Formalization and Verification of Interaction Protocols

Doctoral Consortium

ICLP 2005

Sitges – Barcellona

Federico Chesani

Goal of my PhD thesis

- Adopt Logic Programming to:
  - Define specification languages for interaction protocols
  - Provide a semantics for protocol definitions
  - Be able to verify interactions
  - Be able to prove properties about protocols (starting from the specifications)
  - Be able to "execute" a protocol specification

Outline of the presentation

- Protocols and Protocol Definition Languages
- The SOCS project, the SCIFF language and the proof procedure
- Proving properties – current approach
- Future research directions

Interaction protocols

- Protocols have been widely studied in distributed systems
- Protocols play the principal role in heterogeneous systems
- Protocols are key components in multi-agent systems
- Security protocols are a main issue for networks applications

Formalisms for specifying protocols

- To cite some:
  - Finite state machine diagrams
  - Coloured Petri Nets
  - AUML / AML
  - Message Sequence Charts
  - Model Checking languages (ProMeLa & AVISPA project)

Protocol Definition Languages – some desired features

- Easy to use for humans (graphic formalisms?)
- Sufficient expressiveness
- Formal semantics
- Ability to abstract away from participants' internals
- Executability of the definition (support for direct implementation of peers)
- Ability to prove specific/general properties
- Use of the same formalism for each step of the design and implementation process
- Provide a single framework for the protocol designer - programmer

My research work

- Within the SOCS european project, and with my colleagues:
  - Definition of a general framework for agent societies
  - Definition of a protocol definition language (extendible to generic protocols)
  - Tools have been implemented for specifying, testing and verifying protocols
  - Several application domains studied

The SOCS abductive framework

- A language for protocol definition (the SCIFF language)
- An abductive proof procedure that it is able to determine if a given interaction is compliant with a given protocol definition
- A tool (SOCS-SI) that can be used for on-the-fly conformance checking

The SCIFF language: Events

- happened events (ground)

$$\mathbf{H}(Desc, Time)$$

- Desc (term)
- Time (integer)
- E.g.: $\mathbf{H}(tell(bob, alice, bid(pen, 1\ \$), auc1), 3)$
  Bob tells Alice that he bids 1$ for the pen in auction *auc*1 at time 3
- Events compose a *history* $\mathbf{HAP}$

The SCIFF language: Expectations

- Events that should / should not happen

$$\mathbf{E}(Desc, Time) \qquad \mathbf{EN}(Desc, Time)$$

- Eg $\mathbf{E}(tell(alice, bob, answ(A, pen, 1\$), auc1), T_{Ans}), T_{Ans} > 3$
  Alice should answer to Bob's bid, after time 3
- Eg $\mathbf{EN}(tell(B, alice, bid(pen, P), auc1), T_{Bid}), T_{bid} > 3, P < 1\$$
  No agent should place a bid to Alice for the pen in auction 1 for less than 1$, after time 3
- Expectations compose the set EXP (Δ)

The SCIFF language: Syntax

- Social Organization Knowledge Base (SOKB)
  - clauses *Atom* ← *Cond*
  - *Cond:* conjunction of literals, constraints, expectations
- Social Integrity Constraints (ICs)
  - *Body* → *Head*
  - *Body:* conjunction of literals defined in SOKB, H, E, EN and CLP constraints
  - *Head:* a disjunction of conjunction of E, EN literals and CLP constraints

A protocol example

```
H( tell( A, B, openauction(Item,TEnd,TDeadline), D), TOpen)
--->
E( tell( A, B, closeauction, D), Tend)
/\ Tend > Topen.
```

- If agent A tells to agent B that an auction has been opened, then A is expected to tell (later) to B that the auction is closed.

Fulfillment and Violation

H(tell(A, B, openauction(Item,TEnd,TDeadline), D), TOpen)
--->
E(tell(A, B, closeauction, D), Tend)
/\ Tend > Topen.

- Each positive expectation that is confirmed by a corresponding event is declared "fulfilled" (the opposite holds for negative expectations)
- If some expectations are not fulfilled, hence the protocol has been violated.

