



Parole chiave:

*Agenti*

*Negoziazione*

*Contract Net Protocol*

*Multi-Agent Systems*

## ***RINGRAZIAMENTI***

*Desidero ringraziare la prof. Sonia Bergamaschi e l'ing. Jonathan Gelati per l'aiuto fornito durante la realizzazione di questa tesi.*

*Inoltre, ringrazio sinceramente la mia famiglia, per il loro sostegno durante tutto il mio percorso di studio.*

# Indice

<b>INTRODUZIONE.....</b>	<b>7</b>
<b>1-AGENTI E SISTEMI MULTI-AGENTE.....</b>	<b>9</b>
1.1 CHE COSA SONO GLI AGENTI?.....	9
1.2 PROPRIETA' DEGLI AGENTI.....	10
1.3 TIPOLOGIE DI AGENTI .....	11
1.4 GLI AGENTI E L'AMBIENTE.....	13
1.5 SISTEMI MULTI-AGENTE .....	13
1.6 INTEROPERABILITA' ATTRAVERSO GLI AGENTI DI MEZZO.....	16
1.7 AGENT COMMUNICATION LANGUAGES.....	17
1.8 LO STANDARD FIPA.....	18
<b>2-NEGOZIAZIONE FRA AGENTI INTELLIGENTI E PROTOCOLLI.....</b>	<b>22</b>
2.1 CARATTERISTICHE DELLA NEGOZIAZIONE.....	22
2.2 APPROCCI AL PROCESSO DI NEGOZIAZIONE .....	23
2.3 LA TEORIA DEI GIOCHI APPLICATA ALLA NEGOZIAZIONE .....	24
2.4 IL CONTRACT NET PROTOCOL.....	25
2.4.1 Vantaggi del CNP.....	27
2.4.2 Svantaggi del CNP.....	28
2.5 FIPA CONTRACT NET INTERACTION PROTOCOL.....	29
2.6 FIPA ITERATED CONTRACT NET INTERACTION PROTOCOL .....	33
2.7 FIPA CANCEL META-PROTOCOL .....	35
<b>3-PROTOCOLLI DI NEGOZIAZIONE ESTESI.....</b>	<b>37</b>
3.1 IMPLEMENTAZIONE DEL CONTRACT NET PROTOCOL BASATA SUL CALCOLO DEI COSTI MARGINALI.....	37
3.2 PROTOCOLLO DI NEGOZIAZIONE PER AGENTI BOUNDED RATIONAL SELF-INTERESTED.....	45
3.2.1. Protocollo multi-leveled commitment .....	47
3.2.2 Struttura dei messaggi di contrattazione.....	48
3.2.3 La razionalità limitata degli agenti.....	50

3.2.4 Nuovi approcci alla negoziazione.....	51
3.2.5 Tragedy of the Commons: il problema della congestione.....	52
3.3 PROTOCOLLO DI NEGOZIAZIONE TASK-SWAP.....	53
3.3.1 Utilità e costi marginali.....	55
3.3.2 Il ciclo di negoziazione.....	56
3.3.3 Strategia di negoziazione.....	59
3.4 OCMS-CONTRACT .....	60
<b>4-NUOVI MECCANISMI ALLA BASE DEI PROTOCOLLI DI</b>	
<b>NEGOZIAZIONE.....</b>	<b>65</b>
4.1 LA NEGOZIAZIONE MULTI-LEVELLED .....	65
4.2 PROTOCOLLI DI NEGOZIAZIONE MULTI-LINKED.....	70
4.3 PROTOCOLLI DI NEGOZIAZIONE MULTI-STEP.....	74
<b>5-CONCLUSIONI E LAVORO FUTURO.....</b>	<b>79</b>
<b>GLOSSARIO.....</b>	<b>81</b>
<b>BIBLIOGRAFIA.....</b>	<b>83</b>

## Indice delle Figure

Figura 1: Vista di una parte della tipologia di un agente.....	11
Figura 2 : Brokering vs. Matchmaking nei MAS.....	17
Figura 3 : Struttura Agent Platform.....	20
Figura 4 : Ciclo di vita di un agente.....	21
Figura 5 : Diagramma di sequenza del FIPA Contract Net Interaction Protocol.....	31
Figura 6 : Diagramma di sequenza del FIPA Iterated CNIP.....	34
Figura 7 : Diagramma di sequenza del FIPA Cancel Meta-Protocol.....	35
Figura 8 : Fasi di annuncio, di offerta e di assegnazione.....	40
Figura 9 : Struttara dei messaggi del contractor e del contractee.....	49
Figura 10 : Negoziazione strutturata su due livelli .....	68
Figura 11 : Protocollo di negoziazione cooperativa per l'allocazione dei task.....	76

## Introduzione

Lo studio dei meccanismi e dei principi di negoziazione all'interno di sistemi distribuiti, è da molti anni oggetto di ricerche e di studi da parte della disciplina scientifica dell'intelligenza artificiale. In particolare, all'interno della teoria degli agenti intelligenti, vengono studiate le capacità che queste entità software hanno di interagire fra di loro allo scopo di pervenire a determinati risultati.

La negoziazione, fra agenti intelligenti, può essere definita come l'insieme di attività ed operazioni (processo) che consentono a due o più a soggetti (negoziatori), ognuno con un proprio obiettivo da perseguire (funzione di utilità), di scambiarsi opinioni (offerte e contro-offerte) nel rispetto di un insieme di regole condivise (protocollo di negoziazione), con lo scopo di giungere ad un accordo finale. I protocolli svolgono quindi, all'interno della negoziazione, un ruolo fondamentale in quanto hanno il compito di coordinare e controllare le operazioni degli agenti coinvolti.

Il nostro lavoro si concentra sullo studio di protocolli di negoziazione per l'allocazione dei task, all'interno dei sistemi distribuiti multi-agente. Attraverso tali protocolli gli agenti interagiscono con l'obiettivo di minimizzare i costi totali derivanti dalla gestione dei loro task, e massimizzare la propria funzione di utilità. Lo scopo del nostro elaborato è quello di descrivere le tappe fondamentali del processo evolutivo dei protocolli di negoziazione precedentemente delineati, attraverso la descrizione di quelli che hanno contribuito in maniera più marcata alla definizione delle moderne tecnologie di negoziazione. In particolare, per ciascun protocollo, si descriveranno le tipologie di agenti intelligenti coinvolti nella negoziazione, la struttura dei messaggi scambiati e le fasi caratterizzanti del protocollo. Ampio spazio troveranno, inoltre, le descrizioni delle strategie proposte per far fronte ai problemi ed ai rischi derivanti dall'operare in ambienti aperti, come Internet, soggetti a repentini cambiamenti. La struttura del testo si sviluppa a partire dall'analisi del modello del *Contract Net Protocol (CNP)* di Smith definito nel 1980, fino a giungere alla definizione dei meccanismi di negoziazione che stanno alla base dei moderni protocolli. Durante questo percorso si delinearanno gli elementi di

innovazione dei vari protocolli e si confronteranno le diverse soluzioni proposte rispetto alle problematiche legate alla negoziazione.

Il testo include cinque capitoli di cui, di seguito, daremo una breve descrizione:

### **1. Agenti e sistemi multi-agente.**

Tale capitolo delinea, in maniera sintetica, gli elementi di base necessari alla comprensione del resto dell'elaborato. Vengono introdotti, infatti, i concetti di agente intelligente, di sistema multi-agente (*Multi-Agent-Systems, MAS*) e vengono descritti i linguaggi di comunicazione fra gli agenti. In tale sezione è presente inoltre la descrizione dell'ente di standard più autoritario nell'ambito della teoria degli agenti intelligenti: *Foundation for intelligent Physical Agents (FIPA)*.

### **2. Negoziazione fra agenti intelligenti e protocolli.**

Dopo una breve introduzione sul concetto di negoziazione, in tale capitolo viene definito il modello di funzionamento del CNP e di alcuni protocolli d'interazione FIPA.

### **3. Protocolli di negoziazione estesi.**

In tale capitolo si descrivono in maniera dettagliata i protocolli che hanno rappresentato le più significative estensioni del CNP. Viene introdotto il concetto di agente interessato a se stesso (*self-interested, SI*) e limitato razionalmente (*bounded rationality, BR*).

### **4. Nuovi meccanismi alla base dei protocolli di negoziazione.**

Il quarto capitolo descrive i nuovi meccanismi introdotti per la negoziazione fra agenti intelligenti, su cui si basano alcuni protocolli vengono presentati nel medesimo capitolo.

### **5. Conclusioni e lavoro futuro.**

In tale sezione trovano spazio alcune osservazioni sullo studio effettuato e vengono individuati gli elementi di un potenziale futuro lavoro di approfondimento.



# Capitolo 1

## Agenti e sistemi multi-agente

Lo scopo di questo primo capitolo è di fornire le conoscenze di base necessarie alla comprensione dei capitoli successivi. Le nozioni qui riportate, risultano importanti per riuscire a comprendere ed a contestualizzare il resto del testo. Di seguito si descriveranno gli agenti, le loro proprietà, i sistemi multi-agente e la struttura delle relative piattaforme.

Inoltre troverà ampio spazio la descrizione dell'ente di standard *Foundation for Intelligent Physical Agents* (FIPA), il quale oggi rappresenta un punto di riferimento per tutta la teoria degli agenti.

### 1.1 Che cosa sono gli agenti?

La *teoria degli agenti*, rappresenta una delle più interessanti teorie proposte nell'ambito degli studi riguardanti l'intelligenza artificiale, la quale studia il tentativo di riprodurre il comportamento umano all'interno di entità software. La definizione esatta di agente ha impegnato non poco i ricercatori del settore, dando corso ad un dibattito non ancora concluso. Esistono, infatti, varie definizioni, tra le quali di seguito si è scelto di riportare quella enunciata da Russel e Norving nel 1995, dove gli agenti sono definiti come:

*“entità capaci di percepire ed agire in maniera razionale, ovvero di eseguire compiti in modo giusto puntando alla massimizzazione della propria misura di prestazione, sulla base delle prove fornite da qualsiasi conoscenza predefinita o acquisita.”*

Gli agenti sono, quindi, entità computazionali (programmi, robot ecc...) in grado di agire in modo autonomo, prelevando informazioni dall'ambiente in cui si trovano, ed agendo secondo la propria base di conoscenze allo scopo di scambiare informazioni con altri agenti o con esseri umani e di prendere l'iniziativa per soddisfare i propri obiettivi.

## 1.2 Proprietà degli agenti

Di seguito si elencheranno le proprietà fondamentali che possono appartenere ad un agente:

- **Autonomia:** ossia la capacità del gruppo di operare senza direttive esterne, nel senso che esso è in grado di effettuare le proprie scelte e prendere le proprie decisioni senza l'intervento di un'entità sovra ordinata.
- **Capacità di comunicare:** capacità di comunicare con il mondo esterno, interagire con un utente o con altri agenti.
- **Reattività:** ovvero la capacità di reagire alle modifiche dell'ambiente che li circonda, ogni qual volta tali cambiamenti influenzino il loro obiettivo.
- **Nozioni mentali:** capacità di possedere conoscenze, ricordare esperienze, avere visioni proprie dell'ambiente che li circonda e sugli altri gruppi con cui collabora, a seconda del livello delle sue abilità sociali; inoltre essere in grado di apprendere dalle esperienze, interagendo con l'ambiente e con gli altri tramite rapporti sociali.
- **Persistenza:** capacità di permanere nel tempo, ciò significa che la durata della vita di un agente è superiore alla durata dei compiti che esegue di base. Un agente continua ad esistere con uno stato interno, in modo da poter eseguire interazioni successive, ma non è detto che la caratteristica di persistenza possa essere mantenuta in maniera indefinita. Tipicamente un agente sarà modellato definendo:
  - 1- Un compito durante il quale può compiere diverse interazioni finite.
  - 2- Delle risorse interne che possono essere riprodotte o consumate.
- **Vitalità:** è la capacità di sopravvivenza di un agente, nel senso che, l'agente può riuscire a risolvere situazioni anomale che lo porterebbero, altrimenti, in uno stato d'instabilità che comprometterebbe la sua persistenza.
- **Mobilità:** l'agente ha la capacità di variare i partner di comunicazione e di spostarsi in ambienti diversi da quello di partenza.
- **Abilità Sociale:** ossia la capacità di comunicare con gli altri agenti, magari cooperando nel perseguimento degli obiettivi comuni, scambiando informazioni e conoscenze.

- **Proattività:** capacità di generare eventi nell'ambiente circostante, iniziare delle interazioni con altri agenti, coordinare le attività di differenti agenti stimolandoli a produrre certe risposte.
- **Veridicità:** un agente non deve comunicare false informazioni.
- **Benevolenza:** un agente fa sempre ciò che gli si comanda.

Ovviamente un agente potrà anche non possedere alcune di queste proprietà. Per esempio, come vedremo in seguito, non tutti gli agenti sono entità benevole, ma possono assumere anche atteggiamenti non corretti nei confronti degli altri agenti.

### 1.3 Tipologie di agenti

Il compito di effettuare una distinzione tra le varie tipologie di agenti, risultata tutt'altro che semplice, poiché gli agenti possono essere distinti in base ad un'ampia varietà di caratteri. In primo luogo, si ritiene opportuno effettuare una prima distinzione in base al fatto che un agente possenga o meno le seguenti caratteristiche: autonomia, capacità di cooperare e d'imparare.

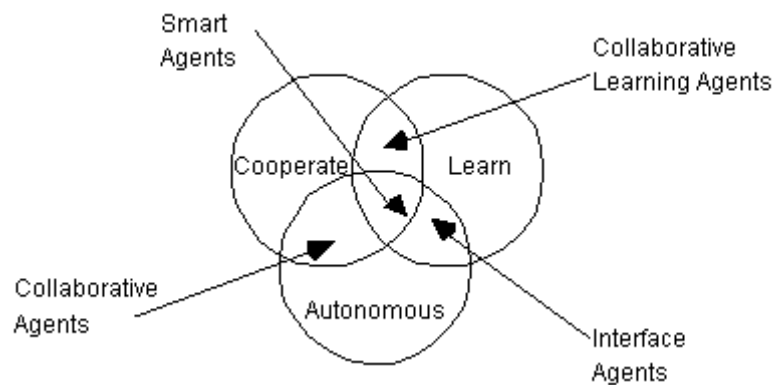


Fig.1 Vista di una parte della tipologia di un agente

La fig.1 mette in evidenza come, usando queste tre caratteristiche minime, sia possibile delineare quattro tipi di agenti: agenti collaborativi, agenti d'interfaccia, agenti imparanti di collaborazione ed agenti intelligenti. Tuttavia questa rappresenta

solo una parziale classificazione alla quale si devono aggiungere altri caratteri di distinzione quali, la mobilità, la reattività ecc...

In generale molti studiosi concordano nell'identificare sette tipologie principali di agenti:

- **Agenti collaborativi (*collaborative agents*):** sono agenti autonomi che hanno la capacità di cooperare con altri agenti; possono possedere o meno capacità di apprendimento. Sono strettamente legati ai sistemi multi-agente (vedi par. 1.5).
- **Agenti d'interfaccia (*interface agents*):** sono agenti che hanno il compito di gestire il dialogo con l'utente. Sono autonomi ed hanno capacità di apprendimento, ma in genere non comunicano con altri agenti. Spesso sono chiamati *assistenti personali (personal assistant)*.
- **Agenti mobili (*mobile agents*):** agenti in grado di migrare da un computer all'altro, attraverso la rete, durante la loro esecuzione. Rappresentano, una percentuale piuttosto piccola dell'intera teoria degli agenti.
- **Agenti delle informazioni (*informazione/internet agents*):** sono agenti per la gestione delle informazioni provenienti da varie fonti, spesso numerose ed eterogenee e prelevate da Internet tramite l'ausilio di motori di ricerca.
- **Agenti reattivi (*reactive agents*):** agenti dotati solamente di reattività e non di proattività. Non possiedono una rappresentazione interna simbolica dell'ambiente esterno, ma agiscono solo secondo un principio richiesta e risposta.
- **Agenti ibridi (*hybrid agents*):** sono agenti che combinano due o più delle tipologie sopra indicate.
- **Agenti intelligenti (*smart agents*):** sono dotati di autonomia, capacità di collaborazione e di apprendimento.

Quest'ultima tipologia, rappresenta l'ambito di interesse del nostro studio, in quanto gli agenti intelligenti sono i soggetti principali che interagiscono nei processi di negoziazione all'interno dei sistemi multi-agente.

## 1.4 Gli agenti e l'ambiente

Molte delle proprietà, precedentemente elencate, derivano dal fatto che un agente deve essere in grado di interagire con l'ambiente. Tale ambiente possiede le seguenti proprietà:

- è parzialmente accessibile e l'agente non ne può avere il controllo totale;
- è in continua evoluzione e subisce repentini cambiamenti indipendentemente dall'azione dell'agente stesso;
- una stessa azione eseguita da un agente può generare effetti diversi se applicata in tempi diversi.

## 1.5 Sistemi multi-agente

Come già affermato precedentemente, gli agenti sono entità software specializzate che si comportano autonomamente a nome dei loro utenti, attraverso ambienti aperti e distribuiti, per risolvere un numero crescente di problemi complessi. Impiegati singolarmente, tuttavia, gli agenti sono destinati a diventare virtualmente inutili. Le applicazioni richiedono delle comunità di agenti che possano agire individualmente e in modo interdipendente.

Un *sistema multi-agente* (*Multi-Agent Systems, MAS*) è un sistema in cui gli agenti intelligenti interagiscono e coesistono tra di loro per soddisfare un certo insieme di obiettivi, allo scopo di portare a termine un certo insieme di compiti. I MAS stanno acquisendo sempre più importanza in ambiti quali l'Ingegneria del Software, le discipline scientifiche e l'innovazione tecnologica. Tale importanza, risiede nelle loro capacità di:

- Risolvere problemi in sistemi altamente distribuiti che sono troppo ampi affinché un unico agente intelligente possa normalmente risolverli, questo a causa di risorse limitate o comunque del fatto che un agente, singolarmente, potrebbe diventare un "collo di bottiglia" nell'elaborazione dell'informazione, rischiando perfino di collassare.
- Permettere l'interconnessione e l'interoperabilità dei sistemi esistenti che necessitano di manutenzione a medio e lungo termine; il modo migliore per

mantenere tali sistemi efficienti, è di riuscire ad incorporarli all'interno di una più ampia comunità di agenti che provvedano ad aggiornarli.

- Fornire soluzioni a problemi che trovano facilmente assolvimento se affrontati da società di agenti interagenti ed autonomi. Per esempio, problemi di programmazione degli eventi, di controllo del traffico aereo, di commercio elettronico ecc...
- Fornire soluzioni che usano efficientemente sorgenti d'informazione distribuite nello spazio, come ad esempio reti di sensori, raccolta informazioni su Internet ecc...

Tutte le capacità possedute da un MAS, sopra elencate, sono strettamente legate alle sue caratteristiche principali. Tali caratteristiche riportate in [2] sono:

- *Efficienza computazionale*: deriva dal carattere altamente distribuito di questi sistemi.
- *Robustezza*: la capacità di sostenere l'imprevedibilità intrinseca di ambienti aperti e complessi nei quali gli agenti interagiscono scambiandosi informazioni.
- *Affidabilità*: la capacità di sopperire ad errori o problemi nei componenti.
- *Estensibilità*: deriva dal fatto che, le capacità di un agente operante su un problema, possono essere modificate.
- *Manutenibilità*: deriva dal fatto che, un sistema di agenti è modulare.
- *Reattività*: dovuta al fatto che, sempre grazie alla modularità, gli agenti sono in grado di trattare anomalie in modo locale, senza propagarle all'intero sistema.
- *Flessibilità*: è dovuta alla capacità degli agenti di avere differenti possibilità di organizzarsi in maniera attiva per risolvere un problema.
- *Riusabilità*: agenti con determinate funzionalità, possono offrire i propri servizi ad altri gruppi di agenti per la risoluzione di diversi problemi

Sia un MAS, che un singolo agente, sono caratterizzati da una propria infrastruttura. L'infrastruttura di un MAS può essere definita come un insieme di servizi, di convenzioni e di conoscenze, che supportano complesse interazioni fra gli agenti [1]. Gli agenti, infatti, hanno bisogno di servizi che gli consentano di comunicare con

altri agenti in ambienti aperti come Internet, e che nello stesso tempo possano assicurare il rispetto delle costrizioni necessarie alla sicurezza dell'interazione. E' importante sottolineare il rapporto che lega l'infrastruttura di un singolo agente, con l'infrastruttura di un MAS al quale gli agenti partecipano: quest'ultima, viene considerata come un dominio indipendente sul quale MAS, servizi e componenti sono in grado di vivere, comunicare, interagire ed interoperare, mentre la singola infrastruttura dell'agente è la parte generica di quest'ultimo che gli consente di essere parte di una società multi-agente.

Dal punto di vista dell'infrastruttura del MAS, gli agenti sono quindi programmi "socialmente consapevoli" che comunicano ed interagiscono tra di loro e con i componenti dell'infrastruttura e, il quale comportamento, è conforme alle regole del MAS. Tutte le capacità di risoluzione di problemi di un agente risultano, invece, come una scatola nera per l'infrastruttura.

Negli ultimi anni, l'emergente importanza e l'utilizzo sempre più diffuso dei MAS, ha spinto alcune industrie e vari gruppi di ricerca ad orientare parte dei loro studi verso la standardizzazione della tecnologia multi-agente. Di seguito saranno riportati alcuni modelli:

- L'*Object Manager Group* (PMG), ha proposto un modello orientato allo sviluppo della tecnologia degli agenti, basato principalmente sulle caratteristiche di un ambiente composto da entità che collaborano usando modelli e politiche d'interazione generale.
- Il gruppo di ricerca del *Knowledge-able Agent Oriented System* (KAOS), ha proposto un modello basato su un'architettura aperta e distribuita per agenti del software, dando un grande contributo alla standardizzazione.
- La *General Magic* è un'attività commerciale che modella un MAS come un mercato elettronico in cui, consumatori e fornitori di beni e servizi, possono incontrarsi e realizzare transazioni commerciali.
- Infine, ricordiamo il modello della *Foundation for Intelligent Physical Agents* (FIPA), il quale si basa sulla specifica dell'ambiente dove gli agenti si trovano ad operare, e della loro piattaforma, la quale descrive un'infrastruttura per lo sviluppo e l'interazione degli agenti stessi. Una

digressione più approfondita sul modello di MAS proposto da FIPA verrà presentata in seguito.

