CHAPTER 8: WORKING TOGETHER

Multiagent Systems

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

Why and how do agents work together?

- Since agents are autonomous, they have to make decisions at run-time, and be capable of dynamic coordination.

- If agents are designed by different individuals, they may not have common goals.

- Overall, they will need to be able to share:
  
  - Information
  
  - Tasks

Chapter 8
Important to make a distinction between:

– *benevolent agents* and
– *self-interested agents*. 
Benevolent Agents

• If we "own" the whole system, we can design agents.

• In this case, we can assume agents are benevolent.

• Our best interest is their best interest.

• In many cooperative (CDPs), problem-solving in benevolent systems is cooperative problem-solving.

• Benevolence simplifies the system design task enormously!

• We will talk about CDSP in this lecture.
Chapter 8
An Introduction to Multiagent Systems
Self-Interested Agents

• If agents represent the interests of individuals or organisations, (the more general case), then we cannot make the benevolence assumption:
  • Agents are assumed to act in their own interests, possibly at the expense of others.
  • Strategic behaviour may be required — we will cover

Game theory

- Strategic behaviour may be required — we will cover
- May complicate the design task enormously.
- Potential for conflict.
- Some of these aspects in later lectures.
Criteria for assessing an agent-based system:

- Coherence
- Coordination
- Criteria for assessing an agent-based system.

We can measure coherence in terms of solution quality, how efficiently resources are used, conceptual clarity and so on. How well the [multiagent] system behaves as a unit along some dimension of evaluation (Bond and Gasser).
Coordination

•

Coordination

metaphorical sense. If the system is perfectly coordinated, agents will not get in each other's way, in a physical or a metaphorical sense. In a perfectly coordinated system, agents will avoid extraneous activity [such as ... synchronizing their activities (Bond and Gasser).] The degree to which [the agents] can avoid...
A group of agents work together to solve problems.

There are three stages:

- Problem decomposition
- Sub-problem solution
- Answer synthesis

Let's look at these in more detail.

Task Sharing and Result Sharing
Chapter 8: An Introduction to Multiagent Systems

Problem Decomposition

How this is done is one design choice.

Clearly there is some processing to do the division.

- Individual program instructions.

- In ACTORS, this is done until we are at the level of
  subproblems get divided up also.

- This is typically a recursive/hierarchical process.

- The overall problem to be solved is divided into smaller sub-problems.

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Another choice is who does the division.

- Who is going to solve the sub-problems?
- Which agents have knowledge of task structure?
- Is it centralized?
Chapter 8: An Introduction to Multiagent Systems

Sub-problem solution

- The sub-problems derived in the previous stage are solved.
- Agents typically share some information during this process.
- A given step may involve two agents synchronizing their actions.
In this stage solutions to sub-problems are integrated.

- Again this may be hierarchical.
- Different solutions at different levels of abstraction.

Solution synthesis
Given this model of cooperative problem solving, we have two activities that are likely to be present:

- **Task sharing**: how do we decide how to allocate tasks to agents;
- **Result sharing**: how do we assemble a complete solution from the different parts (partial results etc.) is distributed.

Given this model of cooperative problem solving, we have two activities that are likely to be present:

- Task sharing: how do we decide how to allocate tasks to agents;
- Result sharing: how do we assemble a complete solution from the different parts (partial results etc.) is distributed.

- Task sharing: how do we decide how to allocate tasks to agents;
- Result sharing: how do we assemble a complete solution from the different parts (partial results etc.) is distributed.

- Task sharing: how do we decide how to allocate tasks to agents;
- Result sharing: how do we assemble a complete solution from the different parts (partial results etc.) is distributed.

The textbook describes these stages in procedural terms from the perspective of an individual agent.

Well-known task-sharing protocol for task allocation is

**The Contract Net**

http://www.cse.liv.ac.uk/~mjw/pubs/imas/
Chapter 8
An Introduction to Multiagent Systems
As a result, it needs to involve other agents.

