Nicola Olivetti (Ed.)

LNAI 4548

Automated Reasoning with Analytic Tableaux and Related Methods

16th International Conference, TABLEAUX 2007
Aix en Provence, France, July 2007
Proceedings

Springer

Nicola Olivetti (Ed.)

Automated Reasoning with Analytic Tableaux and Related Methods

16th International Conference, TABLEAUX 2007
Aix en Provence, France, July 3-6, 2007
Proceedings

Springer

Series Editors

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

Volume Editor

Nicola Olivetti
LSIS - UMR CNRS 6168
Université Paul Cézanne
Domaine Universitaire de Saint-Jérôme
Avenue Escadrille Normandie-Niemen
13397 Marseille Cedex 20, France
E-mail: nicola.olivetti@univ-cezanne.fr

Library of Congress Control Number: 2007929030

CR Subject Classification (1998): I.2.3, F.4.1, I.2, D.1.6, D.2.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-73098-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73098-9 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: 12077922 06/3180 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 4548
Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 4573: M. Kauers, M. Kerber, R. Miner, W. Windsteiger (Eds.), Towards Mechanized Mathematical Assistants. XIII, 407 pages. 2007.

Vol. 4548: N. Olivetti (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. X, 245 pages. 2007.

Vol. 4539: N.H. Bshouty, C. Gentile (Eds.), Learning Theory. XII, 634 pages. 2007.

Vol. 4529: P. Melin, O. Castillo, L.T. Aguilar, J. Kacprzyk, W. Pedrycz (Eds.), Foundations of Fuzzy Logic and Soft Computing. XIX, 830 pages. 2007.

Vol. 4511: C. Conati, K. McCoy, G. Paliouras (Eds.), User Modeling 2007. XVI, 497 pages. 2007.

Vol. 4509: Z. Kobti, D. Wu (Eds.), Advances in Artificial Intelligence. XII, 552 pages. 2007.

Vol. 4496: N.T. Nguyen, A. Grzech, R.J. Howlett, L.C. Jain (Eds.), Agent and Multi-Agent Systems: Technologies and Applications. XXI, 1046 pages. 2007.

Vol. 4483: C. Baral, G. Brewka, J. Schlipf (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 327 pages. 2007.

Vol. 4482: A. An, J. Stefanowski, S. Ramanna, C.J. Butz, W. Pedrycz, G. Wang (Eds.), Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. XIV, 585 pages. 2007.

Vol. 4481: J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N.J. Cercone, D. Ślęzak (Eds.), Rough Sets and Knowledge Technology. XIV, 576 pages. 2007.

Vol. 4476: V. Gorodetsky, C. Zhang, V.A. Skormin, L. Cao (Eds.), Autonomous Intelligent Systems: Multi-Agents and Data Mining. XIII, 323 pages. 2007.

Vol. 4452: M. Fasli, O. Shehory (Eds.), Agent-Mediated Electronic Commerce. VIII, 249 pages. 2007.

Vol. 4451: T.S. Huang, A. Nijholt, M. Pantic, A. Pentland (Eds.), Artifical Intelligence for Human Computing. XVI, 359 pages. 2007.

Vol. 4438: L. Maicher, A. Sigel, L.M. Garshol (Eds.), Leveraging the Semantics of Topic Maps. X, 257 pages. 2007.

Vol. 4429: R. Lu, J.H. Siekmann, C. Ullrich (Eds.), Cognitive Systems. X, 161 pages. 2007.

Vol. 4426: Z.-H. Zhou, H. Li, Q. Yang (Eds.), Advances in Knowledge Discovery and Data Mining. XXV, 1161 pages. 2007.

Vol. 4411: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2007.

Vol. 4410: A. Branco (Ed.), Anaphora: Analysis, Algorithms and Applications. X, 191 pages. 2007.

Vol. 4399: T. Kovacs, X. Llorà, K. Takadama, P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Learning Classifier Systems. XII, 345 pages. 2007.

Vol. 4390: S.O. Kuznetsov, S. Schmidt (Eds.), Formal Concept Analysis. X, 329 pages. 2007.

Vol. 4389: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems III. X, 273 pages. 2007.