## **1.6 Interoperabilità attraverso gli agenti di mezzo**

Come visto in precedenza, gli agenti di un MAS comunicano allo scopo di conseguire nel miglior modo possibile i propri scopi o quelli del sistema a cui appartengono. Tale comunicazione consiste, essenzialmente, nella condivisione o nello scambio d'informazioni e di servizi. All'interno di tali interazioni è possibile individuare tre categorie di agenti che differiscono per il ruolo svolto: fornitori di servizi (*service providers*) ovvero gli agenti che forniscono un determinato servizio, agenti che richiedono servizi (*service requeste*) e gli agenti di mezzo (*middle agents*) [1]. Quest'ultimi sono gli agenti che coordinano le attività fra gli agenti che forniscono e richiedono un servizio in Internet. Il loro principale obiettivo è quello di fornire un servizio base di mediazione tra le società di agenti, di coordinare questo servizio in accordo con un determinato protocollo, convenzione e politica, ed assicurare un certo livello di qualità, sicurezza e affidabilità. Il processo base, supportato da un agente di mezzo, si svolge nel seguente modo:

- a. l'agente fornitore pubblicizza le sue capacità (*capabilities*) all'agente di mezzo, che opportunamente si preoccupa d'immagazzinarle;
- b. l'agente richiedente domanda all'agente di mezzo di individuare l'agente che fornisce un determinato servizio;
- c. l'agente di mezzo elabora la richiesta e ritorna il risultato;

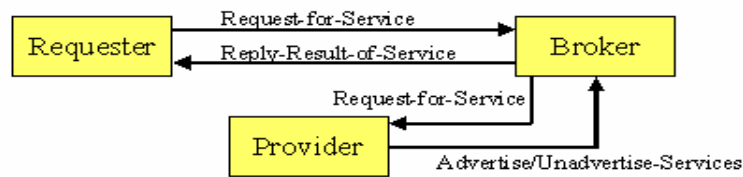
La complessità di tale meccanismo di comunicazione, risiede principalmente nel fatto che ambienti aperti come Internet sono in continua evoluzione, ciò comporta cambiamenti dinamici sia rispetto al contenuto delle risorse che alle loro locazioni. Inoltre ricordiamo che in Internet operano differenti comunità di utenti e società di agenti che inseguono il loro obiettivo il quale spesso rischia di entrare in conflitto con quello di altri. La coordinazione delle attività di mediazione tra le società di agenti, può essere realizzata attraverso due principali vie: *brokering e machmaking*. In [4] i servizi di brokering e matchmaking sono descritti in maniera più dettagliata.



I MAS trovano ampi ambiti di applicazione, tra questi ricordiamo: il recupero e filtraggio delle informazioni, lavori collaborativi, supporto decisionale, mercato elettronico, gestione di servizi interattivi e d'intrattenimento, gestione e supervisione delle reti, automazione di processi, robotica, modellazione e simulazione di sistemi complessi, negoziazione automatica ecc...

### Service Brokering vs. Matchmaking in Multi-Agent Systems

#### Brokering



#### Matchmaking

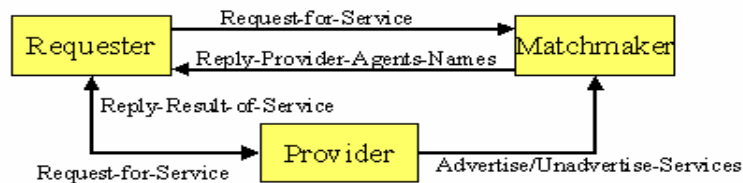


Fig.2 Brokering vs. Matchmaking nei MAS

## 1.7 Agent Communication Languages

Così come nelle società reali degli esseri umani, l'esigenza di un mezzo comune di comunicazione, è essenziale anche fra le società di agenti affinché questi riescano a cooperare ed ad interagire fra loro. Si è venuta così a creare la necessità di una lingua che consenta di comunicare non solo all'interno di un singolo dominio, ma bensì sull'intera rete di Internet. Tale esigenza ha persuaso gli sforzi di alcuni enti di standard a cercare di elaborare un linguaggio di comunicazione comune fra gli (*Agent Communication Languages, ACL*) [3], tra i principali ricordiamo il KQML e il FIPA ACL. Un ACL è un insieme di primitive che consentono ad un agente di dichiarare la propria intenzione. Le primitive sono azioni (*performatives*) che l'agente può usare nel tentativo di comunicare ad altri agenti le proprie intenzioni. Si può quindi affermare che, in un certo senso, l'ACL fornisce un modello concettuale

ad alto livello delle strategie di comportamento dell'agente. La comunicazione in un ACL avviene attraverso messaggi che vengono anche chiamati atti comunicativi (*speech act*) e che si basano sulla teoria del discorso diretto.

Tutti gli ACL forniscono un insieme predefinito di atti comunicativi. In KQML, tali atti sono denominati *performatives*, mentre in FIPA ACL semplicemente *speech act*. Un'altro elemento fondamentale dell'ACL, e di un linguaggio di comunicazione in generale, è la sua ontologia. L'ontologia di un linguaggio rappresenta il suo vocabolario ed è indispensabile allo scopo di uniformare, sotto il punto di vista semantico, la comunicazione fra gli agenti.

KQML e FIPA ACL, se pur con alcune differenze, coincidono concettualmente. Il KQML fu elaborato nel 1993, e pur essendo ancora abbastanza diffuso sta cedendo sempre più il passo al molto più evoluto FIPA ACL.

## **1.8 Lo standard FIPA**

La Fondazione per gli Agenti Fisici Intelligenti è un'organizzazione senza scopo di lucro nata a Ginevra nel 1996, composta da varie società ed università (tra le quali giganti industriali, come NEC, Alcatel, NHK e Siemens), con l'obiettivo di definire uno standard riguardante le tecnologie degli agenti. FIPA ha sviluppato una serie di specifiche riguardanti la tecnologia degli agenti, alle quali possono essere associati i seguenti stadi: preliminare, sperimentale, standard, deprecato o obsoleto. Alcune di queste specifiche sono normative, ovvero obbligatorie per ogni implementazione, altre solo informative.

Le specifiche che FIPA rilascia, sono solitamente classificate per argomenti. E' possibile individuare le seguenti categorie:

- a. *Abstract Architecture*: definisce in modo astratto le entità che compongono un sistema di agenti, concentrandosi principalmente sulla loro interoperabilità.
- b. *Agent Communication*: definisce la struttura del linguaggio di comunicazione fra gli agenti (FIPA ACL); specifica inoltre i più importanti fra i protocolli d'interazione, atti comunicativi e linguaggi usati per rappresentare i contenuti della comunicazione.

- c. *Agent Management*: definisce l'infrastruttura generica sulla quale gli agenti FIPA possono interagire.
- d. *Agent Message Transport*: definisce le modalità di trasmissione dei messaggi scambiati, descrivendo diversi protocolli di trasporto.
- e. *Applications*: descrive ed illustra alcuni esempi di applicazioni concrete di sistemi ad agenti.

Secondo FIPA[5], un agente è un “*processo computazionale che implementa le funzionalità di autonomia e comunicazione di un'applicazione*“. Un agente è l'attore fondamentale di un *Agent Platform* (AP), la quale, è definita come l'ambiente integrato sul quale gli agenti possono esistere ed interoperare. Per potersi distinguere fra loro gli agenti posseggono delle identità. Tali identità sono realizzate attraverso un identificatore di agente (*Agent Identifier*, AID), il quale rappresenta una specie di “*etichetta di riconoscimento*“. Oltre al loro nome, l'AID contiene altri parametri come, per esempio, gli indirizzi dove l'agente è situato fisicamente (su quale host, su quale *piattaforma ecc.*).

Una piattaforma ad agenti fornisce l'infrastruttura fisica sulla quale gli agenti possono interagire. Un'AP è costituita da diverse componenti fra cui, oltre ovviamente all'hardware ed al sistema operativo, da un software di supporto agli agenti, dai componenti per la gestione degli agenti (DF, AMS and MTS) e dagli agenti stessi. I componenti per la gestione degli agenti precedentemente nominati sono:

- Il *Directory Facilitator* (DF) è un componente opzionale di un'AP. Fornisce il servizio di pagine gialle agli altri agenti: gli agenti possono, attraverso il DF, registrare i servizi che sono in grado di fornire o, eventualmente, richiedere di individuare quali servizi gli altri agenti mettono a disposizione e possono offrire.
- L'*Agent Management System* (AMS) è un componente obbligatorio di un'AP. Esiste solo un AMS in un'AP, il quale svolge il ruolo di supervisore degli accessi e dell'utilizzo dell'AP. Inoltre, in esso è presente un direttorio di AID, il quale contiene gli indirizzi di trasporto degli agenti registrati presso la piattaforma. Un AMS offre, quindi, il servizio di pagine bianche agli agenti.

- Il *Message Transport Service (MTS)* è il metodo di comunicazione di default tra gli agenti appartenenti a diverse AP.

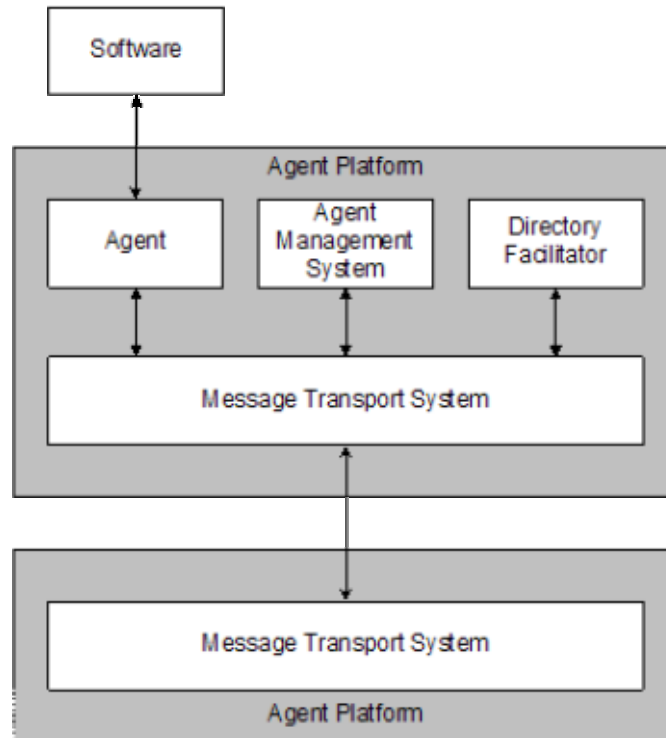


Fig. 3: Struttura di un agent platform

Gli agenti FIPA esistono fisicamente su un'AP e utilizzano i servizi che questa gli offre per realizzare le proprie funzionalità. In tale ambiente, l'agente risulta essere un processo software fisico, che possiede un proprio ciclo di vita. Tale ciclo di vita è rappresentato nella vita fig.4 attraverso il passaggio fra vari stati, che l'agente può assumere durante tale ciclo di vita.

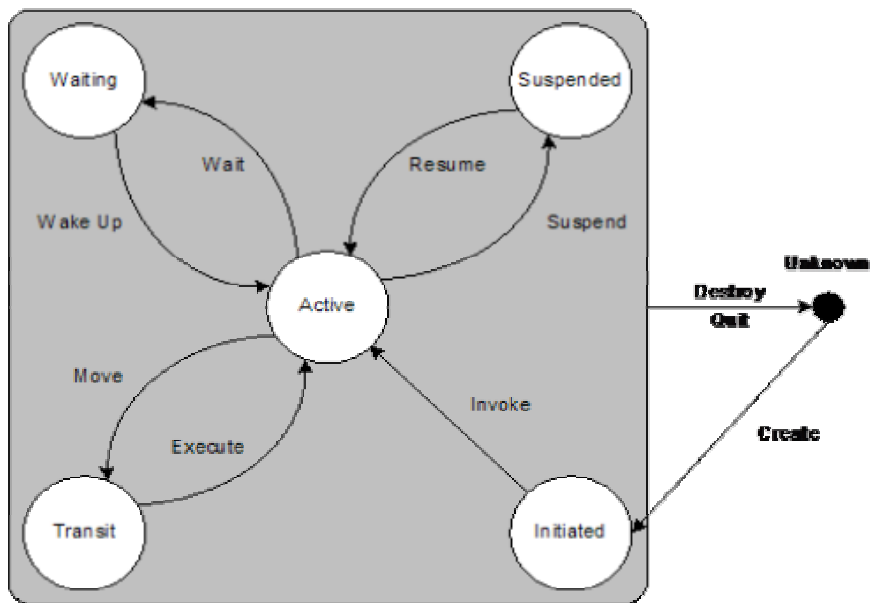


Fig.4 Ciclo di vita di un agente

## Capitolo 2

### Negoziazione fra agenti intelligenti e protocolli

La negoziazione è l'insieme di attività ed operazioni (processo) che permettono a due o più soggetti (negoziatori), ognuno con un interesse economico privato da perseguire (funzione di utilità), di scambiarsi opinioni (offerte e contro-offerte) nel rispetto un insieme di regole condivise (protocollo di negoziazione) con lo scopo di arrivare ad un accordo finale. I negoziatori hanno la capacità di valutare e formulare offerte e di prendere decisioni in base ad una propria strategia ed ad una propria funzione di utilità. Il protocollo di negoziazione ha il compito di coordinare gli agenti e di controllare il rispetto delle regole di negoziazione. Di seguito si analizzeranno protocolli di base per l'allocazione dei task.

#### 2.1 Caratteristiche della negoziazione

La negoziazione è definita sulla base delle seguenti caratteristiche:

- *Protocolli di negoziazione*: sono l'insieme di regole che governano l'interazione [6]. Queste definiscono i tipi di soggetti che possono prendere parte alla negoziazione, lo stato (accettare un offerta, chiudere la negoziazione), gli eventi che causano il cambiamento dello stato e azioni corrette dei partecipanti in particolari situazioni.
- *Oggetti o elementi di una negoziazione*: insieme dei problemi sui quali bisogna raggiungere un accordo. L'oggetto può contenere un singolo parametro (prezzo) oppure centinaia (qualità, tempo, termini, condizioni, ecc..)
- *Modelli decisionali degli agenti*: i partecipanti agiscono seguendo il protocollo di negoziazione per realizzare i loro obiettivi. La complessità del modello dipende dal protocollo di negoziazione, dall'oggetto della negoziazione, dal tipo di decisioni da prendere e dalle operazioni da eseguire.

In alcuni casi il processo di negoziazione avviene attraverso lo scambio di *proposte*, *critiche*, *spiegazioni*. La proposta può rappresentare una soluzione completa o parziale di un problema affrontato da un agente. Inoltre, può essere basata su commenti fatti da altri agenti. Un esempio di una semplice proposta è che un determinato agente richieda che gli venga procurato un determinato servizio. Una risposta può essere formulata in modo più complesso, quindi può contenere delle condizioni da rispettare. Un agente che riceve una proposta può rispondere in due modi differenti: il primo consiste nel fare una critica che può rappresentare l'approvazione o meno della proposta, o un commento alla stessa. La seconda consiste nel formulare una contro proposta che si basa sulla richiesta fatta. Le spiegazioni sono informazioni aggiuntive che chiariscono proposte e contro-proposte.

## **2.2 Approcci al processo di negoziazione**

All'interno dei processi di negoziazione è possibile distinguere differenti tipologie d'approccio.

I principali approcci identificati sono i seguenti:

- ***Approccio euristico per negoziazioni uno-a-uno***. Questo approccio è basato su un particolare tipo di negoziazione detto "*service-oriented negotiation*". Un agente (cliente) richiede un servizio ad altri agenti (fornitori). La negoziazione si sviluppa determinando un contratto sotto certi termini e condizioni. L'offerta e la contro offerta possono ripetersi fino al raggiungimento di un accordo. Questo modello permette ad un agente di valutare il miglior compromesso fra i risultati ottenuti derivanti dalla negoziazione. E' utilizzato per lo sviluppo di sistemi di negoziazione ad agente per i processi di gestione del business.
- ***Approccio euristico per negoziazioni per aste multiple***. In questo caso, si considerano centinaia di postazioni che conducono diversi tipi di aste in linea. L'agente intelligente assisterà a tali aste visitandone i siti. Il ruolo dell'agente intelligente è cercare il miglior affare, rimanendo in un tempo accordato.

- ***Approcci basati sulle argomentazioni (Argumentation Based)***. In molte applicazioni ad agenti, ognuno di essi ha bisogno di interagire con altri per ottenere informazioni, risorse, servizi o per trovare accordi su certe questioni. Il meccanismo più diffuso per gestire queste relazioni è la negoziazione, per la quale i partecipanti si scambiano proposte per cercare di raggiungere accordi mutuamente accettabili. Questo tipo di approccio non si limita solo a questo, ma fa sì che gli agenti inviino informazioni addizionali che possono essere cambiate in relazione alle proposte. Ovvero, in aggiunta ad un rifiuto di una proposta, l'agente può offrire una critica e spiegare perché essa non sia accettabile, allo stesso modo un'altro agente può accompagnare la propria proposta con le motivazioni per le quali dovrebbe essere scelta. Quindi, gli agenti cercano di costruire *argomenti (arguments)*, attraverso i quali illustrano le loro opinioni, in modo da rendere più accettabili le proposte. Gli argomenti cercano di identificare le opportunità per gli scambi, di crearne nuove o di modificare i criteri di valutazione esistenti. Essi hanno il potenziale di incrementare le probabilità e/o la velocità del raggiungimento degli accordi.

### **2.3 La Teoria dei Giochi applicata alla negoziazione**

La ricerca nel ramo *dell'Intelligenza Artificiale Distribuita (Distributed Artificial Intelligence, DAI)* affronta il problema della progettazione di sistemi intelligenti automatizzati capaci di interagire in modo efficace. A questo proposito vari ricercatori si sono occupati di studiare la costruzione di agenti intelligenti adatti a negoziare tra loro combinando tecniche di intelligenza artificiale con metodi e tecniche derivanti da diversi campi che esaminano i comportamenti di multi-entità. In situazioni in cui si presentano problemi relativi all'allocazione di compiti e di risorse da condividere, si consiglia l'applicazione di tecniche della teoria dei giochi. La teoria dei giochi studia modelli matematici di conflitti e cooperazione tra persone. Questi modelli sono rappresentazioni astratte di classi di situazioni della vita reale che coinvolgono individui che hanno differenti obiettivi e preferenze. I modelli della teoria dei giochi [7] sono divisi in due tipi principali: modelli *non-*



*cooperativi*, nei quali le entità attive, i *giocatori (players)*, agiscono individualmente, e modelli *cooperativi*, nei quali i giocatori agiscono in gruppo.

Queste tecniche possono essere usate come base per i protocolli d'interazione fra agenti. Gli agenti automatizzati possono essere modellati come giocatori “non-cooperativi”, se operano per se stessi, o come giocatori “cooperativi”, se i compiti richiedono che gruppi di agenti lavorino insieme. L'uso di tecniche della teoria dei giochi richiede che il processo di negoziazione possa contare su una computazione consistente e una capacità di comunicazione.

Ci sono situazioni in cui la negoziazione avviene attraverso un largo numero di agenti (World Wide Web). Altre ricerche nell'ambito del DAI hanno considerato agenti automatizzati cooperativi, i quali lavorano insieme per raggiungere obiettivi. Ciò ha portato allo sviluppo di protocolli per la cooperazione. In questi casi il numero di agenti non è grande, possono comunicare ed hanno capacità computazionali. E' possibile trovare agenti automatizzati che necessitano di agire con altri agenti in ambienti non strutturati (multimedia, vendita sul WWW). Qui gli agenti che agiscono per interesse personale, non sono in numero elevato, possono comunicare ed utilizzano i modelli informali di comportamento.

## **2.4 Il Contract Net Protocol**

Il protocollo più conosciuto e diffuso è il *Contract Net Protocol (CNP)* sviluppato nel 1980 da Smith [8] e in anni successivi in collaborazione con Davis. Tale protocollo, specifica le interazioni fra gli agenti per realizzare una negoziazione automatizzata e competitiva attraverso l'utilizzo di contratti, per l'allocazione dei task [9], [10]. Esso si basa su un'architettura di nodi indipendenti ognuno dei quali possiede risorse limitate, che possono essere comunicate a tutti gli altri nodi attraverso un efficiente protocollo di negoziazione. Quest'ultimo è indipendente dall'architettura usata. All'interno del CNP possiamo identificare due tipi di agenti: il *manager* e il *contractor*. Il manager è un agente che desidera far eseguire i propri task ad altri agenti. Il contractor, viceversa è un agente che esegue task per conto di altri agenti. Il CNP deve risolvere due questioni fondamentali: primo, il modo in cui un determinato nodo (manager), possa trovare un altro nodo in quel momento inattivo (contractor), in grado di soddisfare un determinato task o sub-task in quel

determinato istante; in secondo luogo, come può un nodo (contractor), trovare una richiesta appropriata che può essere soddisfatta negli istanti in cui tale nodo si trovi ad essere libero. Queste problematiche, inglobano al loro interno, sia problemi di connessione che d'identificazione. Il CNP si basa su una negoziazione caratterizzata dai seguenti elementi:

- Il processo è locale in quanto non vi è un'unità centrale di controllo; ciò implica che ogni nodo (agente) può agire indifferentemente da manager o da contractor.
- Esistono due direzioni di scambio delle informazioni.
- Ogni nodo, che prende parte alla negoziazione, valuta le informazioni in modo soggettivo.
- L'accordo finale, è raggiunto attraverso una *mutua selezione*.

Inoltre, risulta essere più adatto ad ambienti caratterizzati dalle seguenti proprietà:

-I task sono precisi ed espressi in modo gerarchico.

-I task possono essere decomposti in più sub-task.

-Tali sub-task risultano essere mutuamente indipendenti.

