

João Leite
Paolo Torroni (Eds.)

LNAI 3487

Computational Logic in Multi-Agent Systems

5th International Workshop, CLIMA V
Lisbon, Portugal, September 2004
Revised Selected and Invited Papers



Springer

João Leite Paolo Torroni (Eds.)

Computational Logic in Multi-Agent Systems

5th International Workshop, CLIMA V
Lisbon, Portugal, September 29-30, 2004
Revised Selected and Invited Papers



Springer

Series Editors

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

Volume Editors

João Leite
Universidade Nova de Lisboa, Departamento de Informática
Faculdade de Ciências e Tecnologia
Quinta da Torre, 2829-516 Caparica, Portugal
E-mail: jleite@di.fct.unl.pt

Paolo Torroni
Università di Bologna
Dipartimento di Elettronica, Informatica e Sistemistica
Viale Risorgimento 2, 40136 Bologna, Italy
E-mail: paolo.torroni@unibo.it

Library of Congress Control Number: 2005929660

CR Subject Classification (1998): I.2.11, I.2, C.2.4, F.4

ISSN 0302-9743
ISBN-10 3-540-28060-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28060-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11533092 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3487

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence (LNAI)

- Vol. 3632: R. Nieuwenhuis (Ed.), *Automated Deduction – CADE-20*. XIII, 459 pages. 2005.
- Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), *Formal Concept Analysis*. X, 349 pages. 2005.
- Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), *Abstraction, Reformulation and Approximation*. XII, 376 pages. 2005.
- Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), *Conceptual Structures: Common Semantics for Sharing Knowledge*. XI, 467 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. 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. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), *Modeling and Using Context*. XIV, 572 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. Mitrovic (Eds.), *User Modeling*. 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. Stabler, J. Busquets, R. Moot (Eds.), *Logical Aspects of Computational Linguistics*. X, 363 pages. 2005.
- Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Ras, S. Tsumoto (Eds.), *Foundations of Intelligent Systems*. XIII, 700 pages. 2005.
- Vol. 3487: J. Leite, P. Torroni (Eds.), *Computational Logic in Multi-Agent Systems*. XII, 281 pages. 2005.
- Vol. 3476: J. 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.
- Vol. 3452: F. Baader, A. Voronkov (Eds.), *Logic for Programming, Artificial Intelligence, and Reasoning*. XI, 562 pages. 2005.
- Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), *Engineering Societies in the Agents World*. XIII, 349 pages. 2005.
- Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), *Massively Multi-Agent Systems I*. XI, 349 pages. 2005.
- Vol. 3445: G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro (Eds.), *Nonlinear Speech Modeling and Applications*. XIII, 433 pages. 2005.
- Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), *Constraint Solving and Language Processing*. VIII, 205 pages. 2005.
- Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), *Active Mining*. XII, 349 pages. 2005.
- Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), *Constraint Satisfaction and Constraint Logic Programming*. X, 217 pages. 2005.
- Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), *E-Government: Towards Electronic Democracy*. XIII, 311 pages. 2005.
- Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), *Multi-Agent and Multi-Agent-Based Simulation*. X, 265 pages. 2005.
- Vol. 3403: B. Ganter, R. Godin (Eds.), *Formal Concept Analysis*. XI, 419 pages. 2005.
- Vol. 3398: D.-K. Baik (Ed.), *Systems Modeling and Simulation: Theory and Applications*. XIV, 733 pages. 2005.
- Vol. 3397: T.G. Kim (Ed.), *Artificial Intelligence and Simulation*. XV, 711 pages. 2005.

- Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), *Agent Communication*. X, 261 pages. 2005.
- Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), *Adaptive Agents and Multi-Agent Systems II*. VIII, 313 pages. 2005.
- Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), *Applications of Declarative Programming and Knowledge Management*. X, 309 pages. 2005.
- Vol. 3374: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), *Environments for Multi-Agent Systems*. X, 279 pages. 2005.
- Vol. 3371: M.W. Barley, N. Kasabov (Eds.), *Intelligent Agents and Multi-Agent Systems*. X, 329 pages. 2005.
- Vol. 3369: V. R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), *Law and the Semantic Web*. XII, 249 pages. 2005.
- Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), *Argumentation in Multi-Agent Systems*. XII, 263 pages. 2005.
- Vol. 3359: G. Grieser, Y. Tanaka (Eds.), *Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*. XIV, 257 pages. 2005.
- Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), *Programming Multi-Agent Systems*. XIV, 249 pages. 2005.
- Vol. 3345: Y. Cai (Ed.), *Ambient Intelligence for Scientific Discovery*. XII, 311 pages. 2005.
- Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), *Spatial Cognition IV*. XIII, 519 pages. 2005.
- Vol. 3339: G.I. Webb, X. Yu (Eds.), *AI 2004: Advances in Artificial Intelligence*. XXII, 1272 pages. 2004.
- Vol. 3336: D. Karagiannis, U. Reimer (Eds.), *Practical Aspects of Knowledge Management*. X, 523 pages. 2004.
- Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), *Data Mining and Knowledge Management*. XIII, 263 pages. 2005.
- Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), *Advances in Artificial Intelligence – IBERAMIA 2004*. XX, 987 pages. 2004.
- Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), *Knowledge Exploration in Life Science Informatics*. X, 249 pages. 2004.
- Vol. 3301: G. Kern-Isberner, W. Rödter, F. Kulmann (Eds.), *Conditionals, Information, and Inference*. XII, 219 pages. 2005.
- Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), *RoboCup 2004: Robot Soccer World Cup VIII*. XVIII, 678 pages. 2005.
- Vol. 3275: P. Perner (Ed.), *Advances in Data Mining*. VIII, 173 pages. 2004.
- Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), *Machine Translation: From Real Users to Research*. XI, 392 pages. 2004.
- Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), *Grammatical Inference: Algorithms and Applications*. XI, 291 pages. 2004.
- Vol. 3259: J. Dix, J. Leite (Eds.), *Computational Logic in Multi-Agent Systems*. XII, 251 pages. 2004.
- Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), *Engineering Knowledge in the Age of the Semantic Web*. XVII, 517 pages. 2004.
- Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), *Artificial Intelligence and Symbolic Computation*. X, 285 pages. 2004.
- Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), *Natural Language Processing – IJCNLP 2004*. XVIII, 817 pages. 2005.
- Vol. 3245: E. Suzuki, S. Arikawa (Eds.), *Discovery Science*. XIV, 430 pages. 2004.
- Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), *Algorithmic Learning Theory*. XIV, 505 pages. 2004.
- Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), *KI 2004: Advances in Artificial Intelligence*. XI, 467 pages. 2004.
- Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), *Advances in Natural Language Processing*. XII, 488 pages. 2004.
- Vol. 3229: J.J. Alferes, J. Leite (Eds.), *Logics in Artificial Intelligence*. XIV, 744 pages. 2004.
- Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), *Formal Approaches to Agent-Based Systems*. VIII, 290 pages. 2004.
- Vol. 3215: M.G. Negoita, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part III*. LVII, 906 pages. 2004.
- Vol. 3214: M.G. Negoita, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part II*. LVIII, 1302 pages. 2004.
- Vol. 3213: M.G. Negoita, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part I*. LVIII, 1280 pages. 2004.
- Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), *Web Mining: From Web to Semantic Web*. IX, 201 pages. 2004.
- Vol. 3206: P. Sojka, I. Kopeček, K. Pala (Eds.), *Text, Speech and Dialogue*. XIII, 667 pages. 2004.
- Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), *Knowledge Discovery in Databases: PKDD 2004*. XIX, 560 pages. 2004.
- Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), *Machine Learning: ECML 2004*. XVIII, 580 pages. 2004.
- Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), *Inductive Logic Programming*. XI, 361 pages. 2004.
- Vol. 3192: C. Bussler, D. Fensel (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications*. XIII, 522 pages. 2004.
- Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), *Cooperative Information Agents VIII*. XI, 303 pages. 2004.
- Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), *Multiagent System Technologies*. XIII, 341 pages. 2004.
- Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), *Advanced Lectures on Machine Learning*. IX, 241 pages. 2004.

Preface

The notion of agency has recently increased its influence in the research and development of computational logic based systems, while at the same time significantly gaining from decades of research in computational logic. Computational logic provides a well-defined, general, and rigorous framework for studying syntax, semantics and procedures, for implementations, environments, tools, and standards, facilitating the ever important link between specification and verification of computational systems.

