

PROGRAMMA DI RICERCA - MODELLO A
Anno 2005 - prot. 2005011293

PARTE I

1.1 Programma di Ricerca afferente a

Area Scientifico Disciplinare 01: Scienze matematiche e informatiche 60%

Area Scientifico Disciplinare 09: Ingegneria industriale e dell'informazione 40%

1.2 Titolo del Programma di Ricerca

Testo italiano

Specifica e verifica di protocolli di interazione fra agenti

Testo inglese

Specification and verification of agent interaction protocols

1.3 Abstract del Programma di Ricerca

Testo italiano

I servizi forniti in molti domini applicativi sempre più possono essere descritti e realizzati come espressione di un insieme di agenti cooperanti. Rispetto all'approccio più tradizionale della programmazione a componenti, il paradigma ad agenti esplicita aspetti sociali, connessi alla caratteristica degli agenti di essere autonomi e proattivi: le componenti comunicano e si coordinano in maniera dinamica, utilizzando linguaggi ad alto livello, per perseguire scopi comuni (o propri). Tra gli aspetti sociali è particolarmente importante definire "regole comportamentali", espresse mediante protocolli di comunicazione (interazione), finalizzate a controllare l'organizzazione del sistema. I protocolli di comunicazione possono essere utilizzati per regolare l'interazione fra gli agenti specificando un insieme di conversazioni accettabili, ossia sequenze di operazioni (in particolare scambi di messaggi) che si possono svolgere fra agenti diversi.

Con la rapida espansione di internet, gli agenti si trovano sempre più ad operare in ambienti fortemente dinamici. Diventa quindi essenziale che gli agenti soddisfino proprietà che ne garantiscano l'interoperabilità, come ad esempio che un agente sia conforme a un protocollo di interazione, che un agente stia rispettando i suoi vincoli sociali o che un protocollo ottenuto per composizione conservi le proprietà dei protocolli che lo compongono.

Per dimostrare se un protocollo garantisce o meno determinate proprietà è necessario specificarlo in un linguaggio formale. Infatti l'utilizzo di linguaggi formali di specifica consente di sviluppare tecniche di verifica formale di proprietà.

L'obiettivo di questo progetto è quello di sviluppare strumenti e tecniche per la specifica e la verifica di proprietà riguardanti l'interazione fra agenti in un sistema multiagente. Più precisamente i principali obiettivi del progetto sono i seguenti:

Definizione di formalismi per la specifica e la verifica di protocolli di interazione

Verranno analizzate le principali proprietà dei protocolli di interazione che si intendono verificare, e verranno individuati i formalismi più adeguati per esprimere queste proprietà. I formalismi utilizzati per specificare protocolli di comunicazione e per descriverne le proprietà saranno prevalentemente basati sulla logica (temporale e computazionale), ma si prenderanno in considerazione anche formalismi a vincoli, linguaggi formali ed automi.

Verifica automatica delle proprietà

Verranno messe a punto tecniche di verifica per i formalismi sviluppati nel punto precedente. In particolare, per le logiche temporali, saranno sviluppate tecniche di model checking. In altri casi, si applicheranno tecniche basate su procedure di dimostrazione e si indagheranno tecniche basate su risoluzione estesa. Verranno anche sviluppate tecniche di prototipazione rapida per la validazione di protocolli di interazione.

Linguaggi di modellazione e loro traduzione nei linguaggi formali definiti nel progetto

Per lo sviluppo dei sistemi multiagente sono stati proposti numerosi linguaggi di modellazione. Questi linguaggi mancano di semantica formale e quindi non sono utilizzabili direttamente per la dimostrazione automatica di proprietà delle componenti del sistema. Un obiettivo è quello di analizzare i linguaggi di modellazione più diffusi, in particolare formalismi grafici, e di definire tecniche di traduzione nei formalismi sviluppati nel progetto.

I linguaggi, le tecniche e gli strumenti sviluppati verranno valutati in diversi ambiti applicativi, fra cui principalmente: linee guida cliniche, servizi web, commercio elettronico e sicurezza, presentazioni multimediali.

Testo inglese

In many application domains it is getting more and more common describing and realizing the offered services by means of a set of cooperative agents. Compared with the more traditional component based programming approach, the agent based paradigm makes

explicit the social aspects, related to the fact that agents are autonomous and proactive: the different components dynamically communicate and coordinate with each other, by means of higher level languages, to reach some common (or their own) goal. Among the social aspects, specifically relevant is the ability of defining behavioral rules, expressed by means of communication (interaction) protocols, aiming at controlling the organization of the system. Such protocols can be used to rule the agents' interaction, by specifying a set of acceptable conversations, i.e. sequences of operations (in particular message exchanges) that could occur among different agents.

With the fast expansion of Internet, agents are more and more situated in highly dynamic environments. To operate correctly in such environments, it is essential that agents satisfy properties guaranteeing interoperability, such as for instance that an agent is compliant with an interaction protocol, that it is respecting its social facts, that a protocol obtained by composition preserves the properties of the component protocols.

To prove that some properties of a protocol are satisfied, it is necessary to specify the protocol in a formal language. In fact the use of a formal language allows to rely on formal verification techniques.

The aim of this project is to develop tools and methodologies for specification and verification of properties dealing with agent interactions in multi-agent systems. More precisely, the project will be structured as follows.

Definition of formalisms for the specification and verification of interaction protocols

The most significant properties of interaction protocols will be analyzed, so as to single out the most suitable formalisms for expressing these properties. The kind of formalisms used in the project will be mainly based on temporal and computational logics, but other formalisms, such as temporal constraints, formal languages and automata, will be considered.

Automatic property verification

Various approaches to protocol verification will be developed for the formalisms proposed in the project. In particular, model checking techniques, as well as techniques based on theorem proving or on extended resolution will be investigated. We plan also to develop rapid prototyping techniques for validating interaction protocols.

Modeling languages and their translation into the formal languages defined in the project

Different languages have been proposed for the design and implementation of agent interaction protocols. However, these languages lack a formal semantics, and cannot directly be used for automatically proving properties of the designed components (protocols). The goal of this task is to analyze the most widespread modeling languages, in particular graphical languages, and to define techniques for translating them into the formalisms defined in the project.

Languages, techniques and tools developed within the project will be tested on different case studies, among which, in particular, clinical guidelines, web services, electronic commerce and security, multimedia presentations.

1.4 Durata del Programma di Ricerca

24 Mesi

1.5 Settori scientifico-disciplinari interessati dal Programma di Ricerca

INF/01 - Informatica

ING-INF/05 - Sistemi di elaborazione delle informazioni

1.6 Parole chiave

Testo italiano

SISTEMI MULTIAGENTE ; PROTOCOLLI DI INTERAZIONE ; VERIFICA AUTOMATICA ; LOGICA COMPUTAZIONALE ; LOGICA TEMPORALE

Testo inglese

MULTIAGENT SYSTEMS ; INTERACTION PROTOCOLS ; AUTOMATIC VERIFICATION ; COMPUTATIONAL LOGIC ; TEMPORAL LOGIC

1.7 Coordinatore Scientifico del Programma di Ricerca

MARTELLI
(Cognome)

ALBERTO
(Nome)

Professore Ordinario
(Qualifica)

03/06/1944
(Data di nascita)

MRTLRT44H03F952E
(Codice di identificazione personale)

INF/01 - Informatica
(Settore scientifico-disciplinare)

Università degli Studi di TORINO
(Università)

Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI
(Facoltà)

Dipartimento di INFORMATICA
(Dipartimento)

011/6706731
(Prefisso e telefono)

011/751603
(Numero fax)

mrt@di.unito.it
(Indirizzo posta elettronica)

1.8 Curriculum scientifico

Testo italiano

Laureato in Ingegneria Elettronica presso il Politecnico di Milano nel dicembre 1967. Dal 1968 al 1981 ricercatore presso l'Istituto di Elaborazione dell'Informazione del CNR a Pisa. Dal novembre 1981 professore ordinario presso il corso di laurea in Informatica dell'Università di Torino. Dal gennaio 1984 all'ottobre 1989 direttore del Dipartimento di Informatica dell'Università di Torino e coordinatore del Dottorato in Informatica dal 1988 al 1995. Dal 1998 al 2003 presidente del Consiglio di corso di Laurea in Informatica.

È stato responsabile del gruppo di Torino nel progetto CEE Basic ESPRIT "MEDLAR" (Mechanizing Deduction in the Logics of Practical Reasoning) 1989-95, coordinatore nazionale del Progetto MURST 40% "Rappresentazione della conoscenza e meccanismi di ragionamento", 1995-96, e responsabile dell'unità di Torino nel progetto di interesse nazionale "Agenti intelligenti: interazione e acquisizione di conoscenza", 1999-2000.

Attualmente è coordinatore nazionale del progetto PRIN 2003 "Sviluppo e verifica di sistemi multiagente basati sulla logica" ed è rappresentante del gruppo di Torino della Network of Excellence del VI framework REWERSE (Reasoning on the web with rules and semantics).

Ha svolto ricerca nel campo dell'intelligenza artificiale e della programmazione logica. Nell'ambito dell'intelligenza artificiale si è inizialmente occupato di algoritmi per la ricerca euristica e per l'unificazione di termini nella logica del prim'ordine. In seguito si è occupato di ragionamento non monotono, in particolare di Sistemi di Mantenimento della Verità, e di tecniche di ragionamento su azioni e cambiamento, facendo uso principalmente della logica modale. Recentemente ha contribuito alla proposta di una teoria delle azioni basata sulla "Dynamic Linear Time Temporal Logic" (DLTL), una combinazione della logica temporale lineare e della logica dinamica, che consente di esprimere proprietà temporali su azioni, anche complesse.

Si è anche occupato di estensioni di linguaggi di programmazione logica in grado di supportare ragionamenti ipotetici e non monotoni, e basati sulle azioni. Queste ricerche hanno portato alla definizione del linguaggio di programmazione logica DyLOG, basato sulla logica modale, che consente di definire azioni complesse, di rappresentare stati incompleti e di trattare azioni di "sensing".

Gli interessi di ricerca più recenti sono nell'area della specifica e verifica di proprietà di sistemi multiagente, e, in particolare, su come utilizzare formalismi e strumenti basati sulla logica per specificare l'interazione fra agenti e per verificarne proprietà.

Testo inglese

Alberto Martelli received a degree in Electrical Engineering from the Politecnico di Milan in 1967. From 1968 to 1981 he was with the Istituto di Elaborazione dell'Informazione of the National Research Council in Pisa. Since November 1981 is full professor of Computer Science at the University of Torino, where he was director of the Dipartimento di Informatica from 1984 to 1989, and chairman of the PhD program in Computer Science from 1988 to 1995.

He was head of a group in the ESPRIT BRA Project MEDLAR I and II "Mechanizing Deduction in the Logics of Practical Reasoning" (1989-95), "national coordinator" of the MURST 40% project "Knowledge representation and reasoning", 1995-96, and head of a group in the national project "Intelligent agents: Interaction and Knowledge acquisition", 1999-2000.

He is a coordinator of the national project PRIN 2003 "Logic-based development and verification of multi-agent systems", and representative of the Turin group in the VI framework Network of Excellence REWERSE (Reasoning on the web with rules and semantics).

He has carried out research mainly in artificial intelligence and logic programming.

In artificial intelligence he has worked initially on algorithms for heuristic search and unification. Later on he studied the semantic issues concerning nonmonotonic reasoning, in particular Truth Maintenance Systems (TMS), and techniques for reasoning about actions and change, mainly by making use of modal logic. In particular he contributed the development of a theory of actions based on Dynamic Linear Time Temporal Logic (DLTL), a simple combination of temporal logic of linear time and of dynamic logic, which allows to describe temporal properties of complex actions.

He has also worked on extensions logic programming languages to deal with hypothetical and non-monotonic reasoning, and to reason about actions. The outcome of these research activities is a modal logic programming language to reason about actions, called DyLOG, which allows to define complex actions (procedures), to represent incomplete states and to deal with sensing actions.

The most recent research interests are in the area of specification and verification of properties of multiagent systems, with the goal

of using the above mentioned logic based formalisms and tools to specify agent interactions and to prove their properties.

