

Massive: attività svolte, attività future

Viviana Mascardi

Università degli Studi di Genova
Dipartimento di Informatica e Scienze dell'Informazione

14 Luglio 2005

- 1 Attività svolte
- 2 Zoom su attività Coo-BDI, -AgentSpeak, -WS
 - CooBDI
 - CooAgentSpeak
 - CooWS
- 3 Attività future

Contenuti

- 1 Attività svolte
- 2 Zoom su attività Coo-BDI, -AgentSpeak, -WS
 - CooBDI
 - CooAgentSpeak
 - CooWS
- 3 Attività future

Rivediamo le attività previste...

1 Modellazione

1 Modelli formali dei sistemi ad agenti:

- Modellazione dell'interazione fra agenti mediante logiche modali e temporali
- Revisione delle credenze in un ambito multiagente
- Sviluppo di un agente specializzato nel ragionamento temporale a vincoli

2 Linguaggi di Specifica basati sulla logica computazionale:

- Sviluppo di un linguaggio di programmazione logica per modellare azioni ed agenti

2 Validazione e verifica:

- Verifica del comportamento globale dei sistemi multiagente mediante tecniche di model checking e di prototipazione rapida

3 Applicazioni

Modelli formali dei sistemi ad agenti

Tesi di laurea (2004–2005):

- R. Montagna. *Teoria e modellazione logica degli agenti BDI cooperativi*. Tesi di Laurea, DISI, Università di Genova, 2004.

Linguaggi di Specifica basati sulla logica computazionale

Riviste (2004–2005):

- V. Mascardi, M. Martelli, e L. Sterling. *Logic-Based Specification Languages for Intelligent Software Agents*. Theory and Practice of Logic Programming Journal (TPLP), 4(4), Cambridge University Press, pagg. 429–494, 2004.

Linguaggi di Specifica basati sulla logica computazionale

Proceedings di Conferenze e Workshop Internazionali (2004–2005):

- V. Mascardi, L. Bozzo, P. Busetta, D. Ancona. *CooWS: Adaptive BDI Agents Meet Service-Oriented Computing*. Accepted at the IADIS ICWI Conference, 2005.
- D. Ancona, V. Mascardi, J. Hübner, e R. Bordini. *Coo-AgentSpeak: Cooperation in AgentSpeak through Plan Exchange*. In Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004, N. R. Jennings, C. Sierra, L. Sonenberg e M. Tambe editors, ACM Press, pagg. 698–705, 2004.
- D. Ancona e V. Mascardi. *Coo-BDI: Extending the BDI Model with Cooperativity*. In Proceedings of the 1st International Workshop on Declarative Agent Languages and Technologies, DALT 03, Revised Selected and Invited Papers. J. Leite, A. Omicini, L. Sterling and P. Torroni editors, Springer-Verlag, LNAI 2990, pagg. 109–134, 2004.
- V. Mascardi, D. Demergasso, D. Ancona. *Languages for*

Linguaggi di Specifica basati sulla logica computazionale

Tesi di laurea (2004–2005):

- L. Bozzo. *CooWS: un'architettura per servizi web cooperativi*. Tesi di Laurea, DISI, Università di Genova, 2005.
- D. Demergasso. *Coo-AgentSpeak: un linguaggio per agenti deliberativi e cooperativi*. Tesi di Laurea, DISI, Università di Genova, 2005.

Verifica e validazione

Proceedings di Conferenze Internazionali (2004–2005):

- M. Baldoni, C. Baroglio, I. Gungui, A. Martelli, M. Martelli, V. Mascardi, V. Patti, e C. Schifanella. *Reasoning about agents interaction protocols inside DCaseLP*. In Proceedings of the 2nd International Workshop on Declarative Agent Languages and Technologies, DALT'04, J. Leite, A. Omicini, P. Torroni, P. Yolum editors, pagg. 250–265, 2004.

Verifica e validazione

Proceedings di Conferenze Nazionali (2004–2005):

- I. Gungui e V. Mascardi. *Integrating tuProlog into DCaseLP to Engineer Heterogeneous Agent Systems*. Pubblicazione elettronica negli atti del convegno CILC 2004.
<http://www.cs.unipr.it/CILC04/AttiConvegno>.

