

PROGRAMMA DI RICERCA - MODELLO A
Anno 2003 - prot. 2003018831

1.1 Programma di Ricerca di tipo

Interuniversitario

Area scientifico disciplinare Scienze matematiche (70%)

Area scientifico disciplinare Ingegneria industriale e dell'informazione (30%)

1.2 Titolo del Programma di Ricerca

Testo italiano

Sviluppo e verifica di sistemi multiagente basati sulla logica

Testo inglese

Logic-based development and verification of multi-agent systems

1.3 Abstract del Programma di Ricerca

Testo italiano

La diffusione capillare di Internet e World Wide Web ha creato un forte interesse per applicazioni e servizi informatici che ne sfruttino appieno le potenzialità. A questo scopo si richiede alle moderne applicazioni di essere aperte verso gruppi di utenti e componenti software ignoti a priori ed in continuo cambiamento, richiedendo al nuovo software requisiti di adattabilità, scalabilità e sicurezza senza precedenti. Per soddisfare tali requisiti sono stati utilizzati con successo sistemi multiagente, in cui gli agenti sono dotati di autonomia e capacità di ragionamento. Tali agenti, capaci di interagire con altro software o utenti umani per perseguire i propri obiettivi, vengono utilizzati in molti campi portando, per esempio, alla realizzazione di applicazioni mediche, industriali, di telecomunicazioni ed intrattenimento.

Il comportamento dei sistemi multiagente è difficile da prevedere e controllare soprattutto al crescere della complessità delle interazioni. Infatti il comportamento di questi sistemi non è controllato centralmente, ma "emerge" dall'interazione dei loro componenti, che si prevede sempre più "intelligente" e trasparente agli utenti, e coinvolgerà forme di negoziazione, competizione e cooperazione.

Sono pertanto desiderabili sia meccanismi di modellazione e specifica, che permettano di vincolare almeno parzialmente struttura e comportamento globale del sistema (ad esempio restringendo le possibili interazioni senza irrigidire troppo i protocolli) sia strumenti di verifica automatica o semi-automatica dei comportamenti emergenti di sistemi multiagente.

Il progetto si pone quindi i seguenti obiettivi:

- Individuare e sviluppare formalismi logici atti a modellare e specificare il comportamento di agenti, le loro modalità di interazione intelligente ed i comportamenti globali dei sistemi multiagente.

- Definire linguaggi di specifica eseguibili, ottenuti considerando frammenti computazionali delle logiche studiate nel punto precedente.

- Sviluppare tecniche per la verifica di proprietà di sistemi multiagente.

Una scelta naturale è di perseguire questi obiettivi mediante l'uso di logiche non classiche, ad es. epistemiche e temporali, e della logica computazionale. Le logiche temporali sono uno standard nell'analisi dei sistemi distribuiti e con le logiche epistemiche hanno un ruolo rilevante nella modellazione dei sistemi ad agenti (vedasi ad es. il modello BDI). La logica computazionale porta a specifiche eseguibili, semplificando così i compiti di realizzazione e verifica. In particolare, nell'ambito del progetto, verranno utilizzate estensioni della programmazione logica mediante abduzione e ragionamento non-monotono - atti a modellare la conoscenza incompleta che gli agenti hanno sul mondo e l'evoluzione dinamica del sistema - nonché estensioni modali e temporali per modellare il ragionamento su azioni e la dinamica degli stati mentali degli agenti.

Per la verifica automatica, verranno considerate principalmente tecniche di model checking, con l'obiettivo di adattare le tecniche e gli strumenti sviluppati per la verifica di sistemi software distribuiti alla verifica di proprietà di sistemi multiagente. Verranno compiute sperimentazioni in diversi ambiti applicativi.

Testo inglese

The wide diffusion of Internet and of the World Wide Web raised a great interest in applications and software services that fully exploit its potential power. To this aim, modern applications must be open to continuously changing groups of users and software components, possibly not known in advance; they are, therefore, required to be adaptive, scalable, and safe as never before. Multi-agent systems (MAS), made of autonomous and rational agents, have been successfully used to this purpose. Such agents, which can interact with other software systems and/or human users for achieving their own goals, are used in many fields and allow, for instance, the implementation of medical, industrial, telecommunication, and entertainment applications.

It is getting more and more difficult to predict and to control multi-agent system behavior, as their interactions become more and more complex. Actually, a MAS does not have a central control but its behavior emerges from the interaction of its components, interaction that is becoming more and more intelligent and transparent to the users, and will involve forms of negotiation, competition, and cooperation.

Therefore, it would be nice both to have design and specification mechanisms, that allow the global structure and behavior of the system to be constrained at least partially (e.g. to restrict the possible interactions avoiding to make the protocols too much strict), and to have (semi)automatic verification tools for the emerging MAS behavior.

The project has the following targets:

- find and develop logic formalisms for modelling and specifying MAS global behaviour;
- define executable specification languages obtained by considering computational fragments of the logics mentioned above;
- develop techniques for verifying MAS properties.

A straightforward choice is to pursue these goals by means of non classical logics, for instance epistemic and temporal logics, and computational logic. Temporal logics became a standard in the analysis of distributed systems and, together with epistemic logics, they play an important role in modelling agent systems (e.g. see the BDI model). Computational logic allows to write executable specifications, thus simplifying the implementation and verification tasks. In particular, in the project framework, we will use logic programming extensions with abduction and non-monotonic reasoning (suitable to model agent incomplete knowledge of the world and of the dynamic evolution of the system) as well as modal and temporal extensions for modelling reasoning about action and change on the agent mental states.

For automatic verification, we will mainly take into account model-checking techniques, with the aim of adapting the techniques and the tools that were developed for distributed software systems to MAS properties verification.

We will perform an experimentation on various application domains.

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 ; LOGICA COMPUTAZIONALE ; ESTENSIONI DELLA PROGRAMMAZIONE LOGICA ; ABDUZIONE ; LOGICHE MODALI E TEMPORALI ; RAGIONAMENTO SU AZIONI E CAMBIAMENTO ; REVISIONE DELLE CREDENZE ; VERIFICA ; MODEL CHECKING

Testo inglese

MULTI-AGENT SYSTEMS ; COMPUTATIONAL LOGIC ; LOGIC PROGRAMMING EXTENSIONS ; ABDUCTION ; MODAL AND TEMPORAL LOGICS ; REASONING ABOUT ACTIONS AND CHANGE ; BELIEF REVISION ; VERIFICATION ; MODEL CHECKING

1.7 Coordinatore Scientifico del Programma di Ricerca

MARTELLI

ALBERTO

Professore Ordinario

03/06/1944

MRTLRT44H03F952E

INF/01 - Informatica

Università degli Studi di TORINO

Facoltà di SCIENZE MATEMATICHE FISICHE e NATURALI

Dipartimento di INFORMATICA

011/6706731

(Prefisso e telefono)

011/751603

(Numero fax)

mrt@di.unito.it

(Email)

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 Scienze dell'Informazione 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.

E' 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.

Ha svolto ricerca nel campo dell'intelligenza artificiale e della programmazione logica. Negli ultimi anni si è occupato di tecniche di ragionamento su azioni e cambiamento mediante approcci basati sulla logica computazionale. Queste tecniche sono state utilizzate per sviluppare un linguaggio di programmazione per agenti, che consente ad un agente di ragionare sulle proprie conoscenze e di pianificare le proprie azione in base ad obiettivi. Recentemente si è anche occupato dell'uso di tali tecniche per la specifica e la verifica di proprietà di sistemi multiagente.

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 "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 carried out research in artificial intelligence and logic programming. Recently he has studied techniques for reasoning about actions and change, based on computational logics. These techniques have been used for defining an agent programming language, which allows an agent to reason about its beliefs and to make plans to achieve given goals, and, more recently, for specifying and verifying multi-agent systems.

1.9 Pubblicazioni scientifiche più significative del Coordinatore del Programma di Ricerca

- GIORDANO L.; MARTELLI A.; SCHWIND C. (2001). **Reasoning About Actions in a Multiagents Domain** 7th Congress of the Italian Association for Artificial Intelligence, AI*IA'01. vol. LNAI 2175 pp. 237-248
- BALDONI M.; GIORDANO L.; MARTELLI A.; PATTI V. (2001). **Reasoning about Complex Actions with Incomplete Knowledge: A Modal Approach** 7th Italian Conference on Theoretical Computer Science, ICTCS'2001. vol. LNCS 2202 pp. 405-425
- 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)
- 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
- 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)

1.10 Elenco delle Unità di Ricerca

n°	Responsabile Scientifico	Qualifica	Settore Disc.	Università	Dipartimento/Istituto	Mesi Uomo
1.	GIORDANO LAURA	Professore Associato	INF/01	PIEMONTE ORIENTALE	INFORMATICA	14
2.	MARTELLI ALBERTO	Professore Ordinario	INF/01	TORINO	INFORMATICA	12
3.	MELLO PAOLA	Professore Ordinario	ING-INF/05	BOLOGNA	ELETTRONICA, INFORMATICA E SISTEMISTICA-DEIS	12

1.11 Mesi uomo complessivi dedicati al programma

	Numero	Mesi Uomo
Personale universitario dell'Università sede dell'Unità di Ricerca	<i>10</i>	<i>102</i>
Personale universitario di altre Università	<i>2</i>	<i>22</i>
Titolari di assegni di ricerca	<i>1</i>	<i>4</i>
Titolari di borse dottorato e post-dottorato	<i>3</i>	<i>18</i>
Personale a contratto	<i>3</i>	<i>15</i>
Personale extrauniversitario	<i>0</i>	<i>0</i>
TOTALE	19	161

2.1 Obiettivo del Programma di Ricerca

Testo italiano

La diffusione capillare di Internet e World Wide Web e la presenza di componenti software in una moltitudine di oggetti di uso quotidiano richiede alle moderne applicazioni di essere aperte verso insiemi di utenti e componenti software ignoti a priori ed in continuo cambiamento, ponendo sul nuovo software requisiti di adattabilità, scalabilità e autonomia senza precedenti. Il comportamento emergente dei sistemi multiagente è difficile da prevedere e controllare soprattutto al crescere della complessità delle interazioni. Sono desiderabili sia meccanismi di modellazione e specifica, che permettano di vincolare almeno parzialmente struttura e comportamento globale del sistema (ad esempio restringendo le possibili interazioni senza irrigidire troppo i protocolli) sia strumenti di verifica automatica dei comportamenti di sistemi ad agenti. Tra le principali difficoltà tecniche, la necessità di ragionare con informazione incompleta (ad es., nel caso in cui le tipologie di agenti non siano completamente note a priori e la conoscenza dei singoli agenti sul mondo sia imperfetta), di utilizzare un approccio dichiarativo all'interoperabilità di componenti che operano su formati di dati eterogenei (Web Semantico). Il progetto ha l'obiettivo di individuare formalismi, strumenti e tecniche che siano adeguati a modellare, sviluppare e analizzare sistemi ad agenti. Da un lato si vuole modellare la dinamica interna al singolo agente, con le sue credenze, i suoi obiettivi e le sue capacità di ragionamento e pianificazione di azioni. Dall'altro, si vuole modellare la dinamica dell'interazione e della comunicazione fra agenti in un ambiente che evolve dinamicamente e sul quale la conoscenza dei singoli agenti è incompleta. I due livelli sono fra loro strettamente connessi, in quanto le capacità di ragionamento e di pianificazione del singolo agente stanno a supporto delle sue capacità di interazione con gli altri agenti.

Un primo obiettivo del progetto è quindi quello di:

Individuare e sviluppare formalismi logici atti a modellare e specificare il comportamento di agenti, le loro modalità di interazione intelligente ed i comportamenti globali dei sistemi multiagente.

Per la loro espressività e flessibilità, i linguaggi di specifica logici sono particolarmente adatti ad esprimere modelli di comportamento complessi ed articolati come quelli di una società di agenti, come anche a modellarne i sofisticati meccanismi di ragionamento dei singoli agenti. Questi linguaggi di specifica verranno quindi utilizzati per modellare in modo uniforme sia le modalità di interazione tra agenti, facendo riferimento ai modelli e linguaggi esistenti per trattare comunicazione e coordinazione, sia la struttura interna degli agenti stessi, modellando l'evoluzione della conoscenza nei singoli agenti in seguito alle interazioni ed il suo effetto sul comportamento emergente del sistema.