The purpose of the Computational Logic in Multi-agent Systems (CLIMA) international workshop series is to discuss techniques, based on computational logic, for representing, programming, and reasoning about multi-agent systems in a formal way. Former CLIMA editions were conducted in conjunction with other major computational logic and AI events such as CL in July 2000, ICLP in December 2001, FLoC in August 2002, and LPNMR and AI-Math in January 2004.

The fifth edition of CLIMA was held Lisbon, Portugal, in September 29–30, 2004. We, as organizers, and in agreement with the CLIMA Steering Committee, opted for co-location with the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004), wishing to promote the CLIMA research topics in the broader community of logics in AI, a community whose growing interest in multi-agent issues has been demonstrated by the large number of agent-related papers submitted to recent editions of JELIA.

The workshop received 35 submissions – a sensible increase from the previous edition. The submitted papers showed that the logical foundations of multi-agent systems are felt by a large community to be a very important research topic, upon which classical AI and agent-related issues are to be addressed.

In line with the high standards of previous CLIMA editions, the review process was very selective, the final acceptance rate being below 50%. A Program Committee of 24 top-level researchers from 11 countries and 12 additional reviewers selected 16 papers for presentation, authored by 46 researchers worldwide. The workshop program featured an invited lecture by Alessio Lomuscio (University College London) on Specification and Verification of Multiagent Systems, as well as a panel discussion organized by Marina de Vos (University of Bath) on Logic-Based Multi-agent Systems and Industry. Around 50 delegates attended the two-day event.

This book contains a selection, based on a second round of reviewing, of extended CLIMA V papers, and it starts with an invited contribution by Bożena Woźna and Alessio Lomuscio. The papers are divided into four parts: (i) foundations, (ii) architectures, (iii) interaction, and (iv) planning and applications. There follows a brief overview of the book.

Foundations. In the first paper of this book, *A Logic for Knowledge, Correctness, and Real Time*, Woźna and Lomuscio present and exemplify TCTLKD, a logic for knowledge, correctness and real time interpreted on real-time deontic interpreted systems, and extension to continuous time of deontic interpreted systems.

In *Dynamic Logic for Plan Revision in Intelligent Agents*, van Riemsdijk et al. present, with a sound and complete axiomatization, a dynamic logic for a propositional version of the agent programming language 3APL, tailored to handle the revision of plans.

Grossi et al. present in their paper *Contextual Taxonomies* a characterization of the notion of a taxonomy with respect to specific contexts, addressing problems stemming from the domain of normative system specifications for modelling multi-agent systems.

From Logic Programs Updates to Action Description Updates is where Alferes et al. propose a macro language for the language EVOLP and provide translations from some fragments of known action description languages into the newly defined one.

In *Dynamic Logic Programming: Various Semantics Are Equal on Acyclic Programs*, Homola investigates multi-dimensional dynamic logic programming, establishing some classes of programs for which several known semantics coincide.

Architectures. *Declarative Agent Control*, by Kakas et al., extends the architecture of agents based upon fixed, one-size-fits-all cycles of operation by providing a framework for the declarative specification of agent control in terms of *cycle theories*, which define possible alternative behaviors of agents.

In *Metareasoning for Multi-agent Epistemic Logics*, Arkoudas and Bringsjord present an encoding of a sequent calculus for a multi-agent epistemic logic in Athena, an interactive theorem proving system for many-sorted first-order logic, to enable its use as a metalanguage in order to reason about the multi-agent logic as an object language.

In *Graded BDI Models for Agent Architectures*, Casali et al. propose a general model for a graded BDI agent, specifying an architecture able to deal with the environment uncertainty and with graded mental attitudes.

Interaction. Dastani et al., in their article *Inferring Trust*, extend Liau's logic of Belief, Inform and Trust in two directions: with questions, and with a formalization of topics used to infer trust in a proposition from trust in another proposition.

In *Coordination Between Logical Agents*, Sakama and Inoue investigate on the use of answer set programming for belief representation, namely by addressing the problem of finding logic programs that combine the knowledge from different agents, while preserving some properties, useful to achieve agent coordination.