Vol. 4384: T. Washio, K. Satoh, H. Takeda, A. Inokuchi (Eds.), New Frontiers in Artificial Intelligence. IX, 401 pages. 2007.

Vol. 4371: K. Inoue, K. Satoh, F. Toni (Eds.), Computational Logic in Multi-Agent Systems. X, 315 pages. 2007.

Vol. 4369: M. Umeda, A. Wolf, O. Bartenstein, U. Geske, D. Seipel, O. Takata (Eds.), Declarative Programming for Knowledge Management. X, 229 pages. 2006.

Vol. 4342: H. de Swart, E. Orłowska, G. Schmidt, M. Roubens (Eds.), Theory and Applications of Relational Structures as Knowledge Instruments II. X, 373 pages. 2006.

Vol. 4335: S.A. Brueckner, S. Hassas, M. Jelasity, D. Yamins (Eds.), Engineering Self-Organising Systems. XII, 212 pages. 2007.

Vol. 4334: B. Beckert, R. Hähnle, P.H. Schmitt (Eds.), Verification of Object-Oriented Software. XXIX, 658 pages. 2007.

Vol. 4333: U. Reimer, D. Karagiannis (Eds.), Practical Aspects of Knowledge Management. XII, 338 pages. 2006.

Vol. 4327: M. Baldoni, U. Endriss (Eds.), Declarative Agent Languages and Technologies IV. VIII, 257 pages. 2006.

Vol. 4314: C. Freksa, M. Kohlhase, K. Schill (Eds.), KI 2006: Advances in Artificial Intelligence. XII, 458 pages. 2007.

Vol. 4304: A. Sattar, B.-H. Kang (Eds.), AI 2006: Advances in Artificial Intelligence. XXVII, 1303 pages. 2006.

Vol. 4303: A. Hoffmann, B.-H. Kang, D. Richards, S. Tsumoto (Eds.), Advances in Knowledge Acquisition and Management. XI, 259 pages. 2006.

Vol. 4293: A. Gelbukh, C.A. Reyes-Garcia (Eds.), MICAI 2006: Advances in Artificial Intelligence. XXVIII, 1232 pages. 2006.

Vol. 4289: M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenič, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek, M. van Someren (Eds.), Semantics, Web and Mining. X, 197 pages. 2006.

Vol. 4285: Y. Matsumoto, R.W. Sproat, K.-F. Wong, M. Zhang (Eds.), Computer Processing of Oriental Languages. XVII, 544 pages. 2006.

Vol. 4274: Q. Huo, B. Ma, E.-S. Chng, H. Li (Eds.), Chinese Spoken Language Processing. XXIV, 805 pages. 2006.

Vol. 4265: L. Todorovski, N. Lavrač, K.P. Jantke (Eds.), Discovery Science. XIV, 384 pages. 2006.

Vol. 4264: J.L. Balcázar, P.M. Long, F. Stephan (Eds.), Algorithmic Learning Theory. XIII, 393 pages. 2006.

Vol. 4259: S. Greco, Y. Hata, S. Hirano, M. Inuiguchi, S. Miyamoto, H.S. Nguyen, R. Słowiński (Eds.), Rough Sets and Current Trends in Computing. XXII, 951 pages. 2006.

Vol. 4253: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. XXXII, 1301 pages. 2006.

Vol. 4252: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. XXXIII, 1335 pages. 2006.

Vol. 4251: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXVI, 1297 pages. 2006.

Vol. 4248: S. Staab, V. Svátek (Eds.), Managing Knowledge in a World of Networks. XIV, 400 pages. 2006.

Vol. 4246: M. Hermann, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIII, 588 pages. 2006.

Vol. 4223: L. Wang, L. Jiao, G. Shi, X. Li, J. Liu (Eds.), Fuzzy Systems and Knowledge Discovery. XXVIII, 1335 pages. 2006.

Vol. 4213: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006. XXII, 660 pages. 2006.

Vol. 4212: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Machine Learning: ECML 2006. XXIII, 851 pages. 2006.

Vol. 4211: P. Vogt, Y. Sugita, E. Tuci, C.L. Nehaniv (Eds.), Symbol Grounding and Beyond. VIII, 237 pages. 2006.