Un aspetto cruciale che verrà preso in considerazione è l'incompletezza della conoscenza degli agenti sull'ambiente, dovuta al fatto che l'ambiente evolve dinamicamente in seguito alle interazioni degli agenti stessi. La trattazione di questo aspetto richiederà l'adozione di tecniche di ragionamento non-monotono.

Modellazione e specifica verranno affrontate da diversi punti di vista, per coprire un'ampia gamma di aspetti applicativi, che vanno dalla ingegnerizzazione e la verifica di sistemi multiagente eterogenei alla risoluzione distribuita di problemi.

Uno dei principali problemi legati all'uso di linguaggi di specifica formali è dato dalla distanza che spesso esiste fra la specifica e la reale implementazione del sistema multiagente. Per cercare di ovviare a questo problema, il progetto ha anche il seguente obiettivo:

Definire linguaggi di specifica eseguibili, ottenuti considerando frammenti computazionali delle logiche studiate nel punto precedente.

Il punto di partenza per raggiungere questo obiettivo è costituito dai linguaggi di programmazione logica, con opportune estensioni modali, temporali e abduttive. L'uso di linguaggi di specifica eseguibili consente la realizzazione rapida di prototipi di sistemi multiagenti, la cui esecuzione può essere utilizzata per la comprensione e la validazione di proprietà relative all'interazione fra le componenti del sistema.

I linguaggi di specifica logici, avendo una semantica formale, permettono di ragionare - anche in modo automatico - sulle specifiche, e quindi di verificare proprietà del sistema. Tuttavia questi linguaggi possono non essere adeguati per ragionare su alcune proprietà dei sistemi, rendendo necessario l'uso di tecniche diverse. In questo progetto ci si concentrerà su tecniche di "model checking". Un ulteriore obiettivo è quindi:

Definire tecniche di model checking per la verifica delle proprietà dei sistemi multiagente

Ci si propone di adattare tecniche di "model checking", già usate con successo per trattare problemi di verifica di processi concorrenti, in modo da prendere in considerazione aspetti legati alla conoscenza, con riferimento ai modelli identificati nel primo obiettivo del progetto.

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

- Siti web adattativi
- Sistemi ad agenti applicati alla medicina
- Sistemi multiagente per il controllo di sistemi complessi in presenza di guasti

Tutti i gruppi proponenti hanno esperienza decennale sulla programmazione logica e le sue estensioni, nonché sull'uso di logiche non classiche (in particolare, logiche modali, temporali e non-monotone) per la rappresentazione della conoscenza. Individualmente, i gruppi hanno maturato forti competenze specifiche su sistemi multiagente.

Si ritiene quindi che la collaborazione fra le unità del progetto possa risultare in un approccio coerente allo sviluppo di sistemi ad agenti ed, in particolare, alla definizione di una collezione di strumenti per lo sviluppo e la verifica di tali sistemi.

Testo inglese

The fine grained diffusion of Internet and the World Wide Web, and the presence of software components in a multiplicity of common objects, requires applications to be open to users and software components that are necessarily unknown a priori, and continuously changing. This puts completely new requirements of adaptability, scalability and autonomy on the software.

As interactions become more complex, the emerging behavior of multi-agent systems becomes hard to predict and control. We need models and specification languages to (partially) constrain the systems' structure and global behavior without making protocols too rigid. We also need automated tools for verifying the behavior of agent systems. Some of the main technical difficulties lie in the incomplete knowledge about the system (e.g., the kind of agents participating to the system may be unknown and their knowledge about the world may be incomplete) and the need for a declarative and abstract approach to interoperability of heterogeneous components (Semantic Web).

This project aims at identifying formalisms, tools and techniques by which agent systems involving knowledge-based interaction can be modeled, developed and analyzed. Such formalisms, tools and techniques should also facilitate the analysis and verification of such systems. The project will focus, on the one hand, on the belief and goal dynamic, internal to the agent, as well as on its reasoning and planning capabilities and, on the other hand, on the dynamic of the interaction among agents, when they communicate in a dynamically evolving world about which the knowledge of agents is incomplete. The two levels are tightly related, as the capabilities of an agent to interact are supported by its reasoning and planning capabilities.

Accordingly, the first goal of the project is:

Identify and develop formal methods for modeling and specifying intelligent interaction and global behavior in multi-agent systems.

In particular, we shall adopt logic-based specification languages. Because of their expressiveness, such languages are suitable to model the complex and articulated behavior of agent societies as well as for modeling the sophisticated reasoning capabilities of the individual agents. These specification languages should model uniformly both agent interactions (for which we will refer to the formalisms and modeling techniques developed to deal with communication and coordination) and the dynamic of each agent's beliefs, to the extent this influences the emerging behavior of the system.

One of the central issues that has to be addressed is agents incomplete knowledge about the world, which evolves dynamically under the effects of agent interactions. This issue will require the use of non-monotonic reasoning techniques.

Modeling and specification will be tackled from different standpoints, to cover a wide range of aspects, from system engineering and verification to distributed problem solving.

One of the main problems related to many formal specification languages is the gap between specifications and the actual implementation of the system. To solve this problem the project includes a second goal:

Define executable specification languages, using computational fragments of the logics mentioned in the previous goal.

The starting point consists of logic programming languages, with suitable extensions based on modal and temporal logics, and on non-monotonic techniques such as abduction. Executable specification languages should enable rapid prototyping of multi-agent systems, as well as simulation and experimental validation of the interactions within the system. The specification languages should also model the architecture of the system (by which we mean the number and role of the agents constituting the system and their patterns of interaction), abstracting away the detailed behavior of each agent.

Logic-based specification languages have formal semantics, which makes it possible to reason formally - and possibly automatically - about the specified system. However such languages may not be suitable for reasoning about complex properties. This calls for the use of different reasoning mechanisms. In particular, in this project, we will focus on "model checking" techniques. So the project has a further goal:

Define model checking techniques for the verification of the properties of multi-agent systems

Our goal is to adapt or extend existing model checking techniques to deal with knowledge-based multi-agent systems, specified with the languages mentioned in the previous points.

The languages, techniques and tools developed in the previous points will be tested on some significant **case studies**:

- Adaptive web sites
- Multi-agent systems for medical applications
- Multi-agent systems for the control of complex systems in presence of faults

All the units of this project have a long standing experience on modal and temporal logics, non-monotonic reasoning, logic programming and extensions thereof. Individual groups have matured specific competence on multi-agent systems. Consequently, the cooperation among the project's units should result in a coherent approach to multi-agent system development. In particular, we expect the project to identify the main features that a collection of tools for the development and verification of such systems should have.

2.2 Base di partenza scientifica nazionale o internazionale

Testo italiano

La teoria degli agenti è fondamentale in Intelligenza Artificiale, soprattutto nella caratterizzazione di sistemi complessi situati in ambienti dinamici [Sh93, RG91, WJ95]. Gli agenti software sono solitamente progettati come entità altamente autonome (ogni agente ha uno stato e può prendere decisioni basandosi su di esso), in grado di reagire a variazioni dell'ambiente (reattività), in grado di perseguire i propri goal (proattività) interagendo se necessario con altri agenti o con altri software (capacità sociali). Uno dei problemi principali della ricerca in questo ambito è la definizione di linguaggi formali per la specifica del singolo agente (che includa la rappresentazione della sua conoscenza e delle sue forme di ragionamento) e per la modellazione della comunicazione ed interazione fra gli agenti.

Molte teorie sugli agenti si basano su formalismi logici [RG91, CL90b, WJ95, CLMST00, CLMT00, GMS01]. In questo contesto si sono dimostrate particolarmente adeguate alcune estensioni della logica classica come, per esempio, logiche modali, dinamiche, temporali, non-monotone, per formalizzare sia il ragionamento sulle azioni sia la dinamica degli stati mentali. Tuttavia a causa delle difficoltà di verificare che le proprietà garantite dai modelli formali siano garantite pure dai sistemi che li implementano, c'è un divario fra il potere espressivo dei modelli proposti e le realizzazioni pratiche. Un modo per superare tale divario è dato dall'approccio della **logica computazionale**, in cui la logica è anche il linguaggio di programmazione e i programmi degli agenti possono essere specificati come regole logiche. Come conseguenza di tali premesse, le estensioni della programmazione logica sembrano essere valide candidate per la realizzazione di linguaggi di specifica degli agenti [GL93, LRLS97, DLL97, BGMP01].

Agenti e comunicazione

Un aspetto importante dell'approccio ad agenti è il principio per cui gli agenti possono funzionare in modo più efficace in gruppi caratterizzati da cooperazione e divisione del lavoro. I programmi degli agenti sono progettati per cooperare autonomamente l'un con l'altro in modo da soddisfare i propri goal interni e le richieste esterne, condivise, generate in seguito alla loro partecipazione alle società di agenti. Le capacità sociali di un agente derivano dall'uso di modelli e linguaggi che consentono la rappresentazione delle interazioni. Gli agenti autonomi possono comunicare, cooperare e negoziare utilizzando linguaggi e protocolli di comunicazione convenzionali. Negli ultimi anni, lo studio della comunicazione e del dialogo [DG00] ha assunto un'importanza crescente nella teoria degli agenti. Buona parte della ricerca si è focalizzata sulla creazione di linguaggi standard di comunicazione [FIPA00] e sulla definizione di semantiche formali degli atti comunicativi, basata sulla teoria filosofica degli atti linguistici [BS97, EBHM00]. Le comunicazioni non sono viste come mere trasmissioni di informazioni ma azioni che modificano lo stato mentale degli agenti coinvolti.

I linguaggi di comunicazione per agenti sono strutture complesse perché un atto comunicativo deve specificare il contenuto del messaggio, l'ontologia, il tipo di atto comunicativo e molti altri aspetti. La semantica degli atti comunicativi individuali è stata definita considerando questi ultimi come azioni e specificandone precondizioni ed effetti [BS97]. Solo recentemente l'attenzione si è spostata sulla formalizzazione delle politiche di conversazione [PM00], sui protocolli di negoziazione e cooperazione [WP00] e sulla conoscenza e le capacità di ragionamento necessarie agli agenti per coordinare le proprie azioni con il dialogo. Un problema interessante è la modellazione delle conversazioni fra agenti, intese come sequenze di atti comunicativi. Le conversazioni possono essere specificate per mezzo di politiche di conversazione.

I linguaggi per la comunicazione fra agenti più diffusi come KQML [KQML] e FipaACL [FIPA00] sono stati formalmente descritti sia a livello sintattico che semantico. La semantica di questi linguaggi è però basata sul modello BDI, mediante la specifica delle precondizioni e degli effetti che ogni atto comunicativo provoca nello stato mentale degli agenti coinvolti nella comunicazione. Questo approccio ha come principale svantaggio la difficile applicabilità in contesti sociali aperti ed eterogenei (nei quali cioè, non è possibile assumere a-priori che l'architettura interna di ogni agente sia basata su un modello mentalistico). Inoltre, anche nell'ipotesi di agenti BDI, la verifica di una semantica espressa in logica modale risulta di difficile realizzazione. Per questi motivi, lavori recenti [Singh00, Colombetti02] definiscono la semantica di linguaggi per la comunicazione tra agenti a livello sociale, esprimendo per ogni atto comunicativo, gli effetti che esso provoca nella società.

Ragionamento su azioni

Un obiettivo della ricerca sui linguaggi di comunicazione è lo sviluppo di una semantica che possa essere integrata facilmente in una teoria generale dell'agente. Una teoria dell'agente è un modello formale che descrive quali azioni un agente debba eseguire in diverse situazioni (si veda, per esempio, il modello BDI [RG91] e la teoria delle intenzioni [CL95, Sh93]). Si tratta di una teoria normativa, solitamente definita in termini dell'interazione fra attitudini mentali (credenze, desideri, intenzioni) e formalizzata per mezzo di logiche modali. Una teoria dell'agente comprende anche tipi differenti di ragionamento, detti talvolta "ragionamento pratico"; fra questi sono inclusi: la teoria delle azioni e della causalità dell'agente, la sua attività di pianificazione, le sue strategie di revisione delle credenze e il suo comportamento comunicativo. Spesso la teoria semantica dei linguaggi di comunicazione dell'agente è definita in termini di attitudini mentali. Siccome gli atti comunicativi possono essere considerati uno speciale tipo di azione, che modifica lo stato mentale degli agenti piuttosto che il mondo fisico, la teoria delle azioni e della causalità può giocare un ruolo nell'interazione fra agenti dal punto di vista sia della specifica sia della verifica.

