Multi-Agent Systems

Working Together

Summary

- ♢ Working together [Wooldridge Chap 8]
- ♢ Task assignment [Ferber Chap 7]

Working together

Cooperative Distributed Problem Solving

- No agent can solve the problem by itself
- Different capabilities, resources and knowledge

Two different scenarios:

- **Benevolent** agents
  - Agents **help** by design
  - Network of agents with common goal
- **Self-interested** Agents
  - agents act (and possibly cooperate) in their own interest
  - potential for **conflict**

Working together approaches

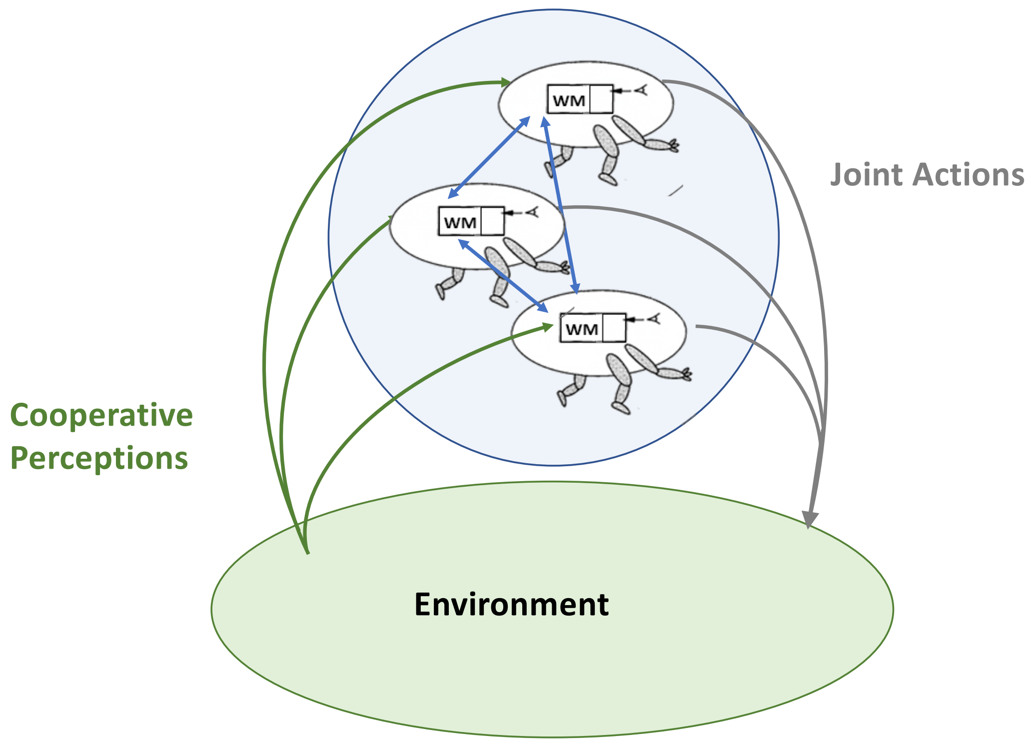
◇ Cooperation in perception and action

- **result sharing (Cooperative Perception)**
- **task sharing (Cooperative Action)**

◇ Coordination (i.e. avoiding interferences)

◇ Multi agent planning

Cooperation in perception and action



Cooperation: Result Sharing

Exchange of information **relevant** to the task

Attitude of agents:

- Passive
- Proactive

Communication model:

- Common model (i.e. Blackboard)
- Message passing (i.e. publish Subscribe)

Result integration (Distributed World Modeling):

- Information fusion
- Cooperative perception

Cooperation: Task Sharing

Division of labour for task sharing requires:

- task decomposition
- task allocation

Task **assignment** is the key process:

- statically defined tasks (roles)
- dynamically discovered tasks

More later

Coordination

♢ Systems that dynamically adapt their behaviour to other agents (special case **lack of conflicts**)

- Coordination in accessing common resources
- Coordination by norms and social laws
- Coordination by mutual modeling
- Coordination through **joint intentions**, the agents commit to a common goal until:
  - goal is achieved
  - goal becomes impossible
  - goal becomes irrelevant

Multiagent Planning

- Centralized planning for **distributed** plans
- Distributed planning
- Distributed planning for distributed plans
  - Coordination through **partial global planning**
- Plan **merging**: (using STRIPS plan representation)