1.9 Pubblicazioni scientifiche più significative del Coordinatore del Programma di Ricerca

1. BALDONI M., BAROGLIO C., GUNGUI I., MARTELLI A., MARTELLI M., MASCARDI V., PATTI V., SCHIFANELLA C. (2005). Reasoning About Agents' Interaction Protocols Inside DCasLP. LECTURE NOTES IN COMPUTER SCIENCE. vol. 3476 pp. 112-131 ISSN: 0302-9743 Post-Proc. of the International Workshop on Declarative Agent Languages and Technologies, DALT'04.
2. BALDONI M., BAROGLIO C., MARTELLI A., PATTI V., SCHIFANELLA C. (2005). Verifying protocol conformance for logic-based communicating agents. LECTURE NOTES IN COMPUTER SCIENCE. ISSN: 0302-9743 Post Proc. of Fifth International Workshop on Computational Logic in Multi-Agent Systems, CLIMA V, to appear.
3. BALDONI M., BAROGLIO C., MARTELLI A., PATTI V. (2004). Reasoning about interaction protocols for web service composition. ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE. vol. 105 pp. 21-36 ISSN: 1571-0661 Proc. of 1st Int. Workshop on Web Services and Formal Methods, WS-FM 2004.
4. BALDONI M., GIORDANO L., MARTELLI A., PATTI V. (2004). Programming Rational Agents in a Modal Action Logic. ANNALS OF MATHEMATICS AND OF ARTIFICIAL INTELLIGENCE. vol. 41 pp. 207-257 ISSN: 1012-2443
5. GIORDANO L., MARTELLI A. (2004). On-the-fly automata construction for dynamic linear time temporal logic. 11th International Symposium on Temporal Representation and Reasoning (TIME'04). (pp. 133-139).
6. GIORDANO L., MARTELLI A., SCHWIND C. (2004). Verifying communicating agents by model checking in a temporal action logic. LECTURE NOTES IN COMPUTER SCIENCE. vol. 3229 pp. 57-69 ISSN: 0302-9743 9th European Conference on Logics in Artificial Intelligence (JELIA'04).
7. BALDONI M., BAROGLIO C., GIORDANO L., MARTELLI A., PATTI V. (2003). Reasoning about communicating agents in the semantic web. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2901 pp. 84-98 ISSN: 0302-9743 Proc. of the 1st International Workshop on Principle and Practice of Semantic Web Reasoning, PPSWR 2003.
8. BALDONI M., BAROGLIO C., MARTELLI A., PATTI V. (2003). Reasoning about Conversation Protocols in a Logic-based Agent Language. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2829 pp. 300-311 ISSN: 0302-9743 AI*IA 2003: Advances in Artificial Intelligence, 8th Congress of the Italian Association for Artificial Intelligence.
9. BALDONI M., BAROGLIO C., MARTELLI A., PATTI V. (2003). Reasoning about self and others: Communicating agents in a modal action logic. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2841 pp. 228-241 ISSN: 0302-9743 Theoretical Computer Science, 8th Italian Conference, ICTCS'2003.
10. GIORDANO L., MARTELLI A., SCHWIND C. (2003). Specifying and verifying systems of communicating agents in a temporal action logic. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2829 pp. 262-274 ISSN: 0302-9743 AI*IA 2003: Advances in Artificial Intelligence, 8th Congress of the Italian Association for Artificial Intelligence.
11. BALDONI M., GIORDANO L., MARTELLI A., PATTI V. (2001). Reasoning about Complex Actions with Incomplete Knowledge: A Modal Approach. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2202 pp. 405-425 ISSN: 0302-9743 Proc. of Theoretical Computer Science, 7th Italian Conference, ICTCS'2001.
12. GIORDANO L., MARTELLI A., SCHWIND C. (2001). Reasoning about actions in a multiagent domain. LECTURE NOTES IN COMPUTER SCIENCE. vol. 2175 pp. 237-248 ISSN: 0302-9743 Proc. of the 7th Congress of the Italian Association for Artificial Intelligence, AI*IA'01.
13. GIORDANO L., MARTELLI A., SCHWIND C. (2001). Reasoning About Actions in Dynamic Linear Time Temporal Logic. LOGIC JOURNAL OF THE IGPL. vol. 9 pp. 298-303 ISSN: 1367-0751
14. GABBAY D. M., GIORDANO L., MARTELLI A., OLIVETTI N., SAPINO M. L. (2000). Conditional Reasoning in Logic Programming. JOURNAL OF LOGIC PROGRAMMING. vol. 44 pp. 37-74 ISSN: 0743-1066
15. GIORDANO L., MARTELLI A., SCHWIND C. (2000). Ramification and Causality in a Modal Action Logic. JOURNAL OF LOGIC AND COMPUTATION. vol. 10(5) pp. 625-662 ISSN: 0955-792X
16. BALDONI M., GIORDANO L., MARTELLI A. (1998). A Modal Extension of Logic Programming: Modularity, Beliefs and Hypothetical Reasoning. JOURNAL OF LOGIC AND COMPUTATION. vol. 8(5) pp. 597-635 ISSN: 0955-792X
17. BALDONI M., GIORDANO L., MARTELLI A., PATTI V. (1998). A Modal Programming Language for Representing Complex Actions. WORKSHOP ON TRANSACTIONS AND CHANGE IN LOGIC DATABASES, DYNAMICS '98. (pp. 1-15).
18. GIORDANO L., MARTELLI A., SCHWIND C. (1998). Dealing with Concurrent Actions in Modal Action Logics. Proc. 13th European Conf. on Artificial Intelligence, ECAI '98. (pp. 537-541).
19. GIORDANO L., MARTELLI A. (1995). A Logical Characterization for Truth Maintenance Systems with Dependency-Directed Backtracking. COMPUTATIONAL INTELLIGENCE. vol. 11 pp. 11-46 ISSN: 0824-7935
20. GIORDANO L., MARTELLI A. (1994). On cumulative default logics. ARTIFICIAL INTELLIGENCE. vol. 66 pp. 161-179 ISSN: 0004-3702

1.10 Elenco delle Unità di Ricerca

Unità	Responsabile Scientifico	Qualifica	Settore Disc.	Università	Dipart./Istituto	Mesi Uomo
I	MARTELLI ALBERTO	Professore Ordinario	INF/01	Università degli Studi di TORINO	Dip. INFORMATICA	12
II	GIORDANO LAURA	Professore Associato	INF/01	Università degli Studi del PIEMONTE ORIENTALE "Amedeo Avogadro"-Vercelli	Dip. INFORMATICA	12
III	LAMMA EVELINA	Professore Ordinario	ING-INF/05	Università degli Studi di FERRARA	Dip. INGEGNERIA	6
IV	MELLO PAOLA	Professore Ordinario	ING-INF/05	Università degli Studi di BOLOGNA	Dip. ELETTRONICA, INFORMATICA E SISTEMISTICA	6

1.11 Mesi uomo complessivi dedicati al programma

Testo italiano

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
Personale universitario dell'Università sede dell'Unità di Ricerca		16	65	66	131
Personale universitario di altre Università		2	8	8	16
Titolari di assegni di ricerca		1	1	3	4
Titolari di borse	<i>Dottorato</i>	5	18	15	33
	<i>Post-dottorato</i>	0			
	<i>Scuola di Specializzazione</i>	0			
Personale a contratto	<i>Assegnisti</i>	0			
	<i>Borsisti</i>	1	0	4	4
	<i>Dottorandi</i>	0			
	<i>Altre tipologie</i>	3	0	14	14
Personale extrauniversitario		0			
TOTALE		28	92	110	202

Testo inglese

		Numero	Mesi uomo 1° anno	Mesi uomo 2° anno	Totale mesi uomo
University Personnel		16	65	66	131
Other University Personnel		2	8	8	16
Work contract (research grants, free lance contracts)		1	1	3	4
PHD Fellows & PHD Students	<i>PHD Students</i>	5	18	15	33
	<i>Post-Doctoral Fellows</i>	0			
	<i>Specialization School</i>	0			
Personnel to be hired	<i>Work contract</i>	0			
	<i>PHD Fellows & PHD Students</i>	1	0	4	4
	<i>PHD Students</i>	0			
	<i>Other tipologies</i>	3	0	14	14
No cost Non University Personnel		0			
TOTALE		28	92	110	202

PARTE II

2.1 Obiettivo del Programma di Ricerca

Testo italiano

I servizi forniti in molti domini applicativi sempre più possono essere descritti e realizzati come espressione di un insieme di agenti cooperanti. Rispetto all'approccio più tradizionale della programmazione a componenti, il paradigma ad agenti esplicita aspetti sociali, connessi alla caratteristica degli agenti di essere autonomi e proattivi: le componenti comunicano e si coordinano in maniera dinamica, utilizzando linguaggi ad alto livello, per perseguire scopi comuni (o propri). Tra gli aspetti sociali è particolarmente importante definire "regole comportamentali", espresse mediante protocolli di comunicazione (interazione), finalizzate a controllare l'organizzazione del sistema. I protocolli di comunicazione possono essere utilizzati per regolare l'interazione fra gli agenti specificando un insieme di conversazioni accettabili, ossia sequenze di operazioni (in particolare scambi di messaggi) che si possono svolgere fra agenti diversi.

Già da tempo l'avvento dei sistemi distribuiti ha portato l'attenzione della comunità scientifica sui protocolli di interazione tra le varie entità coinvolte in una computazione. Con la rapida espansione di internet, poi, ha assunto particolare importanza la specifica di protocolli di comunicazione che garantiscano proprietà generali (ad esempio assenza di deadlock nell'interazione) e particolari al protocollo stesso, ed è sorta l'esigenza di poter verificare se un protocollo garantisca o meno determinate proprietà. A tale scopo linguaggi formali e strumenti di analisi sono considerati come essenziali, soprattutto nell'ambito dei protocolli di sicurezza, dove prove formali di validità di proprietà assumono importanza vitale per il protocollo stesso. In particolare hanno dato ottimi risultati le tecniche di model checking, basate sulle logiche temporali, per cui sono stati sviluppati numerosi strumenti anche a livello commerciale.

Analogamente, nell'ambito della comunità di ricerca su agenti e sistemi multi-agente, si è avvertita la necessità di verificare proprietà di protocolli di comunicazione. Tuttavia le tecniche sviluppate per la verifica di proprietà in sistemi distribuiti non sono direttamente applicabili alla verifica di proprietà di sistemi multiagente a causa delle caratteristiche diverse di queste proprietà, che riguardano prevalentemente le conversazioni fra agenti e si basano sulla semantica delle azioni comunicative e sul loro effetto sullo stato sociale.

L'obiettivo di questo progetto è quello di sviluppare strumenti e metodologie per la specifica e la verifica di proprietà di sistemi multiagente. Più precisamente i principali obiettivi del progetto sono i seguenti:

TASK 1. Definizione di formalismi per la specifica e la verifica di protocolli di interazione

Verranno analizzate le principali proprietà che si intendono verificare, e verranno individuati i formalismi più adeguati per esprimere queste proprietà. Tra i problemi di verifica si possono citare: la verifica (statica) delle proprietà di protocolli, la verifica della conformità di un agente con un protocollo di interazione, la verifica (runtime) che l'agente stia (o non stia) rispettando i suoi vincoli sociali, l'interoperabilità di un insieme di agenti, la verifica che un protocollo ottenuto per composizione conservi le proprietà dei protocolli che lo compongono. I formalismi utilizzati nel progetto per specificare protocolli di comunicazione e per descriverne le proprietà saranno prevalentemente basati sulla logica (temporale e computazionale), ma si prenderanno in considerazione anche formalismi a vincoli, linguaggi formali ed automi.

TASK 2. Verifica automatica delle proprietà

L'utilizzo di linguaggi formali di specifica consente di sviluppare tecniche di verifica formale di proprietà. In questo task verranno messe a punto tecniche di verifica per i formalismi sviluppati nel task precedente. Saranno confrontate varie tecniche al riguardo, in funzione della modalità utilizzata per esprimere il protocollo e le sue proprietà. In particolare, per le logiche temporali, saranno sviluppate tecniche di model checking. In altri casi, si applicheranno tecniche basate su procedure di dimostrazione e si indagheranno tecniche basate su risoluzione estesa. Verranno anche sviluppate tecniche di prototipazione rapida per la validazione di protocolli di interazione.

TASK 3. Linguaggi di modellazione e loro traduzione nei linguaggi formali definiti nel Task 1

Per lo sviluppo dei sistemi multiagente sono stati proposti numerosi linguaggi di modellazione, come ad esempio AUMML o BPEL4WS. Questi linguaggi sono particolarmente adatti all'analisi dei requisiti e al progetto, fasi in cui le tecniche formali mostrano alcuni limiti; ad esempio questi linguaggi possono essere trattati mediante editor grafici che consentono di generare il codice in modo automatico. Tuttavia, poiché mancano di semantica formale essi non forniscono strumenti per la dimostrazione automatica di proprietà delle componenti (protocolli) progettate. Un ulteriore obiettivo del progetto è quello di analizzare i linguaggi di modellazione più diffusi, in particolare formalismi grafici, e di definire tecniche di traduzione nei formalismi sviluppati nel Task 1.

TASK 4. I linguaggi, le tecniche e gli strumenti sviluppati nell'ambito dei punti precedenti verranno valutati in diversi ambiti applicativi, fra cui principalmente:

4.1 Linee guida cliniche: si vuole analizzare l'applicabilità alle linee guida cliniche dei formalismi per la specifica dei protocolli sviluppati nel Task 1, in particolare di formalismi in grado di gestire vincoli temporali.

4.2 Servizi web: verranno sviluppate tecniche di diagnosi di servizi web e verranno studiate tecniche di ragionamento che consentano di selezionare un servizio appropriato.

4.3 Commercio elettronico e sicurezza: gli strumenti sviluppati in questo progetto saranno applicati alla verifica di proprietà di protocolli documentati in letteratura.

4.4 Presentazioni multimediali: si intende modellare presentazioni multimediali come sistemi cooperanti e studiare l'applicabilità di

tecniche di model checking alla verifica di proprietà.

Testo inglese

In many application domains it is getting more and more common describing and realizing the offered services by means of a set of cooperative agents. Compared with the more traditional component based programming approach, the agent based paradigm makes explicit the social aspects, related to the fact that agents are autonomous and proactive: the different components dynamically communicate and coordinate with each other, by means of higher level languages, to reach some common (or their own) goal. Among the social aspects, specifically relevant is the ability of defining behavioral rules, expressed by means of communication (interaction) protocols, aiming at controlling the organization of the system. Such protocols can be used to rule the agents' interaction, by specifying a set of acceptable conversations, i.e. sequences of operations (in particular message exchanges) that could occur among different agents.

The advent of distributed systems has addressed the attention of the scientific community on the interaction protocols between multiple communicating entities. With the fast expansion of Internet, particular importance has been given to the specification of communication protocols that guarantee either general properties (for instance, the deadlock-freedom property) or particular properties (as an example, guarantee of message correct transmission, transmission error detection etc.) of a given protocol. The need for the formal verification of a given protocol has recently emerged as a fundamental requirement. To such purpose formal languages and analysis tools are currently considered aspects of primary importance, especially for security protocols, where formal proofs of properties are indeed fundamental. In particular, the results obtained in model checking have been proved extremely useful for the verification of protocols, where protocol properties can be expressed through formulas in temporal linear logic. At the same time, the research work done within the agents community has pointed out the need of regulating the interaction between agents in a multi-agent system by means of the specification of agents interaction protocols, and of verifying their properties. However the techniques developed for protocol verification in distributed systems cannot be directly applied to the verification of properties of interaction protocols in multi-agent systems, because these properties concern mainly conversations among agents, and are based on the semantics of communicative actions and their effects on the social state.

The aim of this project is to develop tools and methodologies for specification and verification of multi-agent systems properties. More precisely, the following tasks will be pursued:

TASK 1. Definition of formalisms for the specification and verification of interaction protocols

The most significant properties of interaction protocols will be analyzed, so as to single out the most suitable formalisms for expressing these properties. Among the verification problems we mention: the (static) verification of protocol properties, the verification of the compliance of an agent with an interaction protocol, the (runtime) verification that an agent is (is not) respecting its social facts (commitments and permissions), interoperability of a set of agents, the verification that a protocol obtained by composition preserves the properties of the component protocols. The kind of formalisms used in the project will be mainly based on temporal and computational logics, but other formalisms, such as temporal constraints, formal languages and automata, will be considered.

TASK 2. Automatic property verification

The formal definition of protocol properties makes it possible to automatically verify the properties of a specific protocol. For this purpose, several techniques exist, whose applicability depends on how protocols and their properties are expressed. Various approaches to protocol verification will be developed for the formalisms proposed in Task 1. In particular, model checking techniques, as well as techniques based on theorem proving or on extended resolution will be investigated. We plan also to develop rapid prototyping techniques for validating interaction protocols.

