Formalizzazione e Verifica di Protocolli di Interazione

Federico Chesani

Tutor: Prof. Paola Mello
DEIS
Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna

Relazione Attività di Dottorato
7 Novembre 2005

Outline

Agenti

Una possibile definizione...

"Un agente è qualsiasi cosa che possa essere vista come un sistema che percepisce il suo ambiente attraverso dei sensori e agisce su di esso mediante attuatori"
Russel, Norvig - *Intelligenza Artificiale*

Da un altro punto di vista...

Un agente è un'entità computazionale caratterizzata da:

- autonomia
- reattività
- proattività
- socialità

Sistemi multi-agente

- Al fine di ottenere applicazioni reali e complesse, si considerano sistemi composti da più agenti (*Multi-Agent Systems, MAS*)
- Laddove l'interazione fra gli agenti è regolamentata in qualche modo, si può parlare di *società* di agenti
- Le proprietà generali del sistema sono in qualche modo date dal *comportamento* degli agenti che ne fanno parte

Sistemi eterogenei. . .

- Non è realistico ipotizzare di conoscere come sono fatti gli agenti (eterogeneità software, hardware, bussiness competitors)
- Le caratteristiche di un sistema devono essere garantite tramite l'introduzione di regole di interazione (*protocolli*)
- Tali regole devono vincolare il meno possibile il comportamento degli agenti
- Deve essere possibile *verificare* che tali regole siano rispettate
- Si può osservare solo il *comportamento esterno*. . .

Protocolli di interazione

- I protocolli sono quindi un possibile mezzo per garantire certe proprietà di sistemi complessi
- Devono essere facili da esprimere (linguaggio grafico?)
- Devono avere una semantica dichiarativa di facile interprestazione, e non ambigua
- Deve essere possibile *dimostrare automaticamente* se determinate proprietà valgono oppure no
- Deve essere possibile stabilire se, a run-time, tali protocolli sono rispettati o violati

Obiettivi della tesi

- Linguaggi per la definizione dei protocolli (sia aspetti semantici che aspetti operazionali)
- Capacità espressiva del linguaggio di definizione
- Proprietà dei protocolli - come esprimerle, come verificarle
- Eseguibilità dei protocolli (dal protocollo all'agente)
- Verifica del rispetto dei protocolli

Obiettivo finale

Realizzare un framework unico, coerente, che permetta di modellare, definire, dimostrarne le proprietà, eseguire, verificare un sistema multiagente.

Partecipando al progetto SOCS...

- Lo scopo è stato modellare società di agenti usando la logica computazionale
- E' stato definito un linguaggio per definire una società ed i protocolli (SCIFF language)
- Il modello è basato sull'abduzione

- L'approccio scelto è del tipo "*tutto ciò che non e' esplicitamente vietato, è consentito*"
- Introdotta l'idea di aspettativa a livello sociale
- La semantica degli atti comunicativi è data in termini delle aspettative che essi generano a livello sociale

Il formalismo $\mathcal{S}$CIFF

- Gli *eventi* nel sistema sono rappresentati come:

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

- Le *aspettative* come:

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

Modellare una società significa definire:

Una base di conoscenza "statica" della società (SOKB)

- Necessario per definire concetti come ad. es. ruoli, o obiettivi sociali.
- Composta di clausole del tipo: $Atom \leftarrow Cond$, dove $Cond$ è una congiunzione di letterali, vincoli (CLPFD), aspettative.

I protocolli di interazione

Sono espressi sotto forma di emphforward rules (dette vincoli sociali di integrità $\mathcal{IC}_S$):

$$Body \rightarrow Head$$

dove *Body* è una congiunzione di letterali definiti nella SOKB, **H**, **E**, **EN** e vincoli CLP . *Head* è una disgiunzione di congiunzioni di **E**, **EN** e vincoli CLP

Esempio di protocollo

In un'asta inglese, se l'ultima offerta e' stata fatta al tempo T_1, ci aspetta che entro τ istanti avvenga un'altra offerta, oppure che l'ultima offerta vinca...

