Chapter 8: Working Together

Multiagent Systems

Why and how do agents work together?

- Benevolent agents

Benevolent agents simplify the system design task.

- Problem-solving (CDPS)

Problem-solving in benevolent systems is cooperative.

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

- If we "own" the whole system, we can design agents to help each other whenever asked.

Important to make a distinction between:

- Self-interested agents

- Benevolent agents

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

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

- Information

- Tasks

- Overall, they will need to be able to share coordination decisions at run-time, and be capable of dynamic coordination.

- Since agents are autonomous, they have to make dynamic decisions on their own.

- Why and how do agents work together?

Working Together
An Introduction to Multiagent Systems

Chapter 8

Self-Interested Agents

• If agents represent the interests of 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 the expense of others.
  - Potential for conflict.
  - May complicate the design task enormously.
  - Can lead to situations where agents may need to coordinate their actions to avoid conflict.

Coherence and Coordination

• Coherence
  - How well the multiagent system behaves as a whole.
  - Criteria for assessing an agent-based system.

• Coordination
  - If the system is perfectly coordinated, agents will not get in each other's way.
  - If the system is not perfectly coordinated, agents will engage in activities (such as synchronizing and aligning their activities) to avoid "extraneous" activity.

Task Sharing and Result Sharing

• How does a group of agents work together to solve problems?
  - Problem decomposition
  - Sub-problem solution
  - Answer synthesis

• A task to which the agents are assigned.

Chapter 8

Self-Interested Agents

• If agents represent the interests of 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 the expense of others.
  - Potential for conflict.
  - May complicate the design task enormously.

Coherence

• Criteria for assessing an agent-based system.
  - How well the multiagent system behaves as a whole.
  - How efficiently resources are used.

• Potential for conflict.
  - May complicate the design task enormously.
  - Agents may need to coordinate their actions to avoid conflict.

Task Sharing and Result Sharing

• How does a group of agents work together to solve problems?
  - Problem decomposition
  - Sub-problem solution
  - Answer synthesis

• A task to which the agents are assigned.
Problem decomposition

- The overall problem to be solved is divided into sub-problems.
- Clearly there is some processing to do the division.
- This is typically a recursive/hierarchical process.
- Smaller sub-problems get divided up also.
- Sub-problems are solved.

Sub-problem solution

- A given step may involve two agents synchronizing.
  - Another choice is who does the division.

Solution synthesis

- Different solutions at different levels of abstraction.
  - Again this may be hierarchical.
  - Clearly there is some processing to do the division.

Who is going to solve the sub-problems?

- Which agents have knowledge of task structure?

- Is it centralized?

Another choice is who does the division.
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**: information (partial results etc.) is distributed. How do we assemble a complete solution from the parts?

### The Contract Net

A well-known task-sharing protocol for task allocation is the contract net.

The contract net includes five stages:

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

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

- **Task 1**
  - **Task 1.1**
  - **Task 1.2**
  - **Task 1.3**

- **result sharing**
- **task sharing**

- **decomposition**
  - **solution**
  - **synthesis**
• If they do choose to bid, then they submit a tender.

  information (if relevant).

- agent must determine quality constraints & price
- expediting costs.
- agent must decide whether it is capable of:

  - Factors:

  - themselves whether they wish to bid for the task.

- Agents that receive the announcement decide for

  bidding.

The announcement is then

  submitted by...”

- meta-task information (e.g., this must be
- any constraints (e.g., deadlines, quality constraints).
- description of task itself (maybe executable).

- Specification must encode:

  - specification of the task to be achieved.

  announcement of the task which includes a

  announcement to the task sends out an

  Announcement.

- In this stage, the agent with the task sends out an

  broadcast

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

Chapter 8 An Introduction to Multiagent Systems

- As a result, it needs to involve other agents.

  deadline, etc)

  isolation (typically because of solution quality.

- realizable if would prefer not to achieve the goal in

  – unrealizable if cannot achieve the goal in isolation —

  • Agent has a goal and either:

  – wants help with.