TASK 3. Modeling languages and their translation into the formal languages defined in Task 1

The design and implementation of agent interaction protocols can be seen as a multi-step process that needs to be engineered and at every stage different languages have been proposed, such as for instance UML-based languages, like AUMML. In general, these languages are particularly suitable for the requisite analysis and design phases, in which formal techniques show some limitations. For instance they can be used through graphical editors that allow to generate the code in an automatic way. However, since they lack a formal semantics, they do not supply tools for automatically proving properties of the designed components (protocols). The goal of this task is to analyze the most widespread modeling languages, in particular graphical languages, and to define techniques for translating them into the formalisms defined in Task 1.

TASK 4. Case studies

Languages, techniques and tools developed within the previous tasks will be experimented in the following case studies:

4.1 Clinical guidelines

This case study aims at analyzing the applicability to clinical guidelines of the formalisms for the specification of protocols developed in Task 1, in particular of the formalisms able to deal with temporal constraints.

4.2 Web services

We intend to develop techniques for the diagnosis of web services, and reasoning techniques for selecting suitable services.

4.3 Electronic commerce and security

The tools developed in the project will be applied to the verification of well-known protocols.

4.4 Multimedia presentations

We intend to model multimedia presentations as cooperating systems, and to develop model checking techniques for verifying their properties.

2.2 Base di partenza scientifica nazionale o internazionale

Testo italiano

I servizi forniti in molti domini applicativi sempre più possono essere descritti e realizzati come espressione di un insieme di agenti cooperanti. Alcuni esempi sono i processi manifatturieri, i web service, i mercati on-line, la gestione distribuita di reti, le presentazioni multimediali. Rispetto all'approccio più tradizionale della programmazione a componenti, il paradigma ad agenti esplicita aspetti sociali, connessi alla caratteristica degli agenti di essere autonomi e proattivi. L'approccio tradizionale si fonda su una visione funzionale, in cui le componenti richiedono determinati input e producono determinati risultati. Il sistema ha un'architettura basata sul principio di decomposizione statica-funzionale dove le interazioni fra le componenti sono date dalle loro dipendenze. In quello ad agenti, invece, le componenti comunicano e si coordinano in maniera dinamica, utilizzando linguaggi ad alto livello, per perseguire scopi comuni (o propri). Tra gli aspetti sociali è quindi particolarmente rilevante la definizione di "regole comportamentali", finalizzate a controllare l'organizzazione del sistema; fra queste i protocolli di comunicazione sono l'esempio più significativo. I protocolli di comunicazione possono essere utilizzati per regolare l'interazione fra gli agenti, e per verificare se un agente può o meno fare parte del sistema o per controllare se un sistema si sta comportando nel modo stabilito. In generale, tale astrazione consente di realizzare e gestire sistemi aperti, in cui nuove componenti possono entrare in gioco dinamicamente. L'ingresso di una componente in un contesto di esecuzione è determinato in base a un ragionamento sul comportamento da essa esibito: l'ingresso è possibile solo se essa rispetta il corpo delle regole del sistema, inteso come società.

Particolare importanza ha assunto la specifica di protocolli che garantiscano proprietà generali, come l'assenza di deadlock nell'interazione, o particolari al protocollo stesso, ad es. garanzia di trasmissione corretta di un messaggio o rilevazione dell'avvenuto errore. Il problema della verifica di proprietà è per esempio molto sentito nel campo dell'E-commerce, dove la specifica chiara ed esatta di protocolli sicuri, che godano cioè di "proprietà di sicurezza", è parte fondamentale della disciplina stessa [Gollmann2003]. Per consentire la verifica di proprietà è necessario utilizzare linguaggi formali e strumenti di analisi. In questo ambito importanti contributi sono stati sviluppati nell'area dei sistemi distribuiti e concorrenti. In particolare si sono rivelate estremamente funzionali alla verifica dei protocolli i risultati ottenuti nell'ambito del model checking [CGP2000, BBFL2001]. A titolo di esempio, citiamo il sistema SPIN [Holzmann1992, SPIN] nel quale le entità interagenti sono definibili come automi a stati finiti, tramite il linguaggio PROMELA (PROtocol MEta Language), e le proprietà particolari di un protocollo possono essere espresse tramite formule in logica lineare temporale. SPIN supporta anche la verifica di alcune proprietà generali dei protocolli, quali ad esempio l'assenza di deadlock. Sempre nell'ambito dei sistemi distribuiti, si sono affermate come strumento per la definizione e la verifica dei protocolli le Coloured Petri Nets (CPNs) [Jensen1997, CCFL99]. Utilizzando il linguaggio formale delle CPNs è possibile specificare graficamente i protocolli, e tramite confronti tra gli Occurrence Graph generati dalle specifiche è possibile verificare sia proprietà generali sia proprietà particolari. Un esempio dell'uso delle CPNs al fine di specificare protocolli e verificarne proprietà è dato in [BGH2004].

Di recente le logiche temporali hanno suscitato interesse nell'area del ragionamento su azioni e pianificazione [Bacchus98, GMS00a, GMS00b, Pistore01, CDV02], così come nella specifica e verifica di sistemi di agenti che comunicano. Per esempio in [Wooldridge02b] gli agenti sono scritti in MABLE, un linguaggio di programmazione imperativa. Le proprietà formali del sistema sono espresse usando una logica BDI linear time quantificata e possono essere verificate automaticamente usando SPIN. [BGS98] estende le tecniche di model checking rendendole applicabili a MAS i cui agenti hanno attitudini BDI. [Guerin02] definisce un framework che fornisce una semantica dichiarativa della comunicazione tra agenti. [PL03] presenta un framework per verificare proprietà temporali ed epistemiche dei MAS utilizzando la logica CTLK. [GMS04] presenta un formalismo per la specifica e verifica basato su una logica temporale dinamica (DLTL [Henriksen99]) e su una teoria temporale del ragionamento su azioni. Infine [YS02] propone un approccio sociale alla specifica e esecuzione di protocolli basato sull'event calculus.

Un altro problema molto studiato, noto come test di conformance, consiste nel verificare se una certa implementazione (politica di interazione di un agente) rispetta un dato protocollo astratto. Il test di conformance è uno strumento che, verificando il rispetto di un protocollo, certifica l'interoperabilità. Sono stati studiati due tipi di conformance: quella a priori, e quella run-time (o compliance) [EMST04, GP03]. Intuitivamente il primo tipo è una proprietà dell'implementazione nel suo complesso, verifica se un agente produrrà mai conversazioni che violano il protocollo, mentre il secondo è una proprietà della conversazione in corso ed è finalizzato a verificare che essa sia legale. In [B05] è stata affrontata la verifica della conformance a priori nel caso specifico in cui i protocolli sono specificati in AUML e le policy di conversazione nel linguaggio logico modale DyLOG [B04], provando che i sequence diagram AUML possono essere tradotti in linguaggi regolari e la verifica della conformance può essere interpretata come la verifica dell'inclusione di due linguaggi.

Le tecniche per la specifica e verifica del comportamento interattivo in sistemi aperti trovano un'immediata applicazione nei servizi web. Sono attualmente allo studio descrizioni formali che consentano forme di interoperazione automatizzata attraverso processi di matchmaking, selezione, specializzazione e composizione automatica. La definizione di nuovi protocolli per aggregazione di protocolli esistenti è interessante dal punto di vista della progettazione [Mallya04, ACKM04] in quanto raffinamento e aggregazione forniscono nozioni di astrazione fondamentali per la riusabilità. Questo è particolarmente importante per i protocolli di E-business poiché un ambito di business realistico richiede l'integrazione di una varietà notevole di protocolli [Mallya04]. Alcune interessanti proposte derivano dalla comunità di Semantic Web [OWLS, MS01], dove nel contesto del progetto OWL-S, sono applicate tecniche derivate dal situation calculus per produrre composizioni personalizzate di servizi. Uno studio preliminare dei benefici (in termini di personalizzazione di selezione e composizione) prodotti dalla descrizione dichiarativa del comportamento comunicativo dei web service è offerto da [B03, WFSM04]. I servizi sono intesi come agenti software che comunicano tramite protocolli predefiniti, pubblici e condivisi, modellati come insiemi di clausole logiche rappresentanti policy. Una descrizione a policy è più ricca di quella usata solitamente nel matchmaking ad input, output, precondizioni ed effetti. Con una specifica logica è possibile ragionare sugli effetti dell'intrattenere conversazioni specifiche ed eseguire automaticamente la selezione e la composizione. Questo approccio può essere visto come un secondo passo nel processo di matchmaking, che restringe l'insieme dei servizi già selezionati ed esegue la personalizzazione. In parallelo all'iniziativa di Semantic Web, anche il mondo dell'industria si è interessato al discorso dell'interoperabilità dei servizi. L'interesse riguardo modelli di orchestrazione e coreografia finalizzati alla composizione automatica ha condotto alla proposta di nuovi linguaggi che consentono la specifica di servizi complessi sulla base di altri più

semplici, es. BPEL4WS [BPEL4WS] e WSCI, la cui semantica non è però ancora definita in modo preciso. I metodi formali proposti derivano dai modelli formali della concorrenza e coordinazione in sistemi distribuiti (es. algebre di processi). Tuttavia gli standard come BPEL4WS consentono per lo più l'automazione di composizioni manuali [MM03,KS03]: le specifiche BPEL4WS sono scritte manualmente e non si ha alcuna composizione basata su processi di ricerca. Ciò riduce il grado di adattabilità dinamica della composizione rispetto alle preferenze dell'utente.

L'uso dei linguaggi formali non è ancora molto diffuso nella fase di progetto dei protocolli di interazione, dove spesso si utilizzano solamente notazioni grafiche informali. Esempio il caso della specifica del protocollo TCP (Transmission Control Protocol [RFC793]), dove la semantica dei messaggi è addirittura espressa in linguaggio naturale, con ampie parti (es. la temporizzazione) non specificate affatto. La realizzazione di protocolli di interazione può in realtà essere vista come un processo a più passi da ingegnerizzare [HK03], per ogni stadio sono stati proposti linguaggi differenti ma le loro relazioni non sono ancora del tutto chiare. Per quel che riguarda la modellazione, sono usati linguaggi basati su UML, come AUML [OPB00], adottato da FIPA. Per quel che riguarda invece la specifica formale, le proposte sono diverse, fra queste automi a stati finiti [BF95], reti di petri [KFD98] e logica temporale. In generale, i linguaggi di modellazione sono adatti all'analisi dei requisiti e al progetto, fasi in cui le tecniche formali mostrano alcuni limiti, es. possono essere mostrati in modo grafico e si può generare il codice in modo automatico. Tuttavia, mancando di semantica formale essi non forniscono strumenti per la dimostrazione automatica di proprietà delle componenti (protocolli) progettate [Ha04]. Una questione aperta è l'integrazione di specifiche scritte in linguaggi diversi [ES03,Alo04,Hu04,Br03] che richiede meccanismi di meta-modellizzazione [QVT]. UML [UML2] stesso è un insieme di sotto-linguaggi integrati. In questo contesto c'è bisogno di rappresentare le trasformazioni fra i linguaggi, rendendole esplicite [QVT,MM02,ES03,Ev03,SG00].

Nell'ambito dello sviluppo di sistemi ad agenti, sono state presentate varie proposte di linguaggi grafici, basate per lo più su macchine a stati finiti [BF95]; più recenti due formalismi basati su di un approccio differente, AUML [Huge2004, BMO2001] e AML [CT2004]. AUML propone un'estensione degli Interaction Diagram di UML al mondo degli agenti. AUML supporta l'eterogeneità delle entità interagenti, permettendo di definire in modo intuitivo gli attori partecipanti all'interazione e i messaggi consentiti nell'ambito del protocollo. Alla semplicità ed intuitività del formalismo grafico non corrisponde però una formalizzazione completa. Vista la rilevanza scientifica del problema della verifica/validazione di diagrammi AUML, non stupisce che in letteratura si trovino già diversi lavori che propongono soluzioni parziali ad esso, es. [AMMR03, B05, Bal04]. AML riprende e completa la proposta AUML, estendendo il linguaggio grafico per la specifica di protocolli; ancora non viene fornita però una semantica completa dei messaggi scambiati (se non in termini UML); pertanto anche AML non supporta la verifica formale di proprietà. A differenza di AUML, AML aggiunge anche un notevole contributo per la definizione dell'architettura degli agenti stessi, estendendo altre notazioni grafiche proposte in precedenza. È plausibile supporre che tecniche di verifica di proprietà basate su model checking potranno essere applicate proprio a partire dall'architettura degli agenti. Va notato, infine, che il formalismo basato sui Message Sequence Charts, al quale si ispirano AUML e AML, è divenuto uno standard nel settore delle telecomunicazioni dell'International Telecommunication Union (ITU) [ITU-TS-Z120]. Solo recentemente è stato discusso l'uso di tale formalismo in riferimento alla verifica di proprietà [GMP2004], con esiti non del tutto positivi.

Le tecniche formali per descrivere e ragionare su sistemi complessi possono essere applicate anche al dominio delle presentazioni multimediali. Una presentazione multimediale può essere caratterizzata in termini dell'insieme di oggetti atomici da presentare, dell'insieme dei dispositivi virtuali usati per riprodurre componenti atomiche, dell'insieme degli eventi che possono causare un cambiamento nella presentazione (es. la terminazione di un oggetto), l'insieme delle regole di sincronizzazione e di quelle di composizione spaziale. In [Be03,Be04] è presentato un modello formale astratto per descrivere il comportamento di una presentazione multimediale in termini di allocazione di risorse e sincronizzazione fra le componenti. Ogni oggetto è visto come un'entità indipendente con un proprio comportamento ed esigenze di allocazione. L'idea chiave è di modellare i singoli oggetti come macchine a stati finiti che codificano un comportamento reattivo. La presentazione complessiva è modellata come una composizione parallela (o sequenziale) di esecuzioni di singole componenti multimediali. Questo modello consente il ragionamento sulle dinamiche dei documenti multimediali e la dimostrazione di loro proprietà. Questa è una novità rispetto ad altre proposte in letteratura [Ca98,Po01]. L'uso di un'estensione delle macchine a stati finiti, gli hyperchart, è proposto in [Pa99] dove viene fornito un meccanismo per specificare requisiti degli ipermedia. In [La02] sono discussi gli effetti dell'incertezza ed è proposto un algoritmo di scheduling flessibile.

