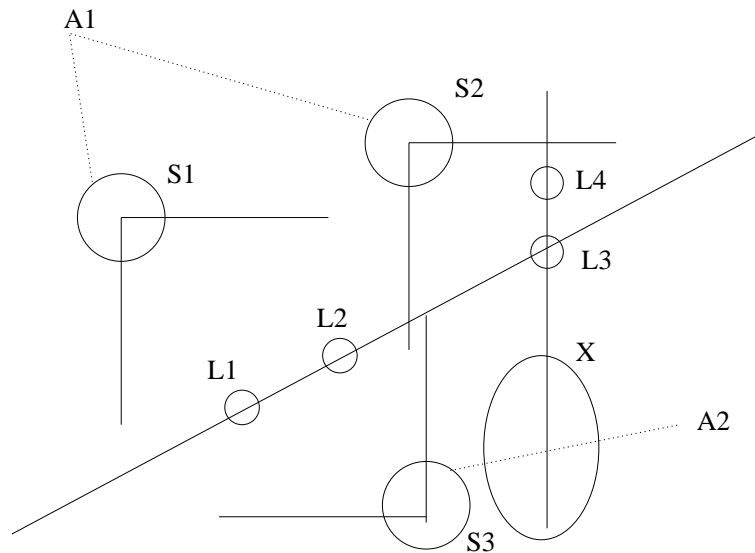


# Collaborazione e Coordinamento II

## Lavorare insieme

# Condivisione Risultati



- ◇  $A_2$  vuole conoscere  $L_1$  e  $L_3$
- ◇  $A_1$  vorrebbe usare  $S_3$  per avere dati su  $X$
- ◇  $A_2$  non sa a priori quali letture di  $S_1$  e  $S_2$  possono essere utili

## Condivisione Risultati

CNP (Acquaintance)

◇ Alloca classi di task:DSNE

◇ Non va bene se diverse classi di task hanno diverse relazioni a run-time

⇒ Scambio di soluzioni approssimate

## Condivisione Risultati

Funzionalmente accurato/cooperativo

Pro

- Accuratezza
- Completezza
- Precisione
- Velocità

Contro

- Distrazione
- Costo Comunicazione
- Costo Fusione
- Algoritmo Comunicazione
- Algoritmo Fusione

## Repository Condiviso

- ◇ Struttura comune condivisa, accessibile in lettura e scrittura
- ◇ Agenti scambiano soluzioni possibili tramite il repository (Design)
- ◇ Agenti rivisitano e criticano soluzioni precedenti (Negoziazione)
  - Initiate sol
  - Extend sol
  - Critique sol
  - Relax Expectations

## Repository Condiviso II

◇ DCSP ma

- Agenti non conoscono i vincoli degli altri
- Agenti possono rilassare la soluzione condivisa

◇ Euristiche utilizzate per decidere come procedere

## Ricerca Euristica Distribuita e Vincolata

- ◇ Contenzioso sulle risorse → DCSP
- ◇ Market Oriented Programming
  - Ogni risorsa un agente (Auctioneer)
  - Auction per decidere a chi allocare la risorsa
- ◇ DCHS
  - Ogni agente aggrega le richieste ricevute per la risorsa
  - L'euristica per la soluzione del DCSP usa questa informazione

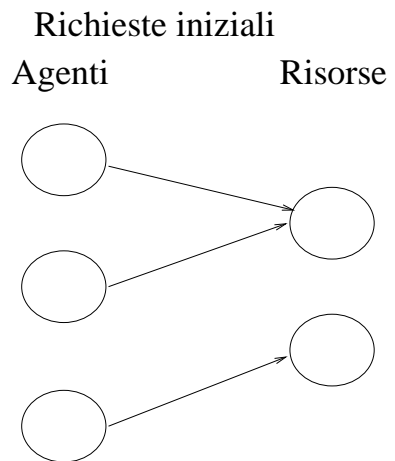
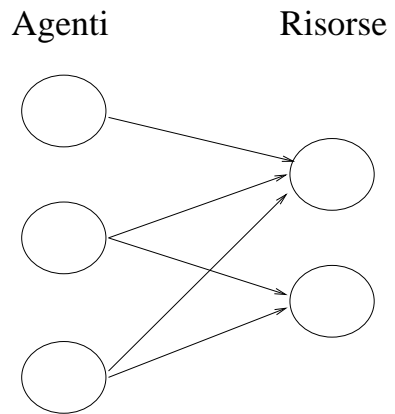
## DCHS [Sycara]

