Lecture Notes in Artificial Intelligence 3900

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Francesca Toni Paolo Torroni (Eds.)

Computational Logic in Multi-Agent Systems

6th International Workshop, CLIMA VI
London, UK, June 27-29, 2005
Revised Selected and Invited Papers

Springer

Volume Editors

Francesca Toni
Imperial College London
Department of Computing
180 Queen’s Gate, SW7 2BZ London, UK
E-mail: ft@doc.ic.ac.uk

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: Applied for

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-33996-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-33996-0 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: 11750734 06/3142 5 4 3 2 1 0

Preface

Computational Logic in Multi-Agent Systems (CLIMA) is a series of workshops aimed at promoting activity and exchange in the intersection of two vivid research areas. Since 2000, CLIMA has provided an opportunity to researchers to present their work on the application of general and declarative theories grounded on computational logic to multi-agent systems specification, semantics and procedures, and to confront ideas such as autonomy, deliberation, knowledge, commitment, openness, and trust with the computational logic paradigms. This research has encouraged the use of formal approaches to multi-agent systems research, and it has dealt with disparate issues such as implementations, environments, tools, and verification of computational systems.

The sixth edition of CLIMA was held at City University London, UK, on June 27–29, 2005. The workshop lasted three days and included an invited lecture by Robert A. Kowalski (Imperial College London) based on his last book. Sixty delegates from 15 countries (UK, Italy, France, Japan, Norway, The Netherlands, Cyprus, Germany, Canada, Ireland, Lithuania, Poland, Spain, Sweden, and Switzerland) attended the three-day event. Many of them were students.

CLIMA VI was innovative in many respects: beside the regular paper sessions, where the speakers presented papers selected from around 30 submissions, and the invited lecture, it hosted:

- a small tutorial program, with six lectures on cutting-edge CL-based agent technology,
- the first edition of the CLIMA contest, organized by Jürgen Dix and Mehdi Dastani, and
- the SOCS dissemination event reporting important results of an EU-funded project at the intersection of MAS and Logic Programming.

As we felt that we had enough material in our hands, motivated by the success of CLIMA VI we decided to propose a special edition to Springer. This volume features an invited article by Robert Kowalski, five tutorial papers presenting a view on the state of the art in CL-based MAS programming, four papers describing the implemented systems that participated in the contest, introduced by an invited paper by Jürgen Dix, Mehdi Dastani, and Peter Novak, a selection of technical papers, and an article about SOCS. All in all, this book is a state-of-the-art survey, authored by 56 researchers worldwide. A Program Committee of 26 top-level researchers and 40 additional reviewers contributed with their hard work and very fruitful comments and suggestions to the technical quality of this book: each of the 14 technical papers, 4 contest papers, and 7 invited papers enjoyed 3 to 5 reviews, in a 2-round process in which authors could reply to the reviewers' comments and argue in favor of their claims.

This volume opens with an invited article by Kowalski, presenting abductive logic programming (ALP) and its application in intelligent agents research. In

The Logical Way to Be Artificially Intelligent, Kowalski shows that ALP can be used to model reactive, proactive and preactive thinking, which can be performed by an agent as needed, or in advance, transforming high-level goals and beliefs into lower-level condition-action rule form. Kowalski also shows how ALP, proposed as a framework for constructing artificial agents, can also be used as a cognitive model of human agents.

The book continues with three sets of papers, covering *foundational aspects of agency*, *agent programming*, and *agent interaction and normative systems.* The articles of these sections are partly extended versions of papers presented at CLIMA VI, are partly novel, invited contributions that present, in a didactic style, advanced topics in CL-based MAS research.

Foundational Aspects of Agency

Setting the foundations of a theory of agency involves reasoning about notions such as agent knowledge, trust, beliefs, competence, abilities, and their relation with the environment, e.g., through actions or access to information. To this end, several formalisms have been used, adapted and refined, such as epistemic and doxastic logics and the situation calculus. This part of the book presents a selection of papers presented at CLIMA VI dealing with such foundational aspects of agency.

In *Ability in a Multi-Agent Context: a Model in the Situation Calculus*, Cholvy et al. provide a model of the notion of ability and its relation with the notion of action in a multi-agent context in the situation calculus. The authors provide a formal definition of the notion of ability of an agent to perform an action as the combination of its competences and some favorable conditions that allow it to perform that action, through the intermediary notion of "theoretical ability." The article is also concerned with the notion of ability of a group of agents. It deals with the dynamic notion of "occasional ability" (depending on the state of the world in which it is evaluated) and with the nontrivial problem of inferring the ability of the group from the abilities of the individuals of the group.

