

# Collaborazione nei sistemi multi agente

## Lezione 14

## Sommario

- ◇ Condivisione dei ruoli [Ferber Cap 7]

## Soluzione cooperativa di problemi distribuiti (cenni)

- ◇ Cooperative Distributed Problem Solving (CDPS)
- ◇ Scenario particolare di coordinamento
- ◇ Caratteristiche principali del CDPS:
  - Rete di agenti con goal comune
  - Nessun agente può risolvere il problema da solo
  - Diverse capacità, risorse ed informazioni
  - Assunzione di benevolenza, unico progettista (altrimenti problema di raggiungimento dell'accordo)

# Cooperazione e Coordinamento

Ipotesi: Agenti self-interested

◇ Cooperazione: sistemi che devono interagire fra di loro per completare i compiti (task) di interesse

- **suddivisione compiti**
- condivisione informazioni

◇ Coordinamento: sistemi che devono dinamicamente adattare il proprio comportamento a quello degli altri

- condivisione di risorse
- obiettivi in comune

## Condivisione dei Task e di informazioni

### ◇ Suddivisione dei task

- Decomposizione dei task
- Allocazione dei task

### ◇ Condivisione di informazioni

- Agenti scambiano informazioni **rilevanti** ai loro task
- Proattiva
- Reattiva

## Decomposizione problema

- ◇ Tipicamente gerarchica
- ◇ Granularità dei sottoproblemi difficile da definire
  - Livelli di decomposizione → Livelli di astrazione

In pratica viene definita a priori

## Modi di allocazione

### ◇ Predefinita

- Centralizzata

- Imposta
- tramite Trader

- Distribuita

- Rete di Conoscenze
- Contract Net Protocol

### ◇ Emergente

## Allocazione centralizzata

### ◇ Imposta

- Struttura gerarchica di subordinamento
- Allocazione rigida e definita a priori
- Come chiamata a procedura in linguaggi imperativi