  – In this stage, an agent recognizes it has a problem it

  Recognition

  • wants help with.

  – In this stage, an agent recognizes it has a problem it

  Recognition

  • wants help with.

  – In this stage, an agent recognizes it has a problem it

  Recognition
• Agent that sent task announcement must choose between bids and decide who to "award the contract" to.
• The result of this process is communicated to agents that submitted a bid.

The successful contractor then expedites the task.

The Contract Net via FIPA Performatives

Proposal
Used to initiate acceptance or rejection of a proposal.
• accept, reject:

Proposal
Used for making a proposal or declining to make a proposal, refuse:
• propose, refuse:

Inform
Used for announcing a task.
• cfp (call for proposals): contractor met:

The FIPA ACL was designed to be able to capture the

Awarding & Expediting

May involve another contractor...

The result of this process is communicated to agents between bids & decides who to "award the contract" to.
Chapter 8: An Introduction to Multiagent Systems

2. Issues for Implementing Contract Net

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

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

2.4

Deciding how to bid

At time t a contractor i is scheduled to carry out τ_i

The marginal cost of carrying out \( \tau \) will be:

\[ c_i(\tau) - c_i(\tau_i) \]

These will cost \( c_i(\tau) \) to carry out, which is for a set of tasks \( \tau(s) \).

Then if receives an announcement of task specification

Contractor i also has resources \( e_i \)

otherwise not
did for the work. To do the new work, then it is rational for the agent to
think of the cost of giving another person a ride to
done for free.

In fact, if can be zero — the additional tasks can be
due to synergies; this is often not just:

\[ c_i(\tau) \]

Result Sharing

- In result sharing agents provide each other with

\[ Result Sharing \]

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
Result Sharing in Blackboard Systems

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

Result Sharing in Subscribe/Notify Pattern

- Common design pattern in OO systems:
  - Subscribers inform objects when relevant information changes.
  - Objects required to know about the interests of other objects.
  - Information proactively shared between objects.
  - When event happens, original object is notified.

Result Sharing

- A group of agents may have inconsistencies in their:
  - Handling inconsistency

Handling Inconsistencies

- The Centibots robots collaborate to map a space and find objects.

- Inconsistent goals may arise because agents are built by different people with different objectives.
  - May be due to sensor faults or noise or just different views of the world.
  - Goals or intentions may be inconsistent.
  - Beliefs may be inconsistent.

- A group of agents may have inconsistencies in their:
  - Handling Inconsistency

- Common design pattern in OO systems:
Three ways to handle inconsistency (Durfee et al.).

- Don't allow it. For example, in the contract net the only view that matters is that of the manager agent.
- Resolve inconsistency. Agents discuss the inconsistent information/goals until the inconsistency goes away.
- Build systems that degrade gracefully in the face of inconsistency. We will discuss this later (argumentation).

Inconsistency:

In an agent system, we can design the norms and walk along.

On 34th Street, which side of the sidewalk do you walk on? Another example: bus arrives. Who gets on the bus first? A group of people is walking at the bus stop. The rules of behavior. Example: agents are often regulated by (often unwritten) social norms.

Von Martialis suggested that positive coordination is:

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

Non-requested coordination relationships can be as follows.

- Action equality: we both plan to do something, and by recognizing this one of us can be saved the effort.
- Consequence: What I plan to do will have the side-effect of achieving something you want to do.
- Favor: What I plan to do will make it easier for you to do what you want to do.

Coordination is managing dependencies between agents.

Example:

First: paper to photostore, who gets to use the machine? We both arrive at the copy room with a stack of paper to photostore. Who gets to use the machine? We both arrive at the copy room at the same time. What do we do if we arrive at the door at the same time? We walk through the same door. We are walking such that we will both want to leave the room through the same door.