1. Ogni agente conosce la **topologia** del problema
2. Ogni agente propaga i suoi vincoli e se inconseguenze **back track** altrimenti determina la **richiesta** per le risorse
3. Se richiesta  $\neq$  richiesta-precedente  $\rightarrow$  manda la richiesta alle risorse coinvolte
4. Ogni risorsa calcola le richieste aggregate e le rimanda indietro (con possibili conflitti)

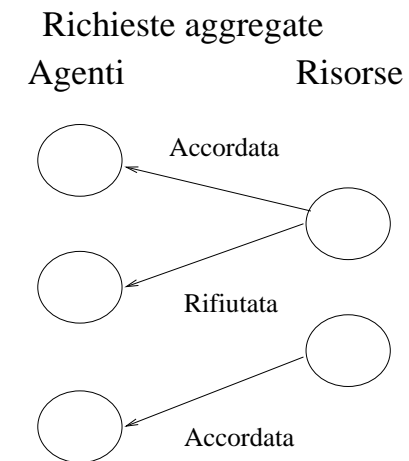
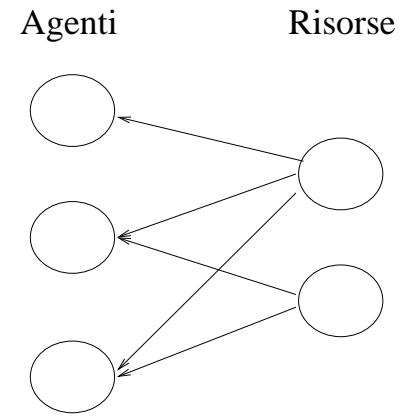


5. Ogni agente ordina le sue variabili (risorsa, tempo) in base alle richieste ricevute
6. Ogni agente richiede alla risorsa l'intervallo temporale preferito
7. Le risorse **accordano** le richieste, se possibile, altrimenti le negano
8. Ogni agente processa la risposta: se **accordata** → vai a 2 altrimenti vai a 6

# DCHS



Prenotazione



## Strutture Organizzative

- ◇ Definire struttura statica per decidere quando comunicare (DSNE)
- ◇ Ruoli → Definire responsabilità e pattern comunicativi
- ◇ Agente **conosce** solo informazioni relative al suo ruolo
- ◇ Come definire la struttura é un problema in sé

# Strategie di Comunicazione

- ◇ Troppa comunicazione
  - Distrazione
  - Inondare memoria
  - Inondare messaggi
- ◇ Poca comunicazione
  - Non si può trovare la soluzione
  - Soluzione peggiore
- ◇ Comunicazione iterativa (chiedere info → ricevere info)
  - Meno overhead
  - Più certezza
- ◇ Perdita Messaggi
  - Murmuring
  - Riconoscimento dei piani

## Struttura dei task

- ◇ Task connessi con relazioni
  - Abilita (Rescue: Poliziotto libera la strada ad un ambulanza)
  - Facilita (Rescue: Due vigili del fuoco spengono lo stesso incendio)
  - Annulla (DVMT: un agente toglie il controllo di un sensore ad un altro agente)
  - Inibisce (DVMT: un agente prende il controllo esclusivo di un sensore)
  
- ◇ TAEMS (Decker) Le relazioni fra task sono utilizzate per decidere strategia di comunicazione:
  - Es. se A facilita B agente alloc. ad A schedula A prima di ogni altra attività

## Combinare Task Sharing e Result Sharing

- ◇ FELINE (Woolridge et al., 1991)
- ◇ Agenti esperti cooperano alla risoluzione di un problema
- ◇ Ogni agente é un sistema basato su regole
  - Insieme di fatti
  - Insieme di regole → conoscenza dominio
- ◇ Ogni agente ha un environment model
  - una entry per se stesso e ciascun agente conosciuto

# Feline

## ◇ Ogni entry

- Skills → ipotesi da confermare o confutare
- Interests → ipotesi che l'agente vuole processare

## ◇ Messaggi

- Tripla ⟨mittente, destinatario, contenuto⟩
- Contenuto ⟨tipo, attributo, valore⟩
  - ⟨richiesta, ipotesi, —⟩
  - ⟨risposta, ipotesi, valore⟩
  - ⟨informazione, ipotesi, valore⟩