Tesi di laurea (2004–2005):

- I. Gungui. *Integrazione di agenti logici in D-CaseLP*. Tesi di Laurea, DISI, 2005.

Contenuti

- 1 Attività svolte
- 2 Zoom su attività Coo-BDI, -AgentSpeak, -WS
 - CooBDI
 - CooAgentSpeak
 - CooWS
- 3 Attività future

CooBDI: Data structures

- **Plans**: extended with an access specifier ranging over *private*, *public* and *only(TrustedAgents)*.
- **Intentions**: extended with a set of plan instances *RelevantInstances* and a set of agent identifiers *WaitingOnAgents*. One-to-one relation with a main desire.
- **Events**: cooperation events and ordinary ones.
- **Cooperation strategy** specified by:
 - a set of trusted agent identifiers
 - a plan retrieval policy ranging over *always* and *noLocal*
 - a plan acquisition ranging over *discard*, *add* and *replace*

CooBDI: Execution cycle

1. Process the event queue
2. Process suspended intentions
3. Process active intentions

The mechanism for retrieving relevant plans (adopted in steps 1 and 3) is characterized by

- i. The intention is suspended.
- ii. The local relevant plans for the desire are generated and associated to the intention.
- iii. According to the cooperation, the set S of the agents expected to cooperate is defined.
- iv. A plan request for the desire is created and it is sent to all the agents in S .

CooAgentSpeak: Motivations

- The work on CooBDI has proposed an approach to plan exchange which applies to BDI agents in general
- The introduction of special illocutionary forces for plan exchange between AgentSpeak agents [F. Moreira, R. Vieira, and R. H. Bordini. *Extending the operational semantics of a BDI agent-oriented programming language for introducing speech-act based communication*. In Proc. of DALT-03, 2003] has been recently implemented in Jason [R. Bordini, J. F. Hübner, et al. *Jason: A Java-based agentSpeak interpreter used with saci for multi-agent distribution over the net*, manual, first release edition, 2004. <http://jason.sourceforge.net/>]
- CooAgentSpeak is an instantiation of CooBDI for AgentSpeak in particular; the ideas behind CooAgentSpeak can be elegantly integrated into Jason

CooWS: Motivations from the agents' viewpoint

CooWS:

- provides an hint to implement adaptive BDI-style agents that are situated in a real software environment, the Web;
- demonstrates the feasibility of the CooBDI approach by means of a working prototype;
- provides a (very small!!!) step towards facing the challenge described in the AgentLink III Agent Technology Roadmap, December 2004 (*"While Web Service technologies define conventions for describing service interfaces and workflows, we need more powerful techniques for dynamically describing, discovering, composing, monitoring, managing, and **adapting** multiple services in support of **virtual organisations**"*)

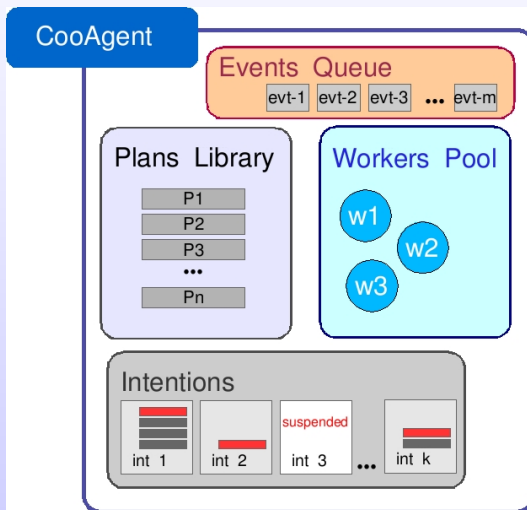
CooWS: Motivations from the web services' viewpoint

CooWS:

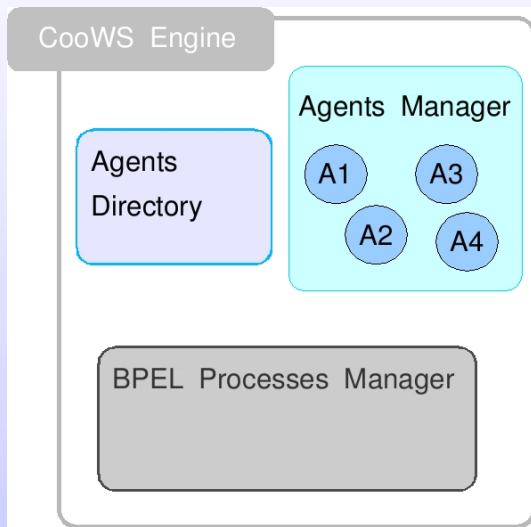
- allows to understand some limitations of current WS technology with respect to adaptivity of procedural knowledge, and gives hints on an architecture both for learning agents and the environment in which they have to act.