Social norms:

Societies are often regulated by (often unwritten) rules of behavior. Example:

Societies are often regulated by (often unwritten) rules of behavior. Another example:

On 34th Street, which side of the sidewalk do you walk on? Another example: bus arrives. Who gets on the bus first? A group of people is walking at the bus stop. The rules of behavior. Example: agents are often regulated by (often unwritten) social norms.

Von Martialis suggested that positive coordination is:

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

Non-requested coordination relationships can be as follows.

- Action equality: we both plan to do something, and by recognizing this one of us can be saved the effort.
- Consequence: What I plan to do will have the side-effect of achieving something you want to do.
- Favor: What I plan to do will make it easier for you to do what you want to do.
Chapter 8 An Introduction to Multiagent Systems

Recall how we described agents before:

\[ A : R \to A_c \]

A function which, given a run ending in a state, gives us an action.

\[ a \in A \times A_c \]

A constraint is then a pair:

A social law is a set of constraints.

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

\[ \forall E \subseteq F \quad \forall a \in A_c \]

We can refine our view of an environment.

\[ \text{A useful social law then a pair:} \]

\[ \text{A useful social law that prevents collisions:} \]

\[ \text{Emergence:} \]

\[ \text{T-shirt game (Shoham and Tennenholtz):} \]

\[ \text{Emergence:} \]

\[ \text{Not necessarily efficient (On \( n \) steps to get to a specific square).} \]
Strategy Update Functions

- **Simple majority**: Agents pick the shirt they have seen the most.
- **Simple majority with types**: Agents come into 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 pick the shirt with the highest cumulative reward.
- **Simple majority with types**: Agents pick the shirt with the highest number of matches.

Joint Intentions

- **The mental states of agents mirror those in BDI**: Also have motivation, want the coach moved.
- **Goals**: Simon wants the coach moved.
- **Motivation**: Bring about some goal, move the coach.
- **Collective commitment to a joint persistent goal**: A group of agents have a collective commitment to a joint persistent goal.

Termination Conditions

- Agents don’t believe that $\phi$ is satisfied, but believe it is possible.
- Agents maintain the goal $\phi$ until a termination condition is reached.
- Joint intentions condition is that it is mutually believed that:
  - the goal $\phi$ is no longer present; or
  - the goal $\phi$ is impossible; or
  - the goal $\phi$ is satisfied; or
  - the agent believes that the goal $\phi$ is satisfied.

- Agents and I have a mutual belief that $d$ is impossible.
- The termination condition is that it is mutually believed that:
  - goal $\phi$ is no longer present; or
  - goal $\phi$ is impossible; or
  - goal $\phi$ is satisfied; or
  - the agent believes that the goal $\phi$ is satisfied.

$\text{http://www.csc.liv.ac.uk/~mjw/pubs/imas/}$
Another approach to coordinate is to explicitly plan

Interaction analysis: do different plans affect one

A group of agents' actions are all interrelated.

Another approach is coordination through explicit plans.

Centralized planning for distributed plans

Could have:

• The agents in a system.
• For example, come up with a large STRIPS plan for all
• What all the agents do.
• Another approach is coordination through explicit plans

Distributed planning for distributed plans

One agent comes up with a plan for everybody.

Mutual belief is achieved by communication.

• Mutual belief is achieved by communication.
• All agents are all appraised of the situation.
• They don't stop working towards the goal until they
• This ensures that the agents are coordinated.
• Instead of adopting a new goal — to make this new
• But it doesn't drop the goal right away.
• Realises that the goal is satisfied, impossible, and so
• The termination condition is achieved when an agent

• The termination condition is achieved when an agent

Termination conditions:

• The goal is satisfied, impossible, and so
• Other conditions can be applied.

Centered planning for distributed plans

• These operators and then do:
Chapter 8: An Introduction to Multiagent Systems

8e

Safety analysis: which interactions are problematic?

Interaction resolution: treat the problematic interactions as critical sections and enforce mutual exclusion. Which interactions are problematic?

These ideas:
- A typical system will need to use a combination of coordinated.
- We discussed a number of ways of having agents do things together.
- This lecture has discussed how to get agents working.

Summary