## Feline II

- ◇ Goal: determinare valore di verità di un'ipotesi
- ◇ Discesa ricorsiva dell'albero di inferenze
- ◇ Se la foglia non é una domanda per l'utente →
  - controllare se l'ipotesi appartiene agli skills di qualche agente conosciuto
  - mandare una  $\langle$ richiesta, ipotesi, —  $\rangle$
  - attendere la  $\langle$ risposta, ipotesi, valore  $\rangle$
- ◇ Quando un nuovo fatto viene generato →
  - controllare se appartiene agli interessi di qualche agente conosciuto: se si →  $\langle$ informazione, fatto, valore  $\rangle$

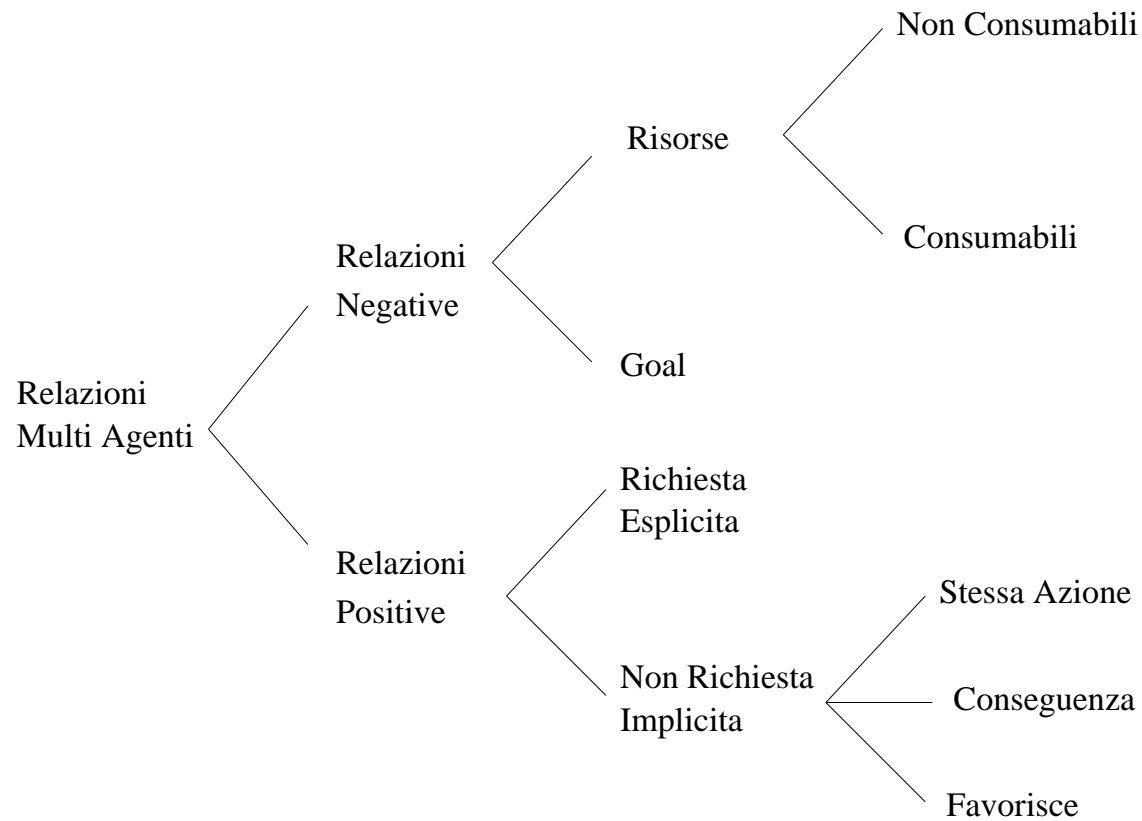


## Come Gestire le Inconsistenze

- ◇ Diversi goal
- ◇ Diverse conoscenze
  - incomplete
  - errate
- ◇ Possibili approcci
  - non permettere che avvengano le inconsistenze
    - CNP 1 solo manager
  - risolvere le inconsistenze
    - negoziazione
  - convivere con le inconsistenze
    - sistemi degradano performance in maniera accettabile
    - FA/C
      - \* soluzioni iterative
      - \* ridondanza

# Coordinamento

◇ Gestire attività interdipendenti tra gli agenti



## Coordinamento II

- ◇ Approcci che affrontano il problema nella sua globalità
  - PGP
  - Joint Intentions
  - Norme Sociali
  - Coordinamento Reattivo

# PGP

- ◇ Progettato per DVMT
- ◇ Agenti scambiano soluzioni parziali iterativamente
- ◇ Task dec.
  - i task sono inerentemente decomposti
  - nessun agente deve necessariamente conoscere l'intero piano
- ◇ Local Plan: ciascun agente é in grado di formulare un piano locale per i suoi goal
  - i piani possono essere ridondanti
  - livello di astrazione necessario

## PGP II

### ◇ MLO

- Meta Level Organization
- Provvede informazioni e flusso di controllo fra gli agenti

### ◇ Partial Global Goal

- agenti devono cercare di capire se i goal locali fanno parte di un unico goal globale
- parziale perché é iterativo

## PGP III

### ◇ PGP Costruzioni e Modifiche

- piani locali vengono integrati
- in base al piano globale si ottimizza lo svolgimento della missione
- si pianificano le comunicazioni per integrare i risultati

Durante lo svolgimento delle azioni:

- i piani locali possono cambiare → cambiamento piano globale;
- riallocazione dei task → proponendo piani diversi.