In *A Computational Model for Conversation Policies for Agent Communication*, Bentahar et al. propose a formal specification of a flexible persuasion proto-

col between autonomous agents, using an approach based on social commitments and arguments, defined as a combination of a set of conversation policies.

The last paper of this section is *Verifying Protocol Conformance for Logic-Based Communicating Agents*, by Baldoni et al., which describes a method for automatically verifying a form of “structural” conformance by translating AUML sequence diagrams into regular grammars and, then, interpreting the problem of conformance as a problem of language inclusion.

Planning and Applications. In the preliminary report *An Application of Global Abduction to an Information Agent Which Modifies a Plan Upon Failure*, Satoh uses a form of abductive logic programming called global abduction to implement an information agent that deals with the problem of plan modification upon action failure.

In *Planning Partially for Situated Agents*, Mancarella et al. use an abductive variant of the event calculus to specify planning problems as the base of their proposal for a framework to design situated agents capable of computing partial plans.

Han and Barber, in *Desire-Space Analysis and Action Selection for Multiple Dynamic Goals*, use macro actions to transform the state space for the agent’s decision problem into the desire space of the agent. Reasoning in the latter allows us to approximately weigh the costs and benefits of each of the agent’s goals at an abstract level.

Hirsch et al. conclude this book with the article *Organising Software in Active Environments*, in which they show how logic-based multi-agent systems are appropriate to model active environments. They do so by illustrating how the structuring of the “agent space” can represent both the physical and virtual structures of an application.

We would like to conclude with a glance at the future of this workshop series. The sixth CLIMA edition is being organized by Francesca Toni and Paolo Torroni, and will take place at the City University of London, UK, in June 27–29, 2005, in conjunction with the EU-funded SOCS Project Dissemination Workshop. CLIMA VI will feature a tutorial program and a competition, besides the usual technical content based on the presentation of papers.

We can not miss this opportunity to thank the authors and delegates, who made of CLIMA a very interesting and fruitful event; our generous Program Committee members who did not skimp on time to help us put together a very rich volume after two rounds of reviewing, discussion, and selection; and our sponsoring institutions, Universidade Nova de Lisboa, Fundação para a Ciência e Tecnologia, FBA, and AgentLink III.