Vol. 4203: F. Esposito, Z.W. Raś, D. Malerba, G. Semeraro (Eds.), Foundations of Intelligent Systems. XVIII, 767 pages. 2006.

Vol. 4201: Y. Sakakibara, S. Kobayashi, K. Sato, T. Nishino, E. Tomita (Eds.), Grammatical Inference: Algorithms and Applications. XII, 359 pages. 2006.

Vol. 4200: I.F.C. Smith (Ed.), Intelligent Computing in Engineering and Architecture. XIII, 692 pages. 2006.

Vol. 4198: O. Nasraoui, O. Zaïane, M. Spiliopoulou, B. Mobasher, B. Masand, P.S. Yu (Eds.), Advances in Web Mining and Web Usage Analysis. IX, 177 pages. 2006.

Vol. 4196: K. Fischer, I.J. Timm, E. André, N. Zhong (Eds.), Multiagent System Technologies. X, 185 pages. 2006.

Vol. 4188: P. Sojka, I. Kopeček, K. Pala (Eds.), Text, Speech and Dialogue. XV, 721 pages. 2006.

Vol. 4183: J. Euzenat, J. Domingue (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 291 pages. 2006.

Vol. 4180: M. Kohlhase, OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. XIX, 428 pages. 2006.

Vol. 4177: R. Marín, E. Onaindía, A. Bugarín, J. Santos (Eds.), Current Topics in Artificial Intelligence. XV, 482 pages. 2006.

Vol. 4160: M. Fisher, W. van der Hoek, B. Konev, A. Lisitsa (Eds.), Logics in Artificial Intelligence. XII, 516 pages. 2006.

Vol. 4155: O. Stock, M. Schaerf (Eds.), Reasoning, Action and Interaction in AI Theories and Systems. XVIII, 343 pages. 2006.

Vol. 4149: M. Klusch, M. Rovatsos, T.R. Payne (Eds.), Cooperative Information Agents X. XII, 477 pages. 2006.

Vol. 4140: J.S. Sichman, H. Coelho, S.O. Rezende (Eds.), Advances in Artificial Intelligence - IBERAMIA-SBIA 2006. XXIII, 635 pages. 2006.

Vol. 4139: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala (Eds.), Advances in Natural Language Processing. XVI, 771 pages. 2006.

Vol. 4133: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), Intelligent Virtual Agents. XIV, 472 pages. 2006.

Vol. 4130: U. Furbach, N. Shankar (Eds.), Automated Reasoning. XV, 680 pages. 2006.

Vol. 4120: J. Calmet, T. Ida, D. Wang (Eds.), Artificial Intelligence and Symbolic Computation. XIII, 269 pages. 2006.

Vol. 4118: Z. Despotovic, S. Joseph, C. Sartori (Eds.), Agents and Peer-to-Peer Computing. XIV, 173 pages. 2006.

Vol. 4114: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence, Part II. XXVII, 1337 pages. 2006.

Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), Mathematical Knowledge Management. VIII, 295 pages. 2006.

Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H.A. Güvenir (Eds.), Advances in Case-Based Reasoning. XIV, 566 pages. 2006.

Vol. 4099: Q. Yang, G. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence. XXVIII, 1263 pages. 2006.

Vol. 4095: S. Nolfi, G. Baldassarre, R. Calabretta, J.C.T. Hallam, D. Marocco, J.-A. Meyer, O. Miglino, D. Parisi (Eds.), From Animals to Animats 9. XV, 869 pages. 2006.

Vol. 4093: X. Li, O.R. Zaïane, Z. Li (Eds.), Advanced Data Mining and Applications. XXI, 1110 pages. 2006.

Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), Knowledge Science, Engineering and Management. XV, 664 pages. 2006.

Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems. XVII, 827 pages. 2006.

Vol. 4087: F. Schwenker, S. Marinai (Eds.), Artificial Neural Networks in Pattern Recognition. IX, 299 pages. 2006.

¥484.00元

Preface

This volume gathers the research papers presented at the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2007) that took place July 3–6, 2007 in Aix en Provence, France. This conference was the 16th in a series of international meetings held since 1992 (the list is on page VIII).

The Program Committee of TABLEAUX 2007 received 43 submissions, 16 of which were accepted for publication in the present proceedings, while 8 were accepted as position papers.