## Joint Intension

- ◇ Alla base del lavoro di squadra (TEAMWORK)
- ◇ Commitments, Responsabilitá, Joint Goals
- ◇ Guidare nel traffico Vs Guidare in colonna
- ◇ Commitment: intenzione **persistente** di creare una certa situazione
- ◇ Convenzione: mezzo per monitorare un commitment

## Joint Intension II

◇ JPG - Joint Persistent Goal definito su WAG (Weak Achievement Goal)

◇ WAG ( $\mu, p, \Theta, q$ )

dove

- $\mu$  é un agente
- $\Theta$  é la squadra
- $p$  completamento di un'azione
- $q$  clausola di irrilevanza

◇ WAG ( $\mu, p, \Theta, q$ )  $\iff$

- $\text{Bel}(\mu, \neg p) \wedge \text{Des}(\mu, p)$  oppure
- $\mu$  sa che  $p$  é compiuto, non ottenibile o irrilevante (perché  $\neg q$ ) e vuole che questo diventi conoscenza mutua in  $\Theta$



# JPG

◇  $JPG(\Theta, p, q) \iff$

1. Tutti i componenti di  $\Theta$  credono mutuamente che  $\neg p$
2. Tutti hanno  $p$  come goal mutuo
3. Tutti credono mutuamente che finché  $p$  non é mutuamente noto per essere compiuto, non raggiungibile o irrilevante allora tutti credono mutuamente di avere  $p$  come WAG

◇ JPG viene istanziato con un protocollo detto di **commitment establishment**

# ARCHON e STEAM

## ◇ ARCHON

- Sistema di controllo produttivo industriale (Jennings 1996)
- Rule Based Systems
- Regole definiscono comportamenti relativi ai JPG

## ◇ STEAM

- Framework generico di coordinamento
- Architettura SOAR (HTN)
- 300 regole indipendenti dal dominio
- Decision theoretic per strategia di comunicazione

## Modello mutuo

- ◇ Modello degli altri agenti (1986 Genesereth et Al.)
- ◇ Approccio simile a Game Theory
- ◇ Si assume conoscenza comune (comunicazione non nec.)
- ◇ MACE
  - Primo testbed sperimentale per MAS (ca. 1987)
  - Usa modelli di conoscenza
    - Classe, Nome, Ruolo, Capacità, Goals, Piani
  - Azioni degli agenti basata sulla valutazione di una funzione LISP

## Normative e Leggi Sociali

- ◇ Modelli di comportamento
- ◇ Evitano Coordinamento
- ◇ Progettate offline
  - Maggior controllo del sistema
  - Spesso piú semplici da realizzare
- ◇ Emergenti
  - Non tutte le caratteristiche del sistema sono conosciute a priori
  - Maggiore adattabilità ai cambiamenti del sistema

## Normative e Leggi emergenti

- ◇ Raggiungere un accordo globale sulle convenzioni tramite informazioni locali
  
- ◇ Shoham, Tennenholtz Tee-Shirt Game
  - Insieme di agenti, due colori di Tee-Shirt (Colori = Strategie)
  - Agenti non vedono il colore degli altri
  - Agenti scelgono un colore random
  - Agenti possono vedersi a due a due ciascun turno
  - Ad ogni turno, Considerando **tutti** gli agenti conosciuti, ciascun agente decide il colore

## Tee-Shirt Game

- ◇ Diverse funzioni di aggiornamento della Strategia
  - Simple Majority
  - Simple Majority, Agent Type
  - Communicate on success
  - Cumulative reward
  
- ◇ Memory restart (evitare minimi locali)
- ◇ Velocità convergenza, stabilità
- ◇ Cumulative reward:
  - Garantisce convergenza
  - Stabile
  - Razionale individualmente