Altro dominio applicativo è costituito dalle linee guida cliniche, definibili come framework per specificare le procedure cliniche e per standardizzarle. In quanto tali, possono essere viste come protocolli. Recentemente, la comunità medica ha incominciato a riconoscere che un trattamento computerizzato delle linee guida fornisce notevoli vantaggi, come la connessione automatica ai database dei pazienti e strumenti per prendere decisioni. Nell'ultima decade, questa osservazione ha motivato lo sviluppo di vari sistemi informatici indipendenti dal dominio per la loro gestione [Pe03]. Inoltre, gli strumenti informatici per la gestione delle linee guida stanno guadagnando importanza nelle aree dell'Intelligenza Artificiale in Medicina e dell'Informatica Medica [TMT01, Ter04].

Tra le proposte di linguaggi di specifica per protocolli basati su logica è doveroso menzionare i risultati di due progetti europei, Alfebiite (Progetto IST-1999-10298, [Alfebiite]), che utilizza la logica deontica [vdT03], e SOCS (Progetto IST-2001-32530, [SOCS]), basato invece su logica computazionale. Entrambi i progetti hanno avuto come scopo la modellazione di sistemi ad agenti. In Alfebiite sono stati applicati modelli formali di attività (regolate da norme) alla definizione, gestione e regolazione delle interazioni tra "infohabitants" nella società dell'informazione. SOCS (Societies Of Computees) ha invece posto l'attenzione sulla modellazione di società di agenti logici (Computees) interagenti in ambienti globali ed aperti, con un approccio basato sulla logica computazionale. Ad es. In [Al03a,Al03b] i protocolli sono espressi tramite vincoli di integrità sociale che specificano le azioni eseguibili (messaggi che possono essere inviati) in seguito ad eventi accaduti in passato. SOCS ha mostrato chiaramente le potenzialità ed i vantaggi di un approccio alla specifica e verifica dei protocolli basato su logica computazionale [GC2004].

Testo inglese

Nowadays, in many application domains it is getting more and more common describing and realizing the offered services by means of a set of cooperative agents. This is the case, for example, for manufacturing processes, web services, on-line markets, distributed network management, and multimedia presentations. Compared with the more traditional component based programming approach,

the agent based paradigm makes social aspects, related to the fact that agents are autonomous and proactive, explicit. The traditional approach is based on a functional view, in which the different components require some specific input, and produce some specific output. The system's architecture is based on the principle of static-functional decomposition, where the interactions among the different components are given by their dependencies. In the agent paradigm, instead, the different components dynamically communicate and coordinate with each other by means of higher level languages, to reach some common (or their own) goal. Among the social aspects, specifically relevant is the ability of expressing behavioural rules, aiming at controlling the organization of the system; communication protocols are the most significant example of such rules. Such protocols can be used to rule the agents' interaction. Therefore, they can be used to check if a given agent can, or cannot, take part into the system, or to check whether the system is behaving as expected. In general, based on this abstraction, open systems can be realized, in which new components can dynamically join the system. The insertion of a new component in an execution context is determined according to some form of reasoning about its behaviour: it will be added provided that it satisfies the body of the rules within the system, intended as a society.

Particular importance has been given to communication protocols that guarantee either general properties, e.g. Deadlock-freedom, or specific properties of a given protocol, e.g. transmission error detection. Property verification is, for instance, an open problem in E-commerce, where the clear and exact specification of safe protocols, i.e. protocols that fulfill security properties, has become an inalienable requirement [Gollmann2003]. Property verification is made possible by the use of formal languages and analysis tools. Important contributions derive from the research areas of distributed and concurrent systems. In particular, the results obtained in model checking [CGP2000, BBFL2001] have been proved extremely useful for the verification of protocols. A notable example is the SPIN system [Holzmann1992, SPIN] where interacting entities can be defined as finite state automata through PROMELA (PROtocol LAnguage Goal), and protocol properties can be expressed through formulas in temporal linear logic. SPIN supports also the verification of protocol general properties, such as deadlock-freedom. Another important tool, for both the definition and the verification of protocols, from the area of Distributed Systems is Coloured Petri Nets (CPNs) [Jensen1997, CCF199]. The formal language of the CPNs allows the graphical specification of protocols; moreover, by comparing the Occurrence Graphs generated from the specification, it is possible to check both general and particular properties. [BGH2004] reports an example of the use of CPNs for protocol specification and property verification.

Recently, temporal logics are receiving a growing attention in the area of reasoning about actions and planning [Bacchus98, GMS00a, GMS00b, Pistoro01, CDV02], as well as in the specification and verification of systems of communicating agents. For instance, in [Wooldridge02b] agents are written in MABLE, an imperative programming language, the formal claims about the system are expressed using a quantified linear time temporal BDI logic, and they can be automatically verified by means of SPIN. [BGS98] extend model checking techniques to make them applicable to multi-agent systems, where agents have BDI attitudes. [Guerin02] defines a framework which gives agent communication a grounded declarative semantics. [PL03] presents a framework for verifying temporal and epistemic properties of MASs based on the logic language CTLK. [GMS04] have developed a formalism for the specification and verification based on a temporal dynamic logic (DLTL [Henriksen99]) and on a temporal theory for reasoning about actions. Finally, [YS02] proposes a social approach to protocol specification based on event calculus.

A different problem deeply investigated in the literature, known as conformance testing, amounts to verify if a given implementation (an agent interaction policy) respects a given abstract protocol. The conformance test is a tool that, by verifying that agents respect a protocol, certifies their interoperability. Two kinds of conformance have been studied: a priori conformance, and run-time conformance (or compliance) [EMST04, GP03]. Intuitively the first kind of conformance is a property of the implementation as a whole, it checks if an agent will never produce conversations that violate the abstract interaction protocol, while the latter is a property of the on-going conversation and it is aimed at verifying if that conversation is legal. In [B05] a priori conformance verification has been faced in the specific case in which protocols are specified in AUML and agent conversation policies are written in the modal logic language DyLOG [B04], proving that AUML sequence diagrams can be translated in regular languages and that conformance verification can be interpreted as the problem of verifying the inclusion of two languages.

Techniques for the specification and verification of the interactive behavior of open agent systems find an immediate application in web services. Formal descriptions that allow forms of automated interoperation through matchmaking, selection, composition and specialization processes are currently being studied. The definition of new protocols by aggregating existing ones is interesting from the point of view of protocol design [Mallya04, ACKM04], as refinement and aggregation provide abstract notions which are fundamental to enable the reuse. This is particularly needed for the design of E-Business protocols because realistic business settings need the integration of an endless variety of business processes [Mallya04]. Some interesting proposals have been drawn by the Semantic Web community [OWLS, MS01], where in the context of the OWL-S project, techniques derived from the situation calculus have been applied to dynamically compose personalized services. By taking the abstraction of web services as software agents, that communicate by following predefined, public and sharable interaction protocols, [B03, WSFM04] propose a preliminary study of the benefits provided by a declarative description of their communicative behavior, in terms of personalization of the service selection and composition. The interaction protocols provided by web services are modelled by a set of logic clauses representing policies. A description by policies is richer than the usual description consisting in the inputs and outputs, preconditions and effects taken into account for the matchmaking. Having a logic specification of the protocol, it is possible to reason about the effects of engaging specific conversations and to perform in an automatic way the selection and composition tasks. This approach can be considered as a second step in the matchmaking process, which narrows a set of already selected services and performs a customization of the interaction with them. In parallel to the semantic web initiative, industry also moved towards the goal of seamless interoperation among web services. The interest about orchestration and choreography models for obtaining automatic composition of web services, led to the definition of new language proposals that support the specification of complex services out of simpler ones, e.g. BPEL4WS [BPEL4WS] and WSCI, whose semantics is not precisely defined yet. The proposed formal methods derive from formal models for concurrency and coordination of distributed systems (e.g. process algebras). However, standards like BPEL4WS mainly enable automation of hand-written compositions [MM03, KS03]: BPEL4WS specifications are written manually and no composition based on a search process takes place. This reduces flexibility in dynamically adapting the composition according to the user's preferences.

The use of formal languages is not yet very much diffused in the design of interaction protocols, task for which informal graphical notations are more often used. Exemplary is the case of the TCP (Transmission Control Protocol [RFC793]) protocol specification. The protocol is described through an informal graphical notation, the semantics of messages is expressed in natural language, and a

wide part of the protocol (e.g. the timing) is even not specified at all. The design and implementation of agent interaction protocols can, actually, be seen as a multi-step process that needs to be engineered [HK03]; at every stage different languages have been proposed but their relations are not yet clear. For what concerns modelling, UML-based languages are mostly used, such as AUML [OPB00], adopted by FIPA. For what concerns formal specification, many are the proposals, including finite state automata [BF95] petri nets [KFD98] temporal logic. In general, modellization languages are suitable to the requisite analysis and design phases (they can be displayed in a graphical way, code can be generated automatically), in which formal techniques show some limitation. However, since they lack a formal semantics, they do not supply tools for automatically proving properties of the designed components (protocols) [Ha04]. An open issue is the integration of specifications written in different languages [ES03,Alo04,Hu04,Br03] that requires meta-modelization mechanisms [QVT]. UML [UML2] itself is a set of integrated sub-languages. In this context there is a need of representing the transformations among the languages, making them explicit [QVT,MM02,ES03,Ev03,SG00].

In the multi-agent system development area, several proposals of graphical languages for protocol definition have been introduced, mostly based on finite states automata [BF95,KITO95]; more recent are two languages, AUML [Hugel2004, BMO2001] and AML [CT2004], based on a different approach. AUML proposes an extension of the Interaction Diagram of standard UML to the agents' world. It supports the heterogeneity of the interacting entities, allowing an intuitive definition of the actors that take part to the interaction, and of the messages allowed in the protocol. Although the AUML graphical formalism is easy and intuitive, the language still misses a complete formalization. Giving the relevance of the problem of verification/validation of AUML diagrams, it is not surprising that we can already find in the literature works that provide a partial solution to this problem, e.g. [AMMR83, B05, Bal04]. AML resumes and completes the AUML proposal by extending the graphical specification language for protocols; however, it still does not supply a complete message semantics and, for this reason, it does not support the formal verification of properties either. Differently from AUML, AML also provides a remarkable contribution for the definition of agent architectures, by extending well known graphical notations. It is reasonable to think that in the future property verifications based on model checking could be applied starting from the architecture itself. Last but not least, it is worth noticing that the formalism based on Message Sequence Charts, to which both AUML and AML are inspired, has become a standard proposed by the International Telecommunication Union (ITU) [ITU-TS-Z120]. Recently, the use of such formalism has been taken into consideration for the verification of properties [GMP2004], though with not fully positive outcomes.

Formal techniques to describe and reasoning about complex systems can also be applied to the domain of multimedia presentations. A multimedia presentation can be characterized in terms of the set of media items which have to present to the user, the set of the virtual devices used to reproduce media components, the set of events which might cause some change in the presentation (e.g. the termination of a media), the set of synchronization rules, and the set of the spatial composition rules. In [Be03,Be04] an abstract formal model has been introduced to describe the behavior of a multimedia presentation in terms of resources allocation and synchronization among the media objects. Each object is treated as an independent entity with its own behavior and resources allocation. The key idea is to model a single media by a finite state machine, that encodes a reactive behavior. The overall presentation is modelled as parallel (or sequential) composition of single media item executions. The proposed model is well suited for reasoning on multimedia documents dynamics, and to prove properties about them. This is a novelty w.r.t other proposals in the literature [Ca98,Po01]. The use of an extension of finite state machines, hypercharts, is proposed in [Pa99] where mechanisms for specifying hypermedia requirements are provided. In [La02] the effects of uncertainty are discussed and a flexible scheduling algorithm is proposed.

One further application domain is supplied by clinical guidelines, that can roughly be defined as frameworks for specifying the "best" clinical procedures and for standardizing them. As such they can be regarded as protocols. In the recent years the medical community has started to recognize that a computer-based treatment of guidelines provides relevant advantages, such as automatic connection to the patient databases and, more interestingly, decision making facilities. In the last decade, this observation has motivated the development of several domain-independent computer systems for guide-lines management [Pe03]. Moreover computer-based tools to manage clinical guidelines are gaining an increasing relevance within the areas of Artificial Intelligence in Medicine and Medical Informatics [TMT01,Ter04].

Among the proposals of protocol specification languages based on logic, it is worth to mention the results achieved in two European projects, Alfebiite (Project IST-1999-10298, [Alfebiite]), which uses deontic logics [vdT03], and SOCS (Project IST-2001-32530, [SOCS]) based, instead, is on computational logics. Both projects have the goal of modeling agents systems. Alfebiite applies formal models of activities (regulated by norms) for the specification, the management and the regulation of interactions between "inhabitants" in the information society. SOCS (Societies Of Computees) is, instead, more focused on modelling open societies of interacting logical agents (Computees), following an approach based on computational logic. For instance, in [AI03a,AI03b] protocols are expressed through social integrity constraints that specify the actions that can be executed (the messages that can be sent) after the occurrence of one or more events. The SOCS project has clearly shown the potentialities and the advantages of using a computational logic approach for the specification and the verification of protocol [GC2004].

