-do loops. The basic elements can be basic actions, test actions, or abstract plans. A test action checks whether a certain formula is derivable from the belief base. An abstract plan is an abstract representation of a plan which can be instantiated with a plan during execution. Thus, an abstract plan cannot be executed directly and should be rewritten into another plan, possibly (and even probably) containing executable basic actions, through application of reasoning rules (see below).

There are three types of basic actions. The first type of basic action is the mental action. This action modifies the beliefs of the agents and is specified in terms of pre- and post-conditions in the form of belief formulae. A mental action can be performed if the pre-condition is derivable from the agent beliefs after which the post-condition must be derivable. The external actions can be performed in the environment of the agents. The effect of these actions is determined by the environment and can be perceived by the agent through sensing. The communication actions pass messages to another agent. A message contains the name of the receiver of the message, the speech act or performative (e.g. inform, request, etc.) of the message, and the content. The content of the message is a belief formula.

In order to reason with goals and plans, 3APL has two types of reasoning rules: goal planning rules and plan revision rules. A goal planning rule, which is a tuple consisting of a goal formula, a belief formula and a plan, indicates that the state represented by the goal formula can be reached by the plan if the belief formula holds. Such a rule is applicable when the agent has a goal unifiable with the goal formula of the rule and when the belief formula of the rule is derivable from the belief base. Application of such a rule will add the plan to the set of plans of the agent.

A plan revision rule, which is a tuple consisting of a belief formula and two plans, indicates that the first plan can be replaced by the second plan if the belief formula holds. Such a rule is applicable when the agent has a plan unifiable with the first plan of the rule and when the belief formula of the rule is derivable from the belief base. The application of the rule will replace the unifiable plan of the agent with the second plan. The plan revision rules are powerful way to handle failed or blocked plans as well as adapting and removing plans of the agent.

A 3APL agent starts its deliberation with a number of goals to achieve. Planning rules are then applied to generate plans for the goals after which the plans are executed to achieve the goals. Plans may start with mental actions for which the pre-conditions are not true. In this case, the plan revision rules are applied to generate alternative plans. During the execution of 3APL agents, there are four cases where it is checked if a certain formula is derivable from the beliefs of the agent. These cases are related to the execution of *test actions* and *mental actions*, and to the application of the *goal planning rules* and *plan revision rules*. In fact, the content of test actions, the pre-condition of mental actions, and the guard of the rules are logical formulae which should be derivable before these actions and rules can be executed or applied. We define the language of these formulae, the *belief query language* ($\mathcal{L}_B$).

Definition 2. *(belief query language) Let $\mathcal{L}$ be the base language. Then, the belief query language $\mathcal{L}_B$ with typical formula β is defined as follows:*

- *if $\phi \in \mathcal{L}$, then $\mathbf{B}(\phi), \neg\mathbf{B}(\phi) \in Disjunction$,*
- *$\top \in Disjunction$,*
- *if $\delta, \delta' \in Disjunction$, then $\delta \vee \delta' \in Disjunction$,*
- *if $\delta \in Disjunction$, then $\delta \in \mathcal{L}_B$,*
- *if $\beta, \beta' \in \mathcal{L}_B$, then $\beta \wedge \beta' \in \mathcal{L}_B$.*

The bold '**B**' is not a modal operator, but is used to represent a query expression. For example $\mathbf{B}\varphi$ represents the query whether φ is derivable from the belief base. We use $\neg\mathbf{B}\varphi$ to represent the query whether φ is *not* derivable from the belief base. This interpretation of $\neg$ corresponds with the interpretation of negation as failure in logic programming. The **B** operator prevents the distribution of this negation over the belief query.

The 3APL semantics is based on an operational semantics which is defined in terms of a transition system. A transition system is a set of derivation rules for deriving transitions. A transition is a transformation of one configuration (or state) into another and it corresponds to a single computation step. In the case of the operational semantics for 3APL agents, the configurations are the mental states of the 3APL agents defined in terms of the belief base, goal base, plan base, and a substitution which assigns terms to variables.

Definition 3. *(configuration) Let $\mathcal{L}_{GB}$ be the goal language and $\mathcal{L}_P$ be the plan language[3]. A configuration of an individual 3APL agent is a tuple $\langle \iota, \sigma, \gamma, \Pi, \theta \rangle$, where ι is an agent identifier, $\sigma \subseteq \mathcal{L}_{BB}$ is the belief base of the agent, $\gamma \subseteq \mathcal{L}_{GB}$ is the goal base of the agent, $\Pi \subseteq \mathcal{L}_P$ is the plan base of the agent and θ is a ground substitution that binds domain variables to domain terms.*

In order to check whether an agent in a certain state has a certain belief or not, one must check if the corresponding belief query formula is derivable from the agent configuration that represents the state of the agent. In the 3APL transition semantics, this is formally expressed through an entailment relation $\models_\tau$[4]. In particular, to check if agent ι believes $\beta \in \mathcal{L}_B$ (a belief query formula) in state $\langle \iota, \sigma, \gamma, \Pi, \theta \rangle$ is specified as $\langle \iota, \sigma, \gamma, \Pi, \theta \rangle \models_\tau \beta$.

The entailment relation $\models_\tau$ is defined recursively for the compound formulae. At the level of atomic formulae the satisfaction relation is defined in terms of propositional satisfaction relation. In particular, if β is of the form $\mathbf{B}\phi$, the satisfaction relation is defined as follows:

$$\langle \iota, \sigma, \gamma, \Pi, \theta \rangle \models_\tau \beta \;\Leftrightarrow\; \sigma \models \phi\tau$$

[3] Since the focus of this paper is the beliefs of the 3APL agents, the goal and plan language are not presented. For a detailed specification of these languages, see [1].

[4] The subscript τ is a substitution under which a formula is derivable from the state.