Multi-agent Systems in Computational Logic: Challenges and Outcomes of the SOCS Project

Francesca Toni

Department of Computing, Imperial College London
180 Queen's Gate, SW7 2BZ London, UK
ft@doc.ic.ac.uk

Abstract. The SOCS project (A computational logic model for the description, analysis and verification of global and open SOcieties of heterogeneous ComputeeS), funded by the European Commission under the Fifth Framework, Future and Emerging Technologies programme, has been one of the main sponsors of CLIMA VI. This short article outlines the project's main challenges and its main outcomes.

1 Introduction

The SOCS project [32] was concerned with the development of a computational logic model for the description, analysis and verification of global and open SOcieties of heterogeneous ComputeeS, where computees are agents realised in computational logic. SOCS was funded by the European Commission under the Fifth Framework, Future and Emerging Technologies programme, within the Global Computing (GC) proactive initiative. GC research provides the foundations for the development of large-scale general purpose computer systems that have dependably predictable behaviour, for the needs of a distributed world [24]. SOCS addressed the challenges of the GC initiative with a consortium composed of six European partners, based in Italy, the UK, and Cyprus.[1] Its original aims were:

- To provide a computational logic model for the description, analysis and verification of global and open societies of heterogeneous computees, intended as abstractions of the entities that populate open and global computing environments;
- To provide prototype implementations of computees and their societies;
- To run experiments based on various scenaria to ground and test the model.

SOCS interpreted the GC challenges under an agent-oriented perspective, with a Logic Programming (LP)-based approach. In particular, the project adopted

[1] These were, respectively, the universities of Pisa, Bologna and Ferrara (Italy), Imperial College London and City University London (UK), and Cyprus University. The project was coordinated by Imperial College London. The project started in January 2002 and finished in June 2005.

variants of Abductive Logic Programming, Constraint Logic Programming and Logic Programming with Priorities, appropriately integrated to deal with agents and GC scenaria.

In this short article, we give an overview of the project, its main challenges and outcomes.

2 The KGP model of agency

The KGP model of agency [25, 28] gives concrete guidelines for the formal specification of the knowledge of computees (via a modular computational logic-based knowledge base, partitioned in modules devoted to the different reasoning tasks of planning, reactivity, goal decision, temporal reasoning and identification of preconditions of actions) and of the behaviour of computees (via a computational logic-based cycle theory providing the flexible, declarative control for the operation of computees [26]). The model can be seen at two different levels: on one hand, it is rather concrete, as it exactly specifies what the internal configuration of the computee is; on the other hand, its control component can be varied to obtain heterogeneous behaviour, and is abstract, in that it can be used also for other agent models, independently of their structure/configuration/design.

We have tested the KGP model satisfactorily on a number of applications (including e-commerce and ambient intelligence scenaria [34]). More insights about how to specify and execute computees can be found in this book in the tutorial paper [30].

3 The SOCS model of agent societies

The SOCS society model [29, 10] gives concrete guidelines for the formal specification of the interaction among computees that form a society, and for the definition of a computational logic-based architecture for computee interactions. A layered architecture is proposed where the society defines the allowed interaction protocols, which in turn are defined by means of *Social Integrity Constraints* (ICs). The society's knowledge is defined as an abductive logic program [5], where ICs are used in order to express constraints on the communication patterns of computees and expected communicative acts ("*expectations*") are expressed as abducible predicates.

Expectations, whose intuition recalls the usual deontic operators of permission, obligation, and prohibition [13], are used to provide a semantics to both agent communication languages and to interaction protocols [3]. The resulting model is based on a declarative (logic) representation and therefore it is easy to understand, and close to an operational model and suitably usable in order to achieve an implementation of societies of computees based on their formal specifications [9]. Finally, thanks to the link between formal specification and implementation, the model provides a good ground for the automatic verification and formal proof of properties [6].

We have tested the society model satisfactorily on a number of applications (including resource exchange [14], e-commerce protocols [4], combinatorial auctions [7]). A repository of protocols specified using ICs is being maintained and is publicly available through the project's home page.[2] More insights about how to specify and execute societies of computees can be found in this book in the tutorial paper [17].