In addition to the contributed papers, the program included three excellent keynote talks by Piero Bonatti of Università di Napoli, by John-Jules Meyer of Utrecht University, and by Cesare Tinelli of the University of Iowa. Finally, the program was completed by three tutorials of deep interest: "The Tableau Work Bench: Theory and Practice" (Pietro Abate and Rajeev Goré), "Tableau Methods for Interval Temporal Logics" (Valentin Goranko and Angelo Montanari), and "Semistructured Databases and Modal Logic" (Serenella Cerrito).

Tableaux and related methods are a convenient formalism for automating deduction in classical as well as in non-classical logics. The papers collected in this volume witness the wide range of logics being covered: from intuitionistic and substructural logics to modal logics (including temporal and dynamic logics), from many-valued logics to nonmonotonic logics, from classical first-order logic to description logics. Some contributions are focused on decision procedures, others on efficient reasoning, as well as on implementation of theorem provers. A few papers explore applications such as model-checking, verification, or knowledge engineering. Finally, some contributions make use of tableaux as a tool for theoretical investigation of logics. This variety of logics and applications illustrates well the flexibility and the ubiquity of analytic tableaux and related proof methods.

I want to express my gratitude to the invited speakers and to the tutorial presenters who really contributed to making a rich and stimulating conference program. I am very grateful to the members of the Program Committee for their assistance in all phases of the conference and to other reviewers who ensured, together with the Program Committee members, a rigorous selection of the papers. Their effort was decisive to keeping the high scientific standard of the conference. I am also grateful to the members of the Steering Committee for their valuable support. I particularly thank my colleagues of the Local Organizing Committee: Belaid Benhamou, Djamal Habet, Philippe Jégou, Richard Ostrowski, Cyril Pain-Barre, Odile Papini, Nicolas Prcovic, Vincent Risch, Pierre Siegel, Cyril Terrioux, Eric Würbel; they worked hard with a truly cooperative spirit,

making the conference a successful event. A final thanks to the Office of Tourism of Aix en Provence and to Promo Sciences for their professional assistance and services.

July 2007 Nicola Olivetti

Organization

Program and Conference Chair

Nicola Olivetti	Paul Cézanne University, France

Program Committee

Peter Baumgartner	NICTA, Australia
Bernhard Beckert	University of Koblenz-Landau, Germany
Patrick Blackburn	INRIA Lorraine, France
Marta Cialdea	University of Roma 3, Italy
Roy Dyckhoff	University of St. Andrews, UK
Christian G. Fermüller	University of Vienna, Austria
Ulrich Furbach	University of Koblenz-Landau, Germany
Didier Galmiche	LORIA, Henri Poincaré University, France
Martin Giese	RISC, Johannes Kepler University, Austria
Rajeev Goré	Australian National University, Australia
Jean Goubault-Larrecq	LSV, ENS Cachan, France
Reiner Hähnle	University of Chalmers, Sweden
Ullrich Hustadt	University of Liverpool, UK
Christoph Kreitz	University of Potsdam, Germany
Carsten Lutz	University of Dresden, Germany
Angelo Montanari	University of Udine, Italy
Ugo Moscato	University of Milano-Bicocca, Italy
Neil V. Murray	ILS Institute, University at Albany, USA
Ilkka Niemelä	Helsinki University of Technology, Finland
Lawrence C. Paulson	University of Cambridge, UK
Camilla Schwind	LIF-CNRS, France
Viorica Sofronie-Stokkermans	Max-Planck-Institut für Informatik, Germany
Arild Waaler	University of Oslo, Norway

Additional Referees

Alessandro Avellone
Matthias Baaz
Thomas Bolander
Mirjana Borisavljević
Torben Braüner
Davide Bresolin
Elie Bursztein
Serenella Cerrito
Marcello D'Agostino
Stéphane Demri
Hans de Nivelle
Camillo Fiorentini
Guido Fiorino
Lev Gordeev
Andrew Haas
Keijo Heljanko
Jan Hladik
Arnold Holger
Jinbo Huang
Yevgeny Kazakov
Kentaro Kikuchi
Dominique Larchey-Wendling
Patrick Maier