2.2.a Riferimenti bibliografici

[ACKM04] G.Alonso,F.Casati,H.Kuno,V.Machiraju. *Web Services*. Springer, 2004

[AI03a] M.Alberti,A.Ciampolini,M.Gavanelli,E.Lamma,P.Mello,P.Torrioni. *Logic based semantics for an agent communication language*. *Int.Work. on Formal Approaches to MAS*, 21-36,2003

[AI03b] M.Alberti,A.Ciampolini,M.Gavanelli,E.Lamma,P.Mello,P.Torrioni. *A Social ACL Semantics by Deontic Constraints*. *CEEMAS 2003, LNAI 2691*, 204-213,2003

[Alo04] G.Alonso. *EDBT Tutorial on Web Services*, 2004

[Alfebiite] *Web site at alfebiite.ee.ic.ac.uk*

- [AMMR03] E.Astesiano, M.Martelli, V.Mascardi, G.Reggio. *A Combination of a Multiview Use-Case Driven Method and Agent-Oriented Techniques*, SEKE'03
- [B05] M.Baldoni, C.Baroglio, A.Martelli, V.Patti, C.Schifanella. *Verifying protocol conformance for logic-based communicating agents*. *Int. Work. CLIMA-V, LNAI*, 2005
- [B03] M.Baldoni, C.Baroglio, L.Giordano, A.Martelli, V.Patti. *Reasoning about communicating agents in the semantic web*. *LNCS 2901*, 84-98, 2003
- [B04] M.Baldoni, L.Giordano, A.Martelli, V.Patti. *Programming Rational Agents in a Modal Action Logic*. *Annals of Mathematics and AI*, 41(2-4):207-257, 2004
- [Bacchus98] F.Bacchus, F.Kabanza. *Planning for temporally extended goals*. *Annals of Mathematics and AI*, 22:5-27, 1998
- [BBFL2001] B.Berard, M.Bidoit, A.Finkel, F.Laroussinie, A.Petit, L.Petrucci, P.Schnoebelen. *Systems and Software Verification. Model-Checking Techniques and Tools*, 2001
- [Bal04] M.Baldoni, C.Baroglio, I.Gungui, A.Martelli, M.Martelli, V.Mascardi, V.Patti, C.Schifanella. *Reasoning about Agents Interaction Protocols inside DCaseLP*, *DALT'04*, 2004
- [BF95] M.Barbuceanu, M.S.Fox. *COOL: a language for describing coordination in multiagent systems*. *ICMAS 1995*
- [BF95] M.Barbuceanu, M.S.Fox. *COOL: a language for describing coordination in multi-agent systems*. *Int. Conf. on Multiagent Systems*, 17-24, 1995
- [BGS98] M.Benerecetti, F.Giunchiglia, L.Serafini. *Model Checking Multiagent Systems*. *J.of Logic and Computation*. 8(3):401-423, 1998
- [BGH2004] J.Billington, G.E.Gallasch, B.Han. *A coloured Petri Net Approach to Protocol Verification*. *ACPN 2003, LNCS 3098*, 210-290, 2004
- [BMO2001] B.Bauer, J.P.Muller, J.Odell. *Agent UML: A Formalism for Specifying Multiagent Interaction*. *Agent-Oriented Software Engineering*, Springer, 91-103, 2001
- [BPELAWS] BPELAWS, www-106.ibm.com/developerworks/library/ws-bpel
- [Br03] M.Broy. *Unifying models and engineering theories of composed software systems*. *Models, Algebras and Logics of Engineering Software*, 1-41, 2003
- [CCFL99] R.S.Cost, Y.Chen, T.Finin, Y.Labrou, Y.Peng. *Modeling agent conversation with colored petri nets*. *Autonomous Agents Work. on Conversation Policies*, 1999
- [CDV02] D.Calvanese, G.De Giacomo, M.Y.Vardi. *Reasoning about Actions and Planning in LTL Action Theories*. *KR'02*, 2002
- [CGP2000] E.Clarke, O.Grumberg, D.Peled. *ModelChecking*. MIT Press, 2000
- [CT2004] R.Cervenka, I.Trencansky. *Agent Modeling Language, Language Specification*. *Draft proposal v.0.9, Whitestein Technologies*
- [EMST04] U.Endriss, N.Maudet, F.Sadri, F.Toni. *Logic-based agent communication protocols*. *LNAI 2922*, 91-107, 2004
- [ES03] J.Euzenat, H.Stuckenschmidt. *The family of languages approach to the semantic interoperability*. *Knowledge Transformation for the Semantic Web*, 2003
- [Ev03] A.Evans, G.Maskeri, P.Sammut, J.S.Willans. *Building families of languages for model-driven system development*. *Work. in Software Model Engineering*, 2003
- [GC2004] M.Alberti, F.Chesani, M.Gavanelli, E.Lamma, P.Mello, P.Torrioni. *The SOCS Computational Logic Approach to the Specification and Verification of Agent Societies*. *Global Computing Work.*, LNAI 3267, 2004
- [GMP2004] B.Genest, A.Muscholl, D.Peled. *Message Sequence Charts*. *ACPN 2003, LNCS 3098*, 537-558, 2004
- [GMS00a] L.Giordano, A.Martelli, C.B.Schwind. *Ramification and Causality in a Modal Action Logic*. *J.of Logic and Computation*, 10(5):625-662, 2000
- [GMS00b] L.Giordano, A.Martelli, C.Schwind. *Reasoning about Actions in Dynamic Linear Time Temporal Logic*. *J.of IGPL*, 9(2):289-303, 2001
- [GMS04] L.Giordano, A.Martelli, C.Schwind. *Verifying Communicating Agents by Model Checking in a Temporal Action Logic*. *JELIA'04, LNAI 3229*, 2004
- [Gollmann2003] D.Gollman. *Analysing Security Protocols*. *FASec 2002, LNCS 2629*, 71-80, 2003

- [GP03] F.Guerin,J.Pitt. *Verification and Compliance Testing*. In LNAI 2650, 98-112,2003
- [Guerin02] F.Guerin. *Specifying Agent Communication Languages*. PhD Thesis, Imperial College, London,2002
- [Ha04] D.Harel,B.Rumpe. *Meaningful Modeling: What's the Semantics of "Semantics"?*. IEEE Computer,2004
- [Henriksen99] J.G.Henriksen,P.S.Thiagarajan. *Dynamic Linear Time Temporal Logic*. Annals of Pure and Applied logic, 96(1-3):187-207,1999
- [HK03] M.P.Huget, J.L.Koning. *Interaction Protocol Engineering*. LNAI 2650, 179-193,2003
- [Holzmann1992] G.Holzmann. *Description and Validation of Computer Protocols*. Prentice Hall,1992
- [Hu04] R.Hull. *Web Services Composition*. EDBT Summer School,2004
- [Huget2004] M.P.Huget. *Agent UML Notation for Multiagent System Design*. Internet Computing, IEEE, 8(4):63-71,2004
- [KFD98] J.L.Koning,G.Francois,Y.Demazeau. *Formalization and pre-validation for interaction protocols in multiagent systems*. ECAI 1998
- [KS03] J.Koehler,B.Srivastava. *Web Service Composition: Current Solutions and Open Problems*. Work. on Planning for Web Services, 28-35,2003
- [ITU-TS-Z120] ITU-TS recommendation Z.120,1996
- [Jensen1997] K.Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, v.1., Monographs in Theoretical Computer Science*,1997
- [Mallya04] A.U.Mallya,M.P.Singh. *A Semantic Approach for Designing E-Business Protocols*. Int. Conf. on Web Services, 742-745,2004
- [MM02] A.Maedche,B.Motik, N.Silva, R.Volz. *MAFRA-A MAPPING FRAMework for Distributed Ontologies*. LNCS 2473,2002
- [MM03] D.J.Mandell,S.A.McIlraith. *Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation*. Int. Semantic Web Conf.,2003
- [MS01] S.McIlraith,T.Son. *Adapting Golog for Programming the Semantic Web*. Int.Symp. on Logical Formalization of Commonsense Reasoning, 195-202,2001
- [OPB00] J.Odell,H.Parunak,B.Bauer. *Extending UML for agents*. In Agent-Oriented Information System Work.,2000
- [OWLS] OWL-S, www.daml.org/services/owl-s/1.1/,2004
- [Pe03] M.Peleg,S.W.Tu,J.Bury,P.Ciccarese,J.Fox,R.A.Greenes,R.Hall, P.D.Johnson,N.Jones,A.Kumar,S.Miksch,S.Quaglini,A.Seyfang,E.H.Shortliffe,M.Stefanelli. *Comparing computer-interpretable guideline models: a case-study approach*. JAMIA 10:52-68,2003
- [PL03] W.Penczek,A.Lomuscio. *Verifying Epistemic Properties of Multi-agent Systems via Bounded Model Checking*. Fundamenta Informaticae, 55(2):167-185,2003
- [Pistore01] M.Pistore,P.Traverso. *Planning as Model Checking for Extended Goals in Nondeterministic Domains*. IJCAI'01, 479-484,2001
- [QVT] QVT-Partners. *MOF Query/Views/Transformations*. OMG Document: ad/2003-08-08
- [RFC793] J.Postel. *Transmission Control Protocol, STD 7, RFC 793, IETF,1981*
- [SG00] C.Schlenoff,M.Gruninger,F.Tissot,J.Valois,J.Lubell,J.Lee. *The process specification language (PSL): Overview and v. 1.0 specification*. 2000. At www.mel.nist.gov/psl/pubs.html
- [SOCS] Web site at www-lia.deis.unibo.it/Research/Projects/SOCS
- [SPIN] netlib.bell-labs.com/netlib/spin/index.html
- [Ter04] P.Terenziani,S.Montani,A.Bottrighi,M.Torchio,G.Molino, G.Correndo. *The GLARE approach to Clinical Guidelines: main features, Computer-based Support for Clinical guidelines and Protocols*. Symposium on Computerized Guidelines and Protocols, 162-166,2004
- [TMT01] P.Terenziani,G.Molino,M.Torchio. *A Modular Approach for Representing and Executing Clinical Guidelines*, AI in Medicine 23, 249-276,2001
- [UML2] UML 2.0 specification. At www.omg.org

[WSFM04] M.Bravetti,G.Zavattaro. *Int.Work. on Web Services and Formal Methods*, ENTCS 105,2004

[Wooldridge02b] M.Wooldridge,M.Fisher,M.P.Huget,S.Parsons. *Model Checking Multi-Agent Systems with MABLE*. AAMAS'02, 952-959,2002

[YS02] P.Yolum,M.P.Singh. *Flexible Protocol Specification and Execution: Applying Event Calculus Planning using Commitments*. AAMAS'02, 527-534,2002

2.3 Descrizione del Programma di Ricerca e del ruolo delle Unità operative locali

Descrizione del Programma di Ricerca

Testo italiano

Il progetto ha l'obiettivo di sviluppare strumenti e metodologie per la specifica e la verifica di proprietà di protocolli di interazione in sistemi multiagente. Rispetto all'approccio più tradizionale della programmazione a componenti, il paradigma ad agenti esplicita aspetti sociali, connessi alla caratteristica degli agenti di essere autonomi e proattivi: le componenti comunicano e si coordinano in maniera dinamica, utilizzando linguaggi ad alto livello, per perseguire scopi comuni (o propri). Tra gli aspetti sociali è particolarmente importante definire "regole comportamentali", espresse mediante protocolli di comunicazione (interazione), finalizzate a controllare l'organizzazione del sistema. I protocolli di comunicazione possono essere utilizzati per regolare l'interazione fra gli agenti specificando un insieme di conversazioni accettabili, ossia sequenze di operazioni (in particolare scambi di messaggi) che si possono svolgere fra agenti diversi.

L'obiettivo del progetto è organizzato nei seguenti 4 task:

TASK 1. Definizione di formalismi per la specifica e la verifica di protocolli di interazione

TASK 2. Verifica automatica delle proprietà

TASK 3. Linguaggi di modellazione e loro traduzione nei linguaggi formali definiti nel task 1

TASK 4. Sperimentazione in domini applicativi

Le attività che verranno svolte nei diversi task sono descritte nel seguito.

TASK 1. Definizione di formalismi per la specifica e la verifica di protocolli di interazione

Queste attività si svolgeranno prevalentemente nel primo anno.

Si intende affrontare il problema della verifica di proprietà di protocolli di interazione da un punto di vista astratto e generale, in modo indipendente sia dal linguaggio di implementazione delle politiche di conversazione sia dal linguaggio di specifica del protocollo astratto. Tra i problemi di verifica si possono citare: la verifica (statica) delle proprietà di protocolli, la verifica della conformità di un agente con un protocollo di interazione, la verifica (runtime) che l'agente stia (o non stia) rispettando i suoi vincoli sociali, l'interoperabilità di un insieme di agenti, la verifica che un protocollo ottenuto per composizione conservi le proprietà dei protocolli che lo compongono.

In questo task verranno innanzitutto analizzate le proprietà riguardanti protocolli di interazione che si intendono verificare, e successivamente verranno proposti formalismi di specifica di tali proprietà che ne permettano la verifica automatica.

Un primo approccio sarà basato sulla definizione di un linguaggio logico-computazionale. Tra le proprietà di interesse per un protocollo di interazione, distingueremo tra proprietà generali (per esempio, terminazione del protocollo, buona definizione del protocollo, assenza di inconsistenze, etc) e proprietà specifiche (per esempio, per il protocollo transazioni commerciali elettroniche NetBill, la proprietà di "good atomicity" è una proprietà specifica).

Si intende inoltre sviluppare un framework logico basato su una formalizzazione in logica temporale dei protocolli che servirà a supportare la definizione di nuovi protocolli tramite la composizione di protocolli esistenti. Il punto di partenza per la specifica e la verifica di protocolli di interazione è la teoria del ragionamento su azioni sviluppata in precedenza che fa uso della logica temporale DLT (Dynamic Linear Time Temporal Logic), logica che estende LTL rafforzando l'operatore "until" indicizzandolo con programmi regolari. Le azioni comunicative sono descritte in termini dei loro effetti e precondizioni. Questa formalizzazione consente di specificare protocolli di interazione per vincolare il comportamento di agenti autonomi e di studiarne le proprietà. Lo scopo di questa ricerca è di estendere questo approccio per trattare problemi di composizione di protocolli, in modo da consentire la creazione automatica di nuovi protocolli facendo interagire protocolli esistenti secondo pattern che possono essere specificati da un insieme di vincoli temporali. La specifica di schemi di composizione richiederà di adottare la versione Prodotto di DLT, che permette di modellare interazioni concorrenti fra agenti.

Oltre ai linguaggi logici citati sopra, intendiamo studiare l'utilizzo di altri linguaggi logici sviluppati in precedenza e già usati con successo in ambiti analoghi, come DyLOG che si basa su una teoria delle azioni modale, e anche di formalismi non logici quali i linguaggi formali e automi.

TASK 2. Verifica automatica delle proprietà

La definizione formale dei protocolli di interazione e delle loro proprietà apre la possibilità di verificare automaticamente tali proprietà per il protocollo in esame. Esistono varie tecniche al riguardo, in funzione della modalità utilizzata per esprimere il protocollo e le sue proprietà. Nell'ambito del progetto, nel corso del primo anno, si indagheranno tecniche basate su risoluzione estesa, per la verifica automatica delle proprietà di un protocollo. A tale scopo si definirà formalmente una proof procedure in

grado di dimostrare proprietà (espresse formalmente in logica) per un protocollo di interazione (ancora espresso in logica computazionale) e di produrre contro-esempi, alla model checking, nel caso in cui la proprietà sia confutata. Poiché la correttezza (soundness e completezza) di tale proof procedure ha influenza sulla effettiva applicabilità nella verifica di proprietà, tale correttezza sarà oggetto di indagine nel corso del progetto.