Task sharing

Task decomposition:

- $\diamondsuit$ hierarchical
- $\diamondsuit$ granularity of subproblems difficult to define
- $\diamondsuit$ decomposition levels $\rightarrow$ abstraction levels

Typically tasks are defined a priori

Allocation modes

Once the need for help on a task is recognized allocation can be:

♢ Predefined

- Centralized
  - Imposed
  - Via Trader
- Distributed
  - Acquaintance Network
  - Contract Net Protocol

♢ Emergent

Centralized Allocation

♢ Imposed

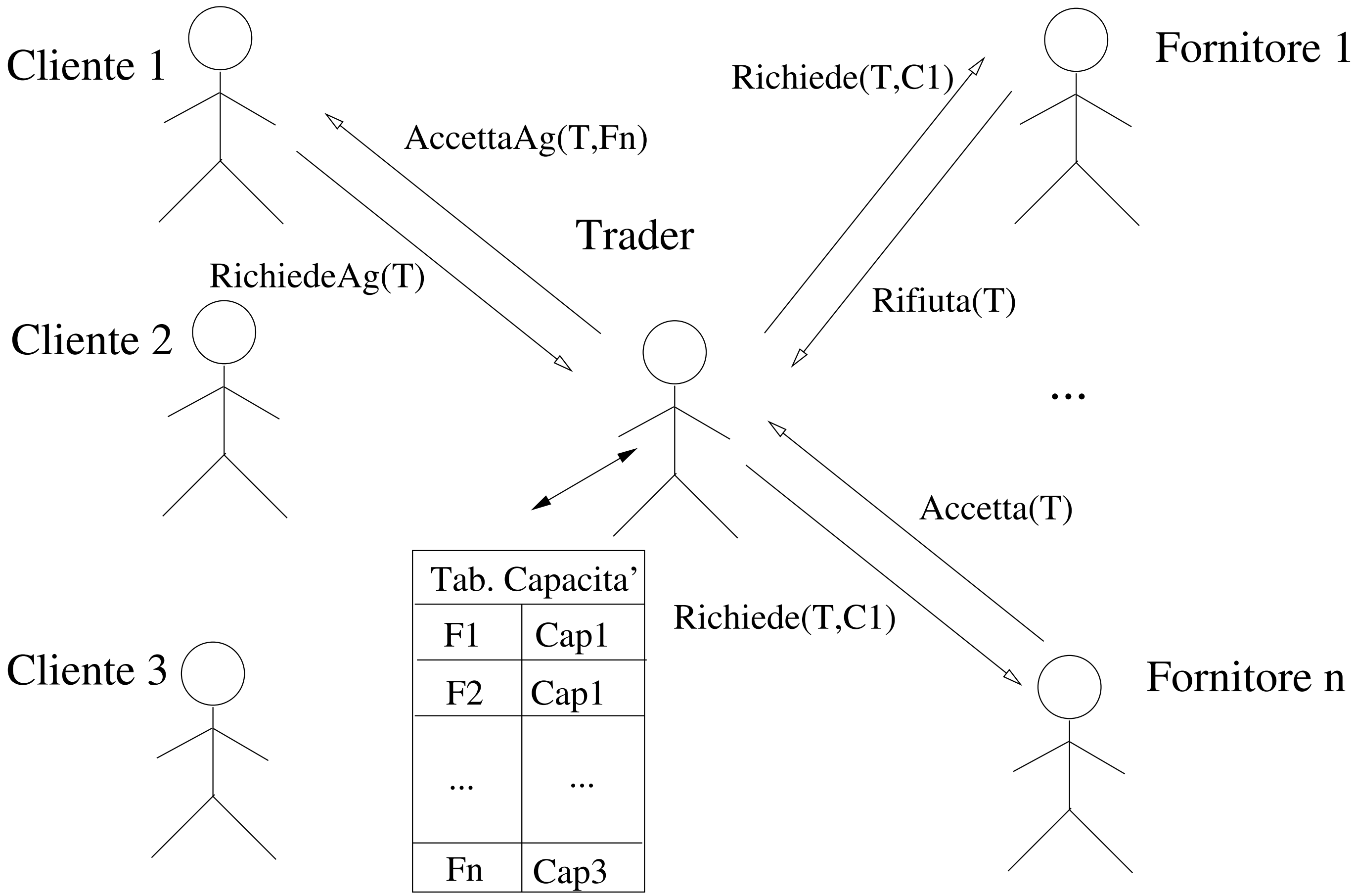
- subordination / hierarchical structure
- rigid allocation (a priori)
- procedure call in imperative languages