Daniel Méry
George Metcalfe
Maja Milicic
Andrea Orlandini
Mario Ornaghi
Gabriele Puppis
Robert Rothenberg
Gernot Salzer
Peter Schotch
Yaroslav Shramko
Evgeny Zolin

Steering Committee

Rajeev Goré (President)	Australian National University, Australia
Bernhard Beckert (Vice-President)	University of Koblenz-Landau, Germany
Stéphane Demri	ENS Cachan, France
Roy Dyckhoff	University of St. Andrews, UK
Ulrich Furbach	University of Koblenz-Landau, Germany
Didier Galmiche	LORIA, Henri Poincaré University, France
Reiner Hähnle	University of Chalmers, Sweden
Nicola Olivetti	Paul Cézanne University, France

Organization

TABLEAUX 2007 was organized by the INCA team (Inference, Constraints and Applications) of LSIS CNRS UMR 6168 (Information and System Sciences Laboratory).

Sponsoring Institutions

LSIS Laboratory CNRS UMR 6168
Université Paul Cézanne
Université de Provence
Université de la Méditerranée
Ville d'Aix en Provence
Communauté du Pays d'Aix
Conseil Général des Bouches du Rhône
Région Provence Alpes Côte d'Azur

Previous Conferences

1992 Lautenbach, Germany
1993 Marseille, France
1994 Abingdon, UK
1995 St. Goar, Germany
1996 Terrasini, Italy
1997 Pont-à-Mousson, France
1998 Oisterwijk, The Netherlands
1999 Saratoga Springs, USA
2000 St. Andrews, UK
2001 Siena, Italy (part of IJCAR)
2002 Copenhagen, Denmark
2003 Rome, Italy
2004 Cork, Ireland (part of IJCAR)
2005 Koblenz, Germany
2006 Seattle, USA (part of IJCAR)

Table of Contents

Invited Talks

Research Papers

System Descriptions

Nonmonotonic Description Logics – Requirements, Theory, and Implementations

Piero A. Bonatti

Dipartimento di Scienze Fisiche
Sezione di Informatica
Università di Napoli Federico II
Napoli, Italy
bonatti@na.infn.it

Abstract. The Semantic Web and a number of modern knowledge-based applications have turned ontologies into a familiar and popular ICT notion. Description Logics (DLs) are one of the major formalisms for encoding ontologies.

Many "users" of such formalisms – that is, ontologies writers – would appreciate DLs to have nonmonotonic features. For example, it would be appealing to describe taxonomies by means of general default properties that may be later overridden in special cases; a similar behavior is supported by all object-oriented languages, after all. However, nonmonotonic extensions of DLs involve many tricky technical problems.

This talk will briefly illustrate some of the major requirements for nonmonotonic description logics and some of the formalisms currently available. Then we shall point out the major problems that still have to be solved in order to apply standard tableaux optimization techniques to nonmonotonic DLs. Since DLs are usually at least PSPACE-hard, such optimization techniques are crucial in making these formalisms usable in practice.

For example, it seems very difficult to find a tableaux system for a fragment of nonmonotonic DLs where a tableau needs not be stored entirely in memory (because it is enough to construct and verify a single branch at each iteration, for example).

Since "traditional" nonmonotonic semantics are not completely satisfactory, it may be possible to solve both semantic shortcomings and optimization problems by adopting suitable new logics.

N. Olivetti (Ed.): TABLEAUX 2007, LNAI 4548, p. 1, 2007.

Our Quest for the Holy Grail of Agent Verification

John-Jules Ch. Meyer

Department of Information and Computing Sciences,
Utrecht University,
The Netherlands
jj@cs.uu.nl

Abstract. Since the inception of agent technology almost two decades ago, researchers have worked both on the formal, theoretical aspects of intelligent agents and on the realisation / implementation of them. However, the link between the two has always remained rather unclear, to this day. Although there is a definite need for the verification of agents, the methods and techniques for this are still in their infancy. We describe our personal ongoing quest for the 'right' approach to agent verification.