April 2005

João Leite
Paolo Torroni

Organization

Workshop Chairs

João Leite, New University of Lisbon, Portugal

Paolo Torroni, University of Bologna, Italy

Program Committee

José Alferes, New University of Lisbon, Portugal

Gerd Brewka, University of Leipzig, Germany

Jürgen Dix, Technical University of Clausthal, Germany

Klaus Fisher, DFKI, Germany

Michael Fisher, The University of Liverpool, UK

James Harland, Royal Melbourne Institute of Technology, Australia

Katsumi Inoue, National Institute of Informatics, Japan

Sverker Janson, Swedish Institute of Computer Science, Sweden

João Leite, New University of Lisbon, Portugal

Yves Lespérance, York University, Canada

John-Jules Ch. Meyer, Utrecht University, The Netherlands

Leora Morgenstern, IBM, USA

Wojciech Penczek, Polish Academy of Sciences, Poland

Jeremy Pitt, Imperial College London, UK

Enrico Pontelli, New Mexico State University, USA

Fariba Sadri, Imperial College London, UK

Ken Satoh, National Institute of Informatics, Japan

Renate Schmidt, The University of Manchester, UK

Tran Cao Son, New Mexico State University, USA

Francesca Toni, University of Pisa, Italy

Wiebe van der Hoek, The University of Liverpool, UK

Paolo Torroni, University of Bologna, Italy

Makoto Yokoo, Kyushu University, Japan

Cees Witteveen, Delft University of Technology, The Netherlands

Additional Reviewers

Federico Banti

Thomas Eiter

Ulle Endriss

Ullrich Hustadt

Magdalena Kacprzak

Olle Olsson

Inna Pivkina
Chiaki Sakama

Kostas Stathis
Maciej Szreter

Gregory Wheeler
Yingqiang Zhang

Secretariat

Filipa Mira Reis

Sílvia Marina Costa

Local Organization

António Albuquerque
Duarte Alvim
Eduardo Barros

Jamshid Ashtari
Joana Lopes
Miguel Maurício

Miguel Morais
Sérgio Lopes

Steering Committee

Jürgen Dix, Technical University of Clausthal, Germany

João Leite, New University of Lisbon, Portugal

Fariba Sadri, Imperial College London, UK

Ken Satoh, National Institute of Informatics, Japan

Francesca Toni, University of Pisa, Italy

Paolo Torroni, University of Bologna, Italy

Sponsoring Institutions



Table of Contents

Foundations

A Logic for Knowledge, Correctness, and Real Time <i>Božena Woźna, Alessio Lomuscio</i>	1
Dynamic Logic for Plan Revision in Intelligent Agents <i>M. Birna van Riemsdijk, Frank S. de Boer, John-Jules Ch. Meyer</i> ...	16
Contextual Taxonomies <i>Davide Grossi, Frank Dignum, John-Jules Ch. Meyer</i>	33
From Logic Programs Updates to Action Description Updates <i>José Júlio Alferes, Federico Banti, Antonio Brogi</i>	52
Dynamic Logic Programming: Various Semantics Are Equal on Acyclic Programs <i>Martin Homola</i>	78

Architectures

Declarative Agent Control <i>Antonis C. Kakas, Paolo Mancarella, Fariba Sadri, Kostas Stathis, Francesca Toni</i>	96
Metareasoning for Multi-agent Epistemic Logics <i>Konstantine Arkoudas, Selmer Bringsjord</i>	111
Graded BDI Models for Agent Architectures <i>Ana Casali, Lluís Godo, Carles Sierra</i>	126

Interaction

Inferring Trust <i>Mehdi Dastani, Andreas Herzig, Joris Hulstijn, Leendert van der Torre</i>	144
Coordination Between Logical Agents <i>Chiaki Sakama, Katsumi Inoue</i>	161

A Computational Model for Conversation Policies for Agent Communication	
<i>Jamal Bentahar, Bernard Moulin, John-Jules Ch. Meyer, Brahim Chaib-draa</i>	178
Verifying Protocol Conformance for Logic-Based Communicating Agents	
<i>Matteo Baldoni, Cristina Baroglio, Alberto Martelli, Viviana Patti, Claudio Schifanella</i>	196
 Planning and Applications	
An Application of Global Abduction to an Information Agent which Modifies a Plan upon Failure - Preliminary Report	
<i>Ken Satoh</i>	213
Planning Partially for Situated Agents	
<i>Paolo Mancarella, Fariba Sadri, Giacomo Terreni, Francesca Toni</i> ...	230
Desire-Space Analysis and Action Selection for Multiple Dynamic Goals	
<i>David C. Han, K. Suzanne Barber</i>	249
Organising Software in Active Environments	
<i>Benjamin Hirsch, Michael Fisher, Chiara Ghidini, Paolo Busetta</i>	265
Author Index	281

A Logic for Knowledge, Correctness, and Real Time^{*}

Bożena Woźna and Alessio Lomuscio

Department of Computer Science,
University College London,
Gower Street, London WC1E 6BT,
United Kingdom
{B.Wozna, A.Lomuscio}@cs.ucl.ac.uk

Abstract. We present TCTLKD, a logic for knowledge, correctness and real time. TCTLKD is interpreted on real time deontic interpreted systems, and extension to continuous time of deontic interpreted systems. We exemplify the use of TCTLKD by discussing a variant of the “railroad crossing system”.

1 Introduction

Logic has a long tradition in the area of formal theories for multi-agent systems (MAS). Its role is to provide a precise and unambiguous specification language to describe, reason about, and predict the behaviour of a system.

While in the early 80’s existing logical formalisms from other areas such as philosophical logic, concurrency theory, etc., were imported with little of no modification to the area of MAS, from the late 80’s onwards specific formalisms have been designed, studied, and tailored to the needs of MAS. Of particular note is the case of epistemic logic, or the logic of knowledge.