Il CNP si basa sullo svolgimento di alcune fasi principali:

1. *Annuncio di un task*. Quando il manager percepisce la necessità di delegare l'esecuzione di un compito, comunica tale necessità agli altri agenti attraverso un annuncio il quale descrive il task e i requisiti che il potenziale contractor deve possedere.
2. *Offerta*. Riguarda la comunicazione da parte di un agente al manager, della propria disponibilità nello svolgimento del task.
3. *Assegnazione*. Fa riferimento alla selezione effettuata dal manager, riguardo all'agente che si aggiudicherà il contratto per l'esecuzione del task.
4. *Avvenuta ricezione*. Rappresenta la fase finale del protocollo ed è caratterizzata dall'accettazione o dal rifiuto del task da parte del contractor.
5. *Report*. Riguarda le comunicazioni che il contractor rivolge al manager, per informarlo sulla conclusione del compito e sul suo stato attuale.
6. *Licenziamento*. Fa riferimento alla conclusione del protocollo, in maniera unilaterale da parte del manager.

7. *Disponibilità*. Rappresenta la dichiarazione di un agente circa la propria mancanza d'occupazione e la conseguente disponibilità ad eseguire task.

Durante il processo di negoziazione si ha il susseguirsi delle fasi precedentemente descritte. In particolare, in situazioni in cui il numero delle interazioni correnti fra gli agenti è accettabile, sarà il manager ad iniziare la negoziazione attraverso la fase di "annuncio di un task". Nel caso in cui, viceversa, il numero di interazioni correnti sia elevato sarà il contractor a dare inizio alla negoziazione attraverso la fase di "disponibilità". Tale meccanismo ha lo scopo di evitare la congestione delle comunicazioni che potrebbe essere causata da un elevato invio di annunci.

Il primo ambito d'applicazione del CNP, riguardò la simulazione di una rete distribuita di sensori acustici. Successivamente, fu applicato a lavori di distribuzione dei compiti fra macchinari all'interno di un piano di produzione, all'allocazione di lavori computazionali tra processori in una rete, e alla programmazione distribuita. In queste applicazioni, gli agenti erano totalmente cooperativi e la selezione di un contractor era basata semplicemente sulla sua attinenza e sulla sua capacità di esecuzione. Per queste applicazioni, non furono proposti modelli formali che formalizzassero le modalità con le quali venivano prese le decisioni di proposta, di offerta e di soddisfacimento di un obiettivo. Gli annunci, le offerte o la stipulazione di un contratto, non erano basati su principi reali di microeconomia come per esempio il prezzo per l'esecuzione di un determinato task o la quantità di domanda.

#### **2.4.1 Vantaggi del CNP**

L'utilizzo del CNP può risultare vantaggioso su molti aspetti, se paragonato con altre strategie di negoziazione. Tali vantaggi sono elencati di seguito:

- I task sono assegnati dinamicamente, il che implica una migliore distribuzione dei task fra gli agenti coinvolti.
- Gli agenti possono entrare o lasciare il sistema, ogni qual volta lo ritengano opportuno.
- I task, risultano essere naturalmente bilanciati fra gli agenti poiché coloro che hanno già stipulato contratti, non effettueranno offerte per l'esecuzione di altri task. Infatti, avendo a disposizione risorse limitate, gli agenti non possono accettare altri contratti finché non hanno portato a termine quelli già stipulati.

### 2.4.2 Svantaggi del CNP

Sfortunatamente, ai vantaggi precedentemente illustrati, corrispondono altrettanti mancanze evidenziate dal CNP. Queste mancanze, rischiano di causare gravi inefficienze, o addirittura l'intero breakdown del sistema. Tali mancanze sono le seguenti:

- Il CNP, assume che tutti gli agenti siano entità benevole, e che mostrino comportamenti amichevoli se integrati in una società. Tale assunzione comporta, inevitabilmente, la mancanza di un meccanismo d'identificazione, prevenzione ed eventuale risoluzione dei conflitti che possono crearsi fra gli agenti di un sistema. Gli agenti, infatti, sono entità che *self-interested (SI)*, cioè mirano a soddisfare i propri obbiettivi nel modo più efficiente possibile, indipendentemente dagli interessi degli altri agenti. Ciò sta a significare che la soluzione finale al loro scopo è la migliore per gli agenti coinvolti ma non per l'intera società di agenti.
- L'infrastruttura della comunicazione non è completamente affidabile. Come abbiamo già detto, quest'ultima si basa su una rete complessa di nodi e di collegamenti fra di questi. Nel caso in cui, per errore, un collegamento fallisca si rischia di creare gruppi di agenti isolati dal resto del sistema e perciò, praticamente inutili. Il sistema dovrebbe avere un meccanismo di restaurazione della comunicazione in caso di fallimento di quest'ultima. Se la comunicazione non potesse essere ristabilita, gli agenti coinvolti dovrebbero essere recuperati.
- Se il tempo limite per le operazioni è troppo basso, il contractor rischia di non avere tempo a sufficienza per inviare una proposta. In tal modo il manager può perdere offerte molto vantaggiose se queste non sono inviate abbastanza velocemente.
- Un'altro argomento di discussione, concerne il time-bound framework. Un partecipante può spedire un'offerta come risposta, ad un annuncio per l'esecuzione di un task. Può accadere, che mentre il contractor attende la risposta alla sua offerta, riceva un nuovo annuncio da un altro agente. Avendo a sua disposizione risorse limitate, il contractor può decidere di

ignorare i nuovi annunci, perdendo così un eventuale potenziale contratto. Viceversa il contractor potrebbe decidere di considerarli, rischiando in tal modo di ottenere due contratti, avendo a disposizione risorse per tener fede ad uno solo.

Tale problema può essere risolto attraverso l'applicazione di tre tipi di protocolli: *Nothing-Guaranteed Protocol (NGP)*, *Acceptance-Guaranteed Protocol (AGP)* e *Finite-time Guaranteed Protocol (FGP)*. Attraverso il NGP, il contractor può valutare altri annunci anche se ha una o più offerte in sospeso. In tal modo non vi è alcuna garanzia che il contractor esegua effettivamente il task contrattato. In tal caso è previsto un messaggio attraverso il quale viene confermata o meno l'esecuzione del task al manager. Viceversa nell'AGP, il contractor non considera i nuovi annunci, l'offerta da lui fatta è così garantita, ma in caso di rifiuto di quest'ultima, l'agente perde l'opportunità di nuovi contratti. A mediare fra queste due soluzioni limite, vi è il FGP il quale definisce un tempo massimo, oltre il quale se il manager non risponde all'offerta del contractor, quest'ultimo è autorizzato a considerare i nuovi annunci eventualmente ricevuti. Il tempo di fine può essere deciso in base a fattori come le circostanze in cui si trova ad operare il contractor, il tempo necessario al manager per rispondere ecc...

## **2.5 FIPA Contract Net Interaction Protocol**

L'utilizzo sempre più diffuso dei protocolli di negoziazione, e l'esigenza di uno standard a cui poter far riferimento, ha spinto FIPA a sviluppare una serie di specifiche standard che descrivono alcuni protocolli di negoziazione base, ispirati dal CNP originariamente proposto da Smith.

Il FIPA Contract Net Interaction Protocol (CNIP) [11], rappresenta la più fedele estensione dell'originale CNP.

Nell'IP un agente chiamato *iniziatore (initiator)* svolge il ruolo di manager, ovvero di colui che desidera far eseguire i propri task ad altri agenti (uno o più). Tali agenti prendono il nome di *partecipanti (participant)* e svolgono il ruolo di contractor. Nel fare eseguire, i propri task ai partecipanti, l'iniziatore contemporaneamente mira ad ottimizzare la funzione che li caratterizza. Tale caratteristica è funzione di un parametro, come il prezzo oppure il tempo di esecuzione. Per un dato task, ogni

partecipante ha la possibilità di rispondere con una proposta o con un rifiuto. Ovviamente la negoziazione continuerà solamente con i partecipanti che hanno presentato una proposta.

L'iniziatore comunica ai partecipanti l'utilizzo di tale protocollo assegnando al parametro "*protocol*" del messaggio ACL il valore di "fipa-contract-net". Ovviamente, in tutta la specifica FIPA, viene sottinteso l'utilizzo del linguaggio di comunicazione FIPA ACL.

Il protocollo d'interazione è composto da quattro fasi principali. Gli agenti, attraverso il susseguirsi di queste ultime dovranno riuscire a negoziare i vari contratti. Tali fasi sono:

- a. L'iniziatore spedisce una *Call for Proposals* (CFP), ovvero una richiesta di proposte per eseguire il task.
- b. Ogni partecipante analizza la CFP e se lo ritiene opportuno, fa una proposta compatibilmente con le specifiche della CFP.
- c. Conseguentemente l'iniziatore sceglie la migliore fra le offerte pervenutegli, e assegna il contratto a quest'ultima.
- d. L'iniziatore si preoccupa di rifiutare le altre proposte pervenutegli.

La fig.5 illustra, attraverso un diagramma di sequenza, il CNIP.

Di seguito descriveremo in maniera più approfondita le fasi del protocollo precedentemente elencate. L'iniziatore invia  $m$  proposte ad altrettanti agenti, attraverso l'utilizzo del messaggio di CFP. Tale messaggio consente di specificare le condizioni imposte dall'iniziatore sull'esecuzione dei task come per esempio, il prezzo e la terminazione (*deadline*). I partecipanti ricevono le CFP e, dopo averle analizzate e valutate come potenziali contratti, generano  $n$  risposte. Di queste  $n$  risposte,  $j=n-i$  sono proposte (*proposal*) per l'esecuzione del task, specificate attraverso i *propose acts* e includenti ulteriori condizioni sull'esecuzione dei compiti. I rimanenti  $i=j-n$  partecipanti rifiutano il task attraverso un messaggio di rifiuto (*refuse*). Le motivazioni che possono spingere un agente a rifiutare sono varie: per esempio l'agente potrebbe non essere in grado di eseguire il compito richiesto, oppure avere la possibilità di contrattare ad prezzo più alto. Al termine del processo, l'iniziatore valuta tutte le  $j$  proposte pervenute e seleziona l'agente o gli agenti per eseguire il task.

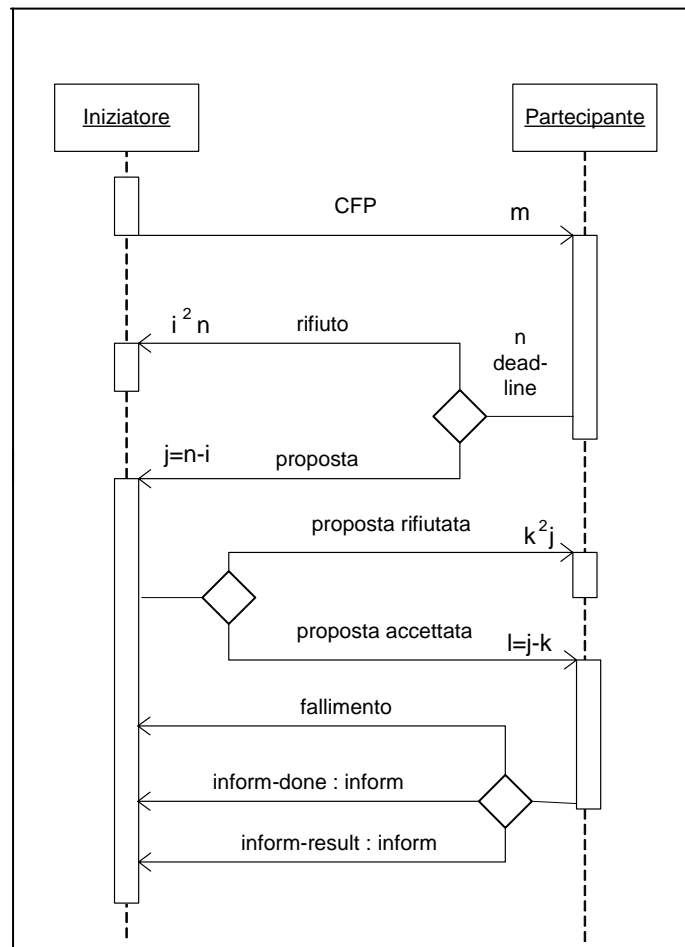


Fig.5 Diagramma di sequenza del FIPA Contract Net Interaction Protocol

Agli  $l$  (con  $l < j$ ) agenti selezionati verrà spedito un messaggio di accettazione della proposta (*accept-proposal*), mentre ai rimanenti  $k$  agenti, verrà inviato viceversa un messaggio di rifiuto (*reject-proposal*). I primi avranno così acquisito un *commitment* per l'esecuzione del task. Una volta completata l'esecuzione dei task, i partecipanti informano l'iniziatore dell'esito dell'esecuzione. Se l'esito è positivo inviano un messaggio di *inform-done* o un messaggio più esplicito di *inform-result*. Se l'esito invece è negativo viene inviato un messaggio di fallimento (*failure*).

Importante è notare che il CNIP richiede che l'iniziatore riesca a riconoscere quando ha ricevuto tutte le risposte, indipendentemente che queste siano rifiuti o proposte. Nel caso in cui, per esempio, un partecipante fallisse la sua risposta (*reply*), l'iniziatore potrebbe rimanere in attesa di quest'ultima per un periodo di tempo

molto lungo. Per scongiurare questo pericolo, la CFP include come abbiamo già visto, tra i suoi parametri anche la terminazione, la quale esprime il tempo limite di attesa dell'iniziatore. Tutti i messaggi pervenuti dopo tale limite, infatti, non vengono neppure considerati, indipendentemente che questi rappresentino proposte o rifiuti. Il valore della terminazione è specificato all'interno del messaggio ACL attraverso l'ausilio del parametro "*reply-by*".

In un ambiente come Internet, o comunque in un qualsiasi sistema aperto, il numero d'interazioni fra gli agenti è sempre molto elevato. E' necessario, quindi, introdurre un meccanismo che consenta agli agenti di distinguere le comunicazioni fra di loro. Nel FIPA CNIP, tale meccanismo è realizzato assegnando ad ogni comunicazione un identificatore. Tale identificatore è deciso dall'iniziatore nella fase di instaurazione dell'interazione ed è espresso mediante il parametro unico e non nullo nel messaggio ACL di "*conversation-id*". Gli agenti coinvolti nella stessa interazione dovranno, quindi, contrassegnare i loro messaggi con l'identificatore di conversazione. L'utilizzo di tale parametro permette agli agenti di poter gestire in maniera strategica le loro conversazioni. L'identificatore, permette, infatti, di poter individuare la "storia" della conversazione, consentendo agli agenti di prendere decisioni anche in base all'evolversi di quest'ultima.

Nel caso di protocolli d'interazione 1:N, o di sub-protocolli, l'iniziatore è libero di decidere se utilizzare il medesimo identificatore o meno.

Di seguito illustreremo alcune situazioni eccezionali, che possono presentarsi durante l'Interaction Protocol e che non sono contemplate nella fig.5.

In ogni fase del CNIP, può accadere che un agente riceva un messaggio di cui non riesca a comprendere il contenuto. In tal caso, l'agente può informare il mittente attraverso un messaggio di *not-understood*. La fig.5 non mostra tale possibilità, poiché questa potrebbe presentarsi in ogni punto del protocollo e da entrambe le parti. L'effetto che un messaggio di *not-understood* può determinare, è la terminazione dell'intera comunicazione. Tale terminazione, comporta inevitabilmente il decadere di ogni commitment fino ad allora assegnato, che deve ritenersi invalido. Poiché il CNIP prevede la possibilità di trasmettere informazioni a più di un partecipante, le risposte che questi invieranno andranno analizzate e valutate separatamente. Nel caso in cui, una o più di queste risultassero tali da



richiedere l'invio di messaggi di not-understood, ciò comunque non giustificherebbe l'interruzione dell'intero CNIP, ma solo dei sotto protocolli interessati.

In ogni istante, inoltre, un agente può, ove lo ritenesse opportuno, decidere di far terminare la comunicazione. Le motivazioni di tale azione, sono da ricercarsi in vari elementi. In generale se un agente decide di cancellare una comunicazione, è perché non ritiene di poterne trarre più vantaggi, e di non avere nessun interesse nel mantenerla. Si cerca, comunque, di eliminare l'interazione, in un modo accettabile per entrambe le parti, utilizzando il *Meta-Cancel Protocol* descritto successivamente. La comunicazione da cancellare, viene identificata ed indicata attraverso l'utilizzo del parametro identificatore di conversazione.

Ovviamente, al di là della specifica FIPA i messaggi possono essere espressi anche attraverso il linguaggio KQML. Un messaggio KQML contiene parametri come: il mittente, il destinatario, reply-with, l'ontologia, il linguaggio ed infine, il contenuto del messaggio ovviamente. Un esempio di messaggio KQML è riportato di seguito:

*“(monitor : from customer : to supplier : reply-with update-111 : ontology standard-units-and-dimensions : language KIF : content (= (q.magnitude (diameter shaft-a) inches) ?x))”*

Il vantaggio che si ottiene dall'utilizzo delle specifiche FIPA, è di poter conseguentemente contare sulla certezza che, una volta creata una rete d'agenti conforme a tali specifiche, questa sarà compatibile con le altre Contract Net. Ciò si realizza attraverso un processo di continua integrazione.

Uno dei più promettenti utilizzi del CNIP è rappresentato dalla sua capacità di poter essere usato nella creazione di mercati elettronici per la vendita e l'acquisto di beni. Il CNIP tuttavia trova utilizzo in una vasta gamma d'applicazioni, sia industriali che commerciali.

## **2.6 FIPA Iterated Contract Net Interaction Protocol**

Il FIPA Iterated Contract Net Interaction Protocol [12] rappresenta un'estensione del FIPA CNIP di base. L'estensione è rappresentata, dall'introduzione di un processo di negoziazione d'offerte iterativo. La fig.6 illustra brevemente il

funzionamento di questo protocollo, che di seguito descriveremo più dettagliatamente.

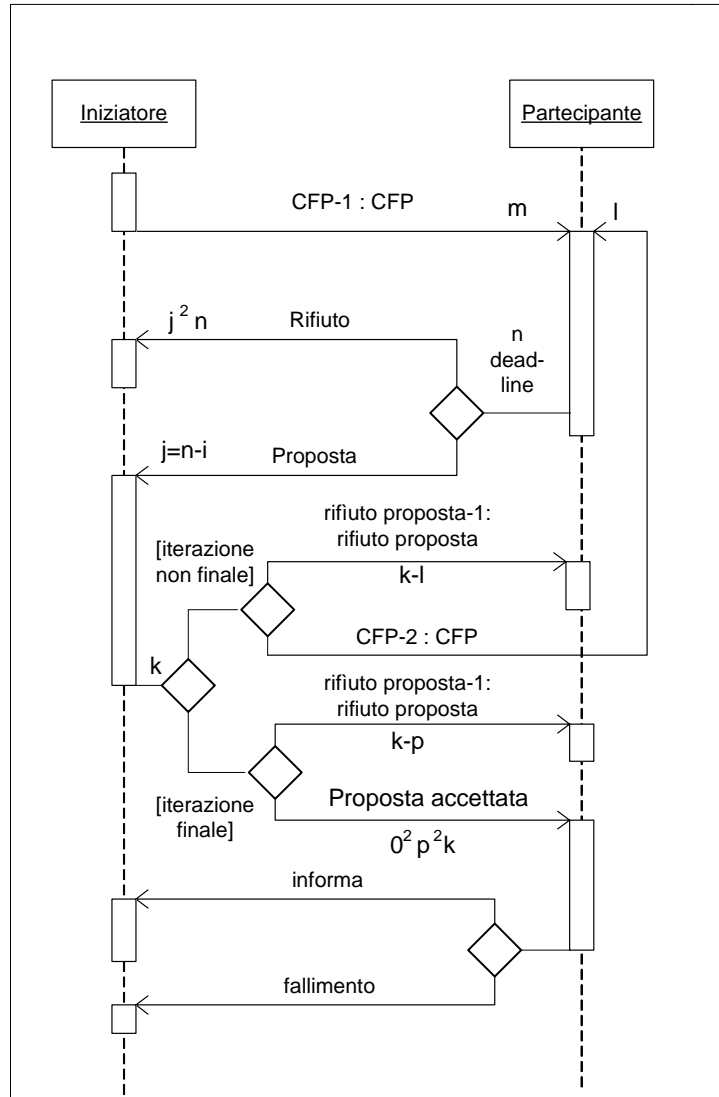


Fig. 6 Diagramma di sequenza del FIPA Iterated CNIP

Come nel FIPA CNIP, l'iniziatore invia  $m$  messaggi di CFP ad altrettanti partecipanti. Solo  $n$  di questi rispondono. Fra queste risposte,  $k$  sono messaggi proposte, ovvero offerte di esecuzione del task da parte dei partecipanti che sono in grado di eseguire il compito sotto le costrizioni imposte dell'iniziatore. Le rimanenti  $j$  risposte, contrariamente, sono rifiuti. Queste  $k$  proposte pervenute all'iniziatore,

possono essere considerate iterazioni finali (*final iteration*), ovvero messaggi che non richiedono ulteriori interazioni. In tal caso  $p$  di queste proposte sono accettate, mentre le rimanenti sono rifiutate. Alternativamente l'iniziatore può decidere di iterare il processo d'offerta (*non-final iteration*). L'Iniziatore invia  $l$  CFP rivisitate ad altrettanti partecipanti mentre rifiuta le rimanenti  $k-l$  proposte. Lo scopo è quello di cercare di ottenere le offerte da parte dei partecipanti, rivisitando le CFP e ottenendo nuove proposte. Il processo termina quando l'iniziatore rifiuta tutte le proposte e non invia nuove CFP, oppure accetta una o più proposte, o ancora nel caso in cui i partecipanti rifiutano tutti di fare offerte.

Per quanto concerne i parametri d'identificazione della conversazione (*conversation-id*) e terminazione (*deadline*), valgono le medesime considerazioni già trattate per il FIPA CNIP, visto precedentemente.