Example

$$
\begin{array}{ll}
 & \mathbf{H}(tell(A, \_, openauction(Item, english(\tau), D), T_{open}) \wedge \\
 & \mathbf{H}(tell(B_1, A, bid(Item, Price_1), D), T_1) \\
\rightarrow & \mathbf{E}(tell(B_2, A, bid(Item, Price_2), D), T_2) \wedge \\
 & Price_2 > Price_1 \ \wedge \ T_2 < T_1 + \tau \\
\vee & \mathbf{E}(tell(A, B_1, answer(win, Item, Price_1), D), T_{win}) \wedge \\
 & T_{win} = T_1 + \tau
\end{array}
$$

Interpretazione abduttiva

Definition (**Programma Logico Abduttivo**)

E' una tripla $\langle P, Ab, IC \rangle$ dove:

- P è un programma logico;
- Ab è un insieme di predicati *abducibili*, non definiti in P;
- IC è un insieme di vincoli di integrità.

Dato un programma logico abduttivo ed un goal G, l'obbiettivo dell'abduzione è trovare un insieme (possibilmente minimo) di abducibili $\Delta \subseteq Ab$, per cui valga:

$$P \cup \Delta \models G \quad (1)$$

$$P \cup \Delta \models IC \quad (2)$$

Interpretazione abduttiva

Definition

Una istanza $\mathcal{S}_{\mathbf{HAP}}$ di una società $\mathcal{S}$ è rappresentata come un programma logico abduttivo (ALP), cioè una tripla $\langle P, Ab, \mathcal{IC}_S \rangle$ dove:

- P è la *SOKB* di $\mathcal{S}$ insieme agli eventi acaduti **HAP**;
- Ab è l'insieme di *predicati abducibili* **E**, **EN**, $\neg$**E**, $\neg$**EN**;
- $\mathcal{IC}_S$ è l'insieme dei vincoli sociali d'integrità $\mathcal{S}$.

$$SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models G \tag{3}$$

$$SOKB \cup \mathbf{HAP} \cup \mathbf{EXP} \models \mathcal{IC}_S \tag{4}$$

"Fulfillment" e "Violation"

Definition (Fulfillment)

Data una istanza di società $\mathcal{S}_{\mathbf{HAP}}$, un insieme di aspettative **EXP** che è $\mathcal{IC}_S - consistent$, $\neg - consistent$ e $\mathbf{E} - consistent$, è *fulfilled* se e solo se per ogni termine (eventualmente ground) p:

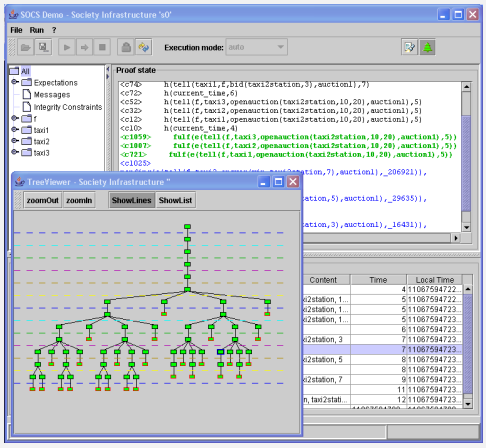
$$\mathbf{HAP} \cup \mathbf{EXP} \cup \{\mathbf{E}(p) \rightarrow \mathbf{H}(p)\} \cup \{\mathbf{EN}(p) \rightarrow \neg\mathbf{H}(p)\} \not\models \bot \tag{5}$$

Definition (Violation)

Data un'istanza di società $\mathcal{S}_{\mathbf{HAP}}$, un'insieme di aspettative sociali **EXP** è *violato* se e solo se esiste un termine (eventualmente ground) p tale che:

$$\mathbf{HAP} \cup \mathbf{EXP} \cup \{\mathbf{E}(p) \rightarrow \mathbf{H}(p)\} \cup \{\mathbf{EN}(p) \rightarrow \neg\mathbf{H}(p)\} \models \bot \tag{6}$$

Il software SOCS-SI



Il software è basato sulla $\mathcal{S}$CIFF *proof procedure*

Caratteristiche

- Verifica della correttezza di una interazione sia on-the-fly che off-line.
- Supporta piattaforme ad agenti come JADE, PROSOCS e TuCSoN, oltre al sistema standard di e-mail e a file di log.
- Tramite la GUI, è possibile accedere ai messaggi scamnbiati, alla lista dei partecipanti, e all'insieme di aspettative generate istante per istante.
- Si può visualiizare l'albero della procedura di prova, ed ispezionare l'insieme di aspettative per ogni nodo generato Facilmente estendibile ad altre piattaforme tramite meccanismo di plug-in