Nguyen advocates the use of modal logic programming to deal with reasoning about epistemic states of agents. In *Reasoning about Epistemic States of Agents by Modal Logic Programming*, he starts from the consideration that an agent should have knowledge about other agents in the system, and when such knowledge is only partial, it should nevertheless be able to reason about their epistemic states, possibly by simulating them, using some assumptions. To this end, Nguyen proposes an SLD-resolution calculus for modal logic programs in the multi-modal logic $KD4I_g5_a$. Such logic is intended for reasoning about belief and common belief of agents. The author provides soundness and completeness results and a formalization of McCarty's wise men puzzle using $KD4I_g5_a$ to demonstrate his ideas.

Epistemic logic frameworks for agents is also dealt within the paper *Strongly Complete Axiomatizations of "Knowing At Most" in Syntactic Structures.* The authors extend the logic language of syntactic structures based on syntactic assignments to model knowledge, with a new operator used to represent the position that an agent "knows at most" a given finite set of formulae. The syntactic approach is presented as a complementary approach to the modal approach, which can be used to model certain types of agents and certain types of situations that are difficult if not impossible to model with the modal approach, e.g., non-ideal – rather than ideal – agents, and situations where one is interested in explicit – rather than implicit – knowledge. In this paper, Ågotnes and Walicki present a strongly complete infinitary system and a strongly complete finitary system for a slightly weaker variant of the language.

Logical Spaces in Multi-Agent Only Knowing Systems presents a weak multi-agent system of "only knowing" and an analysis of the logical spaces that can be defined in it. Logical spaces can be used to express one agent's apprehension of the relations between concepts as understood by another agent, or, as the authors demonstrate, to define a situation in which an agent cannot conceive of a situation in which another agent has certain assumptions. Solhaug and Waaler's logic complements the approach to generalizing Levesque's "All I Know" system made by Halpern and Lakemeyer. The logic is defined entirely at the object level, with no reference to meta-concepts in the definition of the axiom system.

Trustworthiness by Default opens with an epigraph taken from Moses' address to the Israelites, which the authors use to introduce a framework for reasoning about relative trustworthiness. The framework considers sets of information sources as the basic trusted units, and is applied to conflict resolution and belief formation at various degrees of reliability. Klüwer and Waaler show how to construct a lattice of degrees of trustworthiness based on an assignment of relative trustworthiness to information source sets, to derive a priority structure, and apply it to the problem of forming the right opinion. Consolidated with an unquestioned knowledge base, this provides an unambiguous account of what an agent should believe, conditionally on which information sources are trusted.

Decision Procedure for a Fragment of Mutual Belief Logic with Quantified Agent Variables presents a deduction-based decision procedure for a fragment of mutual belief logic with quantified agent variables, $MBQL$. The language of $MBQL$ contains belief, everybody believes and mutual belief modalities, variables and constants for agents. The language of $MBQL$ is especially convenient to describe the properties of rational agents when the number of agents is not known in advance, and helps simplifying expressions when the exact number of agents is known instead. In this article, Pliuškevičius and Pliuškevičienė also propose a sequent calculus with invertible rules MBQ^* for the language of $MBQL$, and a loop-check-free sequent calculus for a fragment of $MBQL$.

Agent Programming

While modal logics have proven very useful to model and reason about agent mental states and their relations, the use of temporal logics and declarative programming is favored by many researchers when it comes to constructing operational agent systems and to implementing MAS based on their logical specifications. This volume includes four papers about state-of-the-art (multi-)agent frameworks based on extensions of logic programming, and one presenting tools for execution and proof based on temporal logic. This part contains four of them, followed by two CLIMA papers about agent programming.

In *Implementing Temporal Logics: Tools for Execution and Proof*, Fisher presents an overview of a selection of tools for execution and proof based on temporal logic, and outlines both the general techniques used and problems encountered in implementing them. The tools considered are mainly theorem-provers and (logic-based) agent programming languages, including clausal temporal resolution and executable temporal logics, and several of their implementations. This tutorial paper concentrates on general principles, with the aim of giving the reader an overview of the ways temporal logics are handled and used as the basis for both programming and verification.

Jason is a multi-agent systems development platform based on an interpreter for an extended version of AgentSpeak: an elegant, logic-based programming language based on the best known and most studied architecture for cognitive agents (the BDI architecture). In the tutorial paper *BDI Agent Programming in AgentSpeak Using* ***Jason***, Bordini and Hübner give an overview of the various features available in ***Jason***. The paper is intended for a general audience although some parts might be clearer for readers familiar with agent-oriented programming. The authors focus on the main features of ***Jason***, so that readers can assess whether ***Jason*** might be of interest, and give plenty of references to other papers and documentation where more detail and examples can be found.