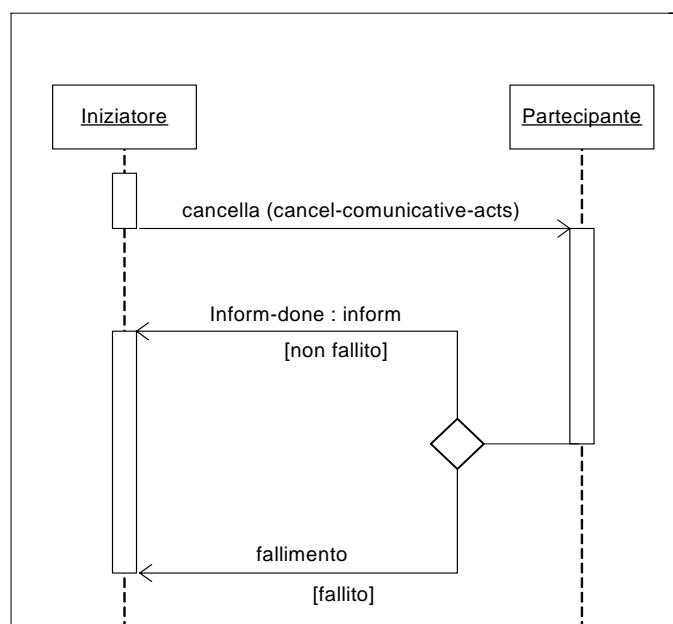


Fig. 7 Diagramma di sequenza del Cancel Meta-Protocol

## 2.7 FIPA Cancel Meta-Protocol

Il protocollo di terminazione della comunicazione, è illustrato in fig.7 e prende il nome di *Cancel Meta-Protocol* [11]. Anche qui, come pure nel CNIP, i partecipanti

al termine del protocollo informano l'iniziatore dell'esito dell'operazione con un messaggio di inform-done, o in caso di fallimento con un messaggio di failure.

## Capitolo 3

### Protocolli di negoziazione estesi

Dall'analisi del modello di base del CNP di Smith [8], sono state evidenziate alcune mancanze ed inadeguatezze, rispetto all'esigenze del processo di negoziazione tra gli agenti nei MAS. Questo ha spinto numerosi studiosi a definire protocolli di negoziazione che estendessero la soluzione proposta dal CNP, al problema dell'allocazione di task, in modo da trovare soluzione alle problematiche delineate.

Sandholm, nel 1993 [13], estese il CNP introducendovi il concetto di agenti *interessati a se stessi* (*self-interested*, *SI*) e presentò una formalizzazione dei processi di offerta e di assegnazione di un contratto, lasciati indefiniti nell'originale CNP.

Sandholm e Lesser, nel 1995 [14], continuarono poi insieme l'estensione del CNP attraverso due vie principali: la razionalità degli agenti è limitata dalla complessità computazionale, inoltre gli agenti possono scegliere la fase e il livello di *commitment* dinamicamente attraverso l'uso di condizioni sulla negoziazione e di penalità in caso di *decommitment*.

Golfarelli, nel 1997 [15], propose il protocollo di negoziazione *Task Swap*, il quale anch'esso estendeva il CNP allo scopo di far fronte ai problemi che si presentano in domini dove lo scambio di task (*Task-Swap*) è il solo possibile mezzo di negoziazione, e gli agenti sono entità eterogenee e SI.

#### **3.1 Implementazione del Contract Net Protocol basata sul calcolo dei costi marginali**

Come già descritto in precedenza nel paragrafo 2.1, il CNP di Smith (1980) trovò applicazione, inizialmente, nella simulazione di reti distribuite di sensori acustici. In queste applicazioni gli agenti erano totalmente cooperativi e la selezione del contractor si basava esclusivamente sulle attinenze che l'agente possedeva rispetto al

task da eseguire, come per esempio la capacità di elaborazione, il carico corrente ecc...

Thomas Sandholm nel 1993 presentò una possibile estensione del CNP. Egli propose un modello formale per le decisioni prese durante le fasi di annuncio, di offerta e di assegnazione, dove gli agenti calcolano localmente i loro costi marginali, ovvero i costi devono supportare per l'esecuzione di determinati insiemi di compiti. Di tale modello è possibile delineare i seguenti elementi d'innovazione rispetto al CNP:

- La scelta del contractor è basata esclusivamente sul calcolo di questi *costi marginali* ovvero sui costi che un agente deve sostenere per allocare i propri task.
- Il modello proposto generalizza il CNP verso il supporto oltre che di agenti cooperativi, anche di agenti competitivi.
- Nell'originario CNP, i task erano negoziati uno alla volta. Tale approccio risulta inadeguato nei casi in cui, l'esecuzione di un task, dipende dall'esito della negoziazione di altri task. La struttura del protocollo proposto, gestisce l'interazione fra i task (*task interaction*), attraverso il raggruppamento di quest'ultimi in insiemi che dovranno essere negoziati come un unico elemento atomico di contrattazione.
- Infine, attraverso questa estensione, si propone una possibile soluzione al problema della congestione causata dai messaggi di annuncio (*announcement message*).

Sandholm sviluppa il protocollo di negoziazione attraverso l'utilizzo del problema d'instradamento dei veicoli. Il problema è strutturato in termini di un certo numero di centri di spedizione, sparsi geograficamente ed appartenenti a differenti compagnie di trasporto. Ogni centro di spedizione rappresenta un agente che è responsabile per le consegne di una certa fabbrica, e ha a disposizione un certo numero di veicoli che devono occuparsi delle effettive consegne. Ogni consegna dovrà essere inclusa nella rotta del veicolo opportuno. Il problema locale di ogni agente è l'individuazione di soluzioni iniziali ottime, in base alle risorse a disposizione. I fattori di cui ogni agente dovrà tener conto saranno, per esempio, per

ogni veicolo il costo per chilometri, la durata dell'itinerario, la massima lunghezza dell'itinerario ecc...Lo scopo è ovviamente quello di minimizzare i costi di trasporto. Nel risolvere tale problema, ogni centro di spedizione rappresentato da un agente, deve quindi per prima cosa risolvere il suo problema locale d'instradamento. Dopo di che un agente ha la possibilità di negoziare con altri centri di consegna. Nel nostro caso specifico, la negoziazione si basa sullo scambio di consegne allo scopo di trarne vantaggi. Tale scambio avviene quando un agente contractor, essendo in grado di eseguire un insieme di task con costi minori rispetto a quelli che dovrebbe sostenere il manager, accetta di fare le consegne per suo conto in cambio di un corrispettivo in denaro. La negoziazione appare, quindi, come uno strumento per individuare e realizzare la migliore soluzione d'instradamento possibile, intendendo per migliore, quella a costo di trasporto più basso. Ad ogni interazione il venditore fissa il prezzo basandosi sulla domanda, mentre il compratore stabilisce la quantità che vuole comprare.

L'applicazione sull'instradamento di veicoli è implementata in un sistema chiamato *TRAnspOrtation COoperation NET* (TRACONET). La negoziazione in TRACONET è automatica, asincrona e diretta. Ad ogni agente coinvolto nella contrattazione, è consentito fare un'offerta per ogni annuncio ricevuto, senza che le decisioni delle altre parti coinvolte gli vengano rivelate. Le negoziazioni sono dirette nel senso che gli annunci non sono spediti indifferentemente a tutti gli agenti, ma a solo determinati potenziali contractor.

Non esiste una gerarchia fissa fra gli agenti. Ogni agente, concordemente con il CNP di base, può ricoprire sia il ruolo di manager che di contractor, ma non può ricoprire i due ruoli contemporaneamente. Ogni agente, inoltre, è libero di negoziare nuovamente i task ricevuti in una precedente negoziazione. Quando un agente fa un annuncio (*announcing*), nel caso in questione cerca di comprare il servizio di trasporto di qualche altro agente, al prezzo massimo specificato nell'annuncio. Viceversa quando un agente fa un'offerta (*bidding*), prova a vendere il proprio servizio al prezzo specificato nell'offerta. Assegnare un contratto (*awarding*), nel nostro contesto, significa comprare il servizio di trasporto di un altro centro di spedizione, mentre aggiudicarsi un contratto (*award taking*), significa vendere il proprio servizio.

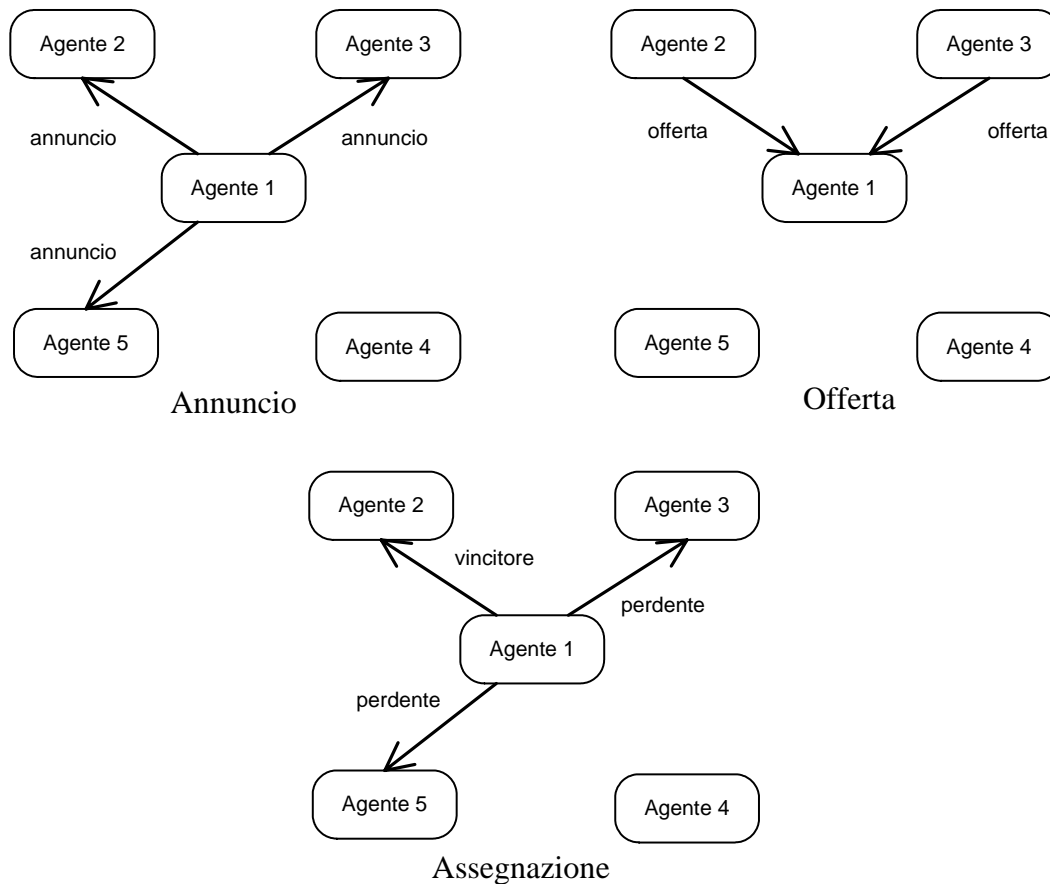


Fig.8 Fasi di annuncio, di offerta e di assegnazione

Diversamente dall'originale CNP, nella fase di assegnazione del contratto è prevista la spedizione di messaggi di rifiuto verso le offerte non selezionate. Tali messaggi hanno la capacità di liberare l'agente offerente dai vincoli che lo legano alla sua offerta e che influenzano la valutazione dei costi marginali di altri task. Un'altra opzione consisterebbe nell'applicazione del FGP descritto al par.2.4.2, dove si definisce un limite di tempo oltre al quale se l'agente offerente non ha ricevuto risposta alla sua offerta, la considera persa. Tuttavia, tale opzione, non sia adatta al sistema di negoziazione asincrono del TRACONET poiché, in tal caso, si forzerebbe il manager a decidere a chi assegnare il contratto entro uno stretto limite di tempo. Come soluzione a questo problema, Sandholm, introduce un messaggio di traffico



(*traffic message*), il quale dovrebbe consentire agli agenti di conoscere in anticipo quali offerte saranno garantite e quali no.

Nel modello proposto da Sandholm gli agenti sono costituiti principalmente da due parti:

- a. *l'ottimizzatore locale (local optimizer)*, il quale fornisce il servizio di ottimizzazione locale. Gli agenti appartenenti alla stessa rete di negoziazione possono usare diversi algoritmi di ottimizzazione locale, messi a punto in base ai requisiti di uno specifico centro di consegna.
- b. *il sistema di contrattazione (bargaining system)*, il quale a sua volta si divide in quattro principali componenti: *l'annunciatore (announcer)*, *l'offerente (bidder)*, *l'assegnatore (awarder)* e *il destinatario dell'assegnazione (award taker)*.

Il sistema di contrattazione deve essere in grado di fornire alcuni servizi principali relativi al calcolo dei costi marginali di un task e all'ottimizzazione del percorso delle consegne che un agente deve eseguire.

In TRACONET, un agente per prima cosa chiama il suo ottimizzatore locale allo scopo di individuare una soluzione al problema dell'instradamento dei veicoli in base alle risorse che ha a disposizione. Basandosi quindi, su questa prima analisi, l'agente inizia la negoziazione con gli altri centri di spedizione per cercare di ottimizzare la soluzione individuata. Un agente, durante la negoziazione, prima di spedire un nuovo annuncio, valuta e gestisce tutti i messaggi ricevuti. Tale meccanismo è attuato allo scopo di evitare il problema della congestione.

Un agente può entrare e uscire dalla rete di negoziazione in maniera dinamica. Quando entra nella rete, per prima cosa deve liberarsi di tutti i messaggi che aveva accumulato in precedenza, dopo di che è pronto per negoziare. Tale protocollo inoltre, consente ad un agente, che ha già effettuato una serie di negoziazioni su un insieme di task, di ricalcolare la propria soluzione d'instradamento, per cercare di riottimizzarla. In tal caso, l'agente dovrà prima uscire dalla negoziazione, effettuare la riottimizzazione e poi possibilmente rientrare nella negoziazione. In caso contrario, i costi marginali calcolati prima della riottimizzazione, non sarebbero più validi.

Analizziamo ora in dettaglio il modello formale proposto da Sandholm per le varie fasi della negoziazione.

La fase di *annuncio* prevede che un agente annunciatore, dopo aver determinato l'insieme di consegne che vuole negoziare, effettui un annuncio per quest'ultima allo scopo di ricevere delle offerte. All'agente è data inoltre la possibilità, ove lo ritenesse opportuno, di poter annunciare più volte il medesimo insieme di consegne. Tale meccanismo fornisce, come vantaggio, la possibilità di determinare migliori soluzioni al problema d'instradamento a scapito però di un allungamento dei tempi di negoziazione. L'agente decide quali task negoziare in base ad un calcolo euristico dei costi marginali che l'agente dovrebbe sostenere nell'allocare tali task, effettuato dall'ottimizzatore locale. Il calcolo dei costi marginali durante la fase di annuncio, tuttavia, non è d'importanza cruciale ai fini della negoziazione. Viceversa, lo è durante la fase di assegnazione del contratto, poiché il valore di tali costi viene valutato tenendo conto delle offerte pervenute all'agente. Come già detto in precedenza, spesso annunciare un task alla volta può non essere sufficiente agli scopi della negoziazione. Questo protocollo propone come soluzione, il raggruppamento di più task in un unico elemento atomico di contrattazione.

La fase di offerta prevede che l'agente offerente legga gli annunci che ha ricevuto dagli altri agenti e, nel caso in cui il prezzo indicato nell'annuncio per eseguire l'insieme di task sia superiore ai costi che l'agente dovrebbe sostenere nell'eseguirli, l'annuncio viene giudicato vantaggioso e viene inviata un'offerta all'annunciatore. Viceversa, l'annuncio viene giudicato non vantaggioso e non viene spedita alcuna offerta. Così come nel CNP, anche in questo modello di protocollo, si è scelto di consentire che un agente possa avere più offerte "sospese" contemporaneamente per diversi contratti, anche se tale scelta implica valutazioni dei costi meno precise.

Di seguito daremo alcune definizioni necessarie per illustrare il modello matematico di valutazione dei costi marginali:

- $T_{cur}$  rappresenta l'insieme di task corrente di un agente;
- $f(T)$  rappresenta la funzione che consente di calcolare il costo totale di una soluzione locale ottima, rispetto ad un dato insieme di task  $T$ ;
- $C_{add}(T)$  rappresenta il costo marginale che si deve supportare nell'aggiungere alla soluzione locale un insieme di task  $T$ ;
- $C_{addmax}(T)$  rappresenta il limite superiore del valore  $C_{add}(T)$ ;

- $C_{addmin}(T)$  rappresenta il limite inferiore del valore  $C_{add}(T)$ .

Come precedentemente illustrato nel paragrafo 2.1, lo scopo nel CNP di base, è cercare di ottimizzare la funzione di utilità di un task. Il calcolo di tale funzione richiede tempi spesso molto lunghi. Questo inconveniente non era stato preso in considerazione nel modello di base del CNP. In tale modello si definisce una funzione di ottimizzazione  $f$ , il cui calcolo richiede tempi esponenziali. Per limitare tali tempi, viene introdotta una funzione  $f'$  che rappresenta un'approssimazione euristica di  $f$ , e tale che  $f(T) \leq f'(T)$  per ogni insieme di task  $T$ . Nel caso considerato, il calcolo di  $f'(T + T_{cur})$ , è subordinato alla conoscenza di  $f(T_{cur})$ . Il problema risiede nel fatto che non è possibile determinare facilmente neppure quest'ultima funzione ma solo una sua approssimazione euristica  $f'(T_{cur})$ . In tale modello, quindi, il calcolo del costo  $C'_{add}(T)$ , è stato ottenuto rispetto ad approssimazioni euristiche e supponendo che le offerte ricevute non garantite, non fossero prese in considerazione dagli agenti. Tali assunzioni tuttavia, rendono il calcolo dei costi marginali *semi-opportunistico*. Ciò implica il rischio che un agente, usando tali approssimazioni, possa stipulare anche contratti non vantaggiosi. Un approccio *sicuro* è rappresentato dall'utilizzo di un euristico limite superiore  $C_{addmax}(T)$ , anche se purtroppo il calcolo di tale valore richiede tempi più lunghi rispetto al calcolo di  $C'_{add}(T)$ .

Il problema della congestione si verifica quando alcuni agenti ricevono annunci più velocemente di quanto possano esaminarli. Tale problema riguarda, essenzialmente solo gli annunci, poiché questi si presentano durante la negoziazione, in numero molto maggiore rispetto ad ogni altro tipo di messaggio. Il problema della congestione è stato risolto imponendo all'agente offerente di considerare solo gli annunci più recenti, cioè gli annunci per i quali il tempo trascorso dal momento della loro ricezione, supera un certo valore stabilito, vengono automaticamente scartati. Questo consente all'agente di evitare di perdere tempo per rispondere ad annunci che rischierebbero di arrivare al manager a negoziazione già conclusa.

Durante la fase di assegnamento del contratto, l'agente manager legge le offerte ricevute da altri agenti. Prima d'iniziare a valutare tali offerte, tuttavia aspetta che sia passato un certo limite tempo da quando ha spedito l'annuncio, in modo tale da

consentire a tutti gli agenti di effettuare un'offerta. Dopo di che risponde agli agenti dai quali ha ricevuto un'offerta, inviando o messaggi di rifiuto o di accettazione. Quest'ultimo tipo di messaggio è inviato agli agenti che offrono i prezzi più economici e perciò più vantaggiosi. Successivamente, l'agente, essendo riuscito a contrattare l'insieme di task, si preoccupa di eliminare tale insieme dalle sue consegne correnti  $T_{cur}$  e dalle relative soluzioni d'instradamento.

Può accadere che una volta scaduto il limite di tempo precedentemente nominato, l'agente non abbia ricevuto ancora alcuna offerta. In tal caso la fase di assegnamento del contratto viene ritardata finché non arriva almeno un'offerta. Nel caso in cui, anche dopo due trascorsi del limite di tempo, nessuna risposta sia pervenuta, viene semplicemente inviato un messaggio di rifiuto a tutti gli agenti a cui si era inviato l'annuncio. Le offerte ricevute in seguito vengono eliminate.

Durante questa fase, all'agente è consentito di poter rivalutare l'utilità ricavabile da un eventuale contratto. Se, tali vantaggi, non dovessero più presentarsi, a causa, ad esempio, di cambiamenti avvenuti nell'ambiente, l'agente potrebbe decidere di rifiutare tutte le offerte pervenutegli rispetto a quella proposta.

Un'offerta si definisce vantaggiosa se  $Crem(T) > cb$ , dove  $Crem(T)$  sono i costi che l'agente manager dovrebbe sostenere nel rimuovere l'insieme di task  $T$  dalla propria soluzione locale, e  $cb$  è il prezzo menzionato nell'offerta. Anche in questo caso definiamo le seguenti grandezze:

- $Crem_{max}(T)$ , rappresenta il limite superiore di  $Crem(T)$ ;
- $Crem_{min}(T)$ , rappresenta il limite inferiore di  $Crem(T)$ .

Similarmente alla fase di offerta, attraverso approssimazioni euristiche e all'assunzione che nessuna delle offerte non garantite sia presa in considerazione dall'agente, si giunge ad ottenere  $Crem(T)$ . Anche in questo caso l'applicazione di tali approssimazioni può far sì che vengano accettate offerte non vantaggiose. Un approccio sicuro implica l'utilizzo di un euristico limite inferiore  $Crem_{min}(T)$  come costo di valutazione per giudicare se un'offerta è ancora vantaggiosa. Ovviamente il suo calcolo richiede tempi più lunghi rispetto a quelli di  $Crem(T)$ .

L'ultima fase del processo di negoziazione è rappresentata dalla fase di ricezione dell'assegnamento del contratto. L'agente una volta che si è aggiudicato il contratto

e quindi l'insieme delle consegne da eseguire, inserisce queste ultime nel suo  $T_{cur}$  e nella sua soluzione locale d'instradamento. Successivamente eseguirà le consegne per conto dell'agente manager, rispettando le condizioni pattuite nel contratto.