- A realises it cannot achieve the goal in isolation (typically because of solution quality, deadline, etc).
- A realises it would prefer not to achieve the goal in isolation — does not have capability.
- A realises it cannot achieve the goal in isolation —
- A has a goal, and either...
- In this stage, an agent recognises it has a problem.

Recognition
In this stage, the agent with the task sends out an announcement.

The announcement is then broadcast.

Specification must encode:

- meta-task information (e.g., "bids must be submitted by...")
- constraints (e.g., deadlines, quality constraints)
- description of task itself (maybe executable)

Specification of the task to be achieved.

Any constraints (e.g., deadlines, quality constraints)
Agentsthatreceivetheannouncementdecidefor themselveswhethertheywishtobidforthetask.

Factors:
- Agentmustdecidewhetheritiscapableof expeditingtask;
- Agentmustdecidewhetheritiscapableof determiningqualityconstrictions&price.

Iftheydochoosetobid,theysubmitatender.

Information(ifrelevant).

Bidding
Agent that sent task announcement must choose between bids & decide who to "award the contract" to.

The result of this process is communicated to agents.

May involve another contract net.

- May involve another contractor net.
  - May involve generating further manager-contractor relationships: sub-contracting.
  - The successful contractor then expedites the task.

Awarding & Expediting

Theresultofthisprocesisiscommunicatedtoagents

The FIPA ACL was designed to be able to capture the contract net.

*The Contract Net via FIPA Performatives*

The FIPA ACL was designed to be able to capture the contract net.

- **Proposal:** Used to indicate acceptance or rejection of a proposal.
  - **Accept:**
  - **Reject:**

- **Proposal:** Used for making a proposal, or declining to make a proposal.
  - **Propose:**
  - **Refuse:**

- **Call for proposals:** Used for announcing a task.

**Call for proposals:**

- **Call for proposals:**

**Contract net:**

- **Contract net:**
inform, failure:

Used to indicate completion of a task (with the result)

or failure to do so.
failure
inform
refuse
accept
propose
propose
propose
propose
... contractor n
contractor 2
contractor 1
manager
OR
http://www.csc.liv.ac.uk/~mjw/pubs/imas/
• Issues for Implementing Contract Net

• How to...?
  - ... specify tasks?
  - ... differentiate between offers based on multiple criteria?
  - ... select between competing offers?
  - ... decide how to bid?
  - ... specify quality of service?
Deciding how to bid

At time $t$ a contractor $i$ is scheduled to carry out $\tau_t^i$. Then $i$ receives an announcement of task specification $\tau_s \subseteq \tau_t^i$, which is for a set of tasks $\tau_t^i \cup \tau_s$. Contractor $i$ also has resources $e$. Contractor $i$ also has resources $e$. The marginal cost of carrying out $\tau_t^i$ will be:

$$\mu_i(\tau_s | \tau_t^i) = (\frac{1}{\tau} \cap (\tau_s) \cap i \cap (\tau_s) \cap i)$$

plus the new tasks:

that is the difference between carrying out what it has already agreed

$$(\frac{1}{\tau} \cap i \cap \tau_t^i) - (\frac{1}{\tau} \cap (\tau_s) \cap i \cap (\tau_s) \cap i) = (\frac{1}{\tau} \cap (\tau_s) \cap i)$$
Chapter 8: An Introduction to Multiagent Systems

• Due to synergies, this is often not just:

\[ \text{infact, it can be zero} \]

\[ (s_i)_\perp \]

\[ c_i \]

• Otherwise not:

\[ \text{Think of the cost of giving another person a ride to} \]

\[ \text{done for free.} \]

\[ \text{As long as } \theta > (i \perp | (s_i)_\perp)_\perp \]

\[ \text{the agent can afford} \]

\[ \text{to do the new work, then it is rational for the agent to} \]

\[ \text{bid for the work.} \]

\[ \text{if } \tau \text{(ts)} \]

\[ \text{in fact, it can be zero — the additional tasks can be} \]

\[ e > \]

\[ \text{not just:} \]

http://www.csc.liv.ac.uk/~mjw/pubs/imas/26
In result sharing, agents provide each other with information as they work towards a solution. In results sharing, agents provide each other with information as they work towards a solution.