Aste combinatorie

- Sono aste in cui viene offerto un insieme di oggetti
- Gli offerenti possono fare offerte per un sottoinsieme o per tutti gli oggetti
- L'astatore deve massimizzare il profitto scegliendo un insieme opportuno di offerte (problema NP-hard)
- Il protocollo studiato era del tipo First Price-Sealed Bid, applicato alle combinatorie

Protocollo TCP/IP

- Collaborazione con H3G al fine di verificare il protocollo TCP
- Modellazione dell'apertura three-hand-shake e dell'apertura "simultanea"
- Verifica off-line (su file di log) fornito da H3G
- Violazioni dovute a diverse implementazioni dello stack tcp/ip (vendors differenti)

Le proprietà di un protocollo. . .

- Due classi diverse di proprietà:
  1. Proprietà generali (terminazione, well-definedness, . . .)
  2. Proprietà specifiche dei singoli protocolli
- Le proprietà specifiche sono uno dei settori più interessanti
- Esistono già sistemi che, per particolari settori applicativi, consentono la verifica di proprietà specifiche (es.: Model Checking per protocolli di sicurezza)
- Attività svolta principalmente sulle proprietà particolari

Verifica di proprietà

Definizione di una proprietà:

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

Una proprietà $\mathcal{P}$ è verificata esistenzialmente se:

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

Una proprietà $\mathcal{P}$ è verificata universalmente se:

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

Come sto procedendo...

- Scelto un approccio per refutazione.
- Il problema diventa:

$$\exists \; \mathbf{HAP}_i \text{ tale che } \neg\mathcal{P} \text{ è verificata ?}$$

In termini abduttivi, data una storia iniziale $\mathbf{HAP}_i$ (eventualmente vuota), si sta cercando un insieme degli eventi $\mathbf{HAP}_f$ tale che:

- $\mathbf{HAP}_i \subseteq \mathbf{HAP}_f \quad \mathbf{HAP}_f \equiv \mathbf{HAP}_i \cup \mathbf{HAP}_\Delta$
- $SOKB \cup \mathbf{HAP}_i \cup \mathbf{HAP}_\Delta \cup \mathbf{EXP} \models G$
- $SOKB \cup \mathbf{HAP}_i \cup \mathbf{HAP}_\Delta \cup \mathbf{EXP} \models \mathcal{IC}_S$

Verifica di proprietà... come?

Ulteriore vincolo di integrità che asserisca che *ad ogni aspettativa positiva deve corrispondere un evento*, e riusare la stessa $\mathcal{S}$CIFFper *abdurre* i fatti:

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

*Eventi che non devono succedere (***EN***)*:

$$\mathbf{EN}(Desc, Time) \rightarrow \neg\mathbf{H}(Desc, Time)$$

Il risultato atteso è una sorta di *modello intensionale* di (tutte) le possibili interazioni ammesse dal protocollo:

$$\mathbf{HAP} \approxeq \left\{H^+, H^-\right\}$$

Alcuni problemi incontrati...

- Alcuni metodi provati ma...
- ...tutti risentono del problema dell'incompletezza

Gli attuali risultati sono stati discussi al Doctoral Consortium della conferenza ICLP 2005 (7 richieste accolte su 17 domande)

Verifica delle proprietà del protocollo NSPK

- Ha come scopo lo scambio di due numeri nonce su cui stabilire una successiva chiave di sessione temporanea
- E' stato utilizzato per circa una decina di anni, creduto "sicuro"
- Lowe ha dimostrato per primo l'esistenza di un attacco "man-in-the-middle"
- SOCS-SI ha confermato che il comportamento dell'intruder è conforme al protocollo $\longrightarrow$ problema di specifica

- Con un primo lavoro (parziale) sulla dimostrazione di proprietà è stato possibile ottenere:
  - refutazione della proprietà di sicurezza
  - generazione del controesempio

Nel prossimo anno...