Concludendo Sandholm, attraverso questa analisi ha esteso, il CNP definendo un modello formale per le fasi di annuncio, offerta e assegnazione basate principalmente sul calcolo dei costi marginali locali. In tale modello vengono inoltre evidenziate, due differenti politiche di determinazione dei costi, una definita sicura e una opportunistica. Quest'ultima consente di velocizzare il processo di negoziazione, mentre a prima richiede tempi più lunghi ma garantisce sempre contratti vantaggiosi per entrambe le parti e quindi porta ad una diminuzione graduale dei costi di negoziazione. L'interazione fra i task è gestita attraverso un raggruppamento euristico di quest'ultimi, in un insieme che rappresenta l'unità atomica della negoziazione. L'implementazione di tale protocollo risulta asincrona, distribuita e comprende anche una soluzione al problema della congestione a causa dei messaggi. Tali risultati sono stati ottenuti, applicando il processo di negoziazione ad un dominio di applicazione concreto come quello dell'instradamento dei veicoli, tuttavia, esso è applicabile a molte altre problematiche di allocazione di task.

### **3.2 Protocollo di Negoziazione per Agenti Bounded Rational Self-Interested**

Una seconda estensione del CNP, fu proposta sempre Sandholm questa volta però in collaborazione con Lesser nel 1995.

Questi ricercatori partirono dalle seguenti considerazioni: gli agenti sono entità SI, a cui si richiede di prendere decisioni in tempo reale e a cui sono fornite limitate risorse computazionali. A partire da queste considerazioni, studiarono le prestazioni e le implicazioni del comportamento degli agenti in una struttura di domini indipendenti.

Ogni agente possiede un insieme di task ed un insieme di risorse che può gestire per portare a compimento i suoi compiti. Tuttavia questi insieme possono subire delle modifiche a causa del dominio degli eventi, che comporta l'arrivo di nuovi task o

l'esaurirsi di determinate risorse. Gli agenti hanno la capacità di sub-contrattare i propri task ad altri agenti attraverso il pagamento di un compenso. Il processo di contrattazione può comportare anche la scomposizione di un task in più sottocompiti gestiti da diversi agenti o, viceversa, il raggruppamento di più task in un unico supertask. L'allocazione di un task è ritenuta vantaggiosa nel caso in cui il contractor è capace di eseguire il task in maniera più economica rispetto al contractee, o nel caso in cui il contractee non sia capace di eseguire il o i task.

Il problema della negoziazione risulta quindi essere composto da due sottoproblemi:

1-il problema d'allocazione globale dei task;

2-il problema di trovare un'ottima combinazione locale, riferita alle risorse e ai task correnti di un agente.

L'obbiettivo di ogni agente risulta, quindi, essere quello di massimizzare il proprio guadagno, intendendo per guadagno la differenza fra il suo reddito e i suoi costi. Le entrate sono rappresentate dalla gestione dei task di altri agenti, le uscite invece dal consumo di risorse per l'esecuzione di tali task. Lo studio di Sandholm e Lesser è stato effettuato, assumendo di operare in domini dove la fattibilità e il costo di gestione di un task non dipendono dai metodi di gestione delle risorse degli altri agenti o da come quest'ultimi allocano i propri task. La fattibilità e il costo di un task, dipendono esclusivamente dagli altri task che un agente possiede.

Il raggiungimento di una buona soluzione per quanto concerne il problema di allocazione globale, è complicato principalmente da due fattori: in primo luogo gli agenti essendo entità SI, possono non mostrare effettivamente tutte le loro informazioni e adottare quindi un comportamento non "sincero"; un secondo limite è rappresentato dal fatto che gli agenti posseggono una *razionalità limitata (Bounded Rational, BR)*, che può causare errori nelle fasi di valutazione sugli effettivi vantaggi di un contratto. I problemi, precedentemente descritti, sono esasperati dalle incertezze che derivano dall'operare in un ambiente aperto dove nuovi agenti e nuovi task possono arrivare o scomparire continuamente e le decisioni prese in un certo istante, alla luce di nuove informazioni, possono non essere più adeguate.

Tutte le problematiche fin qui descritte, non sono state prese in considerazione nel modello originale del CNP. Un primo passo in avanti verso la loro risoluzione era

già stato effettuato attraverso la prima estensione del CNP del 1993 da parte di Sandholm, il quale proponeva una modellazione formale delle fasi di offerta, annuncio e di assegnazione degli agenti. Tuttavia, tale seconda estensione mira ad approfondire gli effetti della limitata razionalità degli agenti SI, all'interno di un processo di negoziazione caratterizzato da numerose incertezze. Si propone, quindi, un nuova tipologia di protocollo, che ingloba al suo interno differenti tipologie di protocolli. L'agente, infatti, in base alle caratteristiche sia dinamiche che statiche della negoziazione, può decidere di modificare alcune opzioni del protocollo. Tali opzioni riguardano:

- la fase della negoziazione durante la quale avviene il commitment; viene presentato un modello nel quale, tale fase, può variare di locazione; ciò consente maggior flessibilità nel prendere le decisioni locali e offre migliori strategie di gestione dei rischi durante la negoziazione, poiché consente agli agenti di recedere dal contratto;
- le deliberazioni locali; si discuteranno i rischi della negoziazione e come i costi di decommitment possono influire sul comportamento di un agente.
- il tentativo di evitare ottimizzazioni locali nello spazio di allocazione dei task utilizzando contratti che coinvolgono più agenti;
- la gestione della congestione da messaggi.

Di seguito si descriveranno più dettagliatamente gli elementi caratterizzanti di questo nuovo protocollo, per la negoziazione tra agenti BRSI, che include ed estende il CNP di Smith.

### **3.2.1 Protocollo multi-leveled commitment**

Nella negoziazione tra gli agenti, commitment, significa che un agente lega se stesso ad un potenziale contratto, in attesa che l'altro agente rifiuti o accetti la sua proposta. Se quest'ultimo accetta l'offerta, entrambe le parti risultano vincolate dal contratto. Il CNP di base, è stato implementato in modo tale che, in caso di poche interazioni, sia l'agente contractee ad annunciare i task, mentre in caso di elevate interazioni il potenziale contractor segnala agli altri agenti la propria disponibilità. In tal modo la scelta del protocollo da utilizzare era basata principalmente sulle caratteristiche dell'ambiente esterno.

Nella prima estensione del CNP di Sandholm (1993), il commitment era logicamente previsto durante la fase di offerta. Qui si propone, invece, un modello di protocollo nel quale si può decidere quando effettuare tale fase separatamente per ogni negoziazione. Questo protocollo dà, quindi, la possibilità di poter scegliere quando eseguire il commitment durante la negoziazione. In tal modo sono consentite tutte le precedenti possibilità e, inoltre, la decisione di quando eseguire la fase di commitment, diventa strategica per il buon esito della negoziazione.

Un altro elemento di estensione rispetto al CNP, è rappresentato dalla possibilità che gli agenti hanno, in questo protocollo, di recedere dal contratto. Tale modello di protocollo si definisce multi-leveled commitment poiché in esso si hanno vari livelli di commitment distribuiti durante il processo di negoziazione. A seconda del livello in cui ci si trova, durante la costruzione del commitment, è possibile recedere o meno dal contratto. In caso di recessione, è prevista una penalità. Il valore della penalità può variare, in base al livello di commitment in cui ci si trova o in maniera proporzionale al tempo di risposta di un agente o ancora può essere condizionato dagli sviluppi di altre negoziazioni o dell'ambiente.

Attraverso l'utilizzo di multipli livelli di commitment, gli agenti sono spinti ad esplorare maggiormente lo spazio degli agenti SI, poiché molti rischi sono evitabili grazie alla possibilità di decommitment.

### **3.2.2 Struttura dei messaggi di contrattazione**

La fig.9 descrive il formato dei messaggi del nuovo protocollo di contrattazione. La negoziazione può iniziare sia con un messaggio da parte del contractor che del contractee. Il messaggio di un contractee specifica le possibili alternative del contratto. All'interno di ogni alternativa, il task può essere diviso in una serie di sottocompiti, specificando i campi che vanno dall'(a) al (d). Ogni alternativa ha la seguente semantica: se il contractor accetta di eseguire tutti i task descritti nel campo (a), e accetta di compiere ogni insieme di task con il livello di qualità specificato nel campo (d), il contractee, s'impegna automaticamente a pagare la quantità specificata nel campo (b), e può recedere dall'accordo solo attraverso il pagamento della penalità specificata nel campo (c). I campi (a) (b) (c) e (d) possono essere espressi in funzione del tempo.



MESSAGGIO DEL CONTRACTOR	MESSAGGIO DEL CONTRACTEE
1. Identificatore del Messaggio	1. Identificatore del Messaggio
2. In-risposta-a (id del messaggio)	2. In-risposta-a (id del messaggio)
3. Mittente	3. Mittente
4. Destinatario	4. Destinatario
5. Terminazione della negoziazione	5. Terminazione della negoziazione
6. Alternativa 1	6. Alternativa 1
6.1. Tempo di validità	6.1. Tempo di validità
6.2. Obblighi in caso di decommitment del partner	6.2. Obblighi in caso di decommitment del partner
6.3. Offerta sotto compenso	6.3. Offerta sotto compenso
6.4. Richiesta risposta sotto compenso	6.4. Richiesta risposta sotto compenso
6.5. Insieme di task 1	6.5. Insieme di task 1
(a) Specifiche (minime) dei task	(a) Specifiche (massime) dei task
(b) Pagamento promesso per il contratto	(b) Pagamento promesso per il contratto
(c) Commitment richiesto al contractor	(c) Commitment richiesto al contractor
(d) Commitment richiesto al contractee	(d) Commitment richiesto al contractee
6.6. Insieme di task 2	6.6. Insieme di task 2
...	...
6.i. Insieme di task i-4	6.m. Insieme di task m-4
...	...
7. Alternativa 2	7. Alternativa 2
...	...
j. Alternativa j-5	n. Alternativa n-5

Fig.9 Struttura dei messaggi del contractor e del contractee

Il contractor può accettare una delle alternative proposte dal contractee, inviando un messaggio in cui le specifiche dei task sono conformi a quelle espresse nel messaggio del contractee. Analogamente a quanto descritto in precedenza, il contractee potrà poi accettare una delle alternative presenti nel messaggio del contractor. Un agente può decidere di far terminare la negoziazione attraverso l'identificatore della negoziazione (campi (0)) e settando il campo (5).

Un altro elemento d'estensione rispetto al CNP è rappresentato dalla possibilità di effettuare delle contro-proposte. Nel caso in cui il contractor non voglia accettare nessuna delle alternative e non desideri neppure terminare la negoziazione, può inviare un messaggio di contro-proposta. Il contractee può accettare questa nuova proposta, o a sua volta contro-proporre. In quest'ultimo caso la negoziazione rischia di andare avanti indefinitivamente.

Il campo (6.1) indica il tempo di validità di un'offerta o di un'alternativa (terminazione). Se il partner della negoziazione non risponde entro tale tempo di validità, l'agente che aveva inviato il messaggio non è più vincolato dalle alternative. Una soluzione alternativa all'utilizzo della terminazione, è quella d'inviare messaggi nei quali il campo (b) (ed eventualmente anche i campi (c) e (d)) è funzione del tempo di risposta. In tal modo, ovviamente, colui che deve rispondere al messaggio è motivato a farlo il più velocemente possibile. Entrambi i meccanismi, richiedono quindi che i tempi di invio e ricezione di un messaggio possano essere verificati da entrambe le parti.

### **3.2.3 La razionalità limitata degli agenti**

Andremo ora ad analizzare più in profondità le implicazioni della limitata razionalità degli agenti.

Altri ricercatori precedentemente, hanno assunto che gli agenti fossero dotati di una perfetta razionalità, il che implicherebbe valutazioni esatte dei costi marginali, cosa praticamente impossibile da realizzare come già dimostrato in [13]. Gli agenti, infatti sono dotati di una razionalità limitata a causa dei costi computazionali e delle risorse, e a causa dei continui mutamenti dell'ambiente che li circonda. Inoltre la negoziazione degli agenti è pressata da altre esigenze:

- il messaggio di conferma o di contro-proposta deve essere inviato entro la terminazione, altrimenti la negoziazione termina;
- i campi (b) e(d) possono essere funzione del tempo, e quindi possono causare grossi svantaggi ad una agente che impiega molto tempo a risponde

I problemi precedentemente elencati, insieme ai limiti computazionali e di risorse a disposizione degli agenti, obbligano quest'ultimi a valutare attentamente i vantaggi e gli svantaggi che derivano dall'impiego di tempi e risorse durante la negoziazione.

Come già visto in precedenza il CNP consente agli agenti di fare offerte per diversi contratti. Tuttavia, avendo risorse limitate, gli agenti non saranno in grado di onorare tutti i contratti. Il CNP non tiene conto dei rischi in cui gli agenti incorrono impegnandosi in attività che non saranno poi in grado di eseguire. In questo protocollo tali rischi sono alleggeriti dal fatto che un agente ha la possibilità di recedere dal contratto attraverso decommitment.

Il valore della penalità in caso di decommitmet influenza il comportamento degli agenti in tal modo: un agente può permettersi d'ignorare la possibilità che il suo partner receda dal contratto se la penalità che dovrebbe pagare è troppo alta, viceversa un agente ignorerà la possibilità di recedere dall'accordo se la penalità che dovrebbe pagare è troppo alta.

### **3.2.4 Nuovi Approcci alla Negoziazione**

Il CNP, come abbiamo già visto, consentiva di negoziare solo un task alla volta. La sua estensione proposta da Sandholm nel 1993 [13], consentiva all'agente annunciatore di negoziare più task contemporaneamente raggruppandoli in un insieme di compiti. Questo nuovo protocollo consente ad entrambe le parti interessate nella negoziazione di raggruppare task, in ogni fase della negoziazione. All'interno di tale modello di protocollo viene anche introdotto il meccanismo di scambio di task, durante il quale un primo agente alloca alcuni task ad un secondo agente, e quest'ultimo alloca alcuni suoi task al primo agente. Tale scambio può essere esplicitamente implementato attraverso un protocollo che consenta fra alternative del messaggio, di esplicitare quali task devono essere contrattati in uscita e quali in ingresso. In alternativa, in protocolli in cui lo swap non è esplicitamente implementato, gli agenti dovranno assumersi dei rischi e realizzare lo swap attraverso una sequenza di contratti che consento di trasferire i task in un solo senso alla volta.

In tale modello di negoziazione si propone l'utilizzo di protocolli multi-agente come soluzione al problema degli ottimi locali. Si descrivono due diverse implementazioni che consentono di realizzare una multi-negoziazione, ovvero la negoziazione tra più di due agenti.

Il prima implementazione proposta è l'*Explicit multiagent contract* dove gli agenti vengono considerati come degli operatori in uno spazio globale di allocazione dei task. Per prima cosa l'agente deve identificare i benefici di un potenziale contratto multi-agente. Secondariamente, il protocollo deve consentire di effettuare un contratto fra più agenti. Questo può essere realizzato facendo circolare fra gli agenti interessati il messaggio di negoziazione. Il contratto diventa valido solo se ogni agente coinvolto sottoscrive quest'ultimo.

Una seconda implementazione è rappresentata dal *Multiagent contract through mutual contracts*. Quest'ultima, ha origine semplicemente dalla considerazione che un contratto multi agente è equivalente ad una sequenza opportuna di contratti mutuali. Ovviamente una simile implementazione comprende rischi per gli agenti coinvolti. Tuttavia, il protocollo fin'ora descritto, fornisce meccanismi per rendere minori tali rischi, rendendo il valore della penalità in caso di decommitment funzione del tempo impiegato nella contrattazione.

### **3.2.5 Tragedy of the Commons: il problema della congestione**

Infine, in questa secondo modello di estensione del CNP, viene ripreso il problema della congestione dei messaggi. Come visto precedentemente il CNP considerò solo parzialmente tale problema: in caso di basso traffico l'agente che ha dei task d'allocare annuncia una serie di proposte, in caso invece d'alto traffico d'interazione, è l'agente che ha le risorse libere ad annunciare la sua disponibilità ad eseguire i task. Il modello proposto da Sandholm nel 1993 [13], risolveva tale problema semplicemente ignorando gli annunci considerati ormai troppo "vecchi", per essere valutati. In questo nuovo protocollo ci si rende conto che, in ambienti aperti come Internet questi strumenti non sono sufficienti ad assicurare dai rischi della congestione. Gli agenti, infatti, essendo entità SI, se ritengono che l'allocazione di determinati task risulti vantaggiosa ai loro scopi, inviano annunci anche se quest'ultimi rischiano di saturare altri agenti. La *Tragedy of the Commons*, così viene definito il problema della congestione, può trovare soluzione attraverso un reciproco controllo fra gli agenti. Quest'ultimi, infatti, possono controllare quanti annunci ricevono da un certo agente. Se questo supera un certo valore dovrà essere punito. Tale meccanismo tra l'altro consente anche le soluzioni precedentemente proposte.

Concludendo, questa seconda estensione del CNP, ha consentito di ampliare l'efficienza del protocollo di negoziazione. Problematiche come la congestione da annunci, o la negoziazione di unità atomiche di task sono state riprese e ampliate rispetto alla prima estensione [13]. Inoltre si sono introdotti nuovi meccanismi di negoziazione, come la dinamicità della fase di commitment ed i contratti multi-agenti. Tutto ciò è stato contestualizzato, in un sistema di agenti SI, tenendo in considerazione inoltre della loro razionalità limitata.

### 3.3 Protocollo di negoziazione Task-Swap

Golfarelli, insieme ad altri studiosi, studiò, nel 1997 [15], un modello di protocollo per la negoziazione in domini in cui lo scambio di task è l'unico tipo di contratto possibile.

Tali ricercatori partirono dalla considerazione che le tecniche di cooperazione adottate fra gli agenti, dipendono strettamente dalla natura di quest'ultimi: un agente "benevolo" (*benevolent agent*) tenderà a massimizzare l'utilità e l'interesse globale della società degli agenti; viceversa, un agente SI, è portato a massimizzare i propri interessi e profitti, quindi sarà disposto a cooperare solo nel caso in cui, tale cooperazione, gli rechi dei vantaggi.

All'interno del protocollo Task-Swap gli agenti sono considerati SI e con limitate capacità di comunicazione; essi posseggono differenti funzioni di valutazione dei costi, hanno diverse abilità, e mirano a raggiungere scopi diversi e spesso in conflitto fra loro. Questa eterogeneità, fa sì che gli agenti non siano in grado di stabilire una comune unità di misura monetaria, e non abbiano gli strumenti per stimare il valore di un servizio fornito da un altro agente. In domini in cui tutte queste considerazioni si verificano, l'utilizzo di un protocollo basato sulla vendita (*sale-based*), come quelli proposti da *Sandholm e Lesser* precedentemente in [13] e [14], diventa improponibile, e il Task-Swap rappresenta l'unico mezzo di negoziazione.

In molti ambiti e settori di applicazione, il protocollo Task-Swap risulta essere, quindi, più adatto e più vantaggioso rispetto al protocollo *sale-based*. In particolare, in [15], si descrive tale protocollo applicandolo alla pianificazione dei percorsi degli agenti autonomi robotici. Lo scopo è di descrivere come, il costo delle operazioni che devono essere eseguite da questa tipologia di agenti, può decrementare attraverso lo scambio dei task che gli sono stati assegnati.

All'interno dello scenario proposto, con il termine di agente, si andrà ad identificare un robot capace di muoversi autonomamente all'interno di un ambiente di cui ha una conoscenza solo parziale. Ogni agente è capace di eseguire azioni specifiche e possiede una personale e incompleta visione dell'ambiente esterno nel quale opera. Tale conoscenza dell'ambiente è organizzata in più strati che corrispondono,

rispettivamente, a differenti livelli d'astrazione. Ciascuno di questi livelli possiede una struttura ad albero, composta da nodi ed archi, che rappresentano rispettivamente posizioni e collegamenti fra queste. All'interno del grafo, al più basso livello di astrazione, i nodi corrispondono a *landmarks* (pietre di confine) che servono per distinguere i luoghi nell'ambiente; nel più alto livello di astrazione ad ogni nodo corrisponde un *cluster*. Un'analisi più dettagliata di tale modello, può essere trovata in [16].

La pianificazione di percorsi (*path planning*) per agenti robotici, mira ad individuare quali percorsi consentono ad un agente di raggiungere una serie di obiettivi collegati fra loro. Lo scopo è quello di determinare all'interno di una rete di luoghi connessi fra di loro, rappresentata attraverso un grafo ad albero, la sequenza opportuna con la quale, tali luoghi, andranno visitati per portare a termine correttamente l'insieme di task nel miglior modo possibile.

In tale contesto esistono solitamente due tipologie di limitazioni: restrizioni sulle risorse a disposizione dell'agente e restrizioni temporanee relative all'esecuzione di un determinato task.

Ad ogni robot è affidata una missione, rappresentata da un insieme di task che devono essere eseguiti con successo. Si assume, in particolare che ogni task:

- implichi un'azione;
- debba essere eseguito su una determinata risorsa. Con il termine risorsa si indica un *landmark* o *cluster*, dove le azioni dovranno essere eseguite; una risorsa può anche essere costituita da un insieme di landmark o cluster, in tal caso i task possono essere eseguiti indifferentemente su uno di questi (per esempio se la risorsa è costituita da un insieme di terminali, il task può essere eseguito su uno qualsiasi di questi);
- può essere costretto ad essere eseguito entro un limitato intervallo di tempo (*time window*); un esempio è rappresentato dal caso in cui determinate risorse sono disponibili per un tempo limitato;
- può essere legato a restrizioni di precedenza rispetto ad altri task: il task A non può, per esempio, essere eseguito prima che siano stati eseguiti i task B e C;

- può richiedere o produrre uno o più oggetti (*objects*), per esempio documenti o floppy disk.