The $\mathcal{KGP}$ model of agency, defined within the SOCS project, gives concrete guidelines for the formal specification of the knowledge of agents based on LP via a modular knowledge base and of the behavior of computees via a cycle theory providing flexible, declarative control of operation. In the tutorial paper *Using the* $\mathcal{KGP}$ *Model of Agency to Design Applications*, Sadri describes the main features of the $\mathcal{KGP}$ and gives user guidance on how the model can be used to develop applications. The paper concentrates on the abstract component of the $\mathcal{KGP}$, which consists of formal specifications of a number of different modules, including the knowledge bases, capabilities, transitions and control. For each of these, Sadri summarizes what is provided by the model, and through the platform implementing the model, and what is left to the users to specify according to the application requirements.

In *Multi-threaded Communicating Agents in Qu-Prolog*, Clark et al. summarize the key features of the multi-threaded Qu-Prolog language for implementing communicating agent applications. Internal threads of an agent communicate using the shared dynamic database used as a generalization of Linda tuple store. Threads in different agents communicate using either a thread-to-thread store and forward communication system or by a publish and subscribe mechanism in which messages are routed to their destinations based on content test subscriptions. The authors illustrate the features using an auction house application, which makes essential use of the three forms of inter-thread communication of Qu-Prolog. The agent bidding behavior is specified graphically as a finite state automaton and its implementation is essentially the execution of its state transition function.

Is an agent that focuses on one goal at a time better than an agent that frequently re-examines his commitments to ensure that he honors only those that are feasible? Or, how can such behaviors be compared with each other? The cycle theories of $\mathcal{KGP}$ agents define declaratively the possible alternative behaviors of agents, depending on their internal state and their perception of the external environment in which they are situated. In *Variety of Behaviors Through Profiles in Logic-based Agents*, Sadri and Toni show how by using this form of control specification one can specify different profiles of agents. In the paper, three different profiles are introduced, called "careful," "focussed," and "full planner" profile. The authors demonstrate how agent profiles would vary agent behaviors and what advantages they have with respect to factors in the application and in the environment, such as time-criticality.

This part is concluded by Knottenbelt and Clark's proposal: a simple event calculus representation of contracts and a reactive BDI agent architecture can be used to enable the monitoring and execution of contract terms and conditions. In *Contract-Related Agents*, the authors use the event calculus to deduce current and past obligations, obligation fulfilment and violation. By associating meta-information with the contracts, the agent is able to select which of its contracts with other agents are relevant to solving its goals by outsourcing. The agent is able to handle an extendable set of contract types such as standing contracts, purchase contracts and service contracts, without the need for a first-principles planner.

Agent Interaction and Normative Systems

A great deal of MAS research is devoted to studying specification and verification of interaction protocols, design of normative systems, representation of contexts, modelling other agents' mental states during interaction, and operational procedures for distributed intelligent reasoning, such as composition of information sources and reasoning using default beliefs about the possible outcomes of agent interaction. A tutorial and six technical papers compose this part, whose focus is not on individual agents but on social agents, their interaction and the norms that govern their systems.

Specification and Verification of Agent Interaction Using Abductive Reasoning, based on Chesani and Gavanelli's tutorial, provides an overview of the theory and tools produced within SOCS to design, define and test agent interaction protocols. The SOCS language for protocol specification is grounded on ALP. Its main element are social integrity constraints, used to specify relationships among happened events (e.g., messages or timeouts), expectations about future events, and predicates defined in the social knowledge base. This language aims to define open, extensible and not over-constrained protocols, following a social approach to agent interaction. A software tool called SOCS-SI allows one to verify *at execution time* if the agents conform to the defined protocols.

A complementary approach to verification of agents' conformance to protocols consists of inspecting the programs that encode their communicative behavior ("policies"), and verifying *a priori*, rather than at execution time, that they will actually produce interactions conforming to the public protocols. In this case, an issue is whether the test preserves the agents' capability of interacting, besides certifying the legality of their possible conversations. In the paper *Verification of Protocol Conformance and Agent Interoperability*, Baldoni and colleagues propose an approach based on the theory of formal languages. The conformance test is based on the acceptance of both the policy and the protocol by a special finite state automaton and it guarantees the interoperability of agents that are individually proven conformant.

