LECTURE 9: WORKING TOGETHER

An Introduction to Multiagent Systems

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Why and how do agents work together?

Important to make a distinction between:

- Self-interested agents.
- Benevolent agents.

1. Working Together
Benevolent agents: If we "own" the whole system, we can design agents to help each other whenever asked. If we can assume agents are benevolent, our best interest is their best interest. In this case, we can solve problems cooperatively distributed problem solving (CDPS).

Problem-solving in benevolent systems is cooperative distributed problem solving (CDPS).

Benevolence simplifies the system design task enormously!
1.2 Self-Interested Agents

If agents represent individuals or organisations, the more general case, then we cannot make the benevolence assumption: agents will be assumed to act to further their own interests, possibly at expense of others.

Potential for conflict. May complicate the design task enormously.
Lecture 9 - An Introduction to Multiagent Systems

2. Task Sharing and Result Sharing

Twomainmodesofcooperativeproblemsolving:

- **Result sharing:** Information (partial results etc.) is distributed.
  - Components of a task are distributed to component agents.

- **Task sharing:**
  - Task sharing.

Two main modes of cooperative problem solving:

- 2 Task Sharing and Result Sharing
Well known task-sharing protocol for task allocation is **Contract Net**:

1. Recognition;
2. Announcement;
3. Bidding;
4. Awarding;
5. Expediting.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
I have a problem...

(a) Recognising the problem
(b) Task announcement
(c) Bidding
(d) Awarding the contract

A1
A2
A3
A4

http://www.csc.liv.ac.uk/~mjw/pubs/tmas/
In this stage, an agent recognises it has a problem it wants help

- realises it cannot achieve the goal in isolation — does not
  have capability;
- realises it would prefer not to achieve the goal in isolation — typically because of solution quality, deadline, etc.

Agent has a goal, and either...
3.2 Announcement

In this stage, the agent with the task sends out an announcement which includes a specification of the task to be achieved.

- Specification must encode:
  - task to be achieved.
  - description of task itself (maybe executable);
  - any constraints (e.g., deadlines, quality constraints);
  - meta-task information (e.g., “bids must be submitted by...”);

The announcement is then broadcast.
Agentsthatreceivetheannouncementdecidethemselveswhethertheywishtobidforthetask.

Factors:
- Agentmustdecidewhetheritiscapableofexpeditingtask;
- Agentmustdeterminequalityconstraints&priceinformation;
- (ifrelevant).

Iftheydochoosetobid,thentheysubmita**tender**.

**Agents** that receive the announcement decide for themselves whether they wish to bid for the task.
Agent that sent task announcement must choose between bids & decide who to "award the contract" to.

The successful contractor then expedites the task.

May involve generating further manager-contractor relationships:

- sub-contracting.
- submittted a bid.
- The result of this process is communicated to agents that submitted a bid.
- The contractor then expedites the task.
- May involve generating further manager-contractor relationships:

3.4 Awarding & Expediting
3.5 Issues for Implementing Contract Net

- How to... 
- ... specify tasks? 
- ... specify quality of service? 
- ... select between competing offers? 
- ... differentiate between offers based on multiple criteria?
The first scheme for cooperative problem solving: the blackboard system.

- Result shared via shared data structure (BB).
- Multiple agents (KSs/KAs) can read and write to BB.
- BB may be structured into hierarchy.
- Mutual exclusion over BB required → bottleneck.
- Not concurrent activity.
- Agents write partial solutions to BB.
- Comparison: LINDA tuple spaces, JAVASPACES.

The first scheme for cooperative problem solving: the blackboard system.

4 Result Sharing in Blackboard Systems
5 Result Sharing in Subscribe/Notify Pattern

- Common design pattern in OO systems: subscribe/notify.
- An object subscribes to another object, saying “tell me when event e happens.”
- When event e happens, original object is notified.
- Information pro-actively shared between objects.
- Objects required to know about the interests of other objects.
- Inform objects when relevant information arises.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/