Un agente, quindi, per portare a termine la sua missione, deve individuare e pianificare un percorso che gli consenta di visitare tutte le risorse necessarie per l'esecuzione dei suoi task, in conformità con le restrizioni imposte. La qualità di un percorso viene valutata in base ad una funzione di costo. Tale funzione può dipendere dalla lunghezza del percorso o dal tempo che esso richiede per essere eseguito. Se indichiamo con  $c_A(p)$  il costo del percorso  $p$  valutato dall'agente  $A$ , lo scopo di quest'ultimo sarà ovviamente, minimizzare tale costo. Maio e Rizzi, nel 1996, proposero un algoritmo capace di determinare il percorso a basso costo, indicato come  $path(S)$ , in grado di consentire l'esecuzione di un dato insieme di task  $S$ ; questo algoritmo verrà riutilizzato da ogni agente, nel protocollo Task-Swap, per determinare il percorso di esecuzione dei task e, nel contempo, valutare l'utilità e l'efficienza del protocollo.

### 3.3.1 Utilità e Costi Marginali

In tale protocollo, viene ripresa la teoria matematica sul calcolo dei costi marginali già trattata in [13], includendovi, inoltre, la stima dell'utilità dello scambio di task per un dato agente. Di seguito daremo una breve descrizione di tale teoria.

Sia  $A$  un agente e  $S$  un suo insieme di task correnti. Dato un task  $s$  viene definito il suo costo marginale relativo (*relative marginal cost*) RMC rispetto all'agente  $A$ , come il costo pagato da  $A$  per eseguire  $s$ . In particolare:

$$rmc(s, A, S) = c_A(path(S)) - c_A(path(S - \{s\})), \text{ se } s \text{ appartiene a } S$$

o

$$rmc(s, A, S) = c_A(path(S + \{s\})) - c_A(path(S)), \text{ se } s \text{ non appartiene a } S$$

Se un agente possiede un task che non è in grado di eseguire allora, l'RMC di quel task, avrà valore finito per l'agente.

E' importante notare la differenza semantica dell'RMC nel protocollo Task-Swap rispetto al protocollo basato sulla vendita. Nel primo, è utilizzato per selezionare i o il task che devono essere inseriti negli annunci dell'agente, nel secondo è uno strumento utilizzato per la stima monetaria di un task.

La convenienza di uno swap, viene valutata attraverso il calcolo della sua utilità. Si definisce utilità (*utility*) per un agente A, nello scambio di un suo task s1 appartenente ad insieme di task S, con un task s2 appartenente ad un agente B come:

$$uti (s1, s2, A, S) = cA (path (S)) - cA (path (S + \{s1\} - \{s2\}))$$

Dalle definizioni sopra enunciate, si può verificare che:

$$uti (s1, s2, A, S) = (rmc (s2, A, S) - rmc (s1, A, S - \{s2\}))$$

Si dice che uno scambio di task è individualmente razionale (*individual rational, IR*) per un agente, se la sua utilità è positiva. Sia il valore del RMC di una dato task, che dell'utilità di uno swap, possono differire a seconda dell'agente che li esegue. Ciò è dovuto principalmente al fatto che agenti diversi hanno differenti capacità e diversi task da eseguire.

### 3.3.2 Ciclo di Negoziazione

La negoziazione fra gli agenti avviene attraverso l'esecuzione di un ciclo di operazioni. Durante ogni sessione di negoziazione, ciascun agente prova a scambiare gli opportuni task correnti. Una volta terminata la sessione, gli agenti individuano un percorso da seguire per l'esecuzione dei task. Dopo che la prima risorsa è stata raggiunta, e il corrispondente task è stato eseguito, inizia una nuova negoziazione per i task rimanenti.

Nella precedente estensione del CNP [15], basata sulla vendita, furono definiti diversi livelli di commitment, nei quali ogni task era associato ad una certa penalità in denaro che l'agente doveva pagare, nel caso in cui, avesse voluto recedere dal contratto. Tale impostazione consentiva ad un agente di poter partecipare a più negoziazioni contemporaneamente.

Nel protocollo Swap-Task, il concetto di penalità non può trovare applicazione; ogni negoziazione è influenzata dall'esito delle precedenti, quindi, l'utilità di uno swap, dipende anche dal fatto alcuni task siano già stati rimossi o aggiunti all'insieme corrente. Per questo, e per altri motivi, in tale protocollo si assume che un agente possa effettuare una sola offerta alla volta.



Similarmente al CNP, anche il protocollo Task-Swap si sviluppa in tre fasi:

- *Fase di annuncio*: l'agente annunciatore formula un annuncio, trasmettendolo agli altri agenti; all'interno di tale annuncio sono contenuti tutti i task che l'agente è interessato a scambiare.
- *Fase d'offerta*: gli agenti che hanno ricevuto l'annuncio, formulano un'offerta (bid) e la inviano all'agente annunciatore. Tale offerta contiene tutti i task posseduti dagli agenti offerenti, che quest'ultimi sono interessati a scambiare con quelli indicati nell'annuncio. Lo swap è proposto solo se è individualmente razionale. L'agente offerente non può fare offerte per altri annunci e non può a sua volta fare annunci, finché non ha ricevuto una conferma o un rifiuto per la sua ultima offerta.
- *Fase di assegnamento*: l'agente annunciatore attende le risposte per un determinato periodo di tempo. Al termine di tale tempo, l'agente seleziona fra tutte le offerte pervenutegli quella che ha utilità maggiore. Se tale valore è positivo, implica che lo swap può essere considerato IR per entrambe le parti coinvolte nella negoziazione. Un messaggio di award è trasmesso all'offerente selezionato e lo swap è confermato. A tutti gli altri agenti offerenti, viene inviato un messaggio di rifiuto dell'offerta (acknowledgment). La sessione di negoziazione è conclusa.

Lo scambio di task può essere limitato da alcune restrizioni di precedenza, come già affermato precedentemente. Consideriamo, per esempio, un agente A al quale sono stati assegnati una coppia di task  $s_1$ ,  $s_2$  e, tale che,  $s_1$  deve essere negoziato prima di  $s_2$ . Se si negozia prima  $s_2$  con un agente B si viene a creare una dipendenza indesiderata fra i percorsi che gli agenti A e B devono compiere. Di seguito descriveremo la soluzione a tale problema. Prima di negoziare, l'agente A deve determinare il suo piano di esecuzione attraverso l'algoritmo *path ()*; in tal modo si pianifica esattamente, in quale istante ogni task deve essere negoziato. Detto  $t$  il tempo programmato per  $s_1$ , nel proporre  $s_2$  l'agente A limita la sua finestra temporale a periodo che va da  $t$  ad infinito, forzando in tal modo l'agente B ad eseguire  $s_2$  dopo  $s_1$ .

Precedentemente, abbiamo affermato che un task, per essere eseguito, può richiedere o produrre uno o più oggetti. Dal punto di vista della pianificazione dei programmi, tali oggetti rappresentano delle restrizioni sul task. E' possibile tradurre tali costrizioni, in una limitazione di precedenza attraverso la distinzione, per ogni oggetto  $o$ , dei seguenti tre tipi di task:

- *sorgenti (source)*: ovvero task che producono ma non richiedono oggetti;
- *pozzi (sinks)*: ovvero task che richiedono ma non producono oggetti;
- *pompe (pumps)*: ovvero task che richiedono e producono contemporaneamente oggetti.

Il generale un task  $s$  è tradotto nel modo seguente:

- se  $s$  è una sorgente per  $o$ , il suo insieme di precedenze rimane immutato;
- se  $s$  è un pozzo per  $o$ , il suo insieme di precedenze vengono aggiunti tutti i task sorgenti e i pompe per  $o$  ;
- se  $s$  è una pompa per  $o$ , il suo insieme di precedenze vengono aggiunti tutti i task sorgenti per  $o$ .

Le tecniche descritte precedentemente per risolvere i problemi derivanti da limitazioni di precedenze dei task, non valgono nel caso in cui tali limitazioni, derivino da limitazioni di oggetti. Consideriamo, per esempio, un agente A al quale è stata assegnata la coppia di task  $s_1, s_2$ , tale che  $s_1$  produce uno o più oggetti necessari a  $s_2$ . Se uno dei due task, attraverso una negoziazione, è assegnato ad un agente B, è necessario che dopo l'esecuzione di  $s_1$  e prima che inizi quella di  $s_2$ , i due agenti s'incontrino, per scambiarsi gli oggetti necessari. Di seguito, si descriveranno alcune tecniche di incontro fra due agenti SI:

- a. *Propongo quindi pago (Propose-hence-pay)*: l'agente annunciatore s'impegna a raggiungere l'agente al quale è stato assegnato il task, che perciò non deve supportare costi addizionali;
- b. *Richiedo quindi pago (Require-hence-pay)*: l'agente che possiede il task che richiede gli oggetti, s'impegna a raggiungere l'altro agente; in tal modo, l'agente che possiede il task che produce gli oggetti non deve supportare costi addizionali.
- c. *Distribuito (Distribuite)*: i due agenti accettano di incontrarsi a metà strada tra i due luoghi dove  $s_1$  ed  $s_2$  devono essere eseguiti; in tal modo i costi di trasferimento degli oggetti sono distribuiti equamente fra le due parti.

Le prime due strategie possono fallire in due casi: quando lo stesso task è scambiato due o più volte tra diversi agenti, e quando i task che rispettivamente producono e richiedono un dato oggetto, sono scambiati con diversi agenti. In entrambi i casi, infatti, può accadere che un agente si trovi a ricevere o a distribuire oggetti che non richiede né produce. La terza strategia non presenta, invece, questo problema poiché il luogo d'incontro è definito a priori indipendentemente dagli agenti coinvolti.

### **3.3.3 Strategia di Negoziazione**

La strategia adottata nel protocollo Task-Swap è descritta di seguito.

L'agente annunciatore, nella fase di selezione dei task da negoziare, non riesce a valutare l'utilità dello swap, conseguentemente adotta un criterio euristico basato sul valore dell'RMC. L'agente sceglierà il task con RMC maggiore, tenendo conto di quando quel task dovrà essere eseguito nel suo piano di esecuzione. Tale considerazione è necessaria poiché, il trasferimento di un task programmato alla fine del percorso, risulterebbe una mossa molto azzardata, in quanto, il suo RMC potrebbe subire sostanziali modifiche durante l'esecuzione degli altri task.

Dopo aver ricevuto l'annuncio l'agente offerente si preoccupa di calcolare le utilità che può ricavare scambiando i task specificati nell'annuncio, con i propri. Dopo aver scartato gli swap non IR, decide come formulare l'offerta. La strategia più "politicamente corretta" sarebbe quella in cui l'agente include, entro la sua offerta, tutti gli swap identificati come IR. Questa strategia, infatti, consente di massimizzare le probabilità di successo di un contratto e fornisce maggiori utilità all'agente annunciatore, a scapito purtroppo dell'utilità dell'agente offerente. In alternativa, quest'ultimo può decidere di includere, nella sua offerta, solo gli scambi per lui più vantaggiosi. In tal modo, l'agente rinuncia ad una maggiore probabilità di successo della negoziazione, a favore di un contratto più vantaggioso.

Il protocollo Task-Swap, presenta alcuni vantaggi rispetto a quello basato sulla vendita, in termini di stabilità del protocollo stesso. Il CNP basato sulla vendita, proposto da Sandholm nel 1993, infatti, non è stabile. Consideriamo, per esempio, un agente annunciatore. Quest'ultimo invia messaggi di annuncio all'interno dei quali specifica i task che vorrebbe negoziare e il massimo prezzo  $v_A$  che è disposto a pagare per quest'ultimi. L'agente offerente, che indicheremo con **B**, per eseguire i

task contenuti nell'annuncio deve sostenere una spesa  $v_B < v_A$ . Tuttavia, l'agente B, può decidere di tentare di incrementare la propria utilità a scapito dell'agente A. Infatti, conoscendo il valore di  $v_A$ , può richiedere nella sua offerta un qualsiasi prezzo  $v^*$  compreso fra  $v_B \leq v^* < v_A$ , senza andare a compromettere il fatto che il contratto sia IR per l'agente A. Ciò significa che l'agente B è stimolato a mentire all'agente A per trarne profitto, a scapito ovviamente di minori probabilità di successo del contratto.

Tale problema, non può presentarsi nel protocollo Task-Swap, poiché non esistendo una comune metrica di valutazione dell'utilità, gli agenti nel fare offerte meno vantaggiose, rischiano sia di avere meno opportunità di successo, sia di fare proposte che non risultino più IR per l'altro agente.

Attraverso risultati sperimentali, in [15] si dimostra che, l'utilizzo del protocollo Task-Swap nella negoziazione fra gli agenti, consente di ridurre i costi di esecuzioni dei task all'interno di una società di agenti SI e limitati razionalmente.

Concludendo, tale protocollo risulta essere un interessante confronto, rispetto ai protocolli di negoziazione precedenti, comunemente basati principalmente sulla vendita di task.

### **3.4 OCSM-Contract**

Nel 1998, Sandholm in [17], dopo aver portato avanti i suoi studi sulla negoziazione fra agenti intelligenti, pubblicò una nuova estensione del CNP proponendo quattro diversi tipi di contratto.

All'interno del suo nuovo lavoro, Sandholm incluse il concetto, già ripreso in [15], di razionalità individuale. Un contratto, in base alla definizione di Sandholm, si può definire individualmente razionale (IR) per un agente, se comporta un miglioramento delle condizioni economiche di quest'ultimo. Riprendendo l'estensione del CNP, da lui stesso pubblicata in [14] basata sulla valutazione dei costi marginali, Sandholm ripropone un protocollo nel quale gli agenti propongono contratti ad altri agenti, ed accettano o rifiutano proposte basando le loro decisioni sulla stima dei costi marginali. Inoltre gli agenti hanno la possibilità di riallocare dei

task che hanno ricevuto, precedentemente, attraverso un contratto, da altri agenti. La negoziazione può terminare in un qualsiasi istante e il valore (definito come la somma dei pagamenti ricevuti scorporati dei costi sostenuti per l'esecuzione dei task) delle operazioni di ogni agente, incrementa in modo uniforme. Ciò consente al benessere sociale degli agenti di aumentare a sua volta in maniera uniforme.

Anche in questo caso si cerca di formulare una soluzione per far fronte al problema di allocazione dei task in un sistema di agenti intelligenti SI e con capacità e risorse limitate. A tale problema viene data la seguente formulazione formale:

*Il problema di allocazione dei task può essere definito attraverso un insieme di task  $T$ , un insieme di agenti  $A$ , una funzione di costo  $C_i$  (la quale esprime il costo che l'agente  $i$  deve sostenere nello svolgere un particolare sottoinsieme di task), ed attraverso un'iniziale allocazione dei task fra gli agenti del sistema.*

Gli studi pubblicati da Sandholm, mirano ad estendere il CNP, attraverso la definizione di quattro tipi di contratto, *contratto per un singolo task (single task contract)*, *contratto di raggruppamento (cluster contract)*, *contratto di scambio (contract swaps)*, *contratto multi agente (multi-agent contract)* e, attraverso l'introduzione di un nuovo tipo di contratto, ovvero il *OCSM-contract*.

Il contratto, che prevede la negoziazione di un task alla volta, abbreviato con O-contracts, e viene definito formalmente nel seguente modo:

*Un O-contract è definito attraverso una coppia  $(T_{i,j}; P_{i,j})$ , dove  $T_{i,j}$  in valore assoluto vale uno.  $T_{i,j}$  rappresenta il insieme di task (in tal caso includente un solo task), che l'agente  $i$  assegna ad un agente  $j$ .  $P_{i,j}$  rappresenta il prezzo contrattato che l'agente  $i$ , deve pagare all'agente  $j$  per la gestione del insieme di task.*

Attraverso una serie di osservazioni Sandholm arriva a concludere che, l' O-contract è in grado di individuare sempre un percorso breve per l'allocazione ottimale dei task, ma nel caso in cui il percorso richiesto sia totalmente IR tale contratto può fallire.

Il contratto di raggruppamento consente di negoziare più task contemporaneamente viene definito nel modo seguente:

*Un contratto cluster (C -contract) è definito attraverso una coppia  $(T_{i,j}; P_{i,j})$ , dove il valore di  $T_{i,j}$  in valore assoluto è maggiore di uno.  $T_{i,j}$  rappresenta il insieme di task che l'agente  $i$  assegna ad un agente  $j$ .  $P_{i,j}$  rappresenta il prezzo contrattato che l'agente  $i$ , deve pagare all'agente  $j$  per la gestione del insieme di task.*

Il C-contract evita alcuni ottimi locali del O-contract (viceversa il O-contract evita alcune local optima del C-contract). Un gruppo può racchiudere un qualsiasi numero di task. L'allocazione di un di un task è definita  $k$  ottimale se nessun contratto cluster vantaggioso dei  $k$  task, può essere realizzato fra due agenti. Si noti che, una  $m$  ottimalità non implica una  $n$  ottimalità anche se  $m < n$ . Ovviamente non vale neppure il contrario. Purtroppo anche il C-contract non può portare ad un'ottima allocazione dei task:

*Esistono istanze del problema di allocazione dei task dove nessun percorso (IR o no) di un C-contract porta a raggiungere, a partire dalle allocazioni iniziali, l'allocazione ottima.*

Il contratto di scambio, che consente lo scambio reciproco di task fra due agenti, viene definito nel modo seguente:

*Un contratto di scambio (S-contract) è definito attraverso una quadrupla  $(T_{i,j}; T_{j,i}; P_{i,j}; P_{j,i})$ , dove  $T_{i,j}$  in valore assoluto è uguale a  $T_{j,i}$  in valore assoluto che è uguale ad uno.  $T_{i,j}$  rappresenta il insieme di task (in tal caso includente un solo task), che l'agente  $i$  assegna ad un agente  $j$ .  $T_{j,i}$  rappresenta il insieme di task (in tal caso includente un solo task), che l'agente  $j$  assegna ad un agente  $i$ .  $P_{i,j}$  rappresenta il prezzo contrattato che l'agente  $i$ , deve pagare all'agente  $j$  per la gestione del insieme di task  $T_{i,j}$ ;  $P_{j,i}$  rappresenta il prezzo contrattato che l'agente  $j$ , deve pagare all'agente  $i$  per la gestione del insieme di task  $T_{j,i}$ .*

L' S-contract evita alcuni ottimi locali del C-contract e del O-contract (viceversa O-contract e C-contract evitano alcune ottimi locali del S-Contract).

Tale protocollo è adatto per la negoziazione sia di agenti cooperativi che per agenti SI. Si suppone, infatti, che i primi si occupino dei task degli altri agenti senza richiedere un compenso ogni qual volta sia necessario per il benessere della comunità degli agenti. I secondi, invece, possono utilizzare il contratto di scambio,

realizzando una specie di baratto. Ovviamente in tal caso, lo scambio di task avverrà solo se questo risulta essere vantaggioso per entrambe le parti in causa. Tuttavia, questo tipo di negoziazione è resa difficile dal fatto che, non sempre esiste la possibilità di avere baratti vantaggiosi per entrambe le parti; inoltre identificare uno scambio con baratto è molto più complesso rispetto ad una qualsiasi altra via di scambio dei task, specialmente in ambienti distribuiti. Sebbene quindi l'S-contract sia necessario per raggiungere un'allocazione ottimale dei task, non è sufficiente.

*Esistono istanze del problema di allocazione dei task dove nessun percorso (IR o no) di un S-contract porta a raggiungere, a partire dalle allocazioni iniziali, l'allocazione ottima.*

Consideriamo, per esempio, agenti che posseggono un differente numero di task nella loro iniziale allocazione. Se l'allocazione ottima richiede che tale numero iniziale di task venga modificato, allora non potrà mai essere raggiunta attraverso un S-protocol, poiché quest'ultimo preserva il numero dei task posseduti da un agente.

Il protocollo multi agente consente, a differenza dei protocolli precedentemente descritti, di trasferire i task fra più di due agenti. Tale contratto viene definito da nel modo seguente:

*Un contratto multi agente (M-contract) è definito attraverso una coppia  $(T; P)$  di una matrice  $A \times A$ , dove gli ultimi tre elementi di  $T$  non sono pieni (altrimenti si avrebbe a che fare di nuovo con un contratto fra due agenti), e per ogni  $i$  e  $j$  sia ha che il modulo di  $T_{i,j}$  è minore o uguale ad uno. Un elemento  $T_{i,j}$  rappresenta l'insieme di task che l'agente  $i$  assegna ad un agente  $j$ .  $P_{i,j}$  rappresenta il prezzo contrattato che l'agente  $i$ , deve pagare all'agente  $j$  per la gestione del insieme di task.*

Al contratto multi agente, mancano alcuni ottimi locali presenti nei O-, C- ed S-contract. (viceversa O-, S-, C-contract non posseggono alcuni ottimi locali del M-contract). Sebbene l'M-contract sia un contratto necessario per ottenere un'ottima allocazione dei task, non è sufficiente:

*Esistono istanze del problema di allocazione dei task dove nessun percorso (IR o no) di un M-contract porta a raggiungere, a partire dalle allocazioni iniziali, l'allocazione ottima.*

Consideriamo per esempio, due agenti che devono negoziare un task. Poiché gli agenti sono solo due l' M-contract non potrà essere usato. Quindi, se l'agente che possiede il task ha un costo di gestione per quest'ultimo più alto rispetto all'altro agente, e tale costo è diverso da zero allora, l'allocazione ottima non potrà essere raggiunta.

Riassumendo i quattro contratti, fin'ora proposti, presentano tutte le caratteristiche necessarie al raggiungimento dell'allocazione ottima dei task attraverso un contratto IR, ma singolarmente non sono sufficienti. Ognuno dei quattro protocolli è carente di alcune ottimizzazioni locali presenti, viceversa, negli altri protocolli.

Fu proposta, quindi, l'idea di fondere in un unico contratto i quattro tipi precedentemente descritti. Tuttavia, anche un protocollo di contrattazione equipaggiato di tutti e quattro i tipi di contratti, mancherebbe di qualche ottimo locale. Sandholm, introdusse un nuovo tipo di contratto di negoziazione chiamato OCSM-contract, il quale combina le caratteristiche dei O-, C-, S-, M-contract in un unico tipo di contratto, dove le idee alla base dei quattro precedenti tipi di contratto possono essere applicate simultaneamente. Si definisce l' OCSM-contract nel modo seguente:

*Un OCSM-contract è definito attraverso una coppia  $(T; P)$  di una matrice  $A \times A$ . Un elemento  $T_{i,j}$  rappresenta l' insieme di task che l'agente  $i$  assegna ad un agente  $j$ .  $P_{i,j}$  rappresenta il prezzo contrattato che l'agente  $i$ , deve pagare all'agente  $j$  per la gestione del insieme di task.*

Attraverso l'utilizzo di tale contratto, esiste sempre un percorso che conduce dall'iniziale allocazione dei task all'ottima allocazione di quest'ultimi. Quindi, attraverso un numero finito di contratti OCSM, si può raggiungere la soluzione ottima al problema dell'allocazione dei task. Questo sta a significare che gli agenti che hanno come prospettiva la massimizzazione del benessere sociale, possono accettare i contratti IR che gli sono offerti senza dover, per esempio, aspettare eventuali contratti più vantaggiosi.