1 Cognitive Agent Programming

The basic ideas of agent programming go back to Aristotle's *practical reasoning* (cf. [5], p. 728), augmented by the modern philosophical concepts of the *intentional stance* by Dennett [16] and that of *intention* by Bratman [10]. In short, practical reasoning is about specifying the decision of an agent by coining it in a rule (a so-called practical syllogism). Together with the idea of treating an entity as a rational agent deliberating its beliefs and goals in order to come up with the next action (Dennett's intentional stance), and the idea that resource-bounded agents should always settle on some of their desires and then stick with these as long as is rationally possible (the concept of an intention), this provides the ingredients of the current agent-oriented programming languages such as Agent0 [42], AgentSpeak(L) [34] and 3APL [24,14]/ 2APL [13] (also cf. [7]). In particular, we are interested in what we call cognitive or BDI agent programming, in which the agent has mental attitudes such as beliefs, desires (goals) and intentions (plans).

For instance, in 3APL and 2APL an agent has a belief base, a goal base and a plan base. The programmer can use Plan Generation (PG) and Plan Revision (PR) rules of the form

$$\gamma|\beta \rightarrow \pi$$

and

$$\pi_1|\beta \rightarrow \pi_2$$

respectively, to let the agent generate plans π, given the agent's beliefs β and goals γ, and revise plans π_1 to π_2 when necessary (indicated by a certain belief condition β) (cf.[7,13]). As one can see, these rules are a direct operationalization of practical reasoning together with the ideas of Dennett and Bratman.

N. Olivetti (Ed.): TABLEAUX 2007, LNAI 4548, pp. 2–9, 2007.

2 Agent Verification

Since the inception of agent technology almost two decades ago, researchers have worked both on the formal, theoretical aspects of intelligent agents and on the realisation / implementation of them [12,35,28,43]. However, the link between the two has always remained rather unclear, to this day. A case in point is the pioneering work of Rao & Georgeff, who developed their famous BDI (beliefs-desires-intentions) logic ([35]) on the one hand, while working on their BDI architecture and programming language AgentSpeak(L) ([34]) to realize agents, on the other. Although the names of the logic and architecture (both containing 'BDI') suggest an intrinsic connection, they are only superficially related. This also holds for the BDI-like logic and the programming language Agent0 in the pioneering first paper on agent-oriented programming by Shoham [42]. More generally, this phenomenon is referred to in the literature as the 'gap' between agent logics and implemented agent systems. One of the problems is that the BDI notions in agent logics generally are not 'grounded' in agent computations: they are rather general notions, modelled by abstract (accessibility) relations in modal logic, and have no apparent association with concrete agent behaviour, and an agent program in particular (cf. [44]). On the other hand one has increasingly come to realize that complex agent-based systems need verification. For instance, in the domain of space exploration one is getting more and more interested in agent systems, but also feels the need for verifying them [36]. Although here it is an absolute necessity in view of huge investments of money and the fact that possibly also human lives are at stake, there are many other examples, where the sheer complexity of agent-based systems to be deployed calls for formal verification. However, since it is hard to connect the practice of agent programming with the formal counterpart of agent logics, it has been hard to establish a solid framework for agent verification.

3 Our Approaches

In the last 15 years or so we have been working on both agent logics (viz. the KARO framework [28]) and agent programming (viz. the agent languages 3APL [24,14] and, more recently, 2APL [13]), and personally I have been always fascinated by the relation between the two, and in particular how agent logics can be used to specify and verify agent programs. Furthermore, having our 'roots' in semantics and correctness of 'traditional' programming in the style of De Bakker [6], when attempting to prove agent programs correct, we have always been very interested to (re-)use methods and techniques from 'traditional' program correctness, and try to adapt and extend them to deal with agent programs. Of course, these adaptations and extensions are needed to cater for the typical agent-oriented features such as the BDI attitudes in particular, not present and dealt with in traditional approaches.

As we saw before one of the biggest problems is to connect ('ground') BDI notions in agent logics with agent programming, i.e. more computational notions.

Agent logics are generally not grounded: notions like beliefs, goals, etc. and, more generally, possible worlds and accessibility relations are very abstract and not connected directly to computational notions. Therefore no direct link between agent logics and agent programming exists, even if both use what seems to be at first sight the same notions (such as beliefs and goals).

In this section we will discuss a number of routes we have taken attempting to bridge the gap between agent programming and logic.