Nel secondo anno verrà realizzato un prototipo della proof procedure per la verifica automatica di proprietà. In particolare, tale prototipo sarà sviluppato utilizzando linguaggi simbolici quali la libreria delle Constraint Handling Rules e confluirà in un prototipo complessivo, sviluppato da più Unità. Tale prototipo complessivo sarà corredato di una parte che realizza la traduzione automatica di protocolli da formalismo grafico AAML-like a formalismo logico, descritta nel task successivo. Il prototipo complessivo sarà utilizzato nel Task 4.

La verifica di proprietà temporali dei protocolli richiede la messa a punto di tecniche di prova per la versione prodotta di DLTL. Ci proponiamo di sviluppare tecniche di model checking e, in particolare, di definire una costruzione on-the-fly per costruire automi di Büchi a partire da formule della logica.

Verranno anche sviluppate tecniche di prototipazione rapida per la validazione di protocolli di interazione.

TASK 3. Linguaggi di modellazione e loro traduzione nei linguaggi formali definiti nel Task 1

Nel corso del primo anno verranno analizzati i principali linguaggi di modellazione con particolare riferimento a UML 2.0 o più specifici quali (orientati agli agenti ovvero alle interazioni) AAML, GRL, (orientati ai processi) PSL, BPMN, per la specifica di sistemi reattivi e distribuiti, con l'obiettivo di trasformare i linguaggi di modellazione in linguaggi con una semantica formalizzata, eventualmente in funzione del contesto, in modo da sfruttare i meccanismi di prova automatica a questi ultimi associati. In particolare si studierà un meccanismo di traduzione automatica di protocolli da un linguaggio grafico nel linguaggio formale basato sulla logica risultante dal Task 1.

I risultati ottenuti nel primo anno verranno integrati nel secondo anno in un prototipo mediante il quale sarà possibile, tramite un'opportuna interfaccia, specificare ogni protocollo in modo grafico ed ottenerne automaticamente la traduzione nel linguaggio logico adottato; il prototipo, eseguendo le procedure di dimostrazione per la verifica automatica di proprietà definite nell'ambito del progetto, sarà in grado di dimostrare (o meno) il soddisfacimento di una proprietà da parte del protocollo dato.

Verrà anche affrontato il problema della validazione tramite prototipazione rapida di specifiche AAML (o BPEL) di protocolli di interazione tra entità autonome che "vivono" nel Web (ad esempio, peer in una rete P2P). Si prevede di utilizzare una rappresentazione testuale dei diagrammi AAML (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 AAML - e da far eseguire in un ambiente di prototipazione rapida sviluppato nel Task 2.

L'ultima fase dell'attività di ricerca riguarderà la verifica sperimentale dei risultati ottenuti in diversi ambiti applicativi. In particolare, gli obiettivi saranno sia la validazione sperimentale dell'approccio adottato nella specifica e nella verifica dei protocolli, sia la verifica del corretto funzionamento del prototipo realizzato nei punti precedenti.

TASK 4 La sperimentazione negli ambiti applicativi individuati sarà svolta prevalentemente nel secondo anno. Gli ambiti applicativi considerati saranno:

4.1 Linee guida cliniche

Nelle linee guida cliniche il tempo ha un ruolo fondamentale: ogni azione deve essere eseguita secondo schemi temporali predefiniti. Quindi, i vincoli temporali tra le azioni sono una parte essenziale delle linee guida cliniche. Gestire questi vincoli è molto difficile, poiché non ci sono solo vincoli sulla durata e precedenza ma anche vincoli sulle ripetizioni e periodicità. Questo caso di studio vuole analizzare l'applicabilità alle linee guida cliniche dei formalismi per la specifica dei protocolli sviluppati nel Task 1.

4.2 Servizi web

Diagnosi di servizi web

Data la specifica di un servizio web in un formalismo logico definito nel Task 1, si intende utilizzare il ragionamento diagnostico basato su modelli per diagnosticare possibili malfunzionamenti del servizio stesso che emergono dalla discrepanza tra il comportamento reale e quello atteso. Un punto di partenza è costituito da una proposta che prevede di modellare i servizi attraverso una forma estesa di Rete di Petri, derivando da tale modello quello che correla il funzionamento corretto o scorretto di parti dei servizi alla correttezza dei valori scambiati nei messaggi tra servizi.

Progettazione di web services cooperanti

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 ingegnerizzazione del sistema verranno usate le tecniche descritte nel Task 3 per validare i protocolli di interazione prima della implementazione dei web service.

Selezione e composizione di servizi web

Si intende introdurre un approccio "white box" alla descrizione dei servizi web, in cui le politiche di interazione fra un servizio e i suoi interlocutori sono rese pubbliche, consentendo ad altri servizi (o agenti) di ispezionarle (razionalmente) applicando tecniche di ragionamento, come ad esempio model checking, planning, deduzione. Interpretando i web service come agenti che comunicano attraverso protocolli di interazione pubblici e condivisi, è possibile vedere il problema della selezione del servizio più appropriato (e più in generale quello della composizione di servizi) come un problema di "personalizzazione", cioè un agente può interagire con un certo servizio in modo tale da soddisfare i propri obiettivi?

4.3 Commercio elettronico e sicurezza

La proof procedure realizzata e il prototipo complessivo saranno applicati per verificare o confutare alcune proprietà di protocolli di interazione documentati in letteratura, in particolare nel settore del commercio elettronico. Per esempio, si considereranno il protocollo per commercio elettronico NetBill, il protocollo TCP-IP e protocolli notoriamente "difettosi", come il protocollo di mutua autenticazione denominato Needham-Schroeder. Nella fase sperimentale, si confronterà la tecnica di verifica automatica basata su risoluzione estesa con tecniche simili (per esempio, basate su procedure abduttive) e con algoritmi e tecniche di model checking.

4.4 Presentazioni multimediali

L'obiettivo del primo anno consiste in uno studio di fattibilità allo scopo di decidere se e a qual punto i formalismi descrittivi dello stato dell'arte possano essere applicati al dominio delle presentazioni multimediali. In particolare sarà studiata l'applicabilità di tecniche di model checking al dominio descritto.

Nel secondo anno la descrizione di sistemi complessi sarà arricchita con l'aggiunta di componenti real-time dotate di sensori, con l'esigenza di raccogliere ed elaborare i dati prodotti in real-time. I valori percepiti modificheranno il comportamento della presentazione: sulla base della loro elaborazione saranno scelti certi oggetti multimediali in insiemi di possibili alternative, per esempio caratterizzate da una diversa qualità o un diverso consumo delle risorse. Particolarmente rilevante, in quest'ottica, la questione della modularità del modello proposto: verrà analizzato come si possano combinare diversi sistemi complessi di presentazione per produrne di nuovi, verranno definiti strumenti adeguati per esprimere i requisiti di composizione e le specifiche QoS per la presentazione risultante e verrà analizzato l'impatto della composizione sulla validità delle proprietà, che erano soddisfatte localmente dalle componenti, allo scopo di definire risultati di persistenza.

Complessivamente i risultati attesi della ricerca condotta sono:

- 1) Articoli scientifici in riviste e congressi nazionali e internazionali;
- 2) prototipi per la verifica di proprietà di protocolli di interazione, fra cui, in particolare,
 - un sistema prototipale che, a partire da una rappresentazione grafica del protocollo, sia in grado di tradurre in un formalismo logico e di verificare automaticamente proprietà del protocollo,
 - un ambiente per la prototipazione rapida;
- 3) l'applicazione degli strumenti sviluppati nel progetto a casi di studio nell'ambito delle applicazioni elencate nel Task 4.

Criteria per la valutazione

Verrà realizzato un sito web in cui saranno resi disponibili gli articoli scientifici ed i prototipi realizzati nell'ambito del progetto e in cui verrà descritto lo stato dell'avanzamento dei lavori.

Ogni 6 mesi si terrà un incontro per promuovere la collaborazione fra i diversi gruppi, ed al termine del progetto sarà organizzato un workshop per presentare i risultati del progetto alla comunità scientifica.

Testo inglese

The aim of the project is to develop tools and methodologies for specifying and verifying interaction protocols in multi-agent systems. Compared with the more traditional component based programming approach, the agent based paradigm makes explicit social aspects, related to the fact that agents are autonomous and proactive: the different components dynamically communicate and coordinate with each other, by means of higher level languages, to reach some common (or their own) goal. Among the social aspects, specifically relevant is the ability of defining behavioral rules, expressed by means of communication (interaction) protocols, aiming at controlling the organization of the system. Such protocols can be used to rule the agents' interaction, by specifying a set of acceptable conversations, i.e. sequences of operations (in particular message exchanges) that could occur among different agents.

The project is structured in 4 tasks:

- TASK 1. Definition of formalisms for the specification and verification of interaction protocols
- TASK 2. Automatic property verification
- TASK 3. Modeling languages and their translation into the formal languages defined in task 1
- TASK 4. Case studies

The activities which will be carried out in the various tasks as follows.

TASK 1. Definition of formalisms for the specification and verification of interaction protocols

These activities will be carried out mainly in the first year.

The problem of verifying interaction protocol properties will be faced from an abstract and general point of view, independently of both the implementation language for the conversation policies, and the specification language for the abstract protocol. Among the verification problems we mention: the (static) verification of protocol properties, the verification of the compliance of an agent with an interaction protocol, the (runtime) verification that an agent is (is not) respecting its social facts (commitments and permissions), interoperability of a set of agents, the verification that a protocol obtained by composition preserves the properties of the component protocols.

The aim of this task is to analyze the kind of properties of interaction protocols that have to be verified, and to propose suitable formalisms allowing to specify those properties and to verify them automatically.

A first approach will define a logic language (based on computational logic) for the specification of interaction protocols and of their properties. The relevant properties of an interaction protocol will be classified in general properties (such as protocol termination, well definedness, absence of inconsistencies, etc.) and specific properties, which are peculiar to a given protocol (such as the good atomicity property of the NetBill e-commerce protocol).

This task aims also at developing a logical framework for reasoning about communicating agents and protocols. The logical framework will be based on a temporal logic formalization of protocols and is aimed to support the definition of new protocols through the composition of existing ones. The starting point of our proposal for the specification and verification of interaction protocols is the theory for reasoning about action previously developed, which makes use of the temporal logic DTL (Dynamic Linear Time Temporal Logic), which extends LTL by strengthening the "until" operator by indexing it with the regular programs. The action theory allows to describe communicative actions in terms of their effects and preconditions, to specify interaction protocols to constrain the behavior of autonomous agents and to formulate their properties. We will extend the approach to deal with the problem of protocol composition, which aims at automatically synthesizing new protocols by making a set of protocols interact with each other, according to given interaction patterns which can be specified by a set of temporal constraints. The specification of the composition schema will require to adopt the Product version of DTL, which allows to model the concurrent interaction among agents.

Besides the previously mentioned logical languages, we will also investigate the use of other previously developed languages, such as DyLOG, which is based on a modal theory of actions and has already been used with success in similar contexts, in addition to non-logical formalisms like formal languages and automata.

TASK 2. Automatic property verification

The formal definition of protocol properties makes it possible to automatically verify the properties of a specific protocol. For this purpose, several techniques exist, whose applicability depends on how protocols and their properties are expressed.

In the context of the project, during the first year, techniques based on extended resolution for automatic verification of properties will be investigated. For this purpose, a proof procedure will be formally defined, which will be able to prove properties (expressed in a formal logic language) of an interaction protocol (expressed, in turn, in a formal logic language), and to synthesize counter-examples (similarly to model checking) if the property is refuted. The correctness (soundness and completeness) of such proof procedure will be investigated during the project.

A prototype of the proof procedure for automatic property verification will be implemented during the second year of the project. In particular, such prototype will be developed using symbolic languages such as Constraint Handling Rules and will be integrated in a complete prototype, developed in collaboration among various units. Such complete prototype will be integrated with a tool for automatic translation of protocols from a graphical formalism to the logical formalism to be developed in Task 3. The complete prototype will be applied to the case studies of Task 4.

The verification of properties expressed in temporal logic can be automated by making use of model checking techniques. In particular we aim at defining an on-the-fly construction for building Büchi automata from formulas of the product version of DTL.

A further goal of this task is to develop rapid prototyping techniques for validating interaction protocols.

TASK 3. Modeling languages and their translation into the formal languages defined in Task 1

During the first year, we will analyze the main modeling languages such as UML 2.0, or more specific languages like AUML, GRL (agent oriented, i.e. interaction-oriented), PSL, BNMN (process oriented), state oriented, for the specification of reactive and distributed systems, including in each case relevant extensions. Specifically, we plan to evaluate how modeling languages can be modeled and, consequently, how to integrate different modeling languages.

The aim will be to develop techniques for transforming modeling languages into languages with a formalized semantics, in order to enable the use of the automatic proof mechanisms associated to them. In particular a mechanism of automatic translation of protocols from a graphical language in the formal logic language (turning out from Task 1) will be designed.

The results achieved during the first year will be integrated in the second year in a prototypical system. This prototype will make it possible, through a proper user interface, to specify each protocol in a graphical way and to automatically obtain its translation in the chosen logical language. The system, exploiting the properties specification formalism, through the execution of proof procedures defined within the project, will be able to demonstrate that a given property holds (or does not hold) for a given protocol.

This task will also tackle the problem of validation by rapid prototyping of AUML and BPEL specifications of interaction protocols between autonomous entities "living" in the Web (for instance, peers in a P2P network). More precisely, we want to start from a textual representation of AUML diagrams (as, for instance, XML) and, from this representation, we want to create automatically a "shell" program in Prolog, which is then completed with further knowledge - unspecified in the AUML diagram - and is finally executed in a rapid prototyping environment (see Task 2).

The last research phase will deal with the experimental testing of obtained results. In particular, it will regard both the experimental validation of the chosen approach, and the experimental check of the correct functioning of the prototype implemented in Tasks 2 and 3.

TASK 4. Case studies

In the second year, languages, techniques and tools developed within the previous tasks will be experimented in the following case studies:

4.1 Clinical guidelines

In clinical guidelines, the time plays a fundamental role: each action has usually to be executed according to pre-defined temporal patterns. Thus temporal constraints between actions are an essential part of clinical guidelines. Managing them is a hard job, because there are not only constraints about duration and precedence, but also constraints about repetitions and periodicity. This case study also aims at analyzing the applicability of the formalisms for the specification of protocols developed in Task 1 to clinical guidelines.

4.2 Web services

Web services diagnosis

Given the specification of a web service in the logical formalism of Task 1, we intend to use model-based diagnostic reasoning to diagnose possible malfunctions in the service itself, which could arise from discrepancies between actual behavior and predicted behavior of the service. A starting point is provided by a proposal in which services are modeled in an extended form of Petri Net, from which a further model is derived which relates correct and incorrect behavior of service components to correctness of values exchanged in messages between services.