How to connect norms specified by means of abstract terms ("persons driving vehicles may not access public parks") to norms specified via more concrete ones ("persons wheeling bicycles are allowed to access public parks")? An answer to this question is found in Grossi and coworkers' contextual taxonomies ("A counts as B in context C") for representing categorizing features of normative systems. *Contextual Terminologies* builds on work done on contextual taxonomies so as to add the possibility to deal with attributes or roles, i.e., binary relations besides concepts. This shift from simple taxonomies to rich description logic terminologies allows one to model more complex scenarios. The formalization is obtained by means of a formal semantics framework to reason within contexts and about contexts and their interplay.

Boella and van der Torre consider the design of normative multi-agent systems composed of both constitutive and regulative norms in their paper *Constitutive Norms in the Design of Normative Multiagent Systems.* They analyze the properties of constitutive norms, in particular their lack of reflexivity, and the trade-off between constitutive and regulative norms in the design of normative systems. As a methodology they use the metaphor of describing social entities as agents and of attributing mental attitudes to them. In this agent metaphor, regulative norms expressing obligations and permissions are modelled as goals of social entities, and constitutive norms expressing "counts as" relations are their beliefs.

Sakama and Inoue address the issue of combining knowledge of different information sources. Suppose a multi-agent system in which each agent has a knowledge base written in a common logic programming language. When two programs do not contradict each other, they may be combined into one by taking the union of programs. In non-monotonic logic programs, however, simple merging does not always reflect the meaning of individual programs. In *Combining Answer Sets of Nonmonotonic Logic Programs*, the authors study the compositional semantics of non-monotonic logic programs, supposing the answer set semantics of extended disjunctive programs. They provide methods for computing program composition and discuss their properties.

Speculative computation was first defined to cope with the incompleteness generated by communication failure or response delays. The idea is to allow the asking agent, while waiting for the slave agents to reply, to reason using default beliefs until replies are sent. *Speculative Constraint Processing with Iterative Revision for Disjunctive Answers* extends the framework proposed by Satoh and Yamamoto for speculative computation and iterative answer revision for yes/no questions. In this paper, Ceberio et al. present an extension of the framework for more general types of questions using constraint logic programming. They equip the framework with a sound operational model, which provably gives a correct answer with respect to the most recent replies.

When two agents have to interact it is important for each agent to know the other agent's intentions because this knowledge allows one to anticipate his future behavior. A method for this is presented in Demolombe and Otermin Fernandez's *Intention Recognition in the Situation Calculus and Probability Theory Frameworks*, and instantiated in the particular context of a pilot that interacts with an aircraft. The method is restricted to contexts where the agent only performs procedures in a given library of procedures, and where the system that intends to recognize the agent's intentions has a complete knowledge of the actions performed by the agent. An original aspect is that the procedures are defined for human agents and not for artificial agents, which makes the problem more complex than the standard one of plan recognition.

The First CLIMA Contest

The first CLIMA contest represented an important step towards collecting important benchmarks, identifying advantages/shortcomings, and advertising the use of CL to the broader MAS audience, and fostering integration of CL into existing agent-oriented software engineering frameworks. Dastani et al. open this section with the article *The First Contest on Multi-Agent Systems Based on Computational Logic*. In this paper, the authors describe the contest scenario and the winning criteria, and compare the performance of the competitors in the difficult task of determining the winning system. The other four short articles contain the description of the competing systems.

Coffey and Gaertner used ant-style pheromone trails as the basis for a pseudo-random walk procedure. Their agents, implemented in the concurrent LP language Qu-Prolog described in this book by Clark et al., explore the world uniformly based on information disseminated globally via a publish/subscribe mechanism. Interesting features of this approach, presented in *Implementing Pheromone-Based, Negotiating Forager Agents*, are the distribution of roles (collector/explorer) and the ability of agents to negotiate so as to increase the performance of collection/delivery task allocation. All in Prolog! (or almost all).

Cares et al. took the challenge from an agent-oriented software engineering perspective. The paper *Extending Tropos for a Prolog Implementation: a Case Study Using the Food-Collecting Agent Problem* uses the contest scenario as a case study to illustrate a method of obtaining a Prolog MAS implementation starting from a Tropos design. This solution includes autonomous behavior, beliefs, multiple role playing, communication and cooperation, and it ranked first in the contest together with the one implemented by Coffey and Gaertner.

In *Reactive Food Gathering*, Logie et al. describe a simple system, implemented as a collection of purely reactive agents, with no internal representation of their environment, which dynamically switch between a number of behaviors depending on interaction with their environment. The agents co-operate indirectly via environmental markers, generating an emerging global behavior that solves the problem.