Focus on epistemic logics in MAS began with the use of the modal logic system *S5* developed independently by Hintikka [1] and Aumann [2] in formal logic and economics respectively. This starting point formed the core basis of a number of studies that appeared in the past 20 years, including formalisations of group knowledge [3, 4, 5], combinations of epistemic logic with time [6, 7, 8], auto-epistemic logics [9, 10], epistemic updates [11, 12], broadcast systems and hypercubes [13, 14], etc. Epistemic logic is no longer a remarkable special case of a normal modal system, but has now become an area of study on its own with regular thematic workshops and conferences.

In particular, combinations of epistemic and temporal logics allow us to reason about the temporal evolution of epistemic states, knowledge of a changing world, etc. Traditionally, this is achieved by combining a temporal logic for discrete linear time [15, 16, 17] with the logic *S5* for knowledge [18]. Various classes

^{*} The authors acknowledge support from the EPSRC (grant GR/S49353), and the Nuffield Foundation (grant NAL/690/G).

of MAS (synchronous, asynchronous, perfect recall, no learning, etc.) can be identified in this framework, and axiomatisations have been provided [19, 20]. More recently, combinations of branching time logic CTL [21, 22, 23] with the epistemic logic $S5$ have been studied, and axiomatisation provided [8].

All efforts above have focused on a discrete model of time, either in its linear or branching versions. While this is useful and adequate in most applications, certain classes of scenarios (notably robotics and networking) require a model of time as a continuous flows of events.

In the area of timed-systems the modal logic TCTL has been suggested as an adequate formalism to model real time. In this paper we propose a logic (which we call TCTLKD) combining the temporal aspects of TCTL with the notions defined by the epistemic logic $S5$, as well as the correctness notion defined in [24]. This combination allows us to reason about the real time evolution of epistemic states, the correct functioning of multi-agent systems with respect to real time, and any combination of these.

Traditionally, the semantics of temporal epistemic logic is defined on variants of interpreted systems to provide an interpretation to the epistemic modalities. These use the notion of *protocol* to provide a basis for the action selection mechanism of the agents. Since we are working on real time, here we shall use the finer grained semantics of timed automata to model the agents' evolution. We then synchronise networks of timed automata to provide a general model of a MAS.

The rest of the paper is organised as follows. In Section 2 we define the concept of interpreted systems on real time by taking the parallel composition of timed automata. In Section 3 we define the logic TCTLKD as an extension to real time of the logic for knowledge and correctness as defined in [24, 25]. In Section 4 we provide a case study analysis to demonstrate its use in applications. We conclude in Section 5 by discussing related and future work on this subject.

2 Interpreted Systems over Real Time

Interpreted systems are traditionally defined as a set of infinite runs on global states [18]. In this model each run is a discrete sentence representing events. At each global state, each agent selects an action according to a (possibly non-deterministic) protocol. In this section we extend (discrete) interpreted systems to real time interpreted systems in two aspects. First, we specify the agents' behaviour by a finer grained semantics: timed automata. Second, by means of parallel composition of timed automata, we define a class of interpreted systems operating on real time.

We begin by recalling the concept of timed automata, as introduced in [26]. Timed automata are extensions of finite state automata with constraints on timing behaviour. The underlying finite state automata are augmented with a set of real time variables.

2.1 Timed Automata

Let $\mathbb{R} = [0, \infty)$ be a set of non-negative real numbers, $\mathbb{R}_+ = (0, \infty)$ be a set of positive real numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$ a set of natural numbers, and \mathcal{X} a finite set of real variables, called *clocks*. The set of *clock constraints* over \mathcal{X} is defined by the following grammar:

$$\mathbf{cc} := \text{true} \mid x \sim c \mid \mathbf{cc} \wedge \mathbf{cc},$$