Fra i vari approcci al ragionamento sulle azioni, uno dei più popolari è ancora il situation calculus, introdotto per catturare il cambiamento nella logica del prim'ordine. Il situation calculus rappresenta il mondo e i suoi cambiamenti tramite una sequenza di situazioni. Ogni situazione rappresenta uno stato del mondo ed è ottenuto da una situazione precedente eseguendo un'azione. In seguito Kowalski ha sviluppato un calcolo diverso per descrivere il cambiamento, detto "event calculus", in cui gli eventi producono dei cambiamenti situati temporalmente ed iniziano e terminano gli effetti delle azioni. La ricerca relativa al situation calculus ha ricevuto nuovo impulso dallo sviluppo di un progetto di "robotica cognitiva" [LLLMRS94].

In questi ultimi anni vari autori [DGL95, BGMP97, GMS99, BGMP01] hanno adottato un diverso approccio al ragionamento sulle azioni, basato sull'uso della logica dinamica e della logica temporale. Da un lato la logica dinamica [Ha84] adotta essenzialmente la stessa ontologia del situation calculus, considerando come primario lo stato del mondo e codificando le azioni come trasformazioni su stati. Le azioni possono essere rappresentate in modo naturale tramite le modalità e gli stati tramite sequenze di modalità. D'altro canto l'adozione di una logica temporale consente la specifica di goal generali, come il conseguimento e il mantenimento di goal, attraverso l'uso di modalità temporali.

La necessità di goal estesi temporalmente è stata motivata in [BK98, KBSD97], dove è stato proposto un approccio alla pianificazione basato su una logica lineare temporale, mentre in [GT99] è stata esplorata l'idea della pianificazione come model-checking in una logica temporale. Altri autori hanno utilizzato il mu-calculus per ragionare sulle azioni [Si98, DGC99, DGR99]. In [GMS00] è stata proposta una teoria dell'azione che si basa sulla logica temporale con tempo lineare. Essa fornisce un modo semplice per vincolare le evoluzioni (potenzialmente infinite) del sistema utilizzando programmi regolari.

Uno dei vantaggi dell'uso di logiche temporali è che essa è uno strumento consolidato per la specifica del comportamento di sistemi distribuiti; per tale strumento è stata sviluppata una ricca teoria ed è stata automatizzata la verifica.

Teoria della Revisione

La teoria della revisione delle credenze affronta il cambiamento nello stato mentale dell'agente come conseguenza dell'introduzione di nuova informazione. Molti aspetti della teoria classica della revisione (basata sui postulati di revisione AGM [AGM85]) sono considerati poco soddisfacenti quando si affronta il caso multiagente, in quanto gli stati mentali degli agenti contengono anche attitudini mentali come goal e intenzioni. Alcune alternative alla teoria AGM per formalizzare il processo di revisione nel caso multiagente sono state esplorate in [HL99, CL90a, CL90b, EBHM98]. In [FH94, Gr98, KS91, GGO00, GGO01] è stato sviluppato uno studio sulle relazioni fra la revisione di credenze e la logica condizionale. La logica condizionale consente una rappresentazione esplicita delle strategie di revisione di un agente, che potrebbe essere sfruttata in un contesto di comunicazione, qualora un agente, che interagisce con altri, volesse ragionare sugli effetti delle proprie azioni attesi sullo stato mentale degli altri agenti.

Logica computazionale per la programmazione di agenti

La modellazione del comportamento locale di un agente e della sua conoscenza del mondo esterno è un compito difficile: in generale è impossibile fare assunzioni sull'unicità della conoscenza (più agenti possono avere visioni differenti del mondo) o sulla completezza della conoscenza (un agente potrebbe avere una conoscenza parziale e incompleta del mondo esterno). E' quindi necessario estendere la logica classica e le tecniche di ragionamento al fine di rendere gli agenti in grado di trattare conoscenza incompleta e lasciarli interagire con gli altri per acquisire le informazioni necessarie alla propria computazione.

Estendendo opportunamente la programmazione logica è possibile scrivere specifiche direttamente eseguibili da un sistema multiagente e facilmente verificabili, modellando in modo efficace sia la conoscenza locale di un agente sia l'interazione globale all'interno del sistema. Vi sono molti lavori che riguardano sistemi di agenti intelligenti basati sulla programmazione logica (si vedano per esempio [Ko95, DP99, SAC99]), in cui tecniche di ragionamento basate sulla programmazione logica sono state sviluppate/estese per trattare conoscenza incompleta. In questo contesto alcuni lavori introducono ed estendono tecniche per il ragionamento ipotetico (es. abduzione) per sistemi multiagente [DST98, CLMS99].

L'approccio della logica computazionale è stato studiato anche nel contesto del ragionamento su azioni. In [BGMP97] è stato definito un linguaggio di programmazione logica (modale), DyLOG, in seguito esteso per trattare azioni complesse e azioni di "sensing" [BGMP01]. Le azioni complesse consentono la definizione di librerie di piani specificati in modo parziale e possono essere definite a partire da quelle atomiche sia utilizzando operatori di composizione (come la composizione sequenziale, la scelta non-deterministica e il test) sia definendo procedure ricorsive. Per via della sua capacità di definire azioni complesse, questo linguaggio è vicino al linguaggio Golog [LRLS97]. DyLOG fa parte di un framework generale di sviluppo di estensioni modali della programmazione logica [BGM98, B98] ed è stato utilizzato in applicazioni pratiche relative alla realizzazione di siti web adattativi [BBCP01].

Verifica del comportamento complessivo del sistema

Mentre molta attenzione è stata posta alla definizione di modelli, linguaggi ed architetture per sistemi multi-agente, non altrettanto è stato fatto per lo studio di strumenti formali per modellare e verificare il comportamento complessivo di sistemi di questo tipo, che è difficile da prevedere e da controllare.

I metodi formali stanno assumendo un'importanza sempre più rilevante nello sviluppo e soprattutto nella verifica dei sistemi software. In questo contesto, la logica temporale gioca un ruolo importante fornendo un linguaggio preciso e espressivo per la specifica dell'evoluzione degli stati di un sistema.

Il model-checking [CGP99] costituisce uno degli strumenti più importanti nel campo dei metodi formali.

Nell'ambito dei sistemi multiagente, un approccio alla progettazione ed alla verifica di tali sistemi è stato proposto in [Wooldridge02]. Gli agenti sono scritti in MABLE, un linguaggio di programmazione per agenti imperativo, ed hanno uno stato mentale che contiene credenze, desideri e intenzioni secondo l'approccio "mentalistico". Il sistema MABLE prevede la possibilità di specificare enunciati formali sul sistema in una logica BDI temporale lineare e con quantificazione. MABLE utilizza il model-checker SPIN per verificare tali enunciati.

Un differente framework per la comunicazione di agenti, basato su di un approccio "sociale" [Singh00] anziché "mentalistico", è stato proposto nella tesi di Guerin [Guerin02]. Esso definisce una semantica "grounded" per la comunicazione fra agenti e prevede linguaggi differenti per la programmazione degli agenti, per la specifica della comunicazione fra agenti e per esprimere proprietà di agenti. Queste ultime vengono espresse in LTL.

Testo inglese

Agent theory is fundamental in Artificial Intelligence, especially for characterizing complex systems situated in dynamic environments [Sh93, RG91, WJ95]. Software agents are usually designed as highly autonomous entities (each agent has a state and uses it to take decisions), able to react to variations in their environments (reactivity), able to achieve their goals (proactiveness) by interacting, if necessary, with other agents (social capabilities). One of the main research problems in this field is the definition of formal languages for specifying single agents (including the representation of their knowledge and their reasoning techniques) and for modeling communication and interaction among agents.

Many theories about agency are based on logic formalisms [RG91, CL90b, WJ95, CLMST00, CLMT00, GMS01]. In this context some extensions of classical logic, e.g. modal, dynamic, temporal logics, and non-monotonic logics, proved to be particularly suitable for formalizing reasoning about action as well as the dynamics of the mental states. However, due to the difficulty of verifying that those properties that are granted by the formal models are granted also by the systems which implement them, there is a gap between the expressive power of the proposed models and the practical implementations. One way of filling such a gap is offered by **computational logic**, in which the logic is also the programming language and agent programs can be written as logic rules. As a consequence, logic programming extensions seem to be a valid candidate for designing agent specification languages [GL93, LRLS97, DLL97, BGMP01].

Agents and communication

One important aspect of the agent approach is the principle by which agents are more effective when they work in groups, subdividing the job and cooperating. Agent programs are designed in order to cooperate autonomously with each other so as to satisfy their inner goals and outer (shared) requests, generated by their participation to the agent society. An agent social

capabilities derive from the use of models and languages that allow them to represent interactions. Autonomous agents can communicate, cooperate and negotiate by means of conventional languages and protocols. Recently, studies about communication and dialogue [DG00] gained a growing importance in agent theory. Most of the research focused on the definition of standard communication languages [FIPA00] and on the definition of formal semantics of communicative acts, based on the philosophical theory of linguistic acts [BS97, EBHM00]. Communication is not seen as a mere information transmission but as the result of actions that modify the mental state of the agents that are involved.

Agent communication languages are complex structures because a communication act must specify the message content, the ontology, the kind of act and many other facets. Individual communication acts semantics has been defined by considering them as actions and by specifying their preconditions and effects [BS97]. Only recently attention moved to formalizing conversation policies [PM00], negotiation and cooperation protocols [WP00], and the knowledge and reasoning capabilities that are necessary to agents in order to coordinate their actions by means of dialogue. An interesting problem is to model agent conversations, considering them as sequences of communicative acts.

For most wide-spread agent communication languages, such as KQML [FLM95] and FipaACL [FIPA00], a formal description has been given both at a syntactic level and at a semantic level. ACL semantics, however, is often based on the BDI model, by means of the specification of the preconditions and the effects that each communication performative produces on the mental state of involved agents. This approach, being based again on the BDI model, suffers the main disadvantage of being hardly applicable in open and heterogeneous agents societies. Under this assumption, in fact, it is not possible to a-priori assume a mentalistic-based model for all agents. Moreover, even if we consider BDI-based agents, the verification of a semantic based on modal logic is very difficult to perform. For these reasons, more recent works [Singh00, Colombetti02] define ACL semantics at a social level, by expressing (for each communicative act) the effects it produces into the agent society.

Reasoning about actions

One aim of research about communication languages is the definition of a semantics that can easily be integrated with a general theory of the agent. A theory of the agent is a formal model that describes which actions an agent must execute in different situations (see, for instance, the BDI model [RG91] and the theory of intention [CL95, Sh93]). It's a normative theory, usually defined in terms of the interaction between mental attitudes (belief, desire, intention) and formalized by means of modal logics. A theory of the agent comprises also other kinds of reasoning, sometimes called "practical reasoning"; among them: the theory of the actions and of causality of the agent, its planning activity, its belief revision strategies and its communicative behavior. Often the communication language semantic theory is defined in terms of mental attitudes. As communicative acts can be considered a special kind of action, that modifies the mental state of the agents rather than the physical world, the theory of action and causality could play a role in the interaction between agents from the point of view both of specification and of verification.

One of the most popular approaches to reasoning about action, is Situation Calculus, introduced in first order logic for capturing change. Situation Calculus represents the world and its changes by means of a sequence of situations. Each situation represents a state of the world and is obtained from a preceding situation after the execution of an action. Kowalski developed a different calculus for describing change, named "event calculus", in which events produce temporally situated changes which begin and end actions effects. Recently, the research about Situation Calculus gained a renewed attention thanks to cognitive robotics research [LLLMRS94].