- Risolvere il problema della dimostrazione di proprietà e dimostare correttezza/completezza della soluzione

- Esecuzione di protocolli
  - Cosa significa "far eseguire un protocollo ad un agente" ?
  - Sotto quali condizioni sintattiche un protocollo può essere "eseguito" ?
  - E' possibile utilizare lo stesso linguaggio usato per specificare un protocollo, al fine di specificare anche la *politica* di un singolo agente?
- Identificazione del colpevole (nelle società aperte questo è ancora un problema irrisolto)

Articoli 2004 I

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
Compliance Verification of Agent Interaction: a Logic-Based Tool.
In R. Trappl, editor, Proceedings of the *17th European Meeting on Cybernetics and Systems Research (EMCSR'2004)*, Vol. II, Symposium "From Agent Theory to Agent Implementation" (AT2AI-4), pp. 570-575, Vienna, Austria, April 13-16, 2004. Austrian Society for Cybernetic Studies.

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. 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)". Modena, Italy, June 14, 2004. pp. 387-392. IEEE Press.

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
AAMAS 2004 Demo
In Proccedings of Agent Architectures and Multi-Agent Systems, Columbia University, New York, NY, USA, July 22 2004.

Articoli 2004 II

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
The SOCS Computational Logic Approach to the Specification and Verification of Agent Societies.
In Post-Proceedings of the *Global Computing 2004 Workshop*(GC 2004), LNAI 3267, pp. 314-339. © Springer-Verlag, 2005.

Articoli 2005 I

M. Alberti, F. Chesani, M. Gavanelli, A. Guerri, E. Lamma, P. Mello, M. Milano and P. Torroni.
Expressing Interaction in Combinatorial Auctions Through Social Integrity Constraints.
In Armin Wolf, ed., Proceedings of the 19th Workshop on (Constraint) Logic Programming, (W(C)LP), University of Ulm, Germany, February 21-23, 2005, pp.53-64.

M. Alberti, F. Chesani, M. Gavanelli, A. Guerri, E. Lamma, P. Mello and P. Torroni.
Applicazione dei vincoli di integrità sociali come strumento dispecifica delle interazioni in aste combinatorie.
In Intelligenza Artificiale, Anno II No. 1, Marzo 2005, pp. 22-29. ISSN 1724-8035.

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
Security protocols verification in Abductive Logic Programming: A case study.
In Alberto Pettorossi, Maurizio Proietti, and Valerio Senni, eds., CILC 2005 - Convegno Italiano di Logica Computazionale. Università degli Studi di Roma Tor Vergata, June 21-22 2005.

Articoli 2005 II

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
Security protocols verification in Abductive Logic Programming: A case study.
In O. Dikenelli, M. Gleizes and A. Ricci eds., Proceedings of *Engineering Societies in the Agents' World (ESAW'05)*, Kusadasi, Aydin, Turkey October 26-28 2005.

M. Alberti and F. Chesani.
The computational behaviour of the SCIFF abductive proof procedure and the SOCS-SI system.
In Intelligenza Artificiale, Anno II No. 3, to appear.

F. Chesani.
Formalization and Verification of Interaction Protocols
Research Abstract. In Proceedings of the 21st International Conference on Logic Programming (ICLP 2005), Doctoral Consortium, Sitges, Spain October 2-5 2005. LNCS 3668, pp. 437-438.

Articoli 2005 III

M. Alberti, F. Chesani, A. Ciampolini, P. Mello, M. Montali, S. Storari and P. Torroni.
Protocol Specification and Verification by Using Computational Logic
In Proceedings of Workshop dagli Oggetti agli Agenti (WOA'05), Camerino, November 14-16 2005.

Articoli 2006

M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello and P. Torroni.
Compliance Verification of Agent Interaction: a Logic-Based Tool.
In Applied Artificial Intelligence, Vol. 20, Nos. 4-5, Taylor & Francis, April 2006.
Special issue edited by Paolo Petta and Jörg P. Müller:"Best of AT2AI-4".

F. Chesani, M. Gavanelli
Specification and verification of agent interaction using SOCS-SI.
LNCS, to appear

Varie

- 2004: Scuola di Dottorato di Bertinoro
- 2005: Tutorial al Convegno Italiano di Logica Computazionale
- 2005: Tutorial a CLIMA (Computational Logic in Multi-Agent Systems

Progetti a cui ho preso parte

- 2003-2005: Progetto europeo IST-32530 *SOCS, Societies Of ComputeesS*
- 2003-2005: COFIN MIUR, "*Svilupppo e verifica di sistemi multiagente basati sulla logica*"