It is generally accepted that result sharing improves problem solving.

- Combining local views can achieve a better overall view.
- Independent pieces of a solution can be cross-checked.
- Shared results can improve the accuracy of results.
- Independent pieces of a solution can be cross-checked.
- Combining local views can achieve a better overall view.
- Shared results can improve the accuracy of results.

Shian” resuls allows the use of parallel resources.

Result Sharing

Sharing results allows the use of parallel resources.
Chapter 8
An Introduction to Multiagent Systems

Result Sharing in Blackboard Systems

- The first scheme for cooperative problem solving was the blackboard system.
- Results shared via shared data structure (BB).
- BB may be structured into hierarchy.
- Multiple agents (KAs) can read and write to BB.
- Mutual exclusion required over BB.
- Not concurrent activity.
- Agents write partial solutions to BB.

LINDA tuplespaces, JAVASPACES

The first scheme for cooperative problem solving was:

http://www.csc.liv.ac.uk/~mjw/pubs/tmas/
Result Sharing in Publish/Subscribe Pattern

In multiagent systems, result sharing often involves the following pattern:

- **Objects** inform other objects when relevant information arises. When an event happens, the original object is notified.
- Information is proactively shared between objects.
- Objects required to know about the interests of other objects.
- When an event happens, another object subscribes to tell me about it.

**Publish/Subscribe**

Common design pattern in multiagent systems.
The Centibots robots collaborate to map a space and find objects.

http://www.csc.liv.ac.uk/~mjw/pubs/tmas/
Handling inconsistency

• A group of agents may have inconsistent beliefs in their
• Inconsistent beliefs arise because agents have different views of the world.
• Inconsistent goals may arise because agents are built by different people with different objectives.

• Inconsistent goals may arise because they can’t see everything.
  – May be due to sensor faults or noise or just different views of the world.

• Inconsistent beliefs arise because agents have goals or intentions.

• Goals or intentions
Three ways to handle inconsistency (Durfee et al.)

- Build systems that degrade gracefully in the face of inconsistency.
- Do not allow it.

We will discuss this later (argumentation).

The inconsistency goes away.

Agents discuss the inconsistent information/goals until

Resolve inconsistency.

For example, in the contract net the only view that matters is that of the manager agent.

An Introduction to Multiagent Systems 2e

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Coordination is managing dependences between agents.

Example

We both arrive at the door at the same time. What do we will do to ensure we can both get through the door?

We want to leave the room through the same door. We are walking such that we will arrive at the copy room with a stack of paper to photocopy. Who gets to use the machine first?
• Von Martial suggested that positive coordination is:
  - Requested (explicit)
  - Non-requested (implicit)

Non-requested coordination relationships can be as:

- Requested (explicit)
- Non-requested (implicit)

von Martial suggested that positive coordination is:

- Favor: What I plan to do will make it easier for you to do.
- Consequence: What I plan to do will have the side-effect of achieving something you want to do.
- Action equality: We both plan to do something, and by recognizing this one of us can be saved the effort.
Societies are often regulated by (often unwritten) rules of behavior.

Social norms

In an agent system, we can design the norms and program agents to follow them, or let norms evolve.

Example:
A group of people is waiting at the bus stop. The bus arrives. Who gets on the bus first?

Another example:
On 34th Street, which side of the sidewalk do you walk along?

Social norms

Chapter 8
An Introduction to Multiagent Systems 2e
http://www.csc.liv.ac.uk/~mjw/pubs/imas/
A social law is a set of constraints. A constraint is then a pair:

\[ \langle \alpha', E' \rangle \]

This constraint says that \( \alpha \) cannot be done in any state in \( E' \).

A social law is a set of constraints. Recall how we described agents before:

\[ \text{Offline design} \]
We can refine our view of an environment. Focal states, \( F \subseteq E \), are the states we want our agents to be able to get to. Focal states, \( F \subseteq E \), are the states we want our agents to be able to get to.