where $x \in \mathcal{X}$, $c \in \mathbb{N}$, and $\sim \in \{\leq, <, =, >, \geq\}$. The set of all the clock constraints over \mathcal{X} is denoted by $\mathcal{C}(\mathcal{X})$. A *clock valuation* on \mathcal{X} is a tuple $v \in \mathbb{R}^{|\mathcal{X}|}$. The value of the clock x in v is denoted by $v(x)$. For a valuation v and $\delta \in \mathbb{R}$, $v + \delta$ denotes the valuation v' such that for all $x \in \mathcal{X}$, $v'(x) = v(x) + \delta$. Moreover, let \mathcal{X}^* be the set $\mathcal{X} \cup \{x_0\}$, where x_0 is a clock whose value is always 0, that is, its value does not increase with time as the values of the other clocks. Then, an *assignment* \mathbf{as} is a function from \mathcal{X} to \mathcal{X}^* , and the set of all the assignments over \mathcal{X} is denoted by $\mathfrak{A}(\mathcal{X})$. By $v[\mathbf{as}]$ we denote the valuation v' such that for all $x \in \mathcal{X}$, if $\mathbf{as}(x) \in \mathcal{X}$, then $v'(x) = v(\mathbf{as}(x))$, otherwise $v'(x) = 0$.

Let $v \in \mathbb{R}^{|\mathcal{X}|}$, the satisfaction relation \models for a clock constraint $\mathbf{cc} \in \mathcal{C}(\mathcal{X})$ is defined inductively as follows:

$$\begin{aligned} v &\models \text{true}, \\ v &\models (x \sim c) \quad \text{iff } v(x) \sim c, \\ v &\models (\mathbf{cc} \wedge \mathbf{cc}') \quad \text{iff } v \models \mathbf{cc} \text{ and } v \models \mathbf{cc}'. \end{aligned}$$

For a constraint $\mathbf{cc} \in \mathcal{C}(\mathcal{X})$, by $\llbracket \mathbf{cc} \rrbracket$ we denote the set of all the clock valuations satisfying \mathbf{cc} , i.e., $\llbracket \mathbf{cc} \rrbracket = \{v \in \mathbb{R}^{|\mathcal{X}|} \mid v \models \mathbf{cc}\}$.

Definition 1 (Timed Automaton). A timed automaton is a tuple $\mathcal{TA} = (3, L, l^0, \mathcal{X}, E, \mathfrak{I})$, where

- 3 is a finite set of actions,
- L is a finite set of locations,
- $l^0 \in L$ is an initial location,
- \mathcal{X} is a finite set of clocks,
- $E \subseteq L \times 3 \times \mathcal{C}(\mathcal{X}) \times \mathfrak{A}(\mathcal{X}) \times L$ is a transition relation,
- $\mathfrak{I} : L \rightarrow \mathcal{C}(\mathcal{X})$ is a function, called a location invariant, which assigns to each location $l \in L$ a clock constraint defining the conditions under which \mathcal{TA} can stay in l .

Each element e of E is denoted by $l \xrightarrow{a, \mathbf{cc}, \mathbf{as}} l'$, where l is a source location, l' is a target location, a is an action, \mathbf{cc} is the enabling condition for e , and \mathbf{as} is the assignment for e .

Note that we deal with “diagonal-free” automata. This is because ultimately we would like to verify MAS specified in this formalism, and the model checking methods for real time systems (based on the Difference Bound Matrices [27], variants of Boolean Decision Diagrams [28, 29], or SAT methods [30, 31, 32]) are problematic when the components of the systems are modelled by “diagonal automata”.

In order to reason about systems represented by timed automata, for a set of propositional variables \mathcal{PV} , we define a valuation function $V_{\mathcal{TA}} : L \rightarrow 2^{\mathcal{PV}}$, which assigns propositions to the locations.

Definition 2 (Dense State Space). *The dense state space of a timed automaton $\mathcal{TA} = (\mathcal{Z}, L, l^0, \mathcal{X}, E, \mathcal{J})$ is a structure $D(\mathcal{TA}) = (Q, q^0, \rightarrow)$, where*

- $Q = L \times \mathbb{R}^{|\mathcal{X}|}$ is the set of all the instantaneous states,
- $q^0 = (l^0, v^0)$ with $v^0(x) = 0$ for all $x \in \mathcal{X}$, is the initial state,
- $\rightarrow \subseteq Q \times (\mathcal{Z} \cup \mathbb{R}) \times Q$ is the transition relation, defined by action- and time-successors as follows:
 - for $a \in \mathcal{Z}$, $(l, v) \xrightarrow{a} (l', v')$ iff $(\exists \mathbf{cc} \in \mathcal{C}(\mathcal{X}))(\exists \mathbf{as} \in \mathcal{A}(\mathcal{X}))$ such that $l \xrightarrow{a, \mathbf{cc}, \mathbf{as}} l' \in E$, $v \in \llbracket \mathbf{cc} \rrbracket$, $v' = v[\mathbf{as}]$ and $v' \in \llbracket \mathcal{J}(l') \rrbracket$ (action successor),
 - for $\delta \in \mathbb{R}$, $(l, v) \xrightarrow{\delta} (l, v + \delta)$ iff $v + \delta \in \llbracket \mathcal{J}(l) \rrbracket$ (time successor).