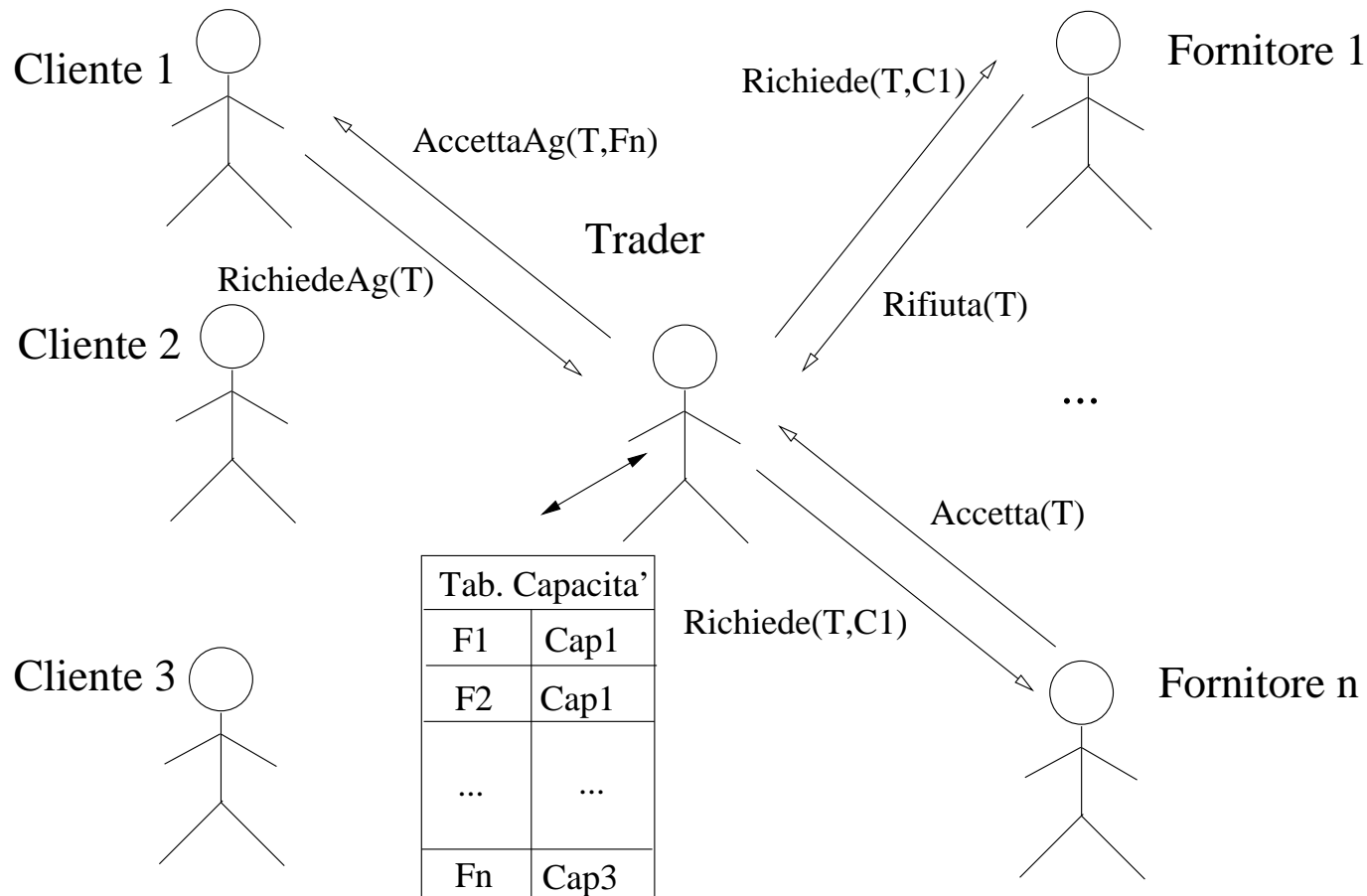
### ◇ Tramite Trader

- Trader agente speciale che gestisce la allocazione
- Tecniche centralizzate per organizzazioni variabili



# Allocazione centralizzata

◇ Trader, Clienti e Fornitori



## Speech Acts Alloc. centralizzata

- ◇ Request(T,X)
- ◇ Accept(T,Y)
- ◇ Impossible(T)
  
- ◇ Request(T,X)
- ◇ Accept(T,Y,X)
- ◇ Refuse(T,Y,X)

## Caratteristiche Alloc. centralizzata

- ◇ Coerenza del sistema assicurata
- ◇ Num. messaggi scambiati =  $\alpha k N (2 + 2\beta N) \Rightarrow O(N^2)$ 
  - $\alpha$  = Clienti potenziali/Clienti totali
  - $\beta$  = Fornitori potenziali/Fornitori totali
  - $k$  = richieste per unità di tempo
- ◇ Molto sensibile ai fallimenti

## Allocazione tramite rete di conoscenza

◇ Acquaintance Network → rete di conoscenza

|         | $Ag_1$ | $Ag_3$ | $Ag_j$ |
|---------|--------|--------|--------|
| $Cap_1$ | 0      | 1      | 1      |
| $Cap_2$ | 0      | 0      | 0      |
| $Cap_i$ | 1      | 0      | 0      |

◇ Ciascuna tabella è corretta, non completa e statica.