## Offline design

- ◇ Problematiche correlate alla progettazione di protocolli
- ◇ Complessità computazionale
- ◇  $Ag : R^E \rightarrow Ac$ 
  - $E$  insieme stati ambiente
  - $R$  insieme di runs
  - $R^E$  insieme di Runs che terminano con uno stato dell'ambiente
  - $Ac$  insieme di azioni
- ◇ Vincolo  $\langle E', \alpha \rangle \ E' \subseteq E, \alpha \in Ac$
- ◇ Legge Sociale  $SL = \{\langle E_1, \alpha_1 \rangle, \dots, \langle E_2, \alpha_2 \rangle\}$
- ◇ Focal State  $F \subseteq E$  stati sempre validi
- ◇ SL detta utile  $\iff$  permette che  $F$  sia raggiungibile da qualsiasi stato di  $F$

## Leggi Sociali in pratica

- ◇ Leggi per controllo del traffico
- ◇ Mondo bidimensionale popolato da robot che trasportano oggetti
  - Collisione = piú di un robot nella stessa posizione
  - Vogliamo trovare un SL utile



## Leggi Sociali in pratica

4	↓	←	←	←
3	→	→	↓	↑
2	↓	←	←	↑
1	→	→	→	↑

- Prossima mossa completamente determinata
- Tutte le caselle sono raggiungibili in  $O(n^2)$  mosse
- Ciclo amiltoniano

## Coordinamento Reattivo

- ◇ Nessuna pianificazione, Nessuno stato
- ◇ Due tecniche fondamentali
  - Potential field:
    - Determinare movimenti agenti
    - Evitare collisioni o interferenze
  - Marcare ambiente
    - Comunicazione Stigmergica
    - Evitare conflitti, migliorare performance

## Sistemi Anti Collisione

◇ Tecniche elementari basate su regole di comportamento (Reynolds 1987):

- Mantieni una distanza minima dagli altri agenti
- Adatta la tua velocità alla velocità media degli agenti
- Vai sempre verso il centro di gravità dei vicini

◇ Collective Robotics (Mataric 1994)

- Comportamenti molto semplici per i singoli robot
- Following, Aggregazione, Dispersione, ...
- Molto difficili da implementare sui robot
- Difficoltà dovuta a limiti sensoriali o all'approccio ?

## Sistemi anti collisione

### ◇ Banchi di pesci

- Sistema software di simulazione
- Ogni agente conosce la direzione di moto dei **vicini**
- $\vec{V}_i(t) = Norm[b[c \sum_{j=1}^{n_i} \alpha_{ij} \vec{R}_{ij}(t-1) + (1-c) \sum_{j=1}^{n_i} \beta_{ij} \vec{V}_{ij}(t-1)] + (1-b) \vec{B}(t)]$

### ◇ Campi di forza

- $U_i = U_{attr_i} + \sum_{i=1}^n U_{rep_i}$   $\vec{F}_i(p) = -\vec{\nabla} U_i(p)$
- Non considera le direzioni di moto degli altri agenti
- Aggiunta di forza di **Sliding**

$$-\vec{F}_{ij}(p) \vec{\nabla} U_{ij}(p) = 0 \quad \|\vec{F}_{ij}(p)\| = \|\vec{\nabla} U_{ij}(p)\|$$

## Marcare l'ambiente – Sincronia

- ◇ Marcatura: segno tangibile nell'ambiente
- ◇ (Hickman and Shiels, 1991):
  - due agenti che cooperano per assemblare un dispositivo elettronico
  - contenitori di componenti elementari dislocati nell'ambiente, un punto di assemblaggio
  - ordinare i componenti nel punto di assemblaggio
- ◇ Risolvere problemi di sincronia
  - Gli agenti prendono contemporaneamente lo stesso tipo di componente
  - Risolvibile con marker nel punto di assemblaggio

## Marcare l'ambiente – Performance

- ◇ Migliorare performance
- ◇ Robot trasportatori di bronzo
  - Robot sanno dove é la base
  - Devono trovare il giacimento di bronzo
  - Portare bronzo dal giacimento alla base fino a che il giacimento non esaurisce
- ◇ Semplici regole:

Rule Explore:

```
If I am not carrying anything  
AND I do not perceive a ore  
THEN explore random
```

## Marcare l'ambiente – Performance II

◇ Ogni agente che trova il giacimento al ritorno lascia una scia di marker

- Marker volatili (taratura del tempo di durata)
- Lascio  $n$  marker da giacimento a base e tolgo  $m$  marker da base a giacimento

◇ Azioni di coordinamento

- Catena di agenti che passano il bronzo di mano in mano
- Equilibrio in sistemi aperti