The SCIFF Proof Procedure

- *S*CIFF: Extension of the IFF abductive proof-procedure [Fung-Kowalski]
  - Generation of expectations
  - Abduction of literals with universally quantified variables
  - Dynamically happening events
  - CLP constraints on variables (both existentially and universally quantified)

SCIFF Properties

- Soundness, for allowed programs
- Completeness, for allowed programs, under some syntactic conditions
- Termination, for acyclic programs

My current research activity

- From the protocol specification → prove protocol properties

- Desired features:
  - Use of a single formalism for defining, proving properties about, testing protocols
  - Properties expressed using the same formalism
  - Ability to generate counter-examples
  - Ability to reason with partially instantiated interactions
  - Executability

Some basic questions…

- How to represent a property?
- Which properties are we going to represent (general/specific/both)?
- What does it mean that "a property holds for a protocol"?
- How are we going to prove properties?

Current approach: re-use the SCIFF approach

- It can abduce literals containing variables
- It can abduce literals with universally quantified variables (EN)
- It supports CLP constraints
- General enough to express a wide variety of protocols (not only MAS!)
- Tools already available for the on-the-fly verification of compliance

- Pragmatic motivations: we have it, it works, we like it!

Representation of the properties

A proposal:

- Properties are represented in terms of events that are expected to happen/ not to happen

$$\mathcal{P} \approxeq \mathbf{E}(p_1) \wedge \ldots \wedge \mathbf{E}(p_n) \wedge \mathbf{EN}(p_{n+1}) \wedge \ldots \wedge \mathbf{EN}(p_m)$$

- In the MAS scenario, properties are defined in terms of which messages should/shouldn't be exchanged, possibly with constraints about the content, time , etc.

A property *P* holds…

Existentially if:

$$\exists \mathbf{HAP}_i \; s.t. \; SOKB \cup \mathbf{HAP}_i \cup \mathbf{EXP} \models \mathcal{IC}_S$$
$$\mathbf{EXP} \text{ is fulfilled, } \neg, \mathbf{E}\text{-consistent}$$
$$SOKB \cup \mathbf{HAP}_i \cup \mathbf{EXP} \models \mathcal{P}$$

Univesally if:

$$\left.\begin{array}{l} \forall_{\mathbf{HAP}} SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{IC}_S \\ \mathbf{EXP} \text{ is fulfilled, } \neg, \mathbf{E}\text{-consistent} \end{array}\right\} \Rightarrow$$
$$\Rightarrow SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{P}$$

The "existential" approach…

- We have decided to adopt an "existential" approach. Given:
  - A protocol definition through ICs
  - A property definition *P*

we would like to answer the question:

Does there exist an interaction compliant to the protocol, s.t. *P* holds?

- In this way it is also possible to disprove properties by refutation:

$\exists H_i$ s.t. it is compliant with $\neg$ P?

H_i would represent the counter-example… much more interesting!

An example…

- Given the protocol:

$H(event_1(X), T_1) \wedge H(event_2, T_2) \rightarrow$
$E(event_3(X), T_3).$

- Given the property

$P \cong E(event_1(X), T_1) \wedge E(event_2, T_2)$

Which are the interactions that are compliant with the protocol, and for which P holds?

An example…

- H_1={h(event$_1$(a), 1), h(event$_2$, 2), h(event$_3$(a), 3) }
- H_2={h(event$_1$(b), 1), h(event$_2$, 2), h(event$_3$(b), 3) }
- H_3={h(event$_1$(a), 1), h(event$_2$, 2), h(event$_3$(a), 3), h(event$_1$(b), 4), h(event$_2$, 5), h(event$_3$(b), 6) }
- …

- How can we generate compliant histories?
  - A proposal: extending the SCIFF by adding the integrity constraint:

$$\mathbf{E}(X,T) \rightarrow \mathbf{H}(X,T)$$

Some issues…

- If we want to disprove properties, we must be sure that:

  if a compliant history does exist, then we are able to generate it

- We need also a way for representing compliant history "intensionally" …

Other ideas for the near future

Besides the main problem of proving properties, other interesting issues are:

- Protocol compositionality and resulting properties
- Protocol executability: under which (syntactic) constraints a protocol can be directly executed by an agent?