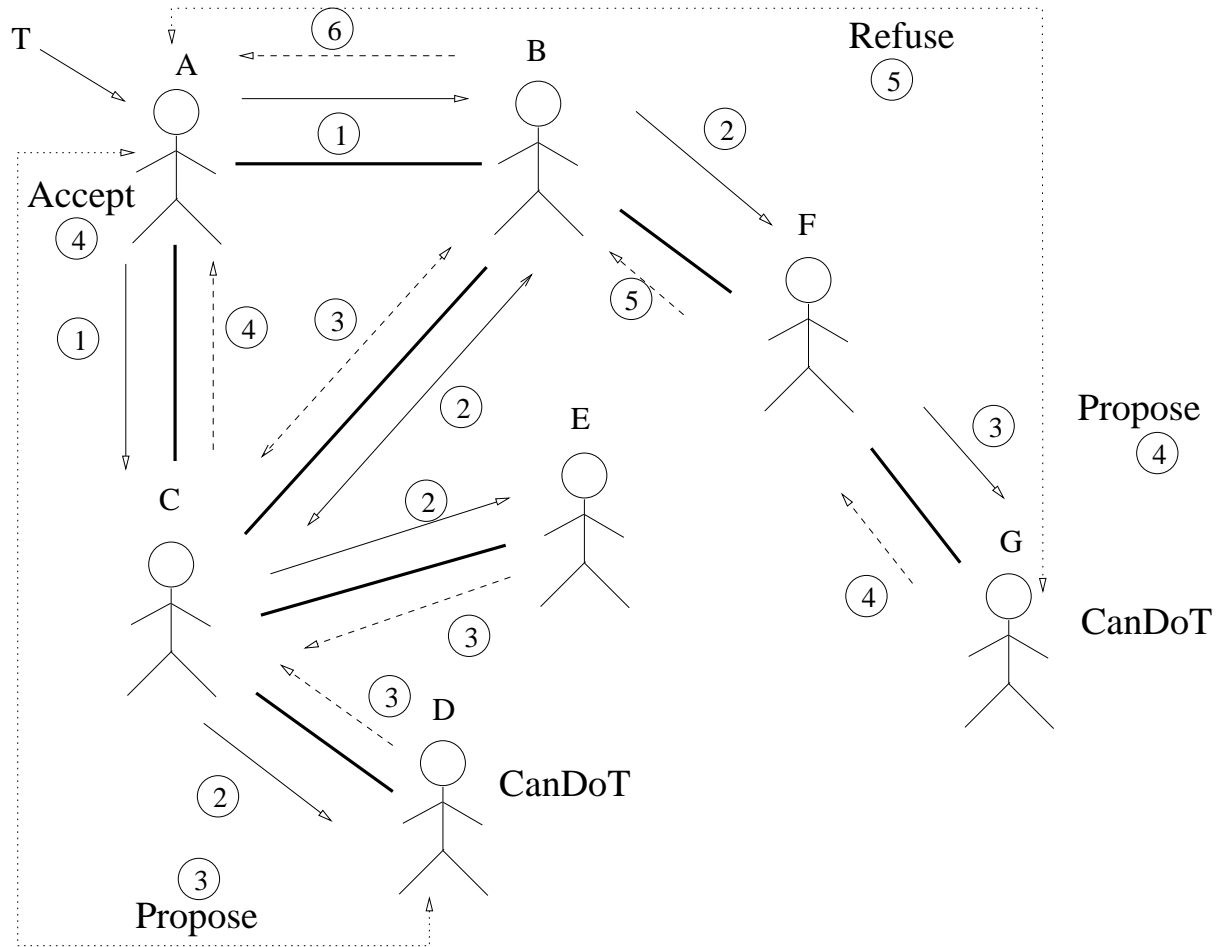
## Allocazione diretta

- ◇ Ogni agente alloca task solo a chi conosce direttamente
- ◇ Per ogni task richiede a tutti i conoscenti di eseguirlo
- ◇ Se nessuno lo può fare:
  - Allocazione forzata
  - Rimando ad un agente centrale
- ◇ Agenti non sanno della presenza degli agenti che non conoscono direttamente

## Allocazione per delega

- ◇ Ricerca in tutta la rete
  - Profondità
  - Ampiezza
  
- ◇ Ricerca in ampiezza parallela
- ◇ Basata su algoritmi di diffusione
  - Visitare tutti gli agenti: Acknowledge (Ho cercato tutto il tuo sottoalbero)
  - Visitarli una sola volta: Marcatura

# Es. di allocazione per delega



## Problemi allocazione per delega

- ◇ Ricerca non ottimizzata (mando messaggi a tutti)
- ◇ Computazione inutile per agenti dopo che almeno un agente ha accettato
- ◇ Nessun Livello di Capacità considerato



# Riorganizzare una rete di conoscenze

## ◇ Cambio Capacità

- Comunicare quando una capacità cambia
  - Broadcast del cambiamento, ma chi sa delle mie capacità ? (necessaria bidirezionalità su rete e maggiore complessità)
  - Sincronizzazione tra cambiamento e richieste da gestire
- Rete adattiva
  - A modifica la sua rete quando riceve un diniego
  - Frequenza dei cambiamenti può essere un problema

## ◇ Inserimento ed eliminazione agenti

## Contract Net Protocol

- ◇ Smith (1979)
- ◇ Semplice e veloce da implementare
- ◇ **Manager** e **Fornitori** colloquiano tramite **offerte**

## CNP alg. single Manager

1. Manager manda una **descrizione** del task a tutti i possibili fornitori (RequestForBid(T,M))
2. Fornitori **valutano** l'offerta e mandano un **proposta** al Manager (Propose(T,Off,Ag) oppure NotInterested(T,Ag))
3. Manager **valuta** le proposte alloca il contratto al miglior fornitore (Award(T,Ag,M))
4. Il fornitore prescelto risponde positivamente o negativamente al Manager (Accept(T,Ag) Refuse(T,Ag))