4 Computational models

Both the KGP model and the society model are equipped with correct computational counterparts ([16] and [15], respectively). These computational models are heavily based upon proof procedures for (various extensions of) logic programming. In particular, the operational model for KGP agents relies upon CIFF [19], a proof procedure for abductive logic programming with constraints, and Gorgias [18], for logic programming with priorities, and the operational model for societies of agents relies upon SCIFF [1], a proof procedure for abductive logic programming with events and expectations. These procedures have been obtained by adapting and suitably extending two existing proof procedures for logic programming, namely Fung and Kowalski's IFF procedure for abductive logic programming [22], for CIFF and SCIFF, and Kakas and Toni's argumentation-based procedure for negation as failure in logic programming [27], for Gorgias. The overall operational models are sound and (in some cases) complete with respect to the abstract KGP model and model for societies of agents, respectively, and form a solid bridge between the models and their implementations within the PROSOCS and SOCS-SI platforms (see section 6 below).

5 Properties

A great deal of the project activities has been devoted to formalising and studying properties of agents and agent systems. The SOCS approach to properties is formal, and it aims at exploiting the potential of the declarative LP paradigm for giving a precise specification of properties and for allowing their formal verification. Moreover, the double declarative and operational reading of LP supports both an abstract description of systems and their (expected) properties, and mechanisms to implement them. Descriptions and mechanisms are closely related to each other so that properties enjoyed by the models are easily reflected in the implementations.

We have compiled a catalogue of concrete properties, demonstrating:

1. The effectiveness of our logic programming approach to modeling computees and their societies. This facilitates the formalisation of formal properties and prediction of behaviour without resorting to empirical methods.

[2] `http://edu59.deis.unibo.it:8079/SOCSProtocolsRepository/jsp/index.jsp`

2. The consequences of some of the design choices. For example, we have identified coherence properties for computees showing some of the benefits that result directly as a consequence of the choice of goal and action selection functions in the computee model. Another example concerns the design of the social infrastructure that provably allows verification of protocol properties automatically [12].
3. The versatility of the computee and society models. For example, we have investigated how we can specify different profiles of behaviour in computees and how such profiles could alter the behaviour of computees [31].

We have identified three broad areas for investigating properties of (societies of) computees [23, 2]:

- Properties of individual computees (agents), including agent profiles [31];
- Properties of the society infrastructure;
- Properties related to protocol conformance [20].

These properties help showing the effectiveness of the computational logic approach in modeling computees and societies, in the sense of facilitating formalisation of properties and prediction of behaviour without the need to resort to empirical methods. They also help exploring the consequences of our design choices.

6 Implementation and Experimentation

We have developed a prototype implementation and platform for computees and societies (PROSOCS [33] and SOCS-SI [11]), which implement the models and have been used for extensive experimentation in the later phases of the project. The experimentation has also been conducted to confirm or disprove properties of the models. The SOCS prototype has also been used to provide a practical basis for the design of combinatorial auctions, which require aggregate behaviour of computational entities and tools.

The PROSOCS platform provides the reasoning and communication capabilities a computee needs in order to operate in a GC environment. The agent developer, as a result, is only required to specify the set of logic theories that describe the background knowledge necessary for the agent to operate within a specific application domain. PROSOCS uses SICStus Prolog for inference-based components (CIFF and Gorgias).

Analogously, SOCS-SI supports the declarative formalisation and the automated verification of the social aspects of a SOCS application. SOCS-SI is general in its scope, and has been interfaced to other implemented agent platforms, such as Jade and tuProlog, and to other non-agent related communication platforms. SOCS-SI uses SICStus Prolog, and in particular its CHR library [21], for the reasoning and verification part [8].

Both PROSOCS and SOCS-SI use JXTA for inter-agent communication and agent discovery, and Java to implement the supporting applications, integrate Prolog and build the GUIs. SOCS-SI and PROSOCS communicate through JXTA.

7 SOCS dissemination meeting at CLIMA VI

The SOCS dissemination event presented several key aspects of the project's approach, and discussed some open issues. The speakers Antonis Kakas,[3] Andrea Bracciali,[4], Marco Alberti,[5] presented the operational models for agents and multi-agent systems and the formal properties of agents and agent systems developed within SOCS. Paolo Torroni[6] attempted to provide guidelines for evaluating intelligent systems of reasoning agents, building on the SOCS experience.

