A demonstration of *SOCS-SI*⋆

Marco Alberti[1], Federico Chesani[2], Marco Gavanelli[1], Evelina Lamma[1], Paola Mello[2], and Paolo Torroni[2]

[1] DI, Università di Ferrara (Italy)
{malberti, mgavanelli, elamma}@ing.unife.it
[2] DEIS, Università di Bologna (Italy)
{fchesani, pmello, ptorroni} @deis.unibo.it

1 Introduction

The demonstration that we propose is meant to show *SOCS-SI* [1], a logic-based tool for verification of compliance of agent interaction to protocols based on the notion of social expectation.

The inputs to *SOCS-SI* are:

- a text file containing the *Social Integrity Constraints* (SICs in the following), used for expressing interaction protocols;
- a text file containing an (abductive) logic program (*Social Organization Knowledge Base*, SOKB in the following), used to express declarative knowledge, such as the value of time deadlines;
- a *history* of events, representing the agent behaviour, recorded from a source of events. So far, the implementation of *SOCS-SI* accepts as sources (*i*) the user prompt, (*ii*) a log file, or (*iii*) a network-based tool for the observation of the agent interaction.

By means of an abductive proof procedure, *SOCS-SI* uses SICs and SOKB to generate *expectations* about the "ideal" social behaviour of agents, i.e., compliant to interaction protocols, given a (partial) history of events. *SOCS-SI* also checks if the actual agent behaviour corresponds to such expectations. Based on that, it either outputs an answer of *fulfillment* (if the agent behaviour is compliant to the interaction protocols) or *violation* (if the agent behaviour is not compliant to interaction protocols).

The demonstration will start with a very brief presentation of the notions of protocols, social semantics and expectations, and compliance. The proof procedure will then be briefly (and informally) introduced, and the functioning of the tool, where SICs, SOKB and history files can be visualized.

The scenario presented in the demonstration will be a "first price sealed bid" auction; for this scenario, we will show an example of fulfillment and one of violation.

More examples will be shown, if time allows.

⋆ This work is partially funded by the Information Society Technologies programme of the European Commission under the IST-2001-32530 project, and by the national MIUR COFIN 2003 projects "Sviluppo e verifica di sistemi multi-agente basati sulla logica" and "La gestione e la negoziazione automatica dei diritti sulle opere dell'ingegno digitali: aspetti giuridici ed informatici".

2 The *SOCS-SI* tool

The Social Infrastructure (SI) tool is an implementation of a logic programming based framework defined within the SOCS project [2]. The main idea of the framework is to exploit abduction in order to verify the compliance of agents behavior in respect to interactions protocols. Protocols are represented by means of integrity constraints, while expectations about future (and past) agent behaviors are mapped onto abducible predicates.

In order to verify compliance, a new proof procedure, *SC*-IFF, has been defined and implemented [3]. The *SC*-IFF is mainly based on the IFF abductive proof procedure [4], with some extensions:

- The set of facts (events) grows dynamically
- It deals with CLP constraints
- It implements the concepts of fulfillment and violation

SOCS-SI is implemented as a Java-Prolog application, where the abductive proof procedure is implemented in Prolog. The Graphical User Interface and the interfaces to the sources of events instead are implemented in Java. Through the GUI it is possible to observe, for each agent, what events are related to a certain agent, as well as its expectations. Different colors highlight expectations of different type: pending, fulfilled and violated. Fig. 1 shows a screenshot of *SOCS-SI*: a protocol violation has been detected, as a consequence of a wrong behavior of an agent. The figure shows the particular case in which an agent fails to take an action which it was expected to take by some temporal deadline. As the deadline expires, *SOCS-SI* promptly detects the violation.

3 Outline of the demonstration

1. (Informal) introduction to the logic-based social framework, to the formalism expressing the interaction protocols, and to the proof procedure used for verification.
2. Demonstration scenario: first price sealed bid auction. In this auction, an auctioneer announces an auction for a single item (which the auctioneer may want to sell or, as in our example, buy) to a set of agents. By some deadline, the agents may place a bid for the item. Then, the auctioneer must decide which bid is the best, and notify by some deadline both the winner and the losers.
3. Description of the SICs used for expressing the interaction protocols. For instance, the following SIC:
   `H(tell(A,B,answer(win,Item,Quote),AuctionId),TWin)--->`
   `EN(tell(A,B,answer(lose,Item,Quote),AuctionId),TLose)`
   $\wedge$ `TLose > TWin.`
   expresses that the auctioneer is not allowed to notify an agent that its bid has lost after notifying that it has won. The protocol is expressed by four such SICs.

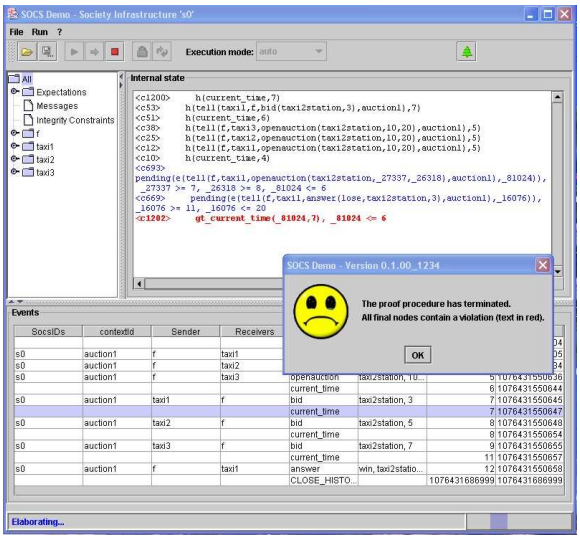

Fig. 1. The *SOCS-SI* Graphic User Interface

4. Running *SOCS-SI*: the GUI will be used to show how to select the interaction protocols and the event source, and how to observe the agent interaction and the proof procedure computation. We plan to show at least two examples: one of fulfillment (the agents will all behave as expected) and one of violation (for instance, the auctioneer may fail to notify the losers, or may notify an agent of both winning and losing the auction).

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Compliance verification of agent interaction: a logic-based tool. In Trappl, R., ed.: Proceedings of the 17th European Meeting on Cybernetics and Systems Research, Vol. II, Symposium "From Agent Theory to Agent Implementation" (AT2AI-4), Vienna, Austria, Austrian Society for Cybernetic Studies (2004) 570–575
2. Societies Of ComputeeS (SOCS): a computational logic model for the description, analysis and verification of global and open societies of heterogeneous computees. Home Page: `http://lia.deis.unibo.it/Research/SOCS/`.
3. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Specification and verification of interaction protocols: a computational logic approach based on abduction. Technical Report CS-2003-03, Dipartimento di Ingegneria di Ferrara, Ferrara, Italy (2003) `http://www.ing.unife.it/aree_ricerca/informazione/cs/technical_reports`.
4. Fung, T.H.: Abduction by Deduction. PhD thesis, Imperial College London (1996)