Cooperating web services design

The goal of this case study is the design and development of a system of cooperating web services, where the cooperation among the services is not only at the level of message exchanges but also at the level of procedural knowledge exchange (through the exchange of BPEL specifications). More precisely, in the engineering phase we will make use of the techniques developed in Task 2 for the validation of interaction protocols before the implementation of the web service.

Web service selection and composition

We plan to introduce a "white box" approach to the description of web services, in which the interaction policies between a service and its interlocutors are made public, thus allowing the other services (or agents) to (rationally) inspect them, by applying reasoning techniques, like, e.g. model checking, planning, deduction. By interpreting web services as agents that communicate through public and shared interaction protocols, the problem of selecting the most appropriate service (and, in general, the problem of the web service composition) can be seen as a "personalization" problem: can an agent interact with a given service so that its goals are achieved?

4.3 Electronic commerce and security

The implemented proof procedure and the complete prototype will be applied to prove or refuse properties of interaction protocols documented in the literature, in particular in the fields of electronic commerce. For instance, the investigation will involve the NetBill protocol, the TCP-IP protocol, a notoriously "faulted" protocol such as the Needham-Schroeder mutual authentication protocol. During the experimental phase, we will compare the automatic verification technique based on extended resolution with related techniques, such as those based on abductive proof procedures and on model checking algorithms and techniques.

4.4 Multimedia presentations

The goal of the first year will be a feasibility study for deciding whether and to which extent the state-of-the-art description formalisms can be applied to the domain of multimedia presentations. In particular, we will study the applicability of model checking techniques to the described domain.

In the second year the description of the complex system will be enriched with the addition of real-time components, in a sensor equipped domain. Real-time produced data will be collected and operated on. Sensed data values will affect the behavior of the presentation; on the basis of the result of some computation on these data, media to be delivered will be chosen among a bunch of candidates, maybe with different quality/resource consumption. Specifically relevant in this perspective will be the study of modularity issues related to our model. We will analyze how different complex presentation systems can be combined to produce a new presentation. With this goal, we will define appropriate composition languages/tools, to express the composition requirements, and the QoS specifications for the resulting presentation. Moreover, we will analyze the impact of the composition on the validity of those properties that were locally satisfied in the component modules, to define persistency results.

To summarize, the expected results of the research activities of the project will be:

- 1) Scientific papers in international journals and conference proceedings, describing the achievements of the project;
- 2) Prototypes for the verification of interaction protocols, among which, in particular,
 - a prototypical system that, starting from the graphical representation of a protocol, will be able to translate it in a logical formalism, and to perform the automatic verification of properties,
 - an environment for rapid prototyping;
- 3) applications of the results of the project to the case studies of Task 4.

Criteria for the assessment of the project

The results of the project will be made available in a web site, which will provide a description of the research activities going on in the project, as well as the papers and prototypes developed within the project.

Every 6 months we will organize a meeting to foster collaboration among the participating groups, and at the end of the second year we will organize a workshop open to the scientific community, to present the results of the project.

Descrizione del ruolo delle Unità operative locali

Testo italiano

Unità I L'esperienza dell'unità di Torino comprende diversi settori: logica computazionale, ragionamento non-monotono, **MARTELLI** agenti intelligenti, sistemi informativi, enterprise modelling, semantic web, (semantic) web services, personalizzazione e presentazioni multi-mediali. Il livello di competenza è testimoniato dalla partecipazione a diversi progetti e iniziative di ricerca a livello nazionale e internazionale. Fra questi, le reti di eccellenza del VI programma quadro **Alberto** REVERSE (Reasoning on the Web with Rules and Semantics, <http://reverse.net>) e INTEROP (Interoperability

Research for Networked Enterprises Applications and Software, <http://www.interop-noe.org>). L'attività all'interno di REWERSE si concentra su specifica, composizione e conformità di politiche e personalizzazione. L'attività in INTEROP si concentra su UEML (Unified Enterprise Modelling Language).

Date le caratteristiche dell'unità l'attività svolta all'interno del progetto si concentrerà sui linguaggi di modellazione e sulla loro traduzione in linguaggi con una semantica formalizzata, sul problema della verifica di proprietà dei protocolli e delle politiche di interazione (es. conformità, interoperabilità), sul matchmaking semantico dei servizi web e sulla loro composizione automatica basata sulla semantica, verifica delle proprietà di presentazioni multimediali con componenti real-time, la cui interazione è specificata tramite linguaggi di modellazione o formali.

L'attività di ricerca che verrà effettuata si articola nei seguenti punti, che riportano l'indicazione del task all'interno del quale si collocano:

[1] studio dei linguaggi di modellazione (con particolare riferimento a UML 2.0 o più specifici) e, in particolare, studio di formalismi di meta-modellazione e di integrazione di linguaggi di modellazione differenti. Trasformazioni dei linguaggi di modellazione in linguaggi con una semantica formalizzata, in modo da sfruttare i meccanismi di prova automatica a questi ultimi associati (TASK 3).

[2]: verifica di proprietà di protocolli di interazione, con particolare riferimento a verifiche statiche di proprietà di protocolli, di conformità, di compliance, di interoperabilità. A questo fine intendiamo studiare una varietà di approcci alternativi, fra cui la logica dinamica lineare temporale (DLTL) e sue estensioni (in collaborazione con l'unità del Piemonte Orientale), linguaggi basati su una teoria delle azioni modale, e anche formalismi non logici quali i linguaggi formali e automi (TASK 2).

[3]: matchmaking semantico per la selezione e la composizione di servizi web. Si intende introdurre una descrizione dichiarativa delle politiche di interazione dei servizi per consentire un'ispezione razionale tramite tecniche di ragionamento. Questa è una novità anche rispetto a linguaggi di markup derivanti dall'area di Semantic Web, come OWL-S, che al momento non consentono di descrivere a livello alto le politiche di interazione (TASK 4).

[4]: sfruttando le competenze specifiche dell'unità nell'ambito delle presentazioni multimediali, testimoniata da una collaborazione di lungo corso con l'università dell'Arizona e quella del Maryland, intendiamo portare le tecniche e gli approcci oggetto di questo progetto in questo innovativo dominio di applicazione. Verranno studiati formalismi di rappresentazione di sistemi complessi di presentazione, che ne consentano la composizione e la customizzazione dinamica (in real-time) (TASK 4).

Unità II
GIORDANO
Laura
L'unità dell'Università of Piemonte Orientale (UPO) include ricercatori con esperienza nel ragionamento temporale, nella logica computazionale e nella logica modale, ma anche esperti nell'area delle linee guida mediche, della diagnosi e dei servizi web. Per questa ragione l'attività dell'unità si orienterà, da un lato, alla definizione di formalismi per la specifica e verifica dei protocolli e, dall'altro, all'uso di questi formalismi nello sviluppo di applicazioni nell'area delle linee guida e dei servizi web.

Più precisamente l'attività dell'unità nel progetto si colloca nell'ambito dei seguenti task:

Task 1. Definizione di formalismi per la specifica e la verifica di protocolli di interazione
L'unità di ricerca intende sviluppare un framework logico per ragionare su agenti che comunicano e protocolli di interazione. Il framework logico sarà basato su una formalizzazione in logica temporale dei protocolli e servirà a supportare la definizione di nuovi protocolli tramite la composizione di protocolli esistenti. In particolare si farà uso di una logica dinamica temporale, DLTL (Dynamic Linear Time Temporal Logic), che è una estensione di LTL, e della sua versione prodotta. Su questo task ci sarà una collaborazione con l'unità di Torino.

TASK 2. Verifica automatica delle proprietà
Tra i problemi di verifica che verranno trattati includono: la verifica (statica) delle proprietà di protocolli e la verifica della conformità di un agente con un protocollo di interazione. Questi problemi di verifica possono essere automatizzati sviluppando tecniche di model checking per la logica temporale usata nella specifica dei protocolli. Inoltre, si intendono sviluppare tecniche di prototipazione rapida per la validazione dei protocolli.

TASK 3. Linguaggi di modellazione e loro traduzione nei linguaggi formali definiti nel task 1
Si vuole definire un mapping fra i linguaggi di modellazione AUML o BPEL4WS ed i formalismi temporali utilizzati nella specifica di protocolli sviluppati nel task 1. Affronteremo inoltre il problema della validazione tramite prototipazione rapida di specifiche AUML e BPEL di protocolli di interazione.

TASK 4. verranno affrontati I seguenti casi di studio:

4.1 Linee guida mediche:

Vincoli temporali saranno usati nella specifica di linee guida cliniche, in particolare nell'ambito del sistema GLARE, sviluppato presso l'Università del Piemonte Orientale, in collaborazione con l'Università di Torino e l'Azienda Ospedaliera S. Giovanni Battista in Torino. Si vuole analizzare l'applicabilità alle linee guida mediche dei formalismi per la specifica dei protocolli sviluppati nel task 1.1. Su questo caso di studio l'unità UPO collaborerà con l'unità di Bologna.

4.2 Servizi Web:

Diagnosi di servizi web

Data la specifica di un servizio web nel formalismo logico descritto al punto 1.1, si intende utilizzare il ragionamento

diagnostico basato su modelli per diagnosticare possibili malfunzionamenti del servizio stesso che emergono dalla discrepanza tra il comportamento reale e quello atteso.

Progettazione di web services cooperanti

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).

L'unità UPO intende inoltre collaborare con l'unità di Torino al caso di studio relativo alla specifica e modellazione di presentazioni multimediali (task 4.4).

**Unità III
LAMMA
Evelina**

L'Unità di Ferrara conduce attività di ricerca nell'ambito della logica computazionale, partecipando attivamente a progetti di ricerca nazionali ed europei. In particolare, il progetto europeo UE IST-2001-32530 SOCS incentrato sulle Società di agenti aperte ed eterogenee, nel quale si è affrontato il tema dei protocolli di interazione tra agenti mediante un approccio basato sulla logica computazionale, e il progetto nazionale progetto COFIN2003 "La gestione e la negoziazione automatica dei diritti sulle opere dell'ingegno digitali: aspetti giuridici e informatici", nel quale si è affrontato il tema della modellazione di protocolli di negoziazione su opere digitali. Sfruttando quindi l'esperienza acquisita nel passato, l'attività dell'Unità di Ferrara sarà rivolta principalmente alla specifica formale e alla verifica automatica i proprietà dei protocolli di interazione tra agenti mediante l'impiego della logica.

In particolare, una volta definite le caratteristiche di un linguaggio per la specifica di proprietà dei protocolli, si vuole esplorare la possibilità di verificare automaticamente tali proprietà applicando procedure automatiche di dimostrazione basate su risolutori logici allo scopo di verificare automaticamente particolari proprietà. A questo proposito, l'Unità di Ferrara potrà collaborare con le altre Unità di Ricerca nella definizione di particolari formalismi per i protocolli e le relative procedure di dimostrazione.

Per verificare sperimentalmente i risultati della ricerca, durante il secondo anno si svilupperà un prototipo della procedura di dimostrazione che consenta la verifica automatica di proprietà; ciò verrà reso possibile dall'integrazione della procedura di dimostrazione nel prototipo complessivo corredato di un modulo di traslazione da formalismo grafico a formalismo logico per i protocolli.

Una volta ultimato il prototipo, si passerà alla fase di verifica sperimentale dei risultati dell'attività di ricerca. In questa fase si metterà alla prova il prototipo ottenuto per specificare e verificare proprietà di protocolli reali appartenenti a diversi ambiti applicativi.

L'attività dell'Unità di Ferrara sarà rivolta principalmente alla specifica formale e alla verifica automatica delle proprietà dei protocolli di interazione tra agenti mediante l'impiego di una logica computazionale. In particolare, l'apporto dell'Unità sarà su:

Definizione di linguaggi formali basati sulla logica computazionale per la specifica di proprietà di protocolli di interazione tra agenti (attività inquadrata nel TASK 1). Studio delle proprietà riguardanti un protocollo di interazione e loro definizione, classificandole, e infine proponendo un linguaggio (basato su logica computazionale) di specifica di tali proprietà che ne permetta la verifica automatica.

Realizzazione di un prototipo per la descrizione e la verifica automatica di protocolli (attività inquadrata nel TASK 2). Definizione di una proof procedure basata su risoluzione estesa in grado di dimostrare proprietà (espresse nel linguaggio definito) per un protocollo di interazione (espresso in un opportuno linguaggio basato su logica) e sua realizzazione prototipale. Tale realizzazione confluirà nel prototipo complessivo, sviluppato congiuntamente alle altre unità del progetto, corredato di una parte che realizza la traslazione automatica di protocolli da formalismo grafico a formalismo logico

Sperimentazione (attività inquadrata nel TASK 4). Verifica sperimentale del prototipo realizzato, per validare sperimentalmente l'approccio adottato e testare il prototipo realizzato. La sperimentazione sarà condotta prendendo in esame alcuni protocolli reali scelti in vari ambiti applicativi come la rappresentazione delle linee-guida cliniche (in collaborazione con l'unità del Piemonte Orientale e l'Unità di Bologna), il commercio elettronico e sicurezza (in collaborazione con l'Unità di Bologna).

**Unità IV
MELLO
Paola**

L'Unità di Bologna è da anni impegnata nell'area della logica computazionale anche attraverso la partecipazione a diversi progetti di ricerca nazionali ed europei. In particolare, il gruppo di ricerca dell'Unità di Bologna attualmente partecipa al progetto europeo SOCS, incentrato sulle Società di agenti aperte ed eterogenee, nel quale si è affrontato il tema dei protocolli di interazione tra agenti mediante un approccio basato sulla logica computazionale. Sfruttando quindi l'esperienza passata, l'attività dell'Unità di Bologna sarà rivolta principalmente alla specifica formale e alla verifica automatica dei protocolli di interazione tra agenti mediante l'impiego della logica.

In particolare, una volta definite le caratteristiche di un linguaggio grafico per la specifica di protocolli, si vuole esplorare la possibilità di tradurre automaticamente protocolli scritti in tale linguaggio grafico in un particolare formalismo logico, le cui caratteristiche saranno anch'esse individuate nell'ambito della ricerca. Ai protocolli espressi in questa forma potranno quindi essere applicate opportune procedure automatiche di dimostrazione basate su risolutori logici allo scopo di verificare automaticamente particolari proprietà. Per verificare sperimentalmente i risultati della ricerca, si svilupperà un prototipo che, a partire dalla specifica di un protocollo nel linguaggio grafico, ne consenta la traslazione automatica nel formalismo logico scelto, e la successiva verifica automatica di proprietà integrando, nel sistema le procedure di dimostrazione automatica definite nell'ambito del progetto.