Acknowledgments

This work was funded by the IST programme of the EC, FET under the IST-2001-32530 SOCS project, within the GC proactive initiative. The author is grateful to all participants of the SOCS consortium for inputs to this article and for their hard work throughout the project. The author is also grateful to Paolo Torroni for some helpful suggestion for an earlier version of this article.

References

1. The *SCIFF* abductive proof procedure home page. `http://lia.deis.unibo.it/research/sciff/`, 2005.
2. M. Alberti, F. Athienitou, A. Bracciali, F. Chesani, U. Endriss, M. Gavanelli, A. Kakas, E. Lamma, W. Lu, P. Mancarella, P. Mello, F. Sadri, K. Stathis, F. Toni, , and P. Torroni. Verifiable properties of societies of computees. Technical report, SOCS Consortium, 2005. Deliverable D13. Available electronically from the SOCS project web site: `http://lia.deis.unibo.it/research/socs/guests/publications/`.
3. M. Alberti, A. Ciampolini, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. A social ACL semantics by deontic constraints. *Lecture Notes in Artificial Intelligence*, 2691:204–213, June 16–18 2003.
4. M. Alberti, D. Daolio, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Specification and verification of agent interaction protocols in a logic-based system. In Hisham M. Haddad, Andrea Omicini, and Roger L. Wainwright, editors, *Proceedings of the 19th Annual ACM Symposium on Applied Computing (SAC 2004). Special Track on Agents, Interactions, Mobility, and Systems (AIMS)*, pages 72–78, Nicosia, Cyprus, March 14–17 2004. ACM Press.
5. M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. An Abductive Interpretation for Open Societies. *Lecture Notes in Artificial Intelligence*, 2829:287–299, September 23–26 2003.
6. M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Specification and verification of agent interactions using Social Integrity Constraints. *Electronic Notes in Theoretical Computer Science*, 85(2), 2003.

[3] University of Cyprus, `http://www2.cs.ucy.ac.cy/~antonis/`
[4] University of Pisa, Italy, `http://www.di.unipi.it/~braccia/`
[5] University of Ferrara, Italy `http://www.ing.unife.it/docenti/malberti.html`
[6] University of Bologna, Italy, `http://lia.deis.unibo.it/~pt/`

7. Marco Alberti, Federico Chesani, Marco Gavanelli, Alessio Guerri, Evelina Lamma, Paola Mello, and Paolo Torroni. Expressing interaction in combinatorial auction through social integrity constraints. *Intelligenza Artificiale*, II(1):22–29, 2005.
8. Marco Alberti, Federico Chesani, Marco Gavanelli, and Evelina Lamma. The CHR-based Implementation of a System for Generation and Confirmation of Hypotheses. In Armin Wolf, Thom Frühwirth, and Marc Meister, editors, *19th Workshop on (Constraint) Logic Programming W(C)LP 2005*, number 2005-01 in Ulmer Informatik-Berichte, pages 111–122, 2005.
9. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. A logic based approach to interaction design in open multi-agent systems. In *Proceedings of the 13th IEEE international Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), 2nd international workshop "Theory And Practice of Open Computational Systems (TAPOCS)"*, pages 387–392, Modena, Italy, June 14–16 2004. IEEE Press.
10. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. The SOCS computational logic approach for the specification and verification of agent societies. 3267:324–339, 2005.
11. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Compliance verification of agent interaction: a logic-based tool. *Applied Artificial Intelligence*, 20(4-5), 2006.
12. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Security protocols verification in Abductive Logic Programming: a case study. In Oguz Dikenelli, Marie-Pierre Gleizes, and Andrea Ricci, editors, *Proceedings of ESAW'05, Kuşadası, Aydın, Turkey, October 26-28, 2005*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2006.
13. Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, Giovanni Sartor, and Paolo Torroni. Mapping deontic operators to abductive expectations. *Computational and Mathematical Organization Theory*, 2006.
14. Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Modeling interactions using *Social Integrity Constraints*: A resource sharing case study. *Lecture Notes in Artificial Intelligence*, 2990:243–262, May 2004.
15. Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. The $\mathcal{S}$CIFF abductive proof-procedure. *Lecture Notes in Artificial Intelligence*, 3673:135–147, 2005.
16. Andrea Bracciali, Neophytos Demetriou, Ulle Endriss, Antonis Kakas, Wenjin Lu, Paolo Mancarella, Fariba Sadri, Kostas Stathis, Francesca Toni, and Giacomo Terreni. The KGP model of agency: Computational model and prototype implementation. *Lecture Notes in Artificial Intelligence*, 3267:340–367, 2005.
17. Federico Chesani, Marco Gavanelli, Marco Alberti, Evelina Lamma, Paola Mello, and Paolo Torroni. Specification and verification of agent interaction using abductive reasoning. In this volume.
18. N. Demetriou and A. C. Kakas. Argumentation with abduction. In *Proceedings of the fourth Panhellenic Symposium on Logic*, 2003.
19. U. Endriss, P. Mancarella, F. Sadri, G. Terreni, and F. Toni. The CIFF proof procedure for abductive logic programming with constraints. *Lecture Notes in Artificial Intelligence*, 3229:680–684, 2004.
20. U. Endriss, N. Maudet, F. Sadri, and F. Toni. Protocol conformance for logic-based agents. Morgan Kaufmann Publishers, August 2003.
21. T. Frühwirth. Theory and practice of constraint handling rules. *Journal of Logic Programming*, 37(1-3):95–138, October 1998.