## CNP Messaggi utilizzati

- ◇ Smith ha definito un linguaggio di specifica per i contratti
  - requisiti
  - forma
  - scadenza per la presentazione

## CNP Caratteristiche

- ◇ Contattare tutti gli agenti è un problema notevole
- ◇ Token-Ring soluzione possibile (tradeoff con velocità)
- ◇ Caching per minimizzare comunicazioni (tradeoff con performance)
- ◇ Data di limite per i contratti:
  - evita comunicazioni inutili di agenti non interessati
  - evita blocchi dovuti a malfunzionamenti di agenti
  - comporta tempo fisso di attesa → inefficienza

## CNP Multi manager

- ◇ Normalmente multi manager → possono interferire tra loro
- ◇ problemi di interferenza dovuti a conoscenza incompleta dei fornitori:
  - relativa al tempo (nessuna predizione dei futuri task) vero anche nel caso singolo manager
  - relativa allo spazio (altre proposte)

|   | $T_a$ | $T_b$ |
|---|-------|-------|
| X | 90    | 80    |
| Y | 80    | 20    |

- relativa al carico di lavoro (vero anche singolo manager)

## CNP problemi di sincronia

◇ Sincronia tra la sottomissione della proposta e l'arrivo dell'award

- Agenti prudenti
  - propongo solo ciò che posso fare
  - se pochi agenti strategia buona
- Agenti sfrontati
  - propongo anche se non ho la possibilità
  - può generare nuove sottomissioni di task
  - buona se tanti agenti
- Agenti moderati
  - decision theoretic per proporre
  - peso le risorse necessarie in base alla possibilità di accettazione della mia proposta

## CNP con sotto contraenti

- ◇ Fornitori possono sub-appaltare contratti
- ◇ Simile a situazione precedente ma con risorse condivise
- ◇ Varie strategie
  - Early commitment
    - Fornitori mandano proposte prima di avere tutte le risorse necessarie
    - Molte riorganizzazioni necessarie
    - Più rapido
  - Late commitment
    - Fornitori mandano proposte solo dopo aver acquisito tutte le risorse necessarie
    - Possibili deadlock
  - Agenzie fisse
    - Meno adattabile più stabilità



## CNP Caratteristiche

### ◇ Pro

- Molto semplice e rapido da implementare
- Dinamico e facilmente adattabile
- Contratto bilaterale → molti parametri considerati nella allocazione

### ◇ Contro

- Molti messaggi  $O(nm)$
- Problemi di sincronia
- Problemi con sub-contraenti

## CNP Variazioni

- ◇ CNP guidato da proposte
- ◇ Fusione di rete di conoscenze e CNP
  - differenziazione sui task (complessità priorità) ed allocazione rete di conoscenze (rapida, poco accurata) CNP (costoso, più accurato)
  - Ricorrere a CNP solo in situazioni di fallimento della rete di conoscenze
- ◇ Rete di conoscenze → cache memory per CNP

# Linguaggio dei contratti

Message: RequestForBids

To: \*

From: A21

DescriptionOfTask:

TypeOfTask: check-feature

QualitiesRequired :

MustHave: Camera

FormOfProposal:

Position : (X,Y)

DateExpiry: 12:00

EndRequestForBids

## Allocazione emergente

- ◇ Allocazione dei task per agenti reattivi
- ◇ Segnali e non messaggi (comm. stigmergica)
- ◇ Behavior agenti dipendono dall'intensità dei segnali...
- ◇ ...e dalla tendenza interna degli agenti (semplice stato)
- ◇ Molto usato in MRS (taglia i costi della comm. diretta)
- ◇ Non orientato all'efficienza ma alla scalabilità
- ◇ Artificial Life, Swarm (Brooks, Mataric)

## Sistema Manta

- ◇ Simula un nido di formiche
- ◇ Tipologia delle entità
  - Beneficiarie (larve, uova, ...)
  - Curatrici (Regina, lavoratori, ...)
  - entità rilevanti del sistema (Cibo, Sorgenti di luce, ...)
- ◇ Ogni stimolo proveniente dall'ambiente è in relazione con uno ed un solo task

## Sistema Manta II

◇ I task hanno un peso per determinarne il grado di attivazione

- FeedBack positivo (avvicinamento al cibo)
- FeedBack negativo (nutrire una larva)

◇ Studiare l'emergenza di divisione dei compiti

- Task: occuparsi di cibo, uova o larve
- Dagli esperimenti è emersa una chiara suddivisione del lavoro