Formalization and Verification of Interaction Protocols

Federico Chesani

DEIS – University of Bologna,
viale Risorgimento 2,
40136 - Bologna, Italy
fchesani@deis.unibo.it

1 Introduction

In recent years, the study of protocols and their properties has been one of the most investigated issues in distributed and multi-process systems research, and they are indeed one of the key component of Multi-Agent Systems. Several formal languages for defining protocols and properties have been proposed within different research communities. Some of the most common objectives of such languages include the ability to: *formalize the protocols* in an easy and clear way for human users; *define the protocols abstracting away from the internal architecture* of the participating peers; be able to specify and investigate *properties*, and help the *implementation* of the peers.

Most of the current research on protocols falls into one of the following four main areas of interest: *protocol formalization*, where languages for specifying protocol has been intensively studied not only in MAS research [3, 5, 9], but also in the broader community of distributed and multi-process systems [6]; *standardization*, aimed at guaranteeing interoperability between heterogeneous agents in open computing environment [2, 5]; *protocol properties*, where tools for proving properties are of utmost importance in the MAS community [4] and in the security protocols community [1]; and finally *specific application domain protocols*, where argumentation and negotiation are examples of domains where the study of protocols is driven by the need to address specific features [8].

2 Goal and current status of the research

Interaction protocols are a necessary component of open and heterogeneous systems. Definition languages, formal semantics, verification tools and proof of properties are the main issues that must be considered to achieve effective interaction protocols. Logic Programming can greatly contribute to tackle these issues, due to its declarative character, as well as its possibility of automatically proving theorems. My doctoral research programme aims to adopt Logic Programming, and in particular Abduction, for solving these problems. To this end, I envisage to pursue the following research directions:

- study of the *state-of-the-art* for protocol definition languages;
- definition of a framework where concepts like protocol, property, compliance and interaction are defined in a coherent and unified way;
- formulation of a language for protocol specification;
- development and implementation of tools for the verification of interaction and the proof of properties;
- development of a methodology for multi agent systems design.

My research activity on protocols has started within the SOCS project, with its definition of a general model for societies of agents, and of a protocol specification language based on *Social Integrity Constraints* (IC_s). An abductive proof-procedure, called SCIFF, has been developed and proven to be sound and complete with respect to its declarative semantics. Using SCIFF it is possible to verify if a certain interaction is *compliant* with a protocol specified using IC_s [7].

My research activity, in particular, has focussed on studying the automatic translation of other protocol definition languages such as AUML into IC_s, and on studying different methods for extending SCIFF in order to prove protocol properties automatically. In particular, we are considering several SCIFF extensions for generating a proof based on refutation. Up to now, we are able to (dis)prove a property only in certain cases. Next, I intend to investigate completeness so as to able to rely on SCIFF for refutation-based property proving.

References

1. David A. Basin, S. Mödersheim, and L. Viganò. An on-the-fly model-checker for security protocol analysis. In *ESORICS*, pages 253–270, 2003.
2. FIPA: Foundation for Intelligent Physical Agents. http://www.fipa.org/
3. N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In C. Castelfranchi and W. L. Johnson, editors, *Proc. of AAMAS-2002*, pages 535–542. ACM Press.
4. F. Guerin and J. Pitt. Proving properties of open agent systems. In C. Castelfranchi and W. L. Johnson, editors, *Proc. AAMAS-2002*, pages 557–558. ACM Press.
5. M. P. Huget. Agent uml notation for multiagent system design. *Internet Computing, IEEE*, Vol. 8(4):63–71, July-Aug. 2004.
6. K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use.*, vol. 1 of *Mon. in Theor. Computer Science. An EATCS Series.* Springer, 2 edition, X 1997.
7. Alberti M., Chesani F., Gavanelli M., Lamma E., Mello P., and Torroni P., Compliance verification of agent interaction: a logic-based software tool. *Applied Artificial Intelligence*, 2005. To appear.
8. P. J. McBurney. *Rational Interaction.* PhD thesis, University of Liverpool, 2002.
9. P. Yolum and M.P. Singh. Flexible protocol specification and execution: applying event calculus planning using commitments. In C. Castelfranchi and W. L. Johnson, editors, *Proc. AAMAS-2002*, pages 527–534. ACM Press.