An approach to reasoning about actions that recently raised attention again is the one based on Dynamic Logic and Temporal Logic. The suitability of the logic model to reason about actions has been observed by many authors [DGL95, BGMP97, GMS99, BGMP01]. On one hand, Dynamic Logic [Ha84] essentially adopts the same ontology of Situation Calculus, considering as primary the state of the world, and coding actions as transformations of the states. Actions can be represented in a natural way by means of modalities. On the other hand, the use of a Temporal Logic allows the specification of general targets, such as the achievement of goals, by means of temporal modalities. The need of temporally extended goals has been motivated in [BK98, KBSD97], where an approach to planning based on a temporal linear logic has been proposed, while in [GT99] the idea of planning as model-checking in a temporal logic has been explored. Other authors have used mu-calculus for reasoning about actions [Si98, DGC99, DGR99]. In [GMS00] a theory about action that is based on a temporal logic with linear time has been proposed. It supplies a simple way for limiting (potentially infinite) system evolutions by using regular programs. An advantage of the use of Temporal Logic is that it is well-known tool for specifying the behavior of distributed systems; for such a tool a rich theory and automatic verification procedures have been developed.

Revision Theory

Belief revision theory considers the agent mental state change as a consequence of the introduction of new knowledge. Many features of the classical revision theory (based on AGM revision postulates [AGM85]) proved to be not very effective in the multi-agent case, since the agent mental states contain also mental attitudes like goal and intention. Some alternatives to AGM theory for formalizing the revision process in the multi-agent case have been explored in [HL99, CL90a, CL90b, EBHM98]. In [FH94, Gr98, KS91, GGO00, GGO01] a study of the relation between belief revision and conditional logic has been developed. Conditional logic allows the explicit representation of an agent revision strategies, which could be exploited in a communication framework, when an agent that interacts with others wants to reason about the desired effects of its own actions on the mental states of the other agents.

Computational logic for agent programming

Modeling an agent local behavior and its knowledge about the external world is a difficult task: generally speaking, it is impossible to take assumptions about the uniqueness of the knowledge (different agents may have different views about the world) or about the completeness of the knowledge (an agent may have incomplete or partial knowledge about the world). It is, therefore, necessary to extend the classical logic and the reasoning techniques so to allow agents to deal with incomplete knowledge and let them interact with the others for acquiring the information that is necessary to their computation.

By properly extending logic programming it would be possible to write specifications that are directly executable by a multi-agent system, and easily verifiable, modeling in an efficient way both an agent local knowledge and the global interaction within the system. There are many works about intelligent agent systems based on logic programming (e.g. Ko95, DP99, SAC99), where reasoning techniques based on logic programming have been developed/extended for dealing with incomplete knowledge. In this context some works introduce and extend hypothetical (e.g. abductive) reasoning techniques for multi-agent systems [DST98, CLMS99].

The computational logic approach has been studied also in the framework of reasoning about action. In [BGMP97] a (modal) logic programming language, DyLOG, has been defined and subsequently extended for tackling complex and sensing actions [BGMP01].

Complex actions allow the definition of partially specified plans and can be defined starting from atomic actions either by using composition operators (such as sequential composition, non-deterministic choice and test) or by defining recursive procedures. Thanks to its capacity of defining complex actions, this language is close to Golog [LRLS97]. DyLOG is a part of a general framework for the development of modal extensions of logic programming [BGM98, B98] and has been used in practical applications, for constructing adaptive web sites [BBCP01].

Verification of the global system behavior

While a lot of attention has been devoted to the definition of multi-agent models, languages and architectures, only a little attention has been devoted to the study of formal tools for modeling and verifying the global behavior of systems of this kind, which is difficult to predict and to control.

Formal methods are assuming a growing relevance in the design and especially the verification of software systems. In this context temporal logic plays an important role by providing a precise and expressive language for the specification of the evolution of the states of a system. Model checking [CGP99] is one of the most important tools in the formal methods field.

An approach to the design and automatic verification of multi-agent systems is presented in [Wooldridge02]. Agents are written in MABLE, an imperative programming language, and have a mental state consisting of beliefs, desires and intentions, according to the "mentalistic" approach [Singh00]. MABLE systems may be augmented by the addition of formal claims about the system, expressed using a quantified, linear temporal BDI logic. MABLE makes use of the SPIN model checker to automatically verify the claims. Guerin's thesis [Guerin02] defines an agent communication framework which is based on the "social" approach to communication rather than on the "mentalistic" approach. It gives agent communication a grounded declarative semantics, and it provides different languages for agent programming, for specifying agent communication and social facts, and for expressing temporal properties. Such temporal properties are expressed in LTL.

2.2.a Riferimenti bibliografici

- [AGM85] C.E. Alchourron, P. Gardenfors, D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *J. of Symbolic Logic*, 50:510-530, 1985
- [BBCP01] M. Baldoni, C. Baroglio, A. Chiarotto, V. Patti. Programming goal-driven web sites using an agent logic language. *LNCS* 1990, p.60-75. Springer, 2001
- [BK98] F. Bacchus, F. Kabanza. Planning for temporally extended goals. *Annals of Math. and AI*, 22:5-27, 1998
- [B98] M. Baldoni. Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension. PhD thesis, Univ. di Torino, 1998
- [BGMP97] M. Baldoni, L. Giordano, A. Martelli, V. Patti. An Abductive Procedure for Reasoning about Actions in Modal Logic Programming. *LNAI* 1216, p.132-150. Springer, 1997
- [BGM98] M. Baldoni, L. Giordano, A. Martelli. A Modal Extension of Logic Programming: Modularity, Beliefs and Hypothetical Reasoning. *J. of Logic and Computation*, 8(5):597-635, 1998
- [BGMP01] M. Baldoni, L. Giordano, A. Martelli, V. Patti. Reasoning about Complex Actions with Incomplete Knowledge: a Modal Approach. *LNCS* 2202, p.405-425. Springer, 2001
- [BS97] P. Bretier, D. Sadek. A rational agent as the kernel of a cooperative spoken dialogue system: implementing a logical theory of interaction. *LNAI* 1193, Springer, 1997
- [CLMS99] A. Ciampolini, E. Lamma, P. Mello, C. Stefanelli. Abductive Coordination for Logic Agents. *Proc. of ACM SAC* 1999, p.134-140, 1999
- [CLMST00] A. Ciampolini, E. Lamma, P. Mello, C. Stefanelli, P. Torroni. An implementation for abductive logic agents. *LNAI* 1792, p.61-71. Springer, 2000
- [CLMT00] A. Ciampolini, E. Lamma, P. Mello, P. Torroni. Expressing Collaboration and Competition among Abductive Logic Agents. *Proc. CLIMA-00*, 2000
- [CL90a] P. Cohen, H. Levesque. Rational interaction as the basis for communication. In *Intentions in Communication*, MIT Press, 1990
- [CL90b] P. Cohen, H. Levesque. Intention is choice with commitment, *Artificial Intelligence*, 42:213-261, 1990
- [CL95] P. R. Cohen, H. J. Levesque. Rational Interaction as the basis for communication. In *Interaction in Communication*, p.221-256. MIT Press, 1995
- [CGP99] E. M. Clarke, O. Grumberg, D. Peled. *Model Checking*. MIT press, 1999
- [Colombetti02] N. Fornara and M. Colombetti, "Operational Specification of a Commitment-Based Agent Communication Language", *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pp 535-542, Bologna, Italy, ACM, July 2002.
- [DGL95] G. De Giacomo, M. Lenzerini. PDL-based framework for reasoning about actions. *LNAI* 992, p.103-114, 1995
- [DGC99] G. De Giacomo, X. J. Chen. Reasoning about nondeterministic and concurrent actions: A process algebra approach. *Artificial Intelligence*, 107:63-98, 1999
- [DGR99] G. De Giacomo, R. Rosati. Minimal knowledge approach to reasoning about actions and sensing. *Proc. of the 3rd Work. on Nonmonotonic Reasoning, Action, and Change*, 1999
- [DLL97] G. De Giacomo, Y. Lesperance, H. J. Levesque. Reasoning about Concurrent Execution, Prioritized Interrupts, and Exogenous Actions in the Situation Calculus. *IJCAI97*, p.1121-1226, 1997
- [DP99] P. Dell'Acqua, L. M. Pereira. Updating Agents. *ICLP Workshop on Multi-Agent Systems in Logic Programming*, 1999
- [DST98] P. Dell'Acqua, F. Sadri, F. Toni. Combining Introspection and Communication with Rationality and Reactivity in Agents. *LNAI* 1489, p.17-32. Springer, 1998
- [DG00] F. Dignum, M. Greaves. Issues in Agent Communication: An Introduction. *LNAI* 1916, p.1-16, 2000
- [EBHM98] R. van Eijk, F. de Boer, W. van der Hoek, J. Meyer. Information-Passing and Belief Revision in Multi-Agent Systems. *LNAI* 1555, 1998
- [EBHM00] R. van Eijk, F. de Boer, W. van der Hoek, J. Meyer. Operational Semantics for agent Communication Languages. *LNAI* 1916, 2000
- [FIPA00] Foundation for Intelligent Agents. *Agent Communication Language*. 2000
- [FH94] N. Friedman, J. Y. Halpern. Conditional Logics of Belief Change. *Proc. of AAAI* 94, p.915-921, 1994
- [FLM95] T. Finin, Y. Labrou, J. Mayfield. KQML as an Agent Communication Language. In *Software Agents*, MIT Press, 1995.
- [GGO00] L. Giordano, V. Gliozzi, N. Olivetti. A Conditional Logic for Iterated Belief Revision. *Proc. of ECAI* 2000, p.28-32, 2000

- [GGO01] L. Giordano, V. Gliozzi, N. Olivetti. Iterated belief revision and conditional logic. *Studia Logica*, 70(1):23-47, 2002
- [GL93] M. Gelfond, V. Lifschitz. Representing Action and Change by Logic Programs. *J. of Logic Programming*, 17:301-321, 1993
- [GMS99] L. Giordano, A. Martelli, C. Schwind. Ramification and Causality in a Modal Action Logic. *J. of Logic and Comp.*, 10(5):625-662, 2000
- [GMS00] L. Giordano, A. Martelli, C. Schwind. Reasoning about Actions in Dynamic Linear Time Temporal Logic. *Proc. of FAPR'00*
- [GMS01] L. Giordano, A. Martelli, C. Schwind. Reasoning about actions in a multi-agent domain. *LNAI 2175*, pp.237-248. Springer, 2001
- [GT99] F. Giunchiglia, P. Traverso. Planning as Model Checking. *Proc. of ECP'99*, p.1-20, 1999
- [Gr98] G. Grahne. Updates and Counterfactuals. *J. of Logic and Computation*, 8(1):87-117, 1998
- [Guerin02] F. Guerin. Specifying Agent Communication Languages. PhD Thesis, Imperial College, London, April 2002.
- [Ha84] D. Harel. First order dynamic logic. In *Extensions of Classical Logic, Handbook of Philosophical Logic II*, p.497-604, 1984
- [HL99] A. Herzig, D. Longin. Belief Dynamics in Cooperative Dialogues. *Proc. of AMSTEOLOGUE 99*, 1999
- [KBSD97] F. Kabanza, M. Barbeau, R. St-Denis. Planning control rules for reactive agents. *Artificial Intelligence*, 95:67-113, 1997
- [KS91] H. Katsuno, K. Satoh. A unified view of consequence relation, belief revision and conditional logic. *Proc. of IJCAI91*, p.406-412.
- [Ko95] R.A. Kowalski. Using meta-logic to reconcile reactive with rational agents. In *Meta-Logic and Logic Programming*, p.227-242. MIT Press, 1995
- [KR00] A.C. Kakas, F. Riguzzi. Learning with abduction. *New Generation Computing*, 18(3), 2000
- [LLLMRS94] Y. Lesperance, H.J. Levesque, F. Lin, D. Marcu, R. Reiter, R.B. Scherl. A logical approach to high-level robot programming. In *Proc. of AAAI Fall Symp.*, 1994
- [LRLS97] H.J. Levesque, R. Reiter, Y. Lesperance, F. Lin, R.B. Scherl. GOLOG: A Logic Programming Language for Dynamic Domains. *J. of Logic Programming*, 31:59-83, 1997
- [PM00] J. Pitt, A. Mamdani. A Protocol-Based Semantics for an Agent Communication Language. *LNAI 1916*. Springer, 2000
- [RG91] A.S. Rao, M.P. Georgeff. Modelling rational agents within a BDI-architecture. In *Proc. KR'91*, p.473-484, 1991
- [Sa93] V.A. Saraswat. *Concurrent Constraint Logic Programming*, MIT Press, 1993
- [SAC99] A. Ciampolini, E. Lamma, P. Mello, C. Stefanelli. Abductive coordination for logic agents. *Proc. of ACM SAC99*, p.134-140, 1999
- [Sh93] Y. Shoham. Agent Oriented Programming. *Artificial Intelligence*, 60(1):51-92, 1993
- [Si98] M.P. Singh. Applying the Mu-Calculus in Planning and Reasoning about Actions. *J. of Logic and Computation*, 1998
- [Singh00] Munindar P. Singh. A social semantics for Agent Communication Languages. In *IJCAI-98 Workshop on Agent Communication Languages*, Springer, Berlin, 2000.
- [Ts93] E.P.K. Tsang. *Foundation of Constraint Satisfaction*, Academic Press, 1993
- [WJ95] M. Wooldridge, N.R. Jennings. Agent Theories, Architectures, and Languages: A survey. *LNAI 890*, p.1-39, 1995
- [WP00] M. Wooldridge, S. Parsons. Languages for Negotiation. *Proc. of ECAI 2000*, 2000
- [Wooldridge02] M. Wooldridge, M. Fisher, M.P. Huget and S. Parsons. Model Checking Multi-Agent Systems with MABLE. In *AAMAS'02*, Bologna, Italy, 2002.