Thanks for the attention!

- Questions?

Bibliography

- General Framework
  - Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Specification and verification of agent interactions using social integrity constraints. *Electronic Notes in Theoretical Computer Science*, 85(2), April 2004.
  - Marco Alberti, Anna Ciampolini, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. A social ACL semantics by deontic constraints. In Vladimir Marik, Jorg Muller, and Michal Pechoucek, editors, *Multi-Agent Systems and Applications III. 3rd International Central and Eastern European Conference on Multi-Agent Systems CEEMAS 2003*, volume 2691 of *LNAI*
  - Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. The SOCS computational logic approach to the specification and verification of agent societies. In Corrado Priami and Paola Quaglia, editors, *Global Computing: IST/FET International Workshop, GC 2004 Rovereto, Italy, March 9-12, 2004 Revised Selected Papers*, volume 3267 of *LNCS*
  - 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 *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2004)*, pages 387-392, Washington, DC, USA, September 2004. IEEE Computer Society.
- Operational Semantics
  - Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Abduction with hypotheses confirmation. In Rossi and Panegai ed., CILC 2004. Also short version as IJCAI2005 poster.
  - Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. The SCIFF abductive proof-procedure. In *IX Congresso nazionale Associazione Italiana per l'Intelligenza Artificiale*, Lecture Notes in Artificial Intelligence, Springer Verlag, 2005.
- Implementation
  - Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Compliance verification of agent interaction: a logic-based tool. In Robert Trappl, editor, *Proceedings of the 17th European Meeting on Cybernetics and Systems Research, Vol. II, Symposium ``From Agent Theory to Agent Implementation'' (AT2AI-4)*, pages 570-575, Vienna, Austria, April 13-16 2004. Austrian Society for Cybernetic Studies.
- Applications
  - Marco Alberti, Marco Gavanelli, Evelina Lamma, Paola Mello, and Paolo Torroni. Modeling interactions using social integrity constraints: a resource sharing case study. In Joao Alexandre Leite, Andrea Omicini, Leon Sterling, and Paolo Torroni, editors, *Declarative Agent Languages and Technologies, First International Workshop, DALT 2003, Melbourne, Australia, July 15, 2003, Revised Selected and Invited Papers*, volume 2990 of *Lecture Notes in Computer Science*, pages 243-262, Melbourne, Australia, 2004. Springer Verlag.
  - 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.

SCIFF semantics…

- SCIFF: abductive semantics

$$SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models G \quad SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models IC$$

- Coherence of set EXP

$$\forall p, \mathbf{E}(p), \mathbf{EN}(p) \notin \mathbf{EXP}$$

$$\forall p, \neg\mathbf{E}(p), \mathbf{E}(p) \notin \mathbf{EXP} \quad \forall p, \neg\mathbf{EN}(p), \mathbf{EN}(p) \notin \mathbf{EXP}$$

- Compliance to protocol

$$\forall p, \mathbf{E}(p) \rightarrow \mathbf{H}(p) \qquad \forall p, \mathbf{EN}(p) \rightarrow not\mathbf{H}(p)$$

SCIFF semantics…

- SCIFF: abductive semantics

$$KB \cup \Delta \models G \qquad KB \cup \Delta \models IC$$

- Coherence of set Δ

$$\forall p, \mathbf{E}(p), \mathbf{EN}(p) \notin \Delta$$

$$\forall p, \neg\mathbf{E}(p), \mathbf{E}(p) \notin \Delta \qquad \forall p, \mathbf{EN}(p), \neg\mathbf{EN}(p) \notin \Delta$$

- Compliance to protocol

$$\forall p, \mathbf{E}(p) \rightarrow \mathbf{H}(p) \qquad \forall p, \mathbf{EN}(p) \rightarrow not\mathbf{H}(p)$$