3.1 Programming KARO Agents

KARO is an expressive formalism to specify agent attitudes ([28]). It is a blend of dynamic and epistemic logic augmented with modalities for motivational attitudes (desires, goals, commitments, ...) Because of the fact that it is based on a logic of *action* rather than *time*, like the other well-known agent logics, we always had the idea that KARO is closer than other agent logics to a computational approach to agents such as agent programming. In [32] we have explored the idea of putting agent programs as actions into the dynamic logic operators to be able to reason about these programs within the KARO framework, but the idea was only elaborated in a rather loose and preliminary fashion. We will see that this idea was later picked up again in a more rigorous manner.

3.2 An Executable Core of KARO

One way, of course, to establish a relation between a program and its logical specification is to use the specification as program itself. This has been the philosophy of METATEM [21]: to execute specifications of agents written in an executable fragment of temporal logic augmented with BDI concepts (see e.g. [22]). Together with the METATEM team we have looked whether the same line could be followed with specifications in the KARO framework, a blend of dynamic and epistemic logic, augmented with other BDI-like concepts (see [28]). We succeeded in mimicking this idea for KARO [29,30], but the concession we had to make is that we had to reduce the very expressive KARO framework to a rather small core, with the drawback that much of the expressiveness was lost.

3.3 The GOAL Method

In [25,23] we considered a very simple agent programming logic with declarative goals (but without intentions or procedural goals / plans!), and tried to give a complete programming theory for it, viz. a programming language together with a formal semantics, as well as a correctness logic. Interestingly, apart from the familiar Hoare triples also ideas from concurrency theory such as the UNITY framework ([11]) were used. It was meant as a kind of proof of concept that one could try to get such a complete theory for agent programming, but, admittedly, the power of the programming language, called GOAL, was rather limited (although deliberately so). Some of these ideas were later used to enhance our programming language 3APL with both declarative and procedural goals [38,14].

3.4 Agent Logics as Program Logics: Grounding BDI-Like Logics

Recently we have taken up our efforts to ground agent logics in a more principled way.

Grounding KARO. In [26,27] this is attempted by making the notions that occur in agent logics (e.g. KARO) such as beliefs and goals less abstract and more computational (e.g. by not basing them on abstract accessibility relations but on certain types of knowledge bases, yielding a 'state-based semantics'), so that reasoning about these notions becomes relevant for reasoning about concrete agent programs. But also this comes with the price of reducing the logic (KARO) to a core.

Relating accessibility and execution (CTL_{APL}). In [15] it is proposed to base the abstract *accessibility relations* of agent logics directly on the (Plotkin-style) operational semantics of agent programs. In this particular proposal the temporal logic CTL* [18] is employed, where the temporal accessibility relation is specified by a transition system for the operational semantics of the agent programming language (APL) at hand, thus establishing the grounding of the logic. By doing this one can use the logical language to express properties of (the behavior of) agent programs written in APL, which may then be verified by model-checking, for instance.

A dedicated PDL version for APLs. In a recent paper [4] we go about in a different way: we consider a (simple) agent programming language SimpleAPL together with its operational semantics. We next devise a PDL-based [20]) agent logic tailored to constructs that are present in SimpleAPL. This is done by employing a function that transforms the basic ingredients of SimpleAPL to expressions in the logic. Next we give a sound and complete axiomatization of this logic. We then show the relevance of the logic for proving properties of programs written in SimpleAPL by proving a theorem exactly relating the transition semantics of the programs appearing in the PDL-like logic and the operational semantics of SimpleAPL. In this way we know that we can use the devised logic for reasoning about SimpleAPL programs, and thus show correctness properties of these programs. We show how this can be done by way of an example. The reasoning can be assisted by automated verfication methods, and actually these (viz. [41]) *were* used in verifying the example mentioned.

3.5 Dynamic Logic for 3APL

We have also proposed a verification logic for the language 3APL directly as for any other traditional programming language without resorting to a connection with agent logics. In [40,37] we propose a dynamic logic for the execution of plans where also it is taken into account that plans may be revised by plan revision (PR) rules. As one may appreciate, the latter renders the execution of plans highly 'non-compositional', which results in the fact that in this situation the standard validity in dynamic logic, $[\pi_1; \pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$, is not valid anymore.