2.3 Numero di fasi del Programma di Ricerca:

2

2.4 Descrizione del Programma di Ricerca

Fase 1

Durata e costo previsto

Durata	Mesi 12	Costo previsto	Euro 75.000
--------	---------	----------------	-------------

Descrizione

Testo italiano

Un agente che opera in un ambiente dinamico ed interagisce con altri agenti deve essere capace di ragionare sui possibili effetti sul mondo delle azioni proprie e degli altri agenti. Deve, inoltre, essere capace di pianificare sequenze di azioni che lo portino a raggiungere gli obiettivi dati, interagendo con gli altri agenti in base a protocolli di comunicazione predefiniti. La sua conoscenza sul mondo, in generale, può essere solo parziale e gli effetti delle sue azioni possono non essere determinabili in modo univoco. L'agente potrà acquisire nuova conoscenza sul mondo mediante delle azioni conoscitive (di sensing) o comunicative, e dovrà rivedere la propria conoscenza sul mondo sulla base di nuove informazioni. Nella sua attività di pianificazione con conoscenza incompleta l'agente dovrà inoltre essere in grado di ragionare in forma ipotetica sugli effetti delle azioni, e di pianificare i tempi di esecuzione delle proprie azioni in un ambiente dinamico.

Il progetto ha l'obiettivo di individuare formalismi, strumenti e tecniche che siano adeguati a modellare, sviluppare e analizzare

sistemi ad agenti. Da un lato si vuole modellare la dinamica interna al singolo agente, con le sue credenze, i suoi obiettivi e le sue capacità di ragionamento e pianificazione di azioni. Dall'altro, si vuole modellare la dinamica dell'interazione e della comunicazione fra agenti in un ambiente che evolve dinamicamente e sul quale la conoscenza dei singoli agenti è incompleta. I due livelli sono fra loro strettamente connessi, in quanto le capacità di ragionamento e di pianificazione del singolo agente stanno a supporto delle sue capacità di interazione con gli altri agenti.

Per raggiungere l'obiettivo del progetto si sfrutterà la lunga esperienza di tutti i partecipanti nell'ambito delle logiche epistemiche e temporali, del ragionamento automatico, della programmazione logica e delle sue estensioni. Ci aspettiamo che una stretta collaborazione tra questi gruppi permetta di raggiungere gli obiettivi del progetto.

Durante la prima fase del progetto ci sarà una stretta interazione tra i gruppi partecipanti, in modo che ogni gruppo possa approfondire la propria conoscenza sull'attività di ricerca degli altri gruppi e individuare le principali caratteristiche, i punti di forza e i limiti dei vari approcci. Verranno considerati inoltre diversi ambiti applicativi, ritenuti particolarmente significativi, per sperimentare le tecniche e gli strumenti sviluppati nel progetto.

Le principali attività di ricerca del progetto possono essere riassunte con i seguenti obiettivi:

1. Modellazione di sistemi ad agenti
 - 1.1 Modelli formali di sistemi ad agenti
 - 1.2 Linguaggi di specifica basati sulla logica computazionale
2. Validazione e verifica
3. Applicazioni

1. Modellazione di sistemi ad agenti

1.1 Modelli formali di sistemi ad agenti

Per la loro espressività e flessibilità, i linguaggi di specifica logici sono particolarmente adatti ad esprimere modelli di comportamento complessi ed articolati come quelli di una società di agenti, come anche a modellarne i sofisticati meccanismi di ragionamento dei singoli agenti. Questi linguaggi di specifica verranno quindi utilizzati per modellare in modo uniforme sia le modalità di interazione tra agenti, facendo riferimento ai modelli e linguaggi esistenti per trattare comunicazione e coordinazione, sia la struttura interna degli agenti stessi, modellando l'evoluzione della conoscenza nei singoli agenti in seguito alle interazioni e l'influenza che la conoscenza degli agenti sugli altri agenti ha sul comportamento emergente del sistema.

Un aspetto cruciale che verrà preso in considerazione è l'incompletezza della conoscenza degli agenti sull'ambiente, dovuta al fatto che l'ambiente evolve dinamicamente in seguito alle interazioni degli agenti stessi. La trattazione di questo aspetto richiederà l'adozione di tecniche di ragionamento non-monotono quali l'abduzione.

L'obiettivo è quello di individuare e sviluppare formalismi logici atti a modellare e specificare il comportamento di agenti, le loro modalità di interazione intelligente e i comportamenti globali di sistemi multiagente.

L'obiettivo principale dell'unità 3 è quello di studiare come descrivere mediante modelli basati su logica, il comportamento individuale e collettivo di agenti che interagiscono in società aperte. Si intende definire modelli di agenti e delle loro possibili interazioni nell'ambito di società, che permettano sia la specifica formale e la verifica di proprietà, sia la concreta implementazione. Gli agenti che verranno presi in considerazione sono caratterizzati da propria conoscenza, capacità computazionali, risorse, obiettivi e regole di comportamento. Tipicamente essi hanno solo conoscenza parziale, incompleta e non precisa della società e dell'ambiente. In questo contesto, si intende studiare l'applicabilità di forme di **ragionamento abduttivo** allo scopo di modellare sia le interazioni tra agenti logici che il comportamento del singolo agente. A questo fine, si considereranno due livelli: il singolo agente e l'interazione tra agenti.

Riguardo al primo punto, saranno identificate ed eventualmente estese varie tecniche di ragionamento ipotetico (con particolare interesse verso l'abduzione) affinché gli agenti possano gestire conoscenza incompleta, ed eventualmente di integrarle tra loro.

Riguardo al secondo punto, gli obiettivi consistono nello studio e nell'identificazione dei modelli più adatti per esprimere l'interazione fra agenti in una società aperta. Ci si concentrerà sulla definizione di un modello logico sufficientemente ricco per permettere la descrizione della società ed in particolare delle interazioni tra gli agenti tramite una descrizione basata su regole.

Le unità 1 e 2 faranno riferimento principalmente a tecniche basate su **logiche modali e temporali**. L'obiettivo è quello di sviluppare una "teoria delle azioni" che permetta di modellare l'interazione di più agenti. Tale teoria delle azioni verrà utilizzata nella modellazione della comunicazione fra agenti intelligenti che interagiscono secondo protocolli predefiniti o pianificando dinamicamente le azioni comunicative da compiere. Il punto di partenza è costituito dalla teoria delle azioni basata sulla logica temporale DLT (Dynamic Linear Time Temporal Logic), una estensione di LTL nella quale la modalità "until" viene indicizzata mediante espressioni regolari (programmi nella logica dinamica). Uno degli obiettivi nella prima fase è quello di estendere questa teoria delle azioni per descrivere il comportamento di un insieme di agenti, ciascuno dotato di uno stato locale, che interagiscono eseguendo azioni comunicative. Nel contesto multiagente, la teoria delle azioni permette di vedere i protocolli di comunicazione fra agenti come piani (non-deterministici) che definiscono i possibili pattern di comunicazione fra gli agenti e di provarne proprietà. Un problema importante è quello di modellare il processo di **revisione delle credenze** che si innesca quando l'agente riceve nuove informazioni in seguito all'interazione con altri agenti. Il problema di rivedere lo stato mentale di un agente è più articolato della semplice revisione di un insieme di credenze, che è il problema trattato dalla teoria della revisione: la comunicazione fra agenti può produrre modifiche non solo delle credenze dell'agente, ma anche dei suoi goal, delle sue intenzioni e dei suoi piani. Questo problema verrà affrontato dalle unità 1 e 2.

1.2 Linguaggi di specifica basati sulla logica computazionale

Uno dei principali problemi legati all'uso di linguaggi di specifica formali è dato dalla distanza che spesso esiste fra la specifica e la reale implementazione del sistema multiagente. Per cercare di ovviare a questo problema, il progetto ha anche l'obiettivo di definire linguaggi di specifica eseguibili, ottenuti considerando frammenti computazionali delle logiche studiate nel punto precedente.

Il punto di partenza per raggiungere questo obiettivo è costituito dai linguaggi di programmazione logica, con opportune estensioni modali, temporali e abduttive. L'uso di linguaggi di specifica eseguibili consente la realizzazione rapida di prototipi di sistemi multiagenti, la cui esecuzione può essere utilizzata per la comprensione e la validazione di proprietà relative all'interazione fra le componenti del sistema. I linguaggi di specifica logici, avendo una semantica formale, permettono di ragionare - anche in modo automatico - sulle specifiche, e quindi di verificare proprietà del sistema.

L'unità 3 intende sviluppare la parte computazionale del modello individuato nel punto precedente. Questa parte computazionale potrà essere ottenuta mediante l'estensione e l'integrazione di proof procedure esistenti. In questo ambito, i linguaggi logici e, in particolare, la programmazione logica, grazie alle sue caratteristiche dichiarative ed operazionali, se integrata con il ragionamento di tipo abduttivo, fornisce un ottimo strumento per modellare le interazioni tra agenti e per la verifica "run-time" di proprietà a livello sociale. Parte integrante dell'attività sarà quindi la definizione di un linguaggio per esprimere l'interazione fra gli agenti.

Obiettivo dell'unità 2 sarà studiare come estendere il linguaggio DyLOG, definito ed implementato in precedenza dall'unità, in modo che possa essere usato per modellare un agente razionale che, situato in un dominio multi-agente, possa dialogare con altri agenti eseguendo e interpretando atti comunicativi primitivi. DyLOG è un linguaggio di programmazione logica basato su una teoria modale dell'azione, che può essere usato per specificare ed eseguire il comportamento di un agente software razionale. Tale agente è dotato della capacità di modificare l'ambiente, di acquisire nuove informazioni dall'esterno, di ragionare sull'effetto delle proprie azioni e pianificare, in base alle proprie conoscenze, le azioni da compiere per raggiungere i propri obiettivi. Lo stato mentale dell'agente verrà arricchito in vari modi. In primo luogo verranno introdotte accanto alle modalità epistemiche che modellano le credenze, modalità per la rappresentazione esplicita dei "goal" dell'agente. In secondo luogo sarà possibile rappresentare "nested attitudes", in modo da fornire agli agenti la capacità di rappresentare e ragionare sulle attitudini mentali degli altri agenti.

Anche l'unità 1 intende sviluppare un linguaggio di programmazione logica per ragionare sulle azioni, derivato dalla teoria sviluppata al punto precedente, basata su logiche modali e temporali. La definizione di tale linguaggio avverrà in stretto contatto con l'unità 2, in modo da sfruttare l'esperienza già maturata con la definizione del linguaggio DyLOG. In questa fase del progetto verrà effettuata prevalentemente una analisi dello stato dell'arte sull'argomento, rinviando la definizione del linguaggio alla seconda fase.