Una volta sviluppato il prototipo, si metterà alla prova il sistema ottenuto per specificare e verificare protocolli reali appartenenti a diversi ambiti applicativi.

Con riferimento ai Task previsti dal progetto, l'attività di ricerca dell'unità di Bologna sarà quindi articolata nei seguenti punti:

° *Definizione di linguaggi formali basati sulla logica computazionale per la specifica di protocolli di interazione tra agenti (TASK 1). L'attività consisterà nell'individuazione delle caratteristiche di un formalismo logico adatto alla specifica di protocolli di interazione tra agenti e nella sua definizione mediante sintassi e semantica formali.*

° *Traduzione di Linguaggi grafici per la specifica di Protocolli in un formalismo logico(TASK 3). Si definiranno le caratteristiche del linguaggio più adatto alla modellazione dell'interazione tra agenti, eventualmente attraverso l'estensione di linguaggi già esistenti. Successivamente si studierà un meccanismo di traduzione automatica di protocolli dal linguaggio grafico adottato nel linguaggio formale basato sulla logica.*

° *Realizzazione di un prototipo per la descrizione e la verifica automatica di protocolli (TASK 2). I risultati ottenuti verranno integrati in un prototipo mediante il quale sarà possibile specificare ogni protocollo in modo grafico ed ottenerne automaticamente la sua traslazione nel linguaggio logico adottato; il prototipo, eseguendo le procedure di dimostrazione per la verifica automatica di proprietà definite nell'ambito del progetto, sarà in grado di dimostrare (o meno) il soddisfacimento di una proprietà da parte del protocollo dato.*

° *Sperimentazione(TASK 4). L'ultima fase dell'attività di ricerca riguarderà la verifica sperimentale dei risultati ottenuti. In particolare, gli obiettivi saranno sia la validazione sperimentale dell'approccio adottato nella specifica e nella verifica dei protocolli, sia la verifica del corretto funzionamento del prototipo realizzato nel punto 2.1. Tale sperimentazione sarà condotta prendendo in esame alcuni protocolli reali scelti in vari ambiti applicativi come la rappresentazione delle linee-guida cliniche (in collaborazione con l'unità del Piemonte Orientale), il commercio elettronico e sicurezza (in collaborazione con l'Unità di Ferrara), per l'e-learning.*

Testo inglese

Unit I
MARTELLI
Alberto
The expertise of the Torino unit encompasses different fields: computational logic, non-monotonic reasoning, intelligent agents, information systems, enterprise modelling, semantic web, (semantic) web services, personalization and multimedia presentations. The quality of the team is witnessed by its participation to various international projects and research initiatives. Among them, the VI framework networks of excellence REVERSE (Reasoning on the Web with Rules and Semantics, <http://reverse.net>) and INTEROP (Interoperability Research for Networked Enterprises Applications and Software, <http://www.interop-noe.org>). The activity within REVERSE focusses on policy specification, composition and conformance and personalized information systems. The activity within INTEROP focusses on UEML (Unified Enterprise Modelling Language).

Given the characteristics of the unit, the activity that it will carry on within the project will focus on modelling languages and their translation into languages with a formalized semantics, on the problem of verifying properties of interaction protocols and of interaction policies (e.g. Conformance, interoperability), on the semantic matchmaking of web services and on their automatic composition guided by semantics, on the verification of properties of multimedia presentations with real-time components, whose interaction is specified either by modelling or by formal languages.

The research activity that will be carried on is structured in the following points, which report the task (of the project) to which they belong:

[1] study of modelling languages (with particular reference to UML 2.0 or more specific languages) and, in particular, study of formalisms for meta-modellization and for integrating different modelling languages. Transformation of modelling languages into languages with a formal semantics, so to exploit the proof mechanisms associated to the latter(TASK 3).

[2]: verification of properties of interaction protocols, with particular reference to static verifications of protocol properties, of conformance, compliance, interoperability. To this aim various approaches will be investigated, among them dynamic linear temporal logic (DLTL) and its extensions (in collaboration with the group at the university of Piemonte Orientale), languages based on a modal action theory, as well as non-logic formalisms, e.g. Formal languages and automata (TASK 2).

[3]: semantic matchmaking for the selection and composition of web services. We mean to introduce a declarative description of the interaction policies of the web services that allows a rational inspection by means of reasoning techniques. This is a novelty also w.r.t. markup languages from the area of Semantic Web, e.g. OWL-S, that at the moment do not allow to describe at a high level the interaction policies (TASK 4).

[4]: exploiting the specific competence of the unit in the multimedia presentations, witnessed by the long-time cooperation with the University of Arizona and the University of Maryland, we mean to port the techniques and the approaches that are the subject of this project into this novel application domain. We mean to study formalisms for the representation of complex presentation systems, which allow dynamic composition and customization (in real-time) (TASK 4).

Unit II
GIORDANO
Laura
The team at the Università' of Piemonte Orientale (UPO) consists of researchers with expertise in temporal reasoning, in computational logic and nonclassical logic, as well as of experts in medical guidelines, web services and diagnoses. For this reason the activity of the team will focus, on the one hand, on the definition of formalisms for protocol specification and verification and, on the other hand, on the use of these formalisms for developing applications in the areas of clinical guidelines and web services.

More precisely the team will address the following tasks within the project:

Task 1. Definition of formalisms for the specification and verification of interaction protocols

The team focuses on the development of a logical framework for reasoning about communicating agents and protocols. The logical framework will be based on a temporal logic formalization of protocols and is aimed to support the definition of new protocols through the composition of existing ones. In particular, we will make use of a dynamic temporal logic DLTL (Dynamic Linear Time Temporal Logic), which is an extension of LTL, and of its product version. On this task, we will have a collaboration with the team in Torino.

TASK 2. Automatic property verification

The kind of verification task the team will address include: the (static) verification of protocol properties as well as the verification of the compliance of an agent with an interaction protocol. These verification tasks will be automated by developing model checking techniques for the temporal logic used for protocol specification. Besides, we will develop techniques for the validation of interaction protocols by rapid prototyping.

TASK 3. Modeling languages and their translation into the formal languages defined in Task 1

We aim at defining a mapping between modeling languages like AUML o BPEL4WS and the temporal formalisms for the specification of protocols developed in task 1. Also, we address the problem of validation by rapid prototyping of AUML and BPEL specifications of interaction protocols.

TASK 4. The team will develop case studies in the following areas:

4.1 Clinical Guidelines:

Temporal constraints will be used in the specification of clinical guidelines, within the system GLARE, developed at the University of Piemonte Orientale in collaboration with the University of Torino and the Azienda Ospedaliera S. Giovanni Battista in Torino. In this case study, the team aims at analysing the applicability of the formalisms for the specification of protocols developed in task 1.1 to clinical guidelines. On this task, we will have a collaboration with the team in Bologna.

4.2 Web services:

Web services Diagnosis: Given the specification of a web service in the logical formalism of task 1.1, we intend to use model-based diagnostic reasoning to diagnose possible malfunctions in the service itself, which could arise from discrepancies between actual behaviour and predicted behaviour of the service.

Cooperating web services design: The goal of this case study is the design and development of a system of cooperating web services, where the cooperation among the services is not only at the level of message exchanges but also at the level of procedural knowledge exchange (through the exchange of BPEL specifications).

UPO team will also have a collaboration with the team in Torino on task 4.4 concerning the modelling of multimedia presentations.

**Unit III
LAMMA
Evelina**

Ferrara Unit is involved in the area of computational logic, through the participation to several national and European research projects. In particular, it is currently involved in the UE project n. IST-2001-32530 SOCS, aiming at modelling open societies of agents, and the national project COFIN2003 "Management and negotiation of digital rights: legal and informatics aspects" where it is investigating the modelling of negotiation of rights on digital items by computational logic.

Then, taking advantage of the past experience, the activity of the Unit of Ferrara will be mainly centered on the formal specification and on the automatic verification of properties of interaction protocols, through the use of computational logic.

In particular, when a protocol is represented in a logic based formalism, and a property for it as well, then some proof procedure could be applied to it. Ferrara will collaborate with other research units, for the definition of a proper language for protocols. In particular, the Unit will be concentrated upon the formal definition of properties in a logic language and upon the definition of a proper proof procedure in order to automatically verify some particular/general properties of protocols. The proof procedure to be defined will be able to prove properties (expressed in a formal logic language) of an interaction protocol (expressed, in turn, in a formal logic language), and to synthesize counter-examples (similarly to model checking) if the property is refuted. With the aim of experimenting this approach, we intend to develop a prototype of the proof procedure for automatic property verification, and integrate such implementation in the overall prototype, equipped with a graphical tool for the specification of a protocol, and a module for the automatic translation of the graphical specification into chosen logical formalism. The availability of this overall protocol will pave the way to its experimentation and real application.

In particular, along the project, the Unit of Ferrara will contribute to:

° Definition of logic-based formal languages for the specification of properties of agent interaction protocols (see project TASK 1). This task will consist in the definition of a logical formalism suitable to the specification of properties of agent interaction protocols aiming at their formal verification.

° Implementation of a prototype for the description and the automatic verification of properties protocols (see project TASK 2). Formal definition of a proof procedure based on computational logic, in order to automatically prove properties (expressed in the defined formal logic language) of an interaction protocol (expressed, in turn, in a formal logic language defined by Bologna Unit). The Unit will implement the proof procedure by using symbolic languages such as Constraint Handling Rules. This implementation will be integrated in the overall prototype, developed jointly with other units of the project equipped with the tool for automatic translation of protocols from a graphical formalism to the logical formalism defined

° Experimentation (see project TASK 4). Experimental testing of obtained results. In particular, experimental validation of the chosen approach, and testing of the prototype itself. Such experimentation will be lead by

considering some chosen real protocols in three main applicative areas such as the clinical guidelines (in collaboration with the Piemonte Orientale Unit and Bologna Unit), the electronic commerce and security protocols (in collaboration with Bologna Unit).

Unit IV
MELLO
Paola

The Bologna Unit is from many years involved in the area of computational logic, through the participation to several national and European research projects. In particular, the research group of Bologna currently participates to the European project SOCS, which is focused on open societies of heterogeneous agents. Within this project the Bologna Unit has faced the topic of agent interaction protocols, following an approach based on computational logic. Then, taking advantage of the past experience, the activity of the Unit of Bologna will be mainly centered on the formal specification and on the automatic verification of interaction protocols, through the use of computational logic.

In particular, once established the features of a proper graphical protocol specification language, we plan to investigate the way of automatically translate protocols written in the graphical language into a logical formalism, whose features will be also matter of investigation. When a protocol is translated in a logic based formalism, then some proof procedure could be applied to it. In particular, we could apply either existing proof procedures, or new proofs defined within the project, aiming at automatically verifying some particular/general properties. In order to experimentally verify the results achieved, during the second year of the project we intend to develop a prototype that, starting from the graphical specification of a protocol, will allow the automatic translation in the chosen logical formalism, and, subsequently, the automatic verification of properties. Once the prototype has been implemented, we plan to test it by specifying and verifying real protocols taken from various application areas.

The research activity of the Bologna Unit will be then subdivided into the following sub-task:

° Definition of logic-based formal languages for the specification of agent interaction protocols (TASK 1). This task will consist in the definition of a logical formalism suitable to the specification of agent interaction protocols and in its definition by means of both formal syntax and semantics.

° Translation of graphical Languages for protocols specification in a logical formalism (TASK 3). Starting from the study of the state of the art in the field of the graphical languages for protocols description, we will define the features of a language suitable for modeling interactions between agents, possibly through the extension of already existing languages. Subsequently a mechanism of automatic translation of protocols from the adopted graphical language in the formal logic language will be designed.

° Implementation of a prototype for the description and the automatic verification of protocols (TASK 2). The results achieved during the first year will be integrated in a prototypical system. This prototype will make possible, through a proper user interface, to specify each protocol in a graphical way and to automatically obtain its translation in the chosen logical language. The system, exploiting the properties specification formalism defined by the unit of Ferrara, through the execution of proof procedures defined within the project, will be able to demonstrate that a given property holds (or not) for a given protocol.

° Experimentation (TASK 4). The last research phase will regard the experimental testing of obtained results. In particular, the goal will be either the experimental validation of the chosen approach, or the experimental check of the correct functioning of the prototype implemented in task 2.1. Such experimentation will be lead by considering some chosen real protocols in several applicative areas such as the clinical guideline representation (in collaboration with the Piemonte Orientale Unit), the electronic commerce and security protocols (in collaboration with the Unit of Ferrara) and e-learning tools.

PARTE III

3.1 Spese delle Unità di Ricerca

	Unità I	Unità II	Unità III	Unità IV
Grandi Attrezzature	0	0	0	0
Materiale inventariabile	15.000	9.000	9.400	15.000
Materiale di consumo e funzionamento	7.000	2.000	4.000	5.000
Spese per calcolo ed elaborazione dati	0	0	0	0
Personale a contratto	6.000	6.000	8.000	10.000
Servizi esterni	0	0	1.200	0
Missioni	31.000	23.000	12.000	13.000
Pubblicazioni	0	0	300	1.000
Partecipazione / Organizzazione convegni	10.000	6.000	3.900	5.000
Altro	2.000	4.000	1.200	1.000
TOTALE	71.000	50.000	40.000	50.000

3.2 Costo complessivo del Programma di Ricerca

	Unità I	Unità II	Unità III	Unità IV
RD+RA (fondi di Ateneo)	21.300	15.000	12.000	15.000
Cofinanziamento di altre amministrazioni pubbliche, private o fondazioni	0	0	0	0
Cofinanziamento richiesto al MIUR	49.700	35.000	28.000	35.000
Costo totale del programma	71.000	50.000	40.000	50.000

	Euro
Costo complessivo del Programma	211.000
Risorse complessivamente disponibili all'atto della domanda (RD + RA)	63.300
Cofinanziamento di altre amministrazioni pubbliche, private o fondazioni	0
Cofinanziamento richiesto al MIUR	147.700

(per la copia da depositare presso l'Ateneo e per l'assenso alla diffusione via Internet delle informazioni riguardanti i programmi finanziati e la loro elaborazione necessaria alle valutazioni; D. Lgs. 196 del 30.6.2003 sulla "Tutela dei dati personali")

Firma _____

Data 15/04/2005 ore 16:29