♢ Via Trader

- Trader is a special agent managing allocation
- Centralized techniques for changing settings

Centralized Allocation via trader

♢ Trader, Clients and Suppliers



Protocol for centralized allocation

Message exchange between **client and trader**

- ♢ `Request(T,X)`
- ♢ `Accept(T,Y)`
- ♢ `Impossible(T)`

Message exchange between **trader and supplier**

- ♢ `Request(T,X)`
- ♢ `Accept(T,Y,X)`
- ♢ `Refuse(T,Y,X)`

Centralized Allocation: features

$\diamondsuit$ coherence can be enforced

$\diamondsuit$ number of messages exchanged $= \alpha k N(2+2\beta N) \Rightarrow O(N^2)$

- α = Potential Clients/total Clients
- β = Potential Suppliers /total Suppliers
- k = requests per time unity

$\diamondsuit$ very sensitive to failures

Allocation via Acquaintance Network

	Ag_1	Ag_3	Ag_j
Cap_1	0	1	1
Cap_2	0	0	0
Cap_i	1	0	0

$\diamondsuit$ Simplifying assumptions:

Tables are correct, complete and static.

Direct Allocation

♢ Every agent allocates tasks only to its direct contacts (sequentially)

♢ If none accepts:

- Force allocation
- Central Agent

♢ If the request is broadcasted among the direct contacts
$\rightarrow$ CNP

♢ Agents are unaware of other agents not in direct contact

Allocation by delegation (tables are incomplete)