Concludendo l'OCSM-contract, rappresenta un potentissimo strumento per risolvere piccole istanze del problema di allocazione dei task. Se il problema si allarga, infatti, il numero di contratti necessari prima di poter raggiungere la soluzione ottimale, raggiunge livelli irrealizzabili.



## Capitolo 4

### Nuovi meccanismi alla base dei Protocolli di Negoziazione

La negoziazione *multi-linked* [17], implica una situazione in cui un agente richiede di negoziare con più agenti, riguardo a più elementi che sono in relazione fra loro, quali per esempio risorse, task ecc...

La negoziazione su uno di questi elementi, in particolare, ha la capacità di influenzare le negoziazioni su gli altri elementi. Una struttura di negoziazione *multi-leveled* [18] implica un processo di negoziazione che si sviluppa su due livelli. Il livello superiore, concerne con le informazioni di alto livello come gli obiettivi degli agenti e le decisioni sul negoziare o meno con altri agenti allo scopo di raggiungere un particolare obiettivo. La negoziazione a tale livello, comporta il difficile scopo della costruzione del commitment e della determinazione dei costi di decommitment. Il livello inferiore concerne la fattibilità e la flessibilità dell'implementazione di una determinata operazione, riguarda, quindi, l'analisi dettagliata dei task candidati, delle azioni, dei dettagli temporali e la specifica dei commitment fra gli agenti. La negoziazione a tale livello si esplica in un raffinamento dei "grezzi" commitment proposti al livello superiore.

In tale capitolo presenteremo, inoltre, il meccanismo di negoziazione *multi-step*[19], che insieme ai precedenti meccanismi, sta alla base dei moderni protocolli di negoziazione fra gli agenti intelligenti. In particolare, presenteremo i protocolli multi-step applicati al problema di allocazione dei task tra agenti cooperativi. Il meccanismo del multi-step si basa sull'utilizzo della teoria del calcolo dei costi marginali e dell'utilità di una determinata operazione, allo scopo di individuare una soluzione che sia in grado di massimizzare l'utilità finale.

#### 4.1 La negoziazione multi-leveled

Solitamente, la negoziazione fra agenti, è sempre stata proposta sviluppata su un unico livello: dalla proposta al commitment finale, tutti gli elementi come, la qualità

raggiunta e il prezzo di ricompensa erano determinati in un unico processo. Questa tipologia di negoziazione, tuttavia rischia di richiedere elevati sforzi da parte degli agenti, nel caso in cui ci si trovi a che fare con le seguenti situazioni: negoziazioni multi-task, dove i task sono raggiungibili attraverso vie differenti con sequenze di attività alcune delle quali possono richiedere risorse sia esterne che interne. Le incertezze derivanti da un ambiente di operazione aperto come internet, possono complicare ulteriormente il processo di negoziazione. Una tecnica per cercare di ridurre la complessità del processo di negoziazione è stata descritta da [19], e riguarda lo sviluppo della struttura della negoziazione su due livelli. Il livello superiore, concerne con informazioni di alto livello come gli obiettivi degli agenti o le scelte di negoziare o meno con altri agenti allo scopo di raggiungere un particolare obiettivo. La negoziazione a tale livello, comporta il raggiungimento di un primo “grezzo” commitment e della determinazione dei costi di decommitment. Il livello inferiore concerne la fattibilità e la flessibilità dell’implementazione di una determinata operazione riguarda, quindi, l’analisi dettagliata dei task candidati, delle azioni da compiere e la specifica dei commitment fra gli agenti. La negoziazione a tale livello si esplica in un raffinamento dei “grezzi” commitment proposti al livello superiore. Di seguito daremo una descrizione più approfondita della negoziazione multi-leveled.

Come già sottolineato, il livello superiore della struttura della negoziazione riguarda la pianificazione delle attività di alto livello dell’agente. Perciò, a tale livello, è richiesta una conoscenza approfondita di ogni attività. E’ possibile individuare due tipi di elementi relativi al processo di decisione (*decision-making process*). Alcuni elementi, hanno la capacità d’influenzare in maniera decisiva la selezione del programma locale di esecuzione dei task e coinvolgono l’utilità trasferita fra i due agenti. Tale prima tipologia, dovrebbe essere negoziata a livello superiore, attraverso la realizzazione di “grezzi” commitment. Esistono, invece, altri elementi, che avendo meno influenza sulla determinazione del programma locale e riguardando principalmente la struttura dettagliata delle attività di basso livello, non richiedono di essere valutati e decisi direttamente al livello superiore. Tali elementi sono:

- le relazioni interne tra sotto task appartenenti a differenti task di alto livello;

-le incertezze, relative alle caratteristiche di esecuzione, che non siano visibili a livello superiore;

-le richieste di risorse interne associate ai task di basso livello.

Considerando le precedenti problematiche, gli agenti hanno l'opportunità di modificare i loro commitment di alto livello attraverso una negoziazione di basso livello e possono, eventualmente riordinare le loro attività di basso livello allo scopo di ottimizzare i commitment e i loro programmi locali per evitare conflitti e raggiungere un'elevata utilità.

Il primo livello della struttura multi-leveled, verrà di seguito descritto attraverso l'uso della struttura MQ, mentre la struttura TAEMS, sarà usata a supporto del processo che avviene al livello inferiore. Ovviamente la struttura multi-leveled può essere applicata su altre strutture adatte alla rappresentazione dei task.

Nella struttura MQ, l'esecuzione dei task, contribuisce in modo qualitativo al raggiungimento di uno o più obiettivi dell'agente. Ogni agente possiede un insieme di MQs dette anche *quantità motivazionali (motivational quantities)* che insegue e accumula. I task MQ, rappresentano un'astrazione di un insieme ordinato di azioni primitive che l'agente può eseguire, e possono possedere una propria terminazione e un tempo d'inizio. Ogni task MQ, consiste di una o più alternative MQ, dove ogni alternativa corrisponde a diversi profili di prestazione dei task. Ogni alternativa richiede un certo tempo di esecuzione, e produce una certa quantità di MQs, chiamati insieme di produzione MQ (MQPS) e consuma una quantità di MQs chiamati invece insieme di consumo MQ (MQCS).

Il linguaggio TAEMS, è un linguaggio di modellazione di domini indipendenti, il quale ci consente di modellare dettagliatamente le attività degli agenti. I task primitivi, detti anche metodi, sono caratterizzati da tre proprietà: qualità (q), durata (d) e costo (c) quest'ultimo rappresentato attraverso una distribuzione probabilistica. Il modello MQ, descrive quindi l'organizzazione delle conoscenze degli agenti sull'utilità dei task, ma non supporta i modelli dettagliati di tali task e delle loro interazioni. Manca, inoltre, nella sua struttura, una rappresentazione delle caratteristiche d'incertezza e delle risorse richieste dai task. Tutto ciò trova invece, descrizione attraverso il linguaggio TAEMS. Ogni agente possiede, delle proprie attività locali, le quali rappresentano un insieme di task MQ, ognuno associato ad un

certo MQPS ed ad un certo MQCS. La fig.10 descrive le attività eseguite dai rispettivi livelli MQ e TAEMS. Il primo passo consiste nella costruzione di piani alternativi per i task MQ, ciascuno con una determinata qualità, durata e caratteristica di costo. Successivamente il programmatore dell'MQ genera un programma parziale contenente i task che l'agente dovrebbe provare ad eseguire, i piani di esecuzione di tali task e il loro ordine di esecuzione.

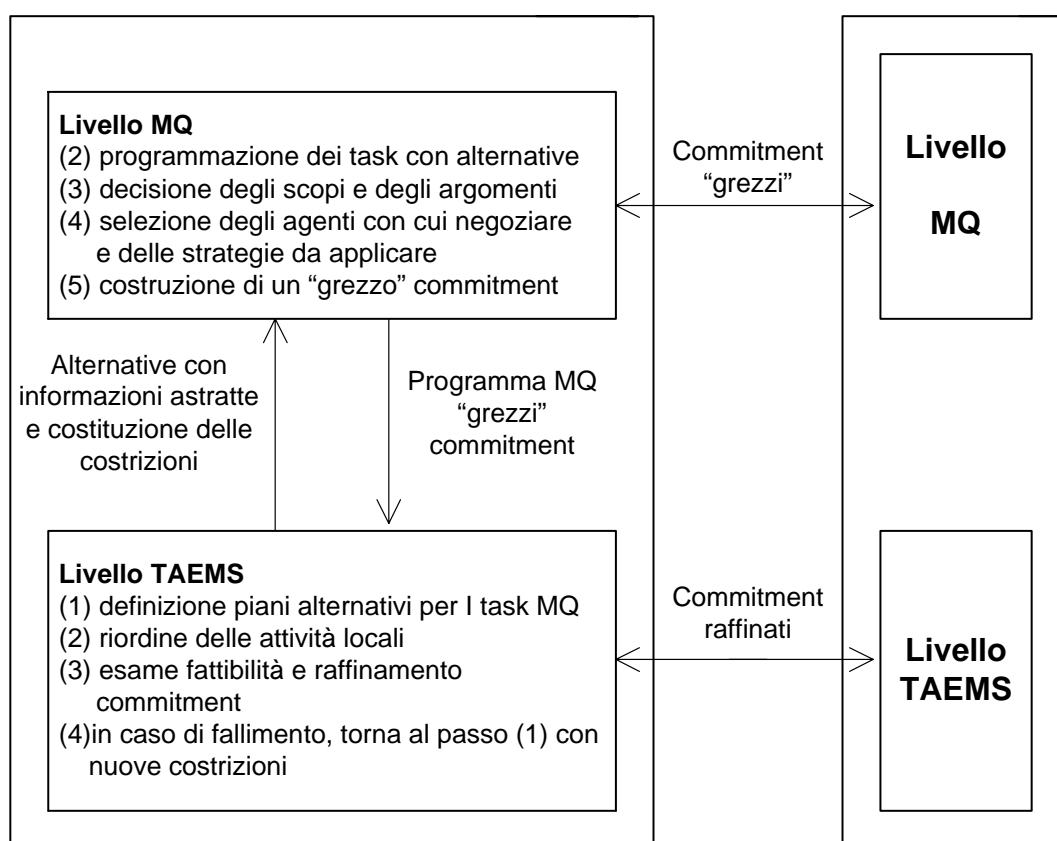


Fig. 10 Negoziazione strutturata su due livelli

Tale programma parzialmente ordinato, consente ulteriori raffinamenti al livello inferiore, dove sono disponibili informazioni più precise e dettagliate. Il programma così ottenuto, rappresenta la migliore scelta d'esecuzione delle attività per massimizzare l'utilità locale e per valutare l'utilità di uno specifico commitment. La terza fase consiste nel fissare gli scopi e gli argomenti della negoziazione. Successivamente, l'agente può decidere con quali altri agenti intende negoziare e la

strategia con la quale intende portare avanti la negoziazione. Come risultato finale delle operazioni compiute a livello MQ, si ha la produzione di un “grezzo“ commitment ovvero contenente solo specifiche parziali. Dopo di ciò, l’agente riordina le proprie attività locali sul livello TAEMS. Durante tale fase l’agente si occupa di ottimizzare il proprio programma locale avvantaggiandosi delle relazioni esistenti fra i metodi/task di basso livello. La fase successiva consiste in un’analisi riguardo alla realizzabilità del proprio programma locale ed in una nuova definizione, più precisa e dettagliata, dei commitment. Tale raffinamento è necessario in particolare quando:

- Esistono conflitti o potenziali conflitti tra commitment ed attività locali causati da ulteriori costrizioni, per esempio sulle risorse, o incertezze legate ad esecuzioni in tempo reale.
- Un’eventuale raffinamento dei commitment consente di ridurre i costi locali o di incrementare l’utilità locale.

Eventi inaspettati, possono causare conflitti nel processo di esecuzione. In tal caso l’agente può richiedere di controllare se tali conflitti possono trovare soluzione attraverso un raffinamento dei commitment. Nel caso in cui, tali conflitti non trovino in ogni modo soluzione, l’agente deve ritornare al livello MQ per costruire nuovi commitment proponendo nuove costrizioni.

La negoziazione al livello MQ presenta le seguenti problematiche:

- Per ogni elemento nella negoziazione, esistono varie caratteristiche che possono essere negoziate. La negoziazione risulta quindi multi dimensionale.
- Per ogni sessione di negoziazione, sono disponibili diversi protocolli. Ogni agente deve essere capace di individuare il protocollo più appropriato.
- Spesso la negoziazione di un elemento può influenzare l’esito di altre negoziazioni. L’agente deve essere in grado di decidere l’ordine opportuno delle negoziazioni e in che modo tali negoziazioni devono avvenire. Tale problema è studiato attraverso i protocolli di negoziazione multi-linked che descriveremo successivamente.

- L'agente deve sostenere le relazioni con gli altri agenti durante la negoziazione. Tale problema viene risolto attraverso l'utilizzo dell'MQ di relazioni, il quale rappresenta le reazioni fra gli agenti.

Un agente può anche decidere di pagare per poter avere una certa flessibilità locale, che gli consenta di gestire le future ed eventuali incertezze. Concludendo, la struttura di negoziazione a più livelli, consente agli agenti di ragionare riguardo ai complicati elementi della negoziazione ed alle sue incertezze, attraverso meccanismi modulari ed efficienti dal punto di vista computazionale. In tal modo si producono commitment sempre più precisi, che aumentano le probabilità di successo di una negoziazione. Inoltre consente agli agenti di ragionare riguardo agli aspetti organizzativi dell'implementazione dei propri obiettivi, e di prendere decisioni riguardanti la negoziazione e la rinegoziazione.

#### **4.2 Protocolli di negoziazione multi-linked**

La negoziazione multi-linked comporta multiple negoziazioni su elementi interdipendenti. In un ambiente distribuito, infatti, un agente può avere la necessità di negoziare attraverso i protocolli opportuni, più elementi come, per esempio, task o risorse. Gli agenti sono capaci di negoziare attraverso un protocollo multi-linked, in domini dove il numero di agenti è finito.

Anche in questo caso, all'interno dei protocolli di negoziazione multi-linked, è possibile delineare un agente *contractor*, il quale esegue task per conto di altri agenti in cambio di una ricompensa, e un agente *contractee*, il quale viceversa possiede dei task che desidera far eseguire da altri agenti pagando loro una ricompensa. Ovviamente, il contratto è stipulato solo se le due parti giungono ad un accordo. All'interno di tale contratto sono specificati il tempo d'inizio e di fine dell'esecuzione del task, il valore della ricompensa ed altri attributi. Nel caso in cui una delle due parti, non riesca a tener fede al contratto, e decida di recedere da quest'ultimo, dovrà pagare all'altro agente una penalità di decommitment.

In [18], si definiscono due tipologie di relazioni tra negoziazioni multiple su elementi interconnessi:

- *direttamente collegati (directly-linked)*: siano A e B due problemi che richiedono negoziazione. Dire che esiste una relazione diretta fra i due, significa affermare che il problema A dipende direttamente dal problema B, poiché quest'ultimo rappresenta una risorsa necessaria per A. Conseguentemente, le caratteristiche di B influenzano direttamente le caratteristiche di A
- *indirettamente collegati (indirectly-linked)*: in questo caso si ha che, A e B sono indirettamente interdipendenti, in quanto competono per una risorsa comune.

I protocolli multi-linked devono essere in grado di gestire entrambi i tipi di relazione. Quando si ha a che fare con negoziazioni multi-linked, si devono gestire varie problematiche. Prima di tutto, l'ordine con il quale si negoziano i vari elementi diventa fondamentale, ed ha effetti diretti sul risultato di quest'ultima. Per esempio, questi potrebbero dover essere negoziati in parallelo o richiedere, viceversa di essere negoziati in modo sequenziale, uno alla volta. La seconda questione, riguarda i valori che devono essere assegnati agli attributi del contratto durante la negoziazione, allo scopo di minimizzare i conflitti fra gli agenti e di massimizzarne invece l'utilità. Esistono numerosi parametri o attributi che devono essere stabiliti nei messaggi dei protocolli di negoziazione. L'agente contractee, deve specificare durante la sua proposta di esecuzione del task, il tempo di inizio e di conclusione (terminazione) della negoziazione, il valore della ricompensa che è disposto a pagare, la penalità di decommitment ecc...L' agente contractor, all'interno della sua risposta, dovrà specificare il tempo in cui promette che il task verrà eseguito correttamente (tempo di fine). Alcuni di questi parametri dipendono dalle caratteristiche delle negoziazioni degli altri elementi interconnessi.

Agli agenti è quindi richiesto di identificare l'appropriato valore per queste caratteristiche, in modo tale da evitare conflitti con altri agenti, e in modo da assicurare l'esistenza di una programmazione locale delle negoziazioni in grado di allocare tutti i task. Il valore di tali attributi, è decisivo per l'esito della negoziazione, e influenza l'utilità raggiunta dagli agenti. Per esempio, più grande è la ricompensa fissata, più grande è la possibilità che il contractor accetti e che il contratto sia stipulato. Questo provoca, tuttavia, effetti opposti sull'utilità dei due agenti: quella del contractee decreterà in maniera inversamente proporzionale al valore della

ricompensa, viceversa quella del contractor aumenterà in modo direttamente proporzionale a quest'ultima. L'ordine delle negoziazioni, e il valore dei loro attributi, diventano elementi strategici all'interno dei protocolli di negoziazione multi-linked.

Lesser, Zhang, e Abdallah, nel 2002, hanno proposto un processo di decisioni che consente agli agenti di gestire la negoziazione multi-linked, scegliendo un'appropriata strategia basata sulle conoscenze relative alle negoziazioni di altri elementi, e alle loro rispettive relazioni. E' data la seguente definizione formale di problema di negoziazione multi-linked:

*Un problema di negoziazione multi-linked è definito attraverso un grafo ad albero  $M=(V,E)$  dove  $V=\{v\}$  è un insieme elementi di negoziazione, ed  $E=\{(u, v)\}$  è un insieme di relazioni binarie su  $V$ .*

*Dire che  $(u, v)$  appartengono ad  $E$ , significa che la negoziazione su  $u$  e la negoziazione su  $v$  sono direttamente collegate. Ogni elemento di negoziazione  $v$  appartenente a  $V$  è associato ad un insieme di attributi  $A_i=\{a_{ij}\}$ . Ogni attributo  $a_{ij}$  può essere già stato determinato o meno. Le relazioni fra gli elementi della negoziazione, possono essere descritte attraverso un grafo ad alberi, composto da un insieme di radici  $\{T_i\}$ . Esiste un operatore di relazione associato ad ogni elemento  $v$  non foglia denotato come  $p(v)$ , il quale descrive le relazioni tra l'elemento  $v$  e i suoi figli. Tale operatore di relazione può assumere i due seguenti valori: AND, OR.*

Dal punto di vista dell'agente esistono due tipi di elementi da negoziare attraverso un protocollo multi-linked: *elementi entranti (incoming negotiation issue)* rappresentati dai task proposti da altri agenti, ed *elementi uscenti (incoming negotiation outgoing)* rappresentati dai task che devono essere assegnati ad altri agenti, o dalle risorse richieste per un task locale.

La soluzione proposta in [18], al problema della negoziazione multi-linked, è stata impostata su un modello matematico di studio della probabilità di successo di una negoziazione, in funzione dei valori degli attributi e dell'ordine di negoziazione degli elementi. La teoria degli alberi è stata impiegata per lo studio delle relazioni fra i vari elementi della negoziazione. Si è presentato un primo algoritmo di ricerca completa per individuare la migliore strategia di negoziazione, il quale si ha



presentato una complessità esponenziale, e quindi non adatta alle applicazioni in tempo reale dove sono coinvolti un grande numero di elementi e di attributi negoziazione. La ricerca di un'ottima strategia di negoziazione include due componenti al suo interno. Una impone di trovare l'ottima programmazione per l'ordine delle negoziazioni, la seconda richiede, invece, di individuare l'assegnamento ottimo per ciascun attributo, per una data programmazione. Basandosi su tali considerazioni, si è proposto un secondo algoritmo di ricerca euristico, il quale si è verificato più efficiente nel caso in cui il numero di elementi e di attributi di negoziazione è elevato, ma meno efficiente rispetto al precedente quando quest'ultimi sono in numero ristretto. La soluzione al problema è risultata, quindi essere una scelta dinamica dell'algoritmo da utilizzare in base all'ampiezza degli elementi coinvolti.

In generale i protocolli di negoziazione multi-linked si basano su tre diverse strategie di negoziazione:

1. *Negoziazione sequenziale*: gli agenti gestiscono gli elementi della negoziazione uno alla volta, prima trattano gli in uscita e poi quelli in ingresso. Il tempo di terminazione dell'operazione promesso dall'agente contractor, coincide con la terminazione richiesta dall'altro agente.
2. *Negoziazione parallela*: gli agenti gestiscono la negoziazione in parallelo. Questo implica maggior flessibilità, verso i task uscenti. Inoltre, in tal modo il tempo di fine di un task entrante è deciso dall'agente contractee.
3. *Negoziazione basata sulle decisioni*: gli agenti gestiscono la negoziazione attraverso l'uso dei due algoritmi precedenti.

Ovviamente, l'utilizzo di un protocollo basato su quest'ultimo tipo di negoziazione, risulta essere migliore rispetto agli altri due, in quanto li consente entrambi.