A useful social law is then one that does not prevent right away to any other focal state \( e \in F \) (though not necessarily from any focal state \( e \in F \) it should be possible to get to another.

Focal states, \( F \subseteq E \), are the states we want our agents to be able to get to.
A useful social law that prevents collisions:

Not necessarily efficient ($O(n^2)$ steps to get to a specific square).
Emergence

• We can also design systems in which social laws emerge.

T-shirt game (Shoham and Tennenholtz):

• T-shirt game (Shoham and Tennenholtz):

What strategy update function should they use?

• They don't get any other information. They only see the shirt their pair is wearing.

Agent meets one other agent, and decides whether or not to change their shirt. During the round they only see the shirt their pair is wearing. In each round, each agent have both a red t-shirt and a blue t-shirt, and wear one. Goal is for everyone to end up with the same color on.

Emergence
Strategy Update Functions

• **Simple majority**: Agents pick the shirt they have seen the most.

• **Simple majority with types**: Agents come in two types. When they meet an agent of the same type, agents pass their memories. Otherwise they act as simple majority.

• **Highest cumulative reward**: Agents can "see" how often other agents (some subset of all the agents) have matched their pair. They pick the shirt with the largest number of matches.

• **Simple majority with types**: Agents pick the shirt they have seen the most.
Joint Intentions

- Just as we have individual intentions, we can have joint intentions.
- A group of agents have a collective commitment to bring about some goal, \( \phi \), "move the couch".
- Also have motivation, \( \psi \), "Simon wants the couch moved".
- Levesque defined the idea of a joint persistent goal (JPG).
• Agents don't believe that \( \phi \) is satisfied, but believe it is possible.
• Agents maintain the goal \( \phi \) until a termination condition is reached.
The terminations condition is that it is mutually believed that:

- goal is impossible; or
- goal is satisfied; or
- the motivation is no longer present.

You and I have a mutual belief that this is d $\phi$ and you believe that if I believe and I believe that you believe and I believe that you believe that I believe and ...
• The termination condition is achieved when an agent realizes that the goal is satisfied, impossible, and so on.

• But it doesn’t drop the goal right away.

• Instead, it adopts a new goal — to make this new knowledge mutually believed.

• This ensures that the agents are coordinated.

• They don’t stop working towards the goal until they are all appraised of the situation.

• Mutual belief is achieved by communication.

Page 44
Chapter 8
An Introduction to Multiagent Systems 2e
http://www.csc.liv.ac.uk/~mjw/pubs/tm/s/
Another approach to coordinate is to explicitly plan for another group of agents.

A group of agents comes up with a centralized plan:

- Distributed planning
  - One agent comes up with a plan for everybody
- Centralized planning for distributed plans

Could have:

- The agents in a system.
- For example, come up with a large STRIPS plan for all agents.
- What all the agents do.

**Multiagent planning**
Agents build up plans for themselves, but take into account the actions of others.

Distributed planning for distributed plans

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Chapter 8
An Introduction to Multiagent Systems

In general, the more decentralized it is, the harder it is...

– Interaction analysis: do different plans affect one another?

### Diffrent agents plan to achieve their goals using these operators and then do:

- Things.
- This places constraints on when other agents can do:

  - New list: during
  - Georgeff proposed a distributed version of STRIPS.

- Specifies what must be true while the action is carried out.

- During

- New list: during

- Georgeff proposed a distributed version of STRIPS.

- Specifies what must be true while the action is carried out.

- Things.

In general, the more decentralized it is, the harder it is...

- http://www.csc.liv.ac.uk/~mjw/pubs/imas/
– Safety analysis: which interactions are problematic?

– Interaction resolution: treat the problematic interactions as critical sections and enforce mutual exclusion.

– Safety analysis: which interactions are problematic?
Chapter 8
An Introduction to Multiagent Systems

Summary

This lecture has discussed how to get agents working together to do things.

• We discussed a number of ways of having agents coordinate.
• A typical system will need to use a combination of these ideas.