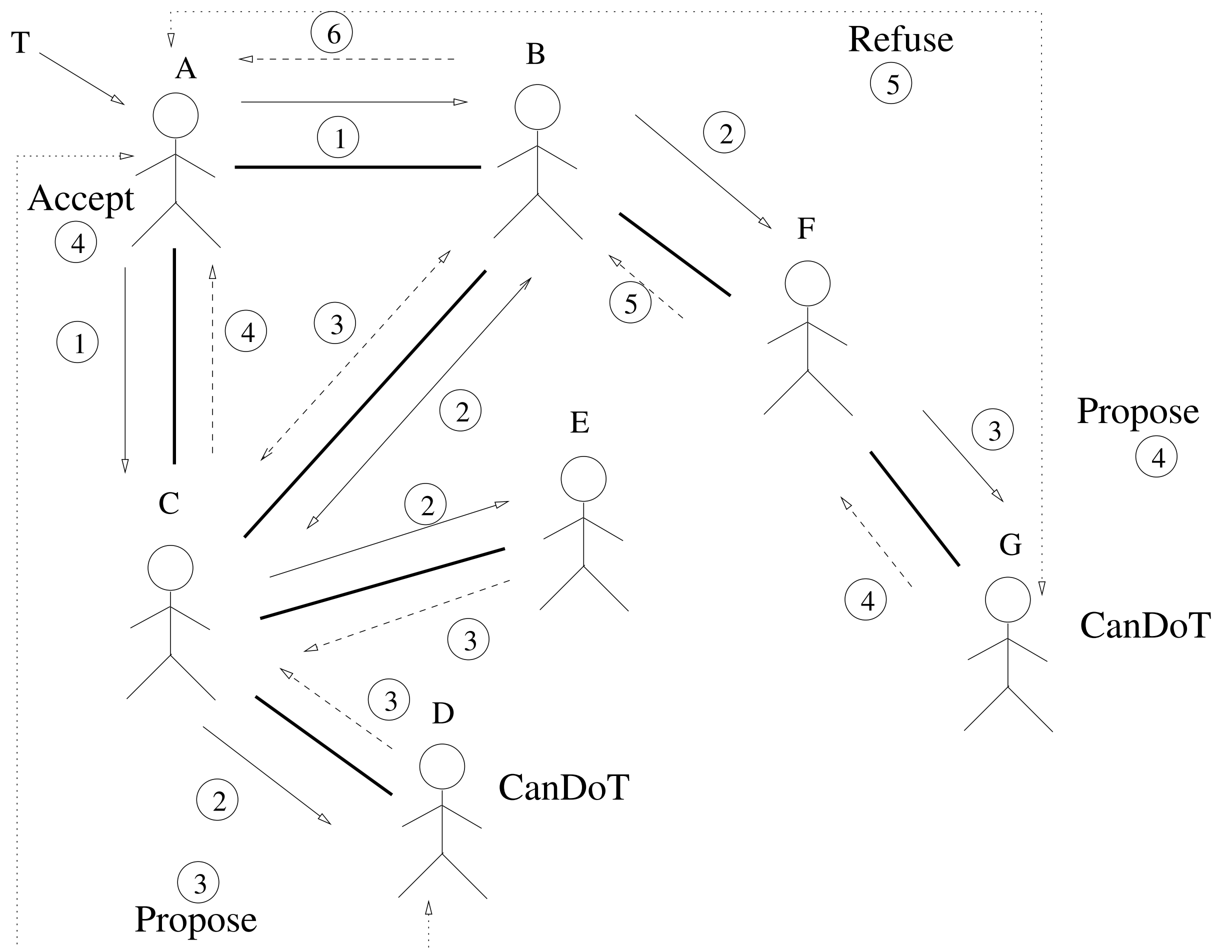
♢ search the whole network

- depth
- breadth

♢ Parallel breadth search (Diffusion algorithms)

- Visit all agents: Acknowledge (searched all the subtree)
- Visit them only once: Marking

Allocation by delegation: example



Allocation by delegation: challenges

◇ Optimized search (send messages to everybody)

◇ Useless computation after at least one agent has accepted

◇ Proposal selection

Rearrange an acquaintance network

♢ Change Capabilities

- Communicate when a capability changes
  - Broadcast the change: who knows my capabilities ? (bidirectionality needed and increased complexity)
  - Syncronization between change and new requests
- Adaptive network
  - Agent changes its network after a denial
  - Frequency of changes becomes critical