22. T. H. Fung and R. A. Kowalski. The IFF proof procedure for abductive logic programming. *Journal of Logic Programming*, 33(2):151–165, November 1997.
23. M. Gavanelli, E. Lamma, P. Torroni, P. Mello, K. Stathis, P. Moraïtis, A. C. Kakas, N. Demetriou, G. Terreni, P. Mancarella, A. Bracciali, F. Toni, F. Sadri, and U. Endriss. Computational model for computees and societies of computees. Technical report, SOCS Consortium, 2003. Deliverable D8. Available electronically from the SOCS project web site: `http://lia.deis.unibo.it/research/socs/guests/publications/`.
24. Global Computing, Future and Emerging Technologies. Co-operation of autonomous and mobile entities in dynamic environments. Home Page: `http://www.cordis.lu/ist/fetgc.htm`, 2000.
25. A. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. A logic-based approach to model computees. Technical report, SOCS Consortium, 2003. Deliverable D4. Available electronically from the SOCS project web site: `http://lia.deis.unibo.it/research/socs/guests/publications/`.
26. A. C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. Declarative agent control. *Lecture Notes in Artificial Intelligence*, 3487:96–110, 2005.
27. A. C. Kakas and F. Toni. Computing negation as failure via argumentation. *Journal of Logic and Computation*, 9:515–562, 1999.
28. A.C. Kakas, P. Mancarella, F. Sadri, K. Stathis, and F. Toni. The KGP model of agency. In R. Lopez de Mantaras and L. Saitta, editors, *Proceedings of the Sixteenth European Conference on Artificial Intelligence, Valencia, Spain (ECAI 2004)*. IOS Press, August 2004.
29. P. Mello, P. Torroni, M. Gavanelli, M. Alberti, A. Ciampolini, M. Milano, A. Roli, E. Lamma, F. Riguzzi, and N. Maudet. A logic-based approach to model interaction amongst computees. Technical report, SOCS Consortium, 2003. Deliverable D5. Available electronically from the SOCS project web site: `http://lia.deis.unibo.it/research/socs/guests/publications/`.
30. Fariba Sadri. Using the KGP model of agency to design applications. In this volume.
31. Fariba Sadri and Francesca Toni. Variety of behaviours through profiles in logic-based agents. In this volume.
32. SOcieties of ComputeeS (SOCS). A computational logic model for the description, analysis and verification of global and open SOcieties of heterogeneous ComputeeS. IST-2001-32530. `http://lia.deis.unibo.it/research/socs/`, 2002-2005.
33. K. Stathis, A.C. Kakas, W. Lu, N. Demetriou, U. Endriss, and A. Bracciali. PROSOCS: A platform for programming software agents in computational logic. *Applied Artificial Intelligence*, 20(4-5), 2006.
34. K. Stathis and F. Toni. Ambient Intelligence using KGP Agents. *Lecture Notes in Artificial Intelligence*, 3295:351–362, 8–10 November 2004.