Sempre l'unità 1 intende studiare la fattibilità dell'uso dell'ambiente di prototipazione rapida CaseLP per integrare i linguaggi sviluppati in questo progetto, partendo dal linguaggio DyLOG.

Complessivamente si prevede di instaurare un'attiva collaborazione fra tutte le unità del progetto, per studiare le relazioni fra i modelli proposti, con l'obiettivo di arrivare ad una integrazione fra le proposte.

2. Validazione e verifica

I linguaggi di specifica logici, avendo una semantica formale, permettono di ragionare - anche in modo automatico - sulle specifiche, e quindi di verificare proprietà del sistema. Questo tema è già stato considerato nel punto precedente e non verrà trattato in questo punto.

In questo punto si intendono affrontare l'uso di altre tecniche, in particolare l'adattamento alla verifica di proprietà di sistemi multiagenti di tecniche di **model checking**, che hanno già avuto molto successo nel campo dei sistemi software distribuiti. Le unità 1 e 2 intendono adattare tecniche di model-checking alla logica Dynamic Linear Time Temporal Logic, descritta al punto 1.1, con l'obiettivo di verificare protocolli di comunicazione fra agenti. La prima fase sarà dedicata allo sviluppo di tecniche per il model-checking specifiche per questa logica. Si intende anche esplorare la possibilità di utilizzare gli strumenti per il model-checking già disponibili per la logica temporale lineare, sviluppando tecniche di trasformazione di formule nella logica DTL in formule LTL standard.

3 Applicazioni

Sono state individuate alcune applicazioni rispetto a cui valutare linguaggi, tecniche e strumenti sviluppati nell'ambito dei punti precedenti. Lo scopo di questa fase sperimentale è quello di comprendere vantaggi e svantaggi degli approcci, la loro efficienza e flessibilità. L'applicazione a problemi "reali" permette, inoltre, di individuare un insieme di proprietà interessanti da verificare.

Siti web adattativi

L'unità 2 utilizzerà i linguaggi definiti ai punti precedenti, in particolare il linguaggio DyLOG, per la realizzazione di siti web in cui l'adattatività sia basata su intenzioni e necessità. In questo caso l'adattatività è intesa essere un'interazione fra il sistema e l'utente finalizzata a proporre e rifinire soluzioni studiate ad hoc, il tutto attraverso un ragionamento che l'agente razionale applica alla descrizione del dominio, interna al sistema, e alla situazione e ai goal dello specifico utente. Nel corso del primo anno di progetto intendiamo approfondire questi argomenti, sviluppando un sistema di supporto alla costruzione e alla validazione di piani di studi. Da un lato ciò ci consentirebbe di dimostrare le potenzialità e la validità delle tecniche sviluppate nella logica computazionale in contesti fortemente applicativi, dall'altro questa applicazione, dalla natura e di interesse reale, è un banco di prova degli strumenti formali offerti dalla logica computazionale che si possono confrontare con strumenti alternativi utilizzati in letteratura.

Sistemi ad agenti applicati alla medicina

L'unità 3 intende occuparsi di sistemi di supporto alla diagnosi medica che richiedono interazioni tra agenti basate su dialoghi, con particolare attenzione alla negoziazione per l'allocazione di risorse. Si ritiene infatti che le problematiche relative alla negoziazione per l'allocazione delle risorse siano tipiche di sistemi distribuiti complessi, come quelli appartenenti all'ambito applicativo citato. Riguardo ai sistemi ad agenti applicati alla medicina, infatti, lo scenario potrebbe comprendere un insieme eterogeneo di agenti fornitori di particolari servizi/risorse (ad esempio, diagnosi, esami clinici, informazioni anagrafiche ecc.). Il piano di un agente utente del sistema potrebbe pertanto richiedere l'allocazione preventiva di alcune risorse, per le quali quindi l'adozione di

particolari tecniche e protocolli di negoziazione può essere un importante strumento per il raggiungimento di obiettivi concreti, quali, ad esempio, la diagnosi di patologie complesse.

Sempre in questo ambito l'unità 1 prenderà in considerazione la definizione di agenti per trattare vincoli temporali nell'esecuzione di azioni nelle linee guida cliniche.

Agenti per il controllo di sistemi complessi in presenza di guasti

Questo ulteriore obiettivo dell'unità 1 consiste nella modellazione di modalità di interazione fra agenti che cooperano alla risoluzione di problemi per il controllo di sistemi complessi, in cui può essere necessario:

- supervisionare il funzionamento del sistema, rilevando eventuali funzionamenti anomali;
- ragionare in modo diagnostico quando questi ultimi sono stati rilevati;
- pianificare ed effettuare azioni per acquisire ulteriori informazioni per discriminare tra ipotesi alternative;
- pianificare le azioni correttive, ad esempio di riconfigurazione del sistema o di riduzione della funzionalità;
- verificare che queste azioni ottengano i risultati previsti e, in caso contrario, rivedere o adattare le soluzioni.

Nella prima fase del progetto l'unità 1 intende modellare alcune delle attività descritte in precedenza nei formalismi logici e come comportamenti di agenti.

Testo inglese

In order to act in a dynamically evolving environment and to interact with other agents, an agent needs the capability of reasoning on the possible effects on the world of his own actions and of the actions of other agents. Moreover, it needs planning capabilities to reach its goals by executing suitable action sequences, while interacting with other agents through predetermined communication protocols. Its knowledge on the world, in general, is partial so that the effects of its actions on the world are not univocally determinable. The agent can acquire new knowledge on the world through knowledge producing actions ("sensing" actions) or communicative actions, and has to change its knowledge about the world when receiving new pieces of information. In its planning activity in presence of incomplete knowledge about the (dynamically changing) world, the agent has to reason hypothetically on the possible effects of actions and to suitably schedule the execution of his actions.

This project aims at identifying formalisms, tools and techniques by which systems involving knowledge-based interaction can be modeled, developed and analyzed. Such formalisms, tools and techniques should also facilitate the analysis and verification of such systems. The project will focus, on the one hand, on the belief and goal dynamic, internal to the agent, as well as on its reasoning and planning capabilities and, on the other hand, on the dynamic of the interaction among agents, when they communicate in a dynamically evolving world about which the knowledge of agents is incomplete. The two levels are tightly related, as the capabilities of an agent to interact are supported by its reasoning and planning capabilities.

To achieve the goal of the project we will exploit the long lasting experience of all participants on modal and temporal logics, automated reasoning, logic programming and extensions thereof. We expect that a close collaboration among them will allow to achieve the goals of the project.

During the first phase of the project there will be a close interaction between the partners, to let every group deepen its knowledge about other groups' research activity and understand the main characteristics, the advantages and the possible weaknesses of the different approaches. We will also consider some case studies in different application areas, which will allow to experiment with techniques and tools developed within the project.

The main tasks of the project can be summarized as follows:

1. Modeling agent based systems
 - 1.1 Formal models of agent based systems
 - 1.2 Specification languages based on computational logics
2. Validation and verification
3. Case studies

1. Modelling agent based systems

1.1 Formal models of agent based systems

Because of their expressiveness, logic specification languages are suitable to model the complex and articulated behavior of agent societies as well as for modeling the sophisticated reasoning capabilities of the individual agents. These specification languages should model uniformly both agent interactions (for which we will refer to the formalisms and modeling techniques developed to deal with communication and coordination) and the dynamic of each agent's beliefs, to the extent this influences the emerging behavior of the system. One of the central issues that has to be addressed is agents incomplete knowledge about the world, which evolves dynamically under the effects of agent interactions. This issue will require the use of non-monotonic reasoning techniques.

The goal is to devise and develop logical formalisms suitable for modeling and specifying the behavior of agents, their intelligent interaction and the global behaviors of multi-agent systems.

*The main aim of unit 3 is to investigate how to specify with logic-based models both the individual and the collective behavior of agents that interact in open societies. The problem of how to bridge this gap between specification and implementation in an expressive and consistent way, is still an open issue. The aim is to supply models of agents and their interactions that not only allow formal specification and property verification, but also lead to a concrete implementation. In this context, unit 3 will investigate the applicability of various kinds of **abductive reasoning**, with the purpose of modeling both the interaction among logic-based agents and the behavior of each single agent. To this purpose, two levels will be taken into consideration: the single agent and the interaction among agents.*

With regard to the first point, the aim will be to identify and possibly extend logic-based hypothetical reasoning techniques (with a special focus on abduction) in order to make agents able to manage incomplete knowledge and possibly to integrate them. The aim of the second point is to study and to identify the most suitable models for expressing agent interaction in an open society, and to deliver a computational counterpart to the logic-based models developed. This computational framework will be obtained by means

of advanced extensions and integration of existing computational logic proof procedures.

Units 1 and 2 will refer mainly to approaches based on **modal** and **temporal logics**. The aim is the development of a "theory of actions" which allows the interaction of multiple agents to be captured. The action theory will be used to model communication among intelligent agents, which interact according to predetermined protocols or which dynamically plan the communicative actions to be executed. The starting point will be an action theory based on temporal logic DLTL (Dynamic Linear Time Temporal Logic), an extension of propositional temporal logic of linear time in which regular programs of dynamic logic can be used for indexing temporal modalities. In this action theory, actions are represented by modalities and temporal queries and constraints can be formulated using the standard temporal operators. One of the aims in the first phase of the project is the extension of this action theory to address the behaviour of a set of agents, with their own local states, which interact by executing communicative actions. In the multi-agent context, action theory allows communication protocols to be modelled as (non-deterministic) plans, which define possible communication patterns among the agents, and properties to be proved for these plans.

One important problem is the modelling of the **belief revision** process which is triggered when an agent receives new information from the environment, as a consequence of its interaction with other agents. The problem of revising the mental state of an agent, which will be tackled in the first phase of the project, is more complex than the simple revision of a set of beliefs: communication between agents may affect not only the beliefs of the agent, but also its goals, intentions and plans. This topic will be tackled by units 1 and 2.

1.2 Specification languages based on computational logic

One of the main problems related to many formal specification languages is the gap between specifications and the actual implementation of the system. To solve this problem the project includes the goal of defining executable specification languages, obtained by considering computational fragments of the logics studied in the previous task.

The specification languages considered in this task will be logic programming languages, suitably extended so as to deal with modal and temporal operators, or abduction. Executable specification languages allow rapid prototyping of multi-agent systems, thus providing a tool for understanding and validating properties related to the interaction among components of the system. Logic specification languages, having a formal semantics, allow to reason, even automatically, on specifications, and thus to formally verify properties of the system.

Unit 3 will deliver a computational counterpart to the logic-based models developed in task 1.1, in the form of advanced extensions and integration of existing computational logic proof procedures. In this context, logic languages, and logic programming in particular, thanks to its declarative and operational aspects, if suitably extended with abductive reasoning, would be a good tool for modelling interaction among agents, integrating local reasoning, and verifying at run-time some global properties of the resulting system. An important part of the activity will be therefore the definition of a language for modelling the structure of agents. Such a language must also allow the user to specify interaction among agents.

One of the aims of unit 2 will be the study of extensions of DyLOG, a language previously defined and implemented by the group, which will allow to model a rational agent, that, when placed in a multi-agent framework, will talk with other agents by executing and interpreting primitive communication acts. DyLOG is a logic programming language based on a modal theory of actions, which can be used to specify and execute the behaviour of a rational agent. The agent can modify the environment, acquire new information, reason on the effects of its actions and plan a sequence of actions to achieve a given goal. The mental state of the agent will be enriched in many ways. First of all, besides the epistemic modalities that allow to model beliefs, we will introduce modalities for explicitly representing agent goals. Then, it will be possible to represent nested attitudes so to allow agents to represent and reason about the mental attitudes of the other agents.

Unit 1 plans as well to develop a logic programming language for reasoning about actions, based on the theory developed in task 1.1. The definition of the language will be done in close contact with unit 2, so as to exploit the experience acquired with the design of language DyLOG. The first phase of the project will be mainly devoted to an analysis of the state of the art, whereas the development of the language will be deferred to the second phase.