♢ Agent insertion and elimination

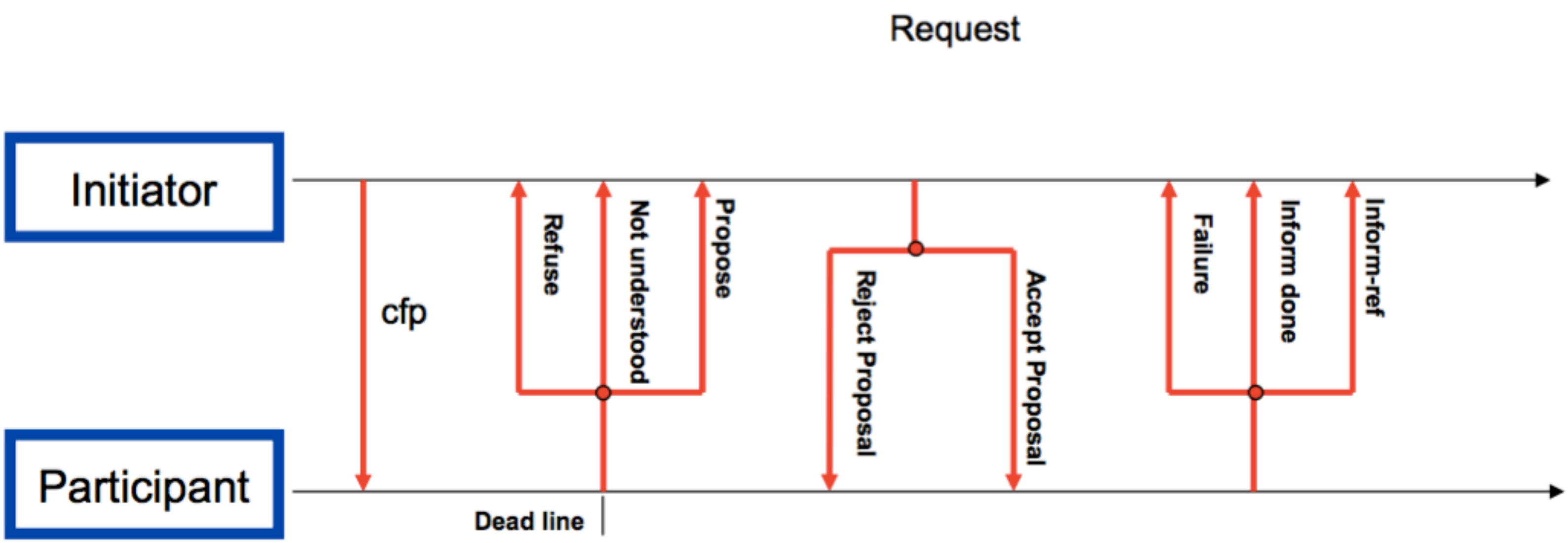
Contract Net Protocol

- ♢ Smith (1979)
- ♢ Simple and easy to implement
- ♢ **Manager** and **Suppliers** communicate via **offers**

CNP alg. single Manager

1. **Announcement**: Manager sends a task **description** to all possible suppliers (RequestForBid(T,M))
2. **Bidding**: Suppliers **evalute** the offer and send a **proposal** to the Manager (Propose(T,Off,Ag) or NotInterested(T,Ag))
3. **Awarding**: Manager **evaluates** the proposals and allocates the contract to the best supplier (Award(T,Ag,M))
4. **Expediting**: The chosen supplier replies positively or negatively to the Manager (Accept(T,Ag) Refuse(T,Ag))

CNP in FIPA ACL



CNP messages

◇ Smith defined a contract specification language

- requirements
- format
- deadline for presentation

Contract language

```
Message: RequestForBids
  To: *
  From: A21
  DescriptionOfTask:
    TypeOfTask: check-feature
  QualitiesRequired :
    MustHave: Camera
  FormOfProposal:
    Position : (X,Y)
  DateExpiry: 12:00
EndRequestForBids
```

CNP Features

♢ Contacting all the agents is problematic

- Token-Ring solution is possible (tradeoff with speed)
- Caching (acquaintance) to minimize communication (tradeoff with performance)

♢ Limit date for the contract:

- avoids useless communications with non interested agents
- avoid blockages due to agent malfunctioning
- fixed waiting time $\rightarrow$ inefficiency

CNP with many managers

♢ many manager → interference

♢ incomplete knowledge about suppliers:

- time relative (no prediction of future tasks), but also for the single manager
- space relative (other proposals)

	T_a	T_b
X	90	80
Y	80	20

- relative to the working load (also for single manager)

CNP synchronization problems

◇ Timing of proposal submission and award arrival

- Cautions Agents
  - propose only doable
  - with few agents is a good strategy
- Brave Agents
  - Propose also when not capable
  - may cause task resubmissions
  - good with many agents
- Moderate Agents
  - decision theoretic approach to make proposals
  - weight resources by evaluating the chances of acceptance of the proposal

CNP with subcontracting

- ♢ Suppliers can sub-contract
- ♢ Like previous situation but with shared resources
- ♢ several strategies

- Early commitment
  - Suppliers send proposal before having necessary resources
  - Several riorganizations needed
  - quicker
- Late commitment
  - Suppliers send proposal after having acquired necessary resources
  - Possible deadlocks
- Fixed agencies
  - less adaptive, more stability

CNP features

◇ Pros

- simple and easy to implement
- dynamic and easily adaptable
- bilateral contract $\rightarrow$ several parameters in the allocation

◇ Cons

- many messages $O(nm)$
- synchronization problems
- challenging with sub-contracting

CNP Variations

◇ CNP: proposal driven

◇ Acquaintance network + CNP

- task differentiation (complexity/priority): allocation in acquaintance network (quick, inaccurate), CNP (expensive, more accurate)
- use CNP only when acquaintance network fails

◇ Acquaintance network → cache memory for CNP

Emergent allocation

- ♢ Task Allocation for reactive agents
- ♢ Signals in place of messages (stigmergic comm.)
- ♢ Agents' behaviors depend on signal strength...
- ♢ ...and internal tendency of the agents (simple state)
- ♢ Used in MRSs (no cost for direct comm.)
- ♢ Not focussed on efficiency, but on scalability
- ♢ Artificial Life, Swarm (Brooks, Mataric, Parker)