Concludendo lo studio di ricerca in [18], dimostra come all'interno delle negoziazioni multi-linked, sia veramente fondamentale che l'agente "ragioni" tenendo in considerazione le relazioni esistenti tra diversi elementi della negoziazione, prendendo decisioni ragionevoli riguardo all'esecuzione della negoziazione. Ciò incrementa l'utilità degli agenti, mentre decrementa le probabilità di decommitment. I protocolli di negoziazione multi-linked, infine, hanno trovato ampia applicazione all'interno di metodologie di compensazione tramite negoziazione. Tali metodologie nascono dall'esigenza che hanno agenti che

operano in ambienti cooperativi, di trasferire utilità ad altri agenti per compensarli di patti svantaggiosi.

### **4.3 Protocolli di negoziazione multi-step**

Come già definito in precedenza (par. 2.2), la negoziazione fra sistemi multi agente, può essere principalmente classificata in negoziazione cooperativa o competitiva (non-cooperativa). Quest'ultima è caratterizzata dal fatto che gli agenti, essendo entità SI, hanno come scopo principale, quello di massimizzare la propria utilità locale; viceversa nella negoziazione fra agenti cooperativi, quest'ultimi cercano, attraverso le loro azioni, di massimizzare l'utilità globale del sistema. Inoltre è possibile distinguere una cooperazione globale fra i sistemi multi agente, che tiene conto dell'utilità di tutti gli agenti del sistema, e una cooperazione locale la quale si limita a tentare di massimizzare la somma dell'utilità degli agenti coinvolti nella negoziazione. A causa dei limiti computazionali e di risorse, che caratterizzano le situazioni reali della negoziazione fra gli agenti intelligenti, la cooperazione globale si dimostra praticamente irrealizzabile. Di seguito, si presenterà un approccio allo studio dei protocolli multi-step, basato sulla negoziazione cooperativa globale fra agenti intelligenti. Un protocollo multi-step, implica un processo di interazione fra gli agenti coinvolti, che si esplica attraverso serie di proposte e contro-offerte allo scopo di decidere se l'agente contractor eseguirà il task per conto dell'agente contractee, nel tempo specificato e con una certa qualità.

Il caso di studio precedentemente, descritto fu affrontato da Zhang in collaborazione con altri studiosi in [20]. Zhang riprese la teoria dei costi e dei guadagni marginali, per proporre un processo di cooperazione locale dove gli agenti hanno la possibilità di negoziare più task, in maniera proporzionale agli sforzi che sono disposti a sostenere durante la negoziazione.

Lo scopo degli agenti è quindi quello di creare una sorta di equilibrio fra costi di negoziazione e i benefici ricavati da quest'ultima. Di seguito illustreremo il modello base di un protocollo multi-step.

Innanzitutto occorre dare le seguenti definizioni:

- *Guadagno marginale di utilità (Marginal Utility Gain) [ NL, C ] (MUG):* rappresenta il guadagno marginale di utilità dell'agente. Per il contractee, aumenta se questo riesce a far eseguire il task *non locali* (NL) dall'agente contractor nei tempi e con la qualità specificata nel commitment C.
- *Costi marginali di utilità (Marginal Utility Cost) [ NL, C ] (MUC):* rappresenta il margine di utilità dei costi, il quale decrementa per il contractor se esegue un task NL nei tempi e con la qualità specificata nel commitment C.

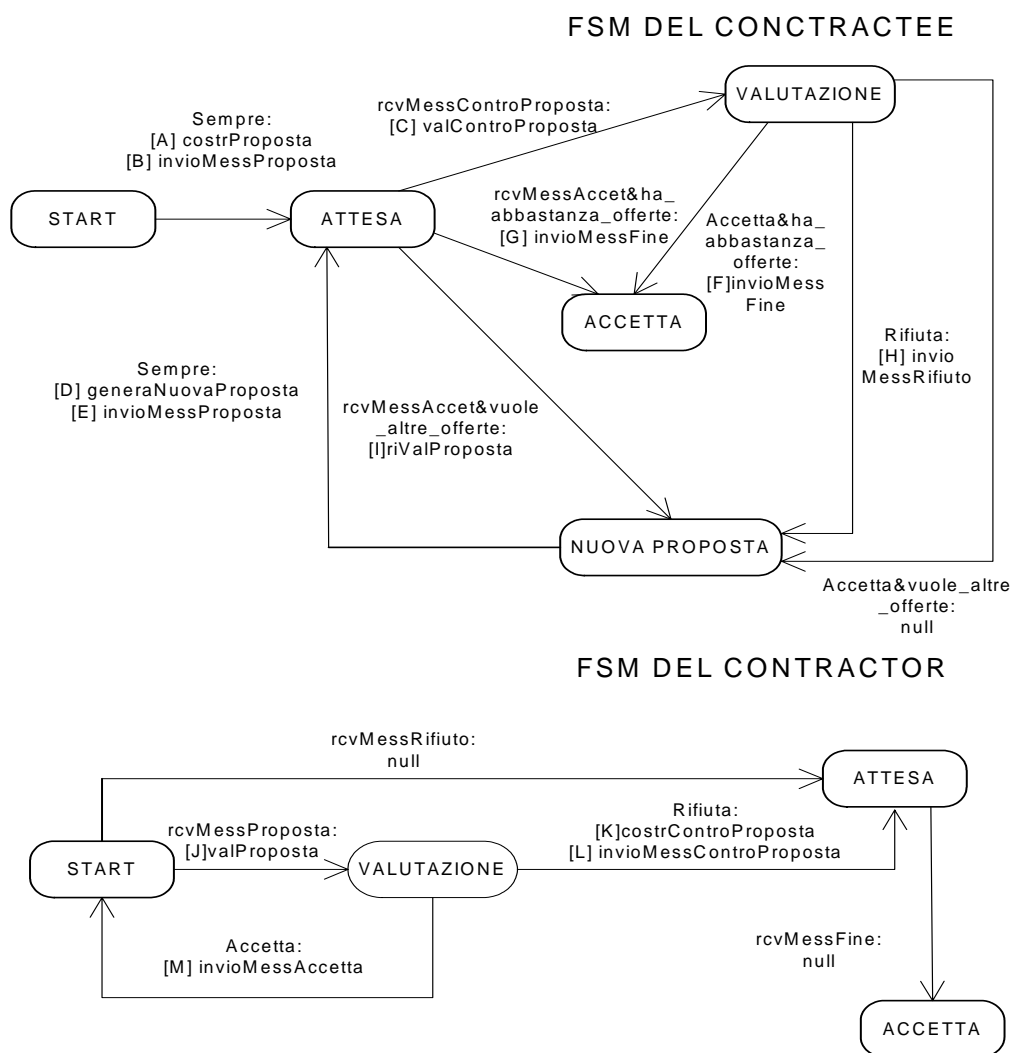


Fig.11 Protocollo di negoziazione cooperativa per l'allocazione dei task

La fig.11 descrive il protocollo di negoziazione cooperativa, generico per l'allocazione dei task. L'agente contractee inizia la negoziazione attraverso l'invio di una proposta (azione A e B: costrProposta e invioMessProposta) all'agente contractor. Quest'ultimo dopo aver ricevuto una proposta (rcvMessProposta), la valuta (azione J: valProposta): se il guadagno di utilità marginale è maggiore dei costi marginali, il contractor accetta la proposta (azione M: inviaMessAccetto); in caso contrario la proposta viene rifiutata e l'agente si preoccupa di costruire una contro proposta (azione K: costrControProposta) e invia quest'ultima all'agente contractee (azione L: invioMessControProposta). Una volta ricevuto il messaggio di contro proposta (rcvMessControProposta), l'agente contractee valuta tale messaggio (azione C: valControProposta): se la contro proposta è giudicata accettabile ed esiste già un numero sufficiente di soluzioni (per soluzione si intende un commitment accettabile ovvero in cui l'MUG è più grande dell'MUC), la negoziazione termina e l'agente contractee informa il contractor che il commitment è stato costruito (azione F: invioMessFine); in caso contrario, se la contro proposta non viene accettata l'agente contractee genera una nuova proposta basandosi sulla precedente (azione D: generaNuovaProposta).

Questo protocollo, utilizza quindi tre funzioni principali: la prima viene usata dal contractee per generare la proposta iniziale, la seconda viene usata dal contractor per generare una contro proposta, e infine con l'ultima il contractee genera una nuova proposta per rispondere alla contro proposta che ha ricevuto.

Il meccanismo di negoziazione fin qui descritto, rappresenta lo scheletro di una famiglia di protocolli di negoziazione, i quali differiscono esclusivamente per il criterio di terminazione adottato. Tali protocolli sono i seguenti:

- *Single-Step*: l'agente contractee invia una proposta di commitment al contractor; il contractor accetta tale proposta se il suo MUG è maggior del MUC; in caso contrario rifiuta la proposta e la negoziazione termina fallendo.
- *Multi-Step-Multiple(n)-Try*: gli agenti contractee e contractor si scambiano una serie di proposte, controproposte e nuove proposte; la negoziazione termina quando si sono individuate almeno "n" soluzioni o si sono raggiunti certi limiti d'interazione. Nel nostro caso considereremo in particolare i casi:
  1. se  $n=1$  si ha il Multi-Step-One-Try;

2. se  $n=2$  si ha il Multi-Step-Two-Try;
  3. se  $n=3$  si ha il Multi-Step-Three-Try;
- *Multi-step-Limited-Effort*: gli agenti contractee e contractor si scambiano una serie di proposte, contro proposte e nuove proposte; la negoziazione termina quando si sono raggiunti certi limiti d'interazione. Se il valore di tali limiti è relativamente grande, tale protocollo ha la possibilità di individuare molte più soluzioni rispetto ai protocolli precedenti.

Zhang, studiò il funzionamento dei diversi protocolli precedentemente elencati, quando quest'ultimi si trovano ad operare in diverse situazioni. In particolare si studiarono le variazioni delle prestazioni dei protocolli al variare della complessità della struttura dei task. I risultati mostrarono che, se la complessità della struttura dei task aumenta, il problema della negoziazione si complica anch'esso. Ciò è dovuto principalmente al fatto che, task più complessi, implicano maggiori costrizioni e conseguentemente causano un restringimento dello spazio di ricerca delle possibili soluzioni. In particolare si notò come il protocollo single step, fornisce le stesse prestazioni degli altri protocolli finché si rimane su bassi livelli di complessità, al crescere di quest'ultima le sue prestazioni decrementano velocemente. Il protocollo Multi-Step-( $n+1$ )-Try presenta invece prestazioni migliori rispetto al Multi-Step-( $n$ )-Try in molte situazioni con un certo numero di costrizioni. Nei casi in cui, le costrizioni sono minori o molto maggiori di tale numero, l'utilizzo di protocolli che vanno al di là del Multi-Step-Three-Try, non comporta alcun vantaggio. Dalla seguente analisi, è quindi possibile affermare che, le prestazioni dei protocolli in questione, dipendono strettamente dal livello di difficoltà della negoziazione a cui devono essere applicati. Inoltre, poiché ogni protocollo comporta sforzi e costi differenti per un agente, diventa molto importante che quest'ultimo scelga il protocollo più opportuno allo scopo di bilanciare i costi e i guadagni della negoziazione. In generale il guadagno di una negoziazione può essere valutato attraverso il guadagno di utilità mentre lo sforzo cresce in maniera proporzionale al numero di step richiesti. I costi della negoziazione hanno quindi effetti diretti sull'utilità di quest'ultima. Ciò è dovuto al fatto che, primo il processo di negoziazione consuma risorse che altrimenti potrebbero essere impiegate per altri task; secondo il processo di negoziazione determina come e quando verrà eseguito il task e ciò va ovviamente ad influire sull'utilità.

Zhang, studiò inoltre la relazione esistente fra i guadagni di utilità che possono essere ottenuti dall'applicazione dei diversi protocolli. Dai risultati, si delineò un fenomeno indicato col termine di fase di transizione durante la quale: se la complessità della negoziazione è bassa nessun protocollo comporta particolari vantaggi di guadagno, se invece la negoziazione si complica notevolmente il protocollo Multi-Step-One-Try risulta più efficiente dal punto di vista dell'equilibrio fra costi e guadagni. Infine, quando la negoziazione è caratterizzata da un livello di difficoltà medio, il protocollo Multi-Step-Three-Try risulta il migliore. Il protocollo Multi-Step-Limited-Effort non è stato testato in nessuna delle precedenti situazioni a causa dei costi troppo elevati che comporta. Nel suo studio Zhang, misurò la complessità di una negoziazione in base alle costrizioni che questa comportava. In tal modo, prima di iniziare la negoziazione, l'agente contractor può informare il proprio partner del suo numero di costrizioni locali. In base a tale valore, il contractee potrà, quindi, valutare il protocollo più opportuno da applicare tenendo conto che:

- il protocollo Multi-Step-One-Try sia, praticamente, sempre più efficiente del protocollo Single Step;
- i protocolli Multi-Step-Two-Try e Multi-Step-Three-Try sono più adatti in condizioni di media -complessità;
- il protocollo Multi-Step-One-Try è più adatto a situazioni con costrizioni molto limitate o molto vincolanti;
- in generale il protocollo da utilizzare deve essere scelto cercando di bilanciare i costi con il guadagno della negoziazione.

## Conclusioni e Lavoro Futuro

La nascita dei protocolli di negoziazione, può essere fatta coincidere con la definizione formale del Contract Net Protocol (CNP) proposto da Smith nel 1980. Questo, infatti, definì alcuni meccanismi di interazione fondamentali che tutt'oggi possono essere ritrovati nei moderni protocolli di negoziazione per agenti intelligenti. Dal nostro studio, in particolare, è emerso come tutti i protocolli analizzati si sviluppino su una struttura, già presente nel CNP, costituita essenzialmente da due soggetti negoziatori, e sviluppata su quattro fasi principali. Tali soggetti sono:

- Un agente che desidera allocare i propri task, allo scopo di minimizzandone i costi di gestione, e di massimizzandone la propria funzione di utilità. Tale agente, può essere indicato con il termine di manager, iniziatore o contractee.
- Un agente che esegue dei task per conto di altri agenti in cambio di un corrispettivo, non necessariamente in denaro. Tale agente, viene indicato con il termine di contractor o partecipante.

Le quattro fasi principali sono:

1. La fase di annuncio, durante la quale l'agente manager invia un messaggio di richiesta per l'esecuzione di un suo task.
2. La fase di offerta, in cui il contractor dopo aver valutato l'annuncio invia un'offerta al manager per l'esecuzione del task.
3. La fase di assegnamento, durante la quale il manager valuta le offerte e decide a chi assegnare il contratto per l'esecuzione del task.
4. La fase di ricezione dell'assegnamento, nella quale il contractor accetta o rifiuta il contratto.

I protocolli definiti da FIPA, rappresentano degli standard per il processo di negoziazione, i quali essenzialmente riprendono fedelmente i principi del CNP precedentemente descritti.

Come risultato delle nostre ricerche si è individuato il seguente percorso evolutivo dei protocolli di negoziazione: Sandholm, nel 1993, estese il CNP introducendovi il concetto di agenti che agiscono per interesse personale (*self-interested*, *SI*) e

presentò una formalizzazione dei processi di offerta e di assegnazione di un contratto, lasciati indefiniti nell'originale CNP; Sandholm e Lesser, nel 1995, continuarono poi insieme l'estensione del CNP attraverso due vie principali: la razionalità degli agenti è limitata dalla complessità computazionale, inoltre gli agenti possono scegliere la fase e il livello di *commitment* dinamicamente attraverso l'uso di condizioni sulla negoziazione e di penalità in caso di *decommitment*; Golfarelli, nel 1997, propose il protocollo di negoziazione *Task Swap*, il quale anch'esso estendeva il CNP allo scopo di far fronte ai problemi che si presentano in domini dove lo scambio di task (*task swap*) è il solo possibile mezzo di negoziazione; ancora Sandholm, nel 1998, introdusse dei nuovi tipi di contratto: *single task contract*, *cluster contract*, *swaps*, e il *multi-agent contract* e un unico tipo di contratto atomico, chiamato *OCSM-contract*, il quale combina l'idea sottesa nei quattro contratti precedentemente descritti. Infine, come meccanismi alla base dei nuovi protocolli di negoziazione, si sono studiati: la negoziazione *multi-linked*, dove i protocolli gestiscono le situazioni in cui un agente richiede di negoziare con più agenti, riguardo a più elementi interdipendenti; la negoziazione *multi-leveled*, dove i protocolli gestiscono negoziazioni multi-task e quindi particolarmente complesse dal punto di vista della definizione dei commitment; la negoziazione *multi-step*, dove i protocolli si basano su differenti cicli di offerte e contro offerte realizzati degli agenti.

Fra tutti i protocolli analizzati, solo il Task-Swap, descritto in [15], non risulta basato sul principio della vendita, ma bensì su un mutuo scambio di task. Forse la motivazione è da ricercare semplicemente nel fatto che, la maggior parte delle applicazioni legate ai protocolli di negoziazione, richiedono negoziazioni basate su contratti di vendita. Il CNP ha trovato, infatti, il suo maggior sviluppo all'interno dei mercati elettronici, dove tra l'altro la componente competitiva degli agenti, predomina su quella cooperativa.

Per quanto concerne gli sviluppi futuri, particolare attenzione potrebbe essere rivolta verso protocolli di negoziazione che tentino d'inglobare al loro interno le varie politiche di negoziazione sviluppate negli ultimi anni. Viceversa, alcuni sforzi potrebbero essere investiti nel tentativo di definire ulteriori standard per quanto concerne le politiche di negoziazione ormai assodate.



## Glossario

**Agent Communication Language (ACL):** linguaggio di comunicazione fra agenti intelligenti. Un ACL è un insieme di primitive che consentono ad un agente di dichiarare la propria intenzione. Le primitive sono azioni (*performatives*) che l'agente può usare nel tentativo di comunicare ad altri agenti le proprie intenzioni.

**Bounded Rationality (BR):** un agente si definisce BR quando possiede una razionalità e una capacità di ragionamento limitata, che gli impedisce di prendere decisioni infallibili, e di effettuare valutazioni prive d'errore. Tutti gli agenti devono essere considerati BR.

**Costi marginali:** rappresentano i costi che un agente deve sostenere nella gestione dei propri task.

**Commitment:** nella negoziazione tra gli agenti, implica un accordo stipulato fra due agenti che li vincola ad un potenziale contratto.

**Decommitment:** nella negoziazione, implica che uno agente receda dagli impegni presi attraverso un commitment per un potenziale contratto.

**Decommitment penalty:** rappresenta la penalità che un agente deve sostenere, nel caso in cui decida di recedere di effettuare decommitment.

**Individual rational:** si dice che uno scambio di task è individualmente razionale per un agente, se l'utilità che se ne ricava è positiva. Più in generale un contratto si può definire individualmente razionale per un agente, se comporta un miglioramento delle sue condizioni economiche.

**Self-interested (SI):** gli agenti si dicono SI, quando mirano a soddisfare i propri obiettivi nel modo più efficiente possibile, indipendentemente dagli interessi degli altri agenti. Agiscono cioè, per interesse personale.

*Sistema Multi-agente (MAS):* è un sistema in cui gli agenti intelligenti interagiscono e coesistono tra di loro per soddisfare un certo insieme di obiettivi, allo scopo di portare a termine un certo insieme di compiti.

## **Bibliografia**

- [1] Katia Sycara. Multi-agent Infrastructure Agent Discovery, Middle Agents for Web Services and Interoperation.
- [2] - Bradshaw, J.M. An Introduction to Software Agents. In: Software Agents. J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pag. 3-46.
- [3] Da Haw Siang Hon Ise. Agent Communication Language, 2001
- [4] M.Klush, K.Sycara. Brokering and Matchmaking for Coordination of Agent Societies: A Survey, 2001.
- [5] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents. Dic. 2002
- [6] M.Beer, M.d'Inverno, M. Luck, N.Jennings, C.Preist, M.Schoroeder. Negotiation in Multi-Agent Systems, 1999
- [7] K.Sycara, D.Zeng. Negotiation As Phenomenon and As Agent Coordination Protocol. The Robotics Institute. Carnargie Mellon University. Pittsburgh., 1998
- [8] Reid G.Smith. The contract net protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transaction on Computer, 205 Vol. C\_29 Dic. 1980; pp 1104-1113.
- [9] G.Itabashi, Y.Haramoto, K.Takahashi. Specification and Analysis of the Contract Net Protocol Based on State Machine Model.IEICE Trans. Fundamental, vol.E 85 N.11, Novembre 2002
- [10] Wenge Guo. Task Allocation: An Introduction. Departement of Computer Scienze North Dakota State University, 2002
- [11] FIPA Contract Net Interaction Protocol Specification. Foundation for Intelligent Physical Agents. Dic. 2002
- [12] FIPA Iterated Contract Net Interaction Protocol Specification. Foundation for Intelligent Physical Agents. Dic. 2002
- [13] T.W.Sandholm. An Implementation of the Contract Net Protocol based on Marginal Cost Calculations. National Conference on Artificial Intelligence, 256-262, 1993.

- [14] T.W.Sandholm, V.R.Lesser. Issue in Automated Negotiation and Elettronic Commerce: Extending the Contract Net Framework. International Conference on Multi-Agent Systems, 328-335, 1995.
- [15] M.Golfarelli, D.Maio, S.Rizzi. A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm. Techincal Report CSITE, N. 005-97, 1997.
- [16] D.Maio, D.Maltoni , S.Rizzi. Dynamic Clustering of Maps In Autonomous Agents. IEEE Transaction on Pattern Analysis and Machine Intelligence, 18(11): 1080-1091, Novembre 1996
- [17] T.W.Sandholm. Contract Types for Satisfying Task Allocation: I Theoretical Results.  
AAAI Spring Symposium: Satisfying Models, Stanford University, California, Marzo 1998.
- [18] X.Zhang, V.Lesser, S.Abdallah. Efficient Ordering and Parameterization of Multi-Linked Negotiation. Computer Science Department. University of Massachusset, 2002
- [19] X.Zhang, V.Lesser, T.Wagner. A Multi-Leveled Negotiation Framework. Computer Science Department. University of Massachusset , 2003
- [20] X.Zhang, V.Lesser, R.Podorozhny. New Results On Cooperative, Multi-Step Negotiation Over Multi-Dimensional Utility Function Computer Science Department. University of Massachusset 2002