Furthermore units 1 and 2 will explore the feasibility of using the rapid prototyping environment CaseLP for integrating the languages developed in this task, starting with DyLOG.

2. Validation and verification

Logic specification languages, having a formal semantics, allow to reason, even automatically, on specifications, and thus to formally verify properties of the system. This topic is already considered in task 1.2 and thus will not be mentioned in this task. Here we will focus on the use of other techniques, in particular by studying how to use the **model checking** approach for the verification of properties of multi-agent systems, by adapting techniques and tools which have successfully been applied in the field of distributed processes.

The aim of units 1 and 2 is to adapt model checking techniques to the logic DLTL, mentioned in task 1.1, for verifying communication protocols among agents. The first phase will be devoted to the development of specific model checking techniques for this logic. A further line of research will be to investigate the possibility of using already available model checking tools, by developing suitable translation techniques from DLTL formulas to standard LTL formulas.

3. Case studies

To evaluate languages, techniques and tools developed in the previous tasks, some application areas have been selected. This will allow to have an experimental evaluation for understanding advantages and disadvantages of the approaches, their efficiency and flexibility.

Adaptive web sites

Unit 2 will use the languages defined in task 1.2, in particular DyLOG, for building web sites, where adaptation is based on

intentions and necessity. In this case adaptation is meant to be an interaction between the system and the user, aimed at building and refining ad hoc solutions, by means of a reasoning process applied by the rational agent to a description of the domain (internal to the system) and to the situation and the goals of the specific user. During the first year of project we want to develop a system that will support students in the process of constructing and validating a study plan. On one hand, this work will allow us to prove the validity of the developed computational logic in highly applicative contexts, on the other hand, this application, which is a real-world application of real interest, is an important test bed for the tools offered by computational logic, that will be compared with other tools that are already used in the field literature.

Agent systems applied to medicine

Unit 3 will take into consideration agent systems for medical diagnosis. This application requires strong agent interactions based on dialogues, with a particular focus on the negotiation for resource allocation. Actually, the negotiation for resource allocation is a typical aspect of complex distributed systems, such as those supporting the solution of complex medical tasks. In those systems, in fact, applicative scenarios could enclose an heterogeneous set of agents, each providing particular services/resources (e.g. diagnosis, clinical tests, personal clinical data, etc.). Thus an agent plan could possibly require the preventive allocation of some resources/services necessary for its execution. In this setting, therefore, the adoption of peculiar techniques and negotiation protocols could be an important tool for achieving concrete goals, such as the diagnosis of complex pathologies.

In the same context, unit 1 will consider the definition of specific agents to deal with temporal constraints in the executions of actions in clinical guidelines.

Agents for the control of complex systems in presence of faults

The modelling and reasoning methodologies will be applied by unit 1 to the case study of agents for the control of complex systems in presence of faults, where several tasks should interact:

- supervising the system behaviour, detecting abnormal behaviour;
- performing diagnostic reasoning after such a fault detection;
- planning and executing actions to acquire further information, in order to discriminate among alternative hypotheses;
- planning actions to repair or reconfigure the system, or to provide limited functionality;
- monitor the execution of such actions, to verify that the expected results are achieved, and, if this is not the case, revising or adapting the solutions.

In the first phase of the project some of the tasks described above will be modelled in the logic formalisms and as agent behaviours. Individual actions, as well as events that are not under control of the agent (e.g. a component breaking down) will be described relying on a uniform representation of basic events and actions.

Risultati parziali attesi

Testo italiano

- Un rapporto, contenente una raccolta di articoli scientifici. Il rapporto presenterà i risultati della ricerca dei singoli gruppi, confrontandone le diverse proposte.
- La definizione ed implementazione preliminare di linguaggi di programmazione basati sulla logica computazionale sviluppati al punto 1.2.
- Descrizione dello stato di avanzamento nella modellazione delle applicazioni del punto 3 mediante l'uso degli strumenti sviluppati nel progetto.

Testo inglese

- A report containing a collection of scientific papers. This report will present research results of the groups and compare the different proposals.
- Definition and preliminary implementation of logic based programming languages developed in task 1.2.
- Progress report on modelling case studies by means of tools and techniques developed in task 1.

Unità di Ricerca impegnate

Unità n. 1

Unità n. 2

Unità n. 3

Fase 2

Durata e costo previsto

Durata Mesi 12 Costo previsto Euro 79.000

Descrizione

Testo italiano

L'attività di ricerca del secondo anno continuerà lungo le linee presentate nella Fase 1, e sarà dedicata soprattutto allo sviluppo dei linguaggi basati su logica computazionale e degli strumenti di verifica definiti nella prima fase. Si procederà comunque, parallelamente, nello studio dei modelli formali. In particolare, saranno realizzati dei prototipi dei linguaggi di specifica basati sulla programmazione logica, che saranno utilizzati nei settori applicativi individuati nel progetto. D'altro canto, si creeranno forti connessioni tra i lavori sulle specifiche formali dell'interazione e quelli sulla verifica, sviluppando metodi di prova e tecniche di verifica per i formalismi definiti nell'ambito del progetto.

Sottolineiamo ancora che uno degli obiettivi principali del progetto è proprio il confronto e l'integrazione delle diverse proposte che saranno sviluppate dai diversi gruppi partecipanti.

Nel seguito si descrivono le attività previste per la seconda fase nei vari obiettivi del progetto.

1. Modellazione di sistemi ad agenti

1.1 Modelli formali di sistemi ad agenti

Tutte le unità proseguiranno nello studio di tecniche basate su **abduzione** e **logiche modali** e **temporali** con l'obiettivo di modellare agenti che possano gestire conoscenza incompleta ed interagire con altri agenti.

L'unità 3 si concentrerà prevalentemente sulla parte computazionale, sviluppando procedure di dimostrazione per il modello sviluppato nella prima fase.

L'obiettivo delle unità 1 e 2, è quello di estendere la teoria delle azioni sviluppata nella prima fase mediante l'introduzione di modalità epistemiche e di altre modalità per rappresentare goal e intenzioni dei vari agenti, nei termini dei quali definire gli effetti e le precondizioni delle azioni comunicative. Le modalità epistemiche forniranno all'agente capacità di introspezione nonché la possibilità di ragionare ed agire in base alla conoscenza dei goal, delle credenze e delle intenzioni altrui. Verrà inoltre approfondito l'uso di questa teoria per specificare protocolli di comunicazione tra agenti.

Riguardo al problema della **revisione delle credenze** un aspetto interessante che verrà studiato dalle unità 1 e 2 è quello della rappresentazione delle credenze che ogni agente ha delle strategie di revisione che hanno gli altri agenti. Questa conoscenza può permettere ad un agente intelligente di pianificare i propri atti comunicativi con gli altri agenti nel modo più proficuo per perseguire i propri goal. La rappresentazione delle strategie di revisione di un agente può essere data in modo naturale mediante formule condizionali. L'uso della logica condizionale deve essere supportata dallo sviluppo di metodi di deduzione. A questo scopo si intendono studiare metodi di prova analitici, come i tableaux ed i calcoli dei sequenti, per i sistemi di logiche condizionali più noti.

1.2 Linguaggi di specifica basati sulla logica computazionale

In questa fase ci si concentrerà soprattutto sulla implementazione dei linguaggi logici definiti nella prima fase. Questo permetterà da un lato di sperimentare tecniche di verifica di proprietà per sistemi multiagente e dall'altra di realizzare prototipi negli ambiti applicativi selezionati.

L'unità 3 intende sviluppare la parte computazionale del modello individuato, ottenuta mediante l'estensione e l'integrazione di proof procedure esistenti. Parte integrante dell'attività sarà la definizione di un linguaggio per esprimere l'interazione fra gli agenti. In particolare, si considererà un linguaggio di comunicazione tra agenti basato sulla logica che contenga tutti gli elementi sintattici richiesti per esprimere i principali atti comunicativi. A tale linguaggio verrà associata una semantica formale, tale che possa essere effettivamente verificata a tempo di esecuzione.

Verrà inoltre definito un formalismo basato sulla logica che consenta di esprimere i protocolli di interazione specifici di una data società di agenti e/o di una certa applicazione. Analogamente al linguaggio di comunicazione, anche in questo caso si ritiene importante poter verificare automaticamente la conformità ai protocolli delle sequenze di interazioni tra agenti generate all'interno della società.

Infine, si prevede la realizzazione, in forma prototipale, di alcune estensioni delle tecniche di ragionamento adottate per modellare le interazioni e la verifica a tempo di esecuzione di alcune proprietà globali del sistema.

Obiettivo dell'unità 2 sarà di proseguire nella realizzazione del linguaggio DyLOG esteso. Più precisamente, la teoria dell'azione verrà estesa per consentire di specificare sia primitive di comunicazione sia politiche di conversazione. Si intende definire una semantica formale della comunicazione che caratterizzi le strategie di decisione seguite dall'agente per rispondere a una comunicazione ricevuta nel contesto di una conversazione, studiando come specificare politiche di conversazione. L'idea è di rappresentare tali protocolli di conversazione tramite definizioni di procedure DyLOG, che, sulla base delle azioni comunicative primitive individuate, definiscano i possibili comportamenti conversazionali dell'agente e quindi semplifichino il processo di selezione della risposta appropriata a una comunicazione ricevuta. A partire dalla definizione formale degli atti comunicativi primitivi, sarà possibile definire e derivare proprietà semantiche di conversazioni.

L'unità 1 intende definire una controparte computazionale delle teorie delle azioni sviluppata nell'obiettivo 1.1 mediante un linguaggio di programmazione logica arricchito con modalità temporali, epistemiche, con azioni di sensing e azioni comunicative.

Esso verrà inoltre arricchito con modalità "goal" e "intenzioni" per renderlo più adatto alla programmazione ad agenti. Lo sviluppo del linguaggio verrà fatto in collaborazione con l'unità 2.

È prevista infine una stretta interazione fra le unità che realizzeranno dei prototipi e l'unità 1 per integrare i prototipi realizzati nell'ambiente di prototipazione rapida CaseLP, disponibile presso questa unità.

2. Validazione e verifica

Il contributo dell'unità 3 a questo obiettivo è stato descritto nel punto 1.2.

In questa fase ci sarà una stretta interazione con le unità 1 e 2, per lo studio e l'adattamento degli algoritmi di model-checking ai linguaggi logici studiati. Queste unità intendono infatti esplorare la utilizzabilità delle tecniche di model checking, sviluppate per le logiche temporali, nella pianificazione e nella verifica di proprietà dei sistemi multiagente modellati. In particolare, si intende proseguire nello sviluppo di tecniche di model checking per la logica delle azioni DTLT iniziato nella Fase 1. Tali tecniche saranno applicate alla verifica di protocolli di interazione fra agenti e a problemi di pianificazione.

3 Applicazioni

Lo scopo di questo obiettivo è quello di comprendere vantaggi e svantaggi degli approcci proposti, la loro efficienza e flessibilità, applicandoli a problemi "reali". Nel seguito non sono dettagliate le partecipazioni delle diverse unità, perché ci si aspetta che tutte le unità possano ugualmente contribuire con gli strumenti sviluppati nel progetto, a seconda delle esigenze dei domini applicativi considerati.

Siti web adattativi

Si intende utilizzare questa applicazione anche nel corso del secondo anno, passando da un sistema in cui un utente interagisce con un singolo agente razionale ad un sistema in cui un utente interagisce con un insieme di agenti razionali che collaborano al fine di determinare una soluzione. Per esempio, se supponiamo che i nostri agenti abbiano capacità specializzate a compiere compiti distinti, potremmo immaginare che un agente che esegue la verifica di un piano, richieda l'aiuto di un agente pianificatore per sostituire un pezzo di piano errato con una proposta di piano corretto. In questo modo si avrebbe come ulteriore vantaggio una maggiore efficienza del sistema dovuta alla condivisione degli agenti specializzati fra più utenti. Come conseguenza degli studi formali iniziati nel primo anno di progetto, gli agenti saranno dotati di protocolli che definiranno i possibili schemi di conversazione. Le proprietà delle conversazioni deriveranno logicamente dalle proprietà degli atti comunicativi individuali.