This part closes with *Strategies for Multi-Agent Coordination in a Grid World Using Petri Nets*, by Nunes Gonçalves and Bittencourt. A distinguishing feature of the authors' solution is the focus on coordination. Their agents implement a strategy to select the most capable agent in the environment so as to execute tasks that they cannot execute themselves. The specification of the multi-agent system is made using Petri Nets.

Project Report

The SOCS dissemination event, affiliated to CLIMA VI, presented several key aspects of the EU-funded European project SOCS (SOcieties of ComputeeS), one of the main sponsors of CLIMA VI. Computees are agents in computational logic. From January 2001 to June 2005, in a joint research effort involving six European academic institutions, SOCS pushed the state of the art in LP and in MAS research, producing as its main results the KGP model of agency and the SOCS social model based on social integrity constraints. During this event, the speakers Toni, Kakas, Bracciali, and Alberti presented the declarative and operational models for agents and multi-agent systems and the formal properties of agents and agent systems developed within SOCS. Torroni discussed possible guidelines for evaluating intelligent systems of reasoning agents, building on the SOCS experience. In the last paper of this volume, Toni presents the challenges and outcomes of SOCS.

Further information about CLIMA VI is available from the website `http://clima.deis.unibo.it/`. General information about the workshop series, with links to past and future events, can be found on the CLIMA workshop series home page, `http://centria.di.fct.unl.pt/~clima/`. The next CLIMA edition is organized by Katsumi Inoue, Ken Satoh and Francesca Toni. It will take place in Hakodate, Japan, on May 8-9, 2006, in conjunction with AAMAS and it will host the second CLIMA contest.

While wishing you a good read, we thank the local organizer, the contest organizers, the website administrators, the Program Committee members, the additional reviewers, the authors and the delegates, who contributed to a very interesting and inspiring event, and the sponsors: the Association for Logic Programming, AgentLink III, and the Fifth Framework EU Programme through the SOCS Project.

March 2006

Francesca Toni
Paolo Torroni

Organization

Workshop Chairs

Francesca Toni, Imperial College London, UK
Paolo Torroni, University of Bologna, Italy

Program Committee

José Alferes, New University of Lisbon, Portugal
Rafael Bordini, University of Durham, UK
Gerd Brewka, University of Leipzig, Germany
Jürgen Dix , Technical University of Clausthal, Germany
Thomas Eiter, Vienna University of Technology, Austria
Klaus Fischer, DFKI, Germany
Michael Fisher, The University of Liverpool, UK
James Harland, Royal Melbourne Institute of Technology, Australia
Katsumi Inoue, National Institute of Informatics, Japan
Antonis Kakas, University of Cyprus, Cyprus
Evelina Lamma, University of Ferrara, Italy
João Leite, New University of Lisbon, Portugal
Paolo Mancarella, University of Pisa, Italy
Paola Mello, University of Bologna, Italy
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
Kostas Stathis, City University London, UK
Wiebe van der Hoek, The University of Liverpool, UK
Cees Witteveen, Delft University of Technology, The Netherlands

CLIMA Steering Committee

Jürgen Dix, Technical University of Clausthal, Germany
Michael Fisher, The University of Liverpool, UK
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

Contest Organizers

Mehdi Dastani, Utrecht University, The Netherlands
Jürgen Dix, Technical University of Clausthal, Germany

Additional CLIMA Reviewers

Marco Alberti
Federico Banti
Jamal Bentahar
Andrea Bracciali
Andreas Brüning
Lisette van der Burgh
Marco Cadoli
Carlos Cares
Federico Chesani
Pierangelo Dell'Acqua
Agostino Dovier
Nivea Ferreira
Marco Gavanelli
Davide Grossi
Jomi Hubner
Ullrich Hustadt
Magdalena Kacprzak
Peep Küngas
Sławomir Lasota
Ambra Molesini
Àlvaro Moreira
Adriaan ter Mors
Yasuo Nagai
Brendan Neville
Peter Novak
Regimantas Pliuškevičius
Ian Pratt-Hartmann
Daniel Ramirez-Cano
Rossella Rubino
Claudio Schifanella
Kostas Stathis
Andrzej Szałas
Giacomo Terreni
Arianna Tocchio
Satoshi Tojo
Krzysztof Trojanowski
Francesco Viganò
Gregory Weeler
Mathijs de Weerdt
Pınar Yolum

Web Support

Fabio Bucciarelli
Federico Chesani

Local Organisation

Kostas Stathis

Sponsoring Institutions

Table of Contents

Foundational Aspects of Agency

Agent Programming

Agent Interaction and Normative Systems

The First CLIMA Contest

Project Report