To our knowledge, there is only one proposal similar to ours, [P. Buhler and J. M. Vidal, 2003. [Semantic Web Services as Agent Behaviors](#). In B. Burg, J. Dale, et al., editors, [Agentcities: Challenges in Open Agent Environments](#), LNCS/LNAI, Springer-Verlag], that discusses agent software with dynamically configured capabilities described with DAML-S, that can represent atomic or orchestrated Web Services.

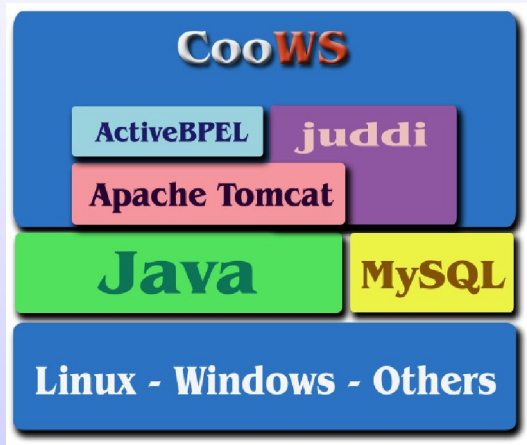
CooWS: Data structures



CooWS: Platform



CooWS: Implementation



Contenuti

- 1 Attività svolte
- 2 Zoom su attività Coo-BDI, -AgentSpeak, -WS
 - CooBDI
 - CooAgentSpeak
 - CooWS
- 3 Attività future

Modelli formali di sistemi ad agenti

Definizione di una semantica per CooWS

DA FARE EX NOVO, MA MI INTERESSA MOLTO!!!

Verifica e validazione 1

Validazione tramite prototipazione rapida di specifiche AUMML di protocolli di interazione tra entità autonome che “vivono” nel Web (ad esempio, peer in una rete P2P).

Nel dettaglio, si prevede di utilizzare una rappresentazione testuale dei diagrammi AUMML (ad esempio, XMI) e di creare automaticamente a partire da questa rappresentazione, dei “gusci” di programmi in Prolog da completare con la conoscenza di dettaglio - non specificata a livello di diagramma AUMML - e da far eseguire in un ambiente di prototipazione rapida adatto allo scopo.

GIÀ IN PARTE FATTO, ANCHE SE IL LINGUAGGIO TARGET NON ERA PROLOG MA “JESS”

Verifica e validazione 2

Validazione tramite prototipazione rapida di specifiche BPEL di protocolli di interazione.

Nel dettaglio, si prevede di creare automaticamente a partire dalle specifiche BPEL, dei programmi in Prolog il cui comportamento osservabile e' lo stesso dei servizi specificati in BPEL, da far eseguire in un ambiente di prototipazione rapida adatto allo scopo.

DA FARE EX NOVO, MA MI INTERESSA MOLTO!!!

Progettazione e sviluppo di sistemi

Progettazione e sviluppo di un sistema di servizi web cooperanti in cui la cooperazione avviene non solo a livello di scambio di dati, ma anche a livello di scambio di conoscenza procedurale (ovvero, scambio di specifiche BPEL).

Nel dettaglio, in fase di inegnerizzazione del sistema verranno usate le tecniche descritte nel punto 2 per validare i protocolli di interazione prima della implementazione dei web service.

GIÀ FATTO PER LA PARTE DI PROGETTAZIONE E SVILUPPO; MANCA LA VALIDAZIONE MEDIANTE LE TECNICHE SVILUPPATE AL PUNTO 2

La presentazione è finita....

....commenti?suggerimenti?critiche????

Grazie!