Sistemi ad agenti applicati alla medicina

Si considererà come caso di studio le interazioni tra agenti basate su dialoghi, nell'ambito dei sistemi di supporto alla diagnosi medica, con particolare attenzione alla negoziazione per l'allocazione di risorse. In questo contesto, si può assumere che ogni agente faccia riferimento ad un piano da eseguire per il raggiungimento del goal attuale. Per la realizzazione di tale piano può essere necessario ottenere l'allocazione di un insieme di risorse. L'allocazione di ogni risorsa si ottiene mediante una fase di negoziazione, realizzata mediante una sequenza di interazioni (o dialogo). Si intende quindi formalizzare la struttura del protocollo di negoziazione fra agenti in termini di linguaggio e conoscenza necessaria per costruire un dialogo fra due agenti.

Inoltre l'agente per il ragionamento temporale a vincoli, proposto dall'unità 1, verrà utilizzato nell'applicazione alle linee guida mediche. Nelle linee guida mediche, le azioni devono usualmente essere eseguite secondo dei pattern temporali predefiniti. I vincoli temporali fra le azioni sono quindi una parte essenziale delle linee guida cliniche ed un agente specializzato è utile per gestirle (ad esempio, per verificarne la consistenza).

Agenti per il controllo di sistemi complessi in presenza di guasti

Nella seconda fase del progetto verrà formalizzata l'attività complessiva di risoluzione del problema, descritto nella prima fase del progetto, in almeno uno dei seguenti modi:

- come comportamento di un singolo agente che ha l'obiettivo di mantenere il sistema funzionante
- come comportamento di un gruppo di agenti specializzati in diverse attività di ragionamento
- come comportamento di un gruppo di agenti specializzati nella conoscenza su diversi sottosistemi

Testo inglese

The research activities in the second year will continue along the lines presented in Phase 1, and will be mainly devoted to the development of the languages and validation tools singled out in the first phase, although the studies on more theoretical aspects will be continued. In particular, the specification languages based on logic programming will be implemented and used to develop prototypes of the case studies. On the other hand, the work on formal specifications of interaction will be strongly connected with the work on verification, by developing proof methods and verifications techniques for the formalisms proposed within the project. We want to remark again that a major goal of the project is to compare and integrate the different proposals which will be developed by the groups in the project.

In the following we present the activities scheduled for the second phase of the project.

1. Modelling agent based systems

1.1 Formal models of agent based systems

All units will continue the study of techniques based on **abduction, modal** and **temporal logics**, with the goal of modelling agents able to manage incomplete knowledge and to interact with other agents.

Unit 3 will focus on the computational part of the model developed in the first phase, by defining proof procedures.

The goal of units 1 and 2 in the second phase of the project is the extension of the action theory, based on logic DLTL, developed in the first phase through the introduction of epistemic modalities and other modalities representing agents goals and intentions. Effects and preconditions of communicative actions will be defined in terms of such goals and intentions. Epistemic modalities provide the agent with introspection capabilities and with the capability of reasoning and acting according to the knowledge about other agents' goals and intentions. A further aim will be to investigate the use of this theory for specifying communication protocols.

Units 1 and 2 will tackle an interesting aspect of belief revision when considered in a multi-agent context: how to represent the beliefs that an agent has about other agents' revision strategies. This knowledge may allow the communicative acts of an intelligent agent to be planned so to achieve its goals in the most profitable way. The representation of the agent's revision strategies can be given in a natural way by means of conditional formulas. The use of conditional logics must be supported by the development of deduction methods for them. To this regard we intend to study analytic proof methods like tableaux and sequent calculi for the most important systems of conditional logics.

1.2 Specification languages based on computational logics

In this phase we will focus mainly on the implementation of the logical languages defined in the first phase. This will allow on one hand to experiment verification techniques for multi-agent systems and on the other hand to realize prototypes for the selected case studies.

Unit 3 will develop a computational counterpart to the proposed logic-based models. This computational framework will be obtained by means of advanced extensions and integration of existing computational logic proof procedures. In particular it will define a logic-based agent communication language whose syntax could allow to express main communicative actions. A formal semantics will be associated with the language, such that it could be actually verified at run-time within the society.

We will also define a proper logic-based formalism for expressing interaction protocols peculiar either of a given society or a specific application. Similarly to the ACL case, also in this case we think that an important point should be to achieve the capability of automatically verifying protocol compliance of agents interaction sequences produced into the society.

Moreover, we will provide implementation (in prototypical form) of some extended reasoning techniques (abductive, constraint-based) designed for modelling interaction and run-time verifying some global properties of the resulting system.

The goal of unit 3 is to extend DyLOG so to allow the specification of communication primitives as well as of conversation policies. Indeed we mean to define a formal semantics of communication that characterizes the decision strategies followed by the agent for answering to a communication, received during a conversation, by studying how to specify conversation policies. The idea is to represent such conversation protocols by means of DyLOG procedures, which, on the basis of the identified primitive communication acts, define the agent possible conversational behaviours, thus simplifying the selection of proper answers. Starting from the formal definition of primitive communication acts, it will be possible to define and derive semantic properties of the conversations.

Unit 1 will define a computational counterpart of the action logic analyzed in task 1.1 of the first phase of the project, by developing a (modal) logic programming language enriched with temporal and epistemic modalities and with sensing and communicative actions. It will also be enriched with "goal" modalities and "intention" modalities, to make it suitable for agent programming. The development of the language will be done in conjunction with unit 2.

As far as rapid prototyping is concerned, in the second phase of the project we will study how the DyLOG language developed by unit 2 can be integrated in the MAS rapid prototyping environment CaseLP, available by unit 1. This integration would allow us to take advantage of the capabilities CaseLP already offers, such as execution of logically distributed and concurrent agents, support to communication among agents, statistical information collection, and so on, for executing agents specified in DyLOG.

2. Validation and verification

The activity of unit 3 regarding verification of properties of multi-agent systems has already been described in task 1.2.

In this phase there will be a close interaction within units 1 and 2. These groups will investigate how model checking techniques developed for temporal logic can be exploited in planning and in proving properties of the modelled multi-agent systems. More precisely, they intend to continue the development of model checking techniques for the action logic DLTL described in the first task and to apply these techniques to the verification of communication protocols between agents and to planning problems.

3 Case studies

The aim of this task is to understand advantages and disadvantages of the proposed approaches, their efficiency and flexibility, by applying them to real-world problems. In the following we do not specify in detail the contributions of the units of the project, because we expect that each one will be able to contribute according to the needs of the application domains considered.

Adaptive web sites

In the second phase we will consider a system where a user will interact with a set of rational agents, that will co-operate to find a solution. For instance, supposing that our agents have specialized functions for executing distinct tasks, we can imagine an agent, whose task is to verify the correctness of a student given study plan, that asks for the contribution of another agent, that is able to perform planning, for substituting a wrong part of study plan with a correct proposal. Another advantage will be a greater efficiency of the system, due to the fact that specialized agents are shared by users. As a consequence of the formal studies carried on during the first year of project, our agents will be supplied with protocols that will define the possible conversation schemes. The conversation properties will logically derive from the individual communication acts properties.

Agent systems applied to medicine

We will consider as a case study agents interactions based on dialogues, with a particular focus on the negotiation for resource allocation, in the context of medical diagnosis systems. In this context we assume that each agent would refer to a plan to be executed in order to reach its current goal. In order to execute its plan, each agent may need to obtain the allocation of a set of necessary resources. Resource allocation is obtained by means of a negotiation phase, consisting of a sequence of interactions (the dialogue). We will formalize the framework for agent dialogue in terms of language and knowledge that are needed to build a dialogue between two agents.

Furthermore, the temporal-reasoning agent proposed by unit 1 will be applied to the case study of clinical guidelines. In clinical guidelines, actions have usually to be executed according to pre-defined temporal patterns. Thus, temporal constraints between actions are an essential part of clinical guidelines, and a specialised agent is useful to manage them (e.g., to check their consistency).

Agents for the control of complex systems in presence of faults

In the second phase of the project, the overall problem solving behaviour will be formalized in at least one of the following ways:

- as the behaviour of a single agent with the goal of a functioning system
- as the behaviour of a group of agents specialized in different reasoning tasks
- as the behaviour of a group of agents specialized in knowledge on different subsystems

Risultati parziali attesi**Testo italiano**

- Un rapporto, contenente una raccolta di articoli scientifici. Il rapporto presenterà i risultati della ricerca dei singoli gruppi, confrontandone le diverse proposte.
- La definizione ed implementazione di linguaggi di programmazione basati sulla logica computazionale sviluppati al punto 1.2.
- Prototipi per le applicazioni elencate al punto 3.
- Prototipo per la verifica di proprietà di sistemi multiagente mediante "model checking".

Testo inglese

- A report containing a collection of scientific papers. This report will present research results of the groups and compare the different proposals.
- Definition and implementation of logic based programming languages developed in task 1.2.
- Prototypes of case studies.
- Prototype of a verifier of properties of multi-agent systems based on model-checking techniques.

Unità di Ricerca impegnate

Unità n. 1

Unità n. 2

Unità n. 3

2.5 Criteri suggeriti per la valutazione globale e delle singole fasi**Testo italiano**

Organizzeremo due workshop alla fine delle due fasi. Si terranno inoltre due incontri, rispettivamente dopo 6 e dopo 18 mesi dall'inizio del progetto, per presentare le attività di ricerca, per verificare lo stato di avanzamento dei lavori, e promuovere la collaborazione tra i diversi gruppi.

I rapporti annuali e la descrizione dei prototipi offriranno informazioni utili per constatare sia la qualità scientifica del progetto, sia l'impatto della cooperazione.

Testo inglese

We will organize two workshops at the end of the two phases. Two other meetings will be held after 6 and 18 months to present the research activities, to assess the progress of the project and to promote collaboration.

The annual reports and description of prototypes will provide useful information to assess both the scientific quality of the project and the impact of the co-operation.

3.1 Spese delle Unità di Ricerca

Unità di Ricerca	Voce di spesa										TOTALE
	Materiale inventariabile	Grandi Attrezzature	Materiale di consumo e funzionamento	Spese per calcolo ed elaborazione dati	Personale a contratto	Servizi esterni	Missioni	Partecipazione / Organizzazione convegni	Pubblicazioni	Altro	
Unità n° 1	9.000	0	2.000	0	10.000	0	22.000	6.000	0	5.000	54.000
Unità n° 2	15.000	0	1.000	0	4.500	0	18.000	6.000	0	5.500	50.000
Unità n° 3	15.000	0	6.000	2.000	10.000	0	7.500	7.500	0	2.000	50.000
	39.000	0	9.000	2.000	24.500	0	47.500	19.500	0	12.500	154.000

3.2 Partecipazione finanziaria

Il coordinatore certifica che il progetto ha carattere di originalità e non è finanziato o cofinanziato da altre amministrazioni pubbliche (art. 4 bando 2003)

SI

3.3 Costo complessivo del Programma di Ricerca e risorse disponibili

Unità di Ricerca	Voce di spesa						
	RD	RA	RD+RA	Cofinanziamento di altre amministrazioni pubbliche	Cofinanziamento richiesto al MIUR	Costo totale del programma	Costo minimo
Unità n. 1	8.100	8.100	16.200	0	37.800	54.000	35.000
Unità n. 2	0	15.000	15.000	0	35.000	50.000	45.000
Unità n. 3	7.500	7.500	15.000	0	35.000	50.000	40.000
	15.600	30.600	46.200	0	107.800	154.000	120.000

	Euro
Costo complessivo del Programma	154.000
Fondi disponibili (RD)	15.600
Fondi acquisibili (RA)	30.600
Cofinanziamento di altre amministrazioni pubbliche (art. 4 bando 2003)	0
Cofinanziamento richiesto al MIUR	107.800

3.4 Costo minimo per garantire la possibilità di verifica dei risultati

Euro 120.000 (dal sistema, quale somma delle indicazioni dei Modelli B)

Euro 120.000 (dal Coordinatore del Programma)

(per la copia da depositare presso l'Ateneo e per l'assenso alla diffusione via Internet delle informazioni riguardanti i programmi finanziati; legge del 31.12.96 n° 675 sulla "Tutela dei dati personali")

Firma _____

Data 29/03/2003 ore 12:22