For $(l, v) \in Q$, let $(l, v) + \delta$ denote $(l, v + \delta)$. A q -run ρ of a \mathcal{TA} is a sequence of instantaneous states: $q_0 \xrightarrow{\delta_0} q_0 + \delta_0 \xrightarrow{a_0} q_1 \xrightarrow{\delta_1} q_1 + \delta_1 \xrightarrow{a_1} q_2 \xrightarrow{\delta_2} \dots$, where $q_0 = q \in Q$, $a_i \in \mathcal{Z}$, and $\delta_i \in \mathbb{R}_+$ for each $i \in \mathbb{N}$. A run ρ is said to be *progressive* iff $\sum_{i \in \mathbb{N}} \delta_i$ is unbounded. A \mathcal{TA} is *progressive* if all its runs are progressive. For simplicity of presentation, we consider only progressive timed automata. Note that progressiveness can be checked as in [33].

2.2 Parallel Composition of Timed Automata

In general, we will model a multi-agent system by taking several timed automata running in parallel and communicating with each other. These concurrent timed automata can be composed into a global timed automaton as follows: the transitions of the timed automata that do not correspond to a shared action are interleaved, whereas the transitions labelled with a shared action are synchronised.

There are many different definitions of parallel composition. We use a *multi-way synchronisation*, requiring that each component that contains a communication transition (labelled by a shared action) has to perform this action.

Let $\mathcal{TA}_i = (\mathcal{Z}_i, L_i, l_i^0, E_i, \mathcal{X}_i, \mathcal{J}_i)$ be a timed automaton, for $i = 1, \dots, m$. To define a parallel composition of m timed automata, we assume that $L_i \cap L_j = \emptyset$ for all $i, j \in \{1, \dots, m\}$, and $i \neq j$. Moreover, by $\mathcal{Z}(a) = \{1 \leq i \leq m \mid a \in \mathcal{Z}_i\}$ we denote a set of numbers of the timed automata containing an action a .

Definition 3 (Parallel Composition). *The parallel composition of m timed automata \mathcal{TA}_i is a timed automaton $\mathcal{TA} = (\mathcal{Z}, L, l^0, E, \mathcal{X}, \mathcal{J})$, where $\mathcal{Z} = \bigcup_{i=1}^m \mathcal{Z}_i$, $L = \prod_{i=1}^m L_i$, $l^0 = (l_1^0, \dots, l_m^0)$, $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$, $\mathcal{J}(l_1, \dots, l_m) = \bigwedge_{i=1}^m \mathcal{J}_i(l_i)$, and a transition $((l_1, \dots, l_m), a, \mathbf{cc}, \mathbf{as}, (l'_1, \dots, l'_m)) \in E$ iff $(\forall i \in \mathcal{Z}(a)) (l_i, a, \mathbf{cc}_i, \mathbf{as}_i, l'_i) \in E_i$, $\mathbf{cc} = \bigwedge_{i \in \mathcal{Z}(a)} \mathbf{cc}_i$, $\mathbf{as} = \bigcup_{i \in \mathcal{Z}(a)} \mathbf{as}_i$, and $(\forall j \in \{1, \dots, m\} \setminus \mathcal{Z}(a)) l'_j = l_j$.*

Note that in the above any automaton is allowed to set a value of any clock, including the ones associated with other agents.

Let PV_i be a set of propositional variables containing the symbol **true**, $V_{\mathcal{TA}_i} : L_i \rightarrow 2^{PV_i}$ be a valuation function for the i th automaton, where $i \in \{1, \dots, m\}$,