Distributed Artificial Intelligence and Intelligent Agents

Agent Coordination

Mihhail Matskin

Agent Coordination

- general models
  - why?
  - how?
  - effective control of distributed search
- common coordination techniques
  - organizational structures
  - meta-level information exchange
  - multi-agent planning
  - explicit analysis and synchronization
  - coordination based on human teamwork
  - commitments and conventions

References - Curriculum

- Wooldridge: ”Introduction to MAS”,
  - Chapter 8

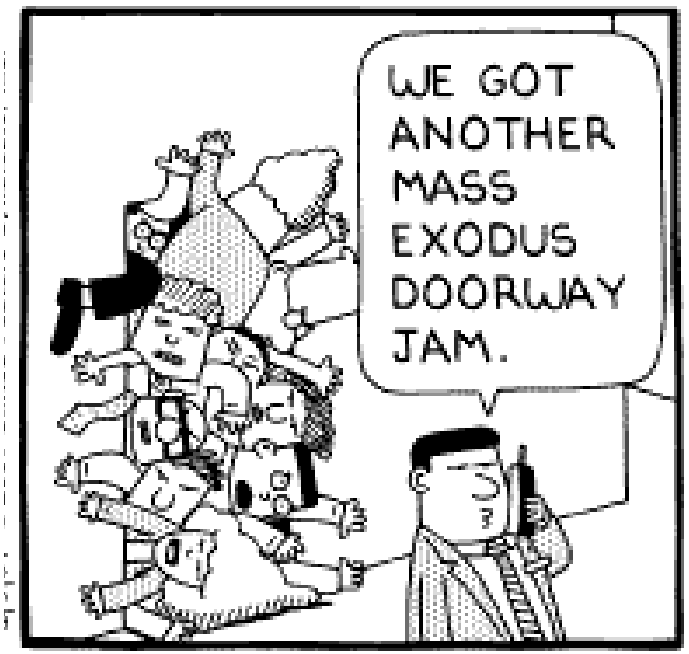
Normal action in multiagent system environments

Coordination Example

Consider an interaction between two robots, *A* and *B*, operating in a warehouse. The robots have been designed by different companies, and they are stacking and unstacking boxes to remove certain goods that have been stored in the building. They need to coordinate their actions to share the work load and to avoid knocking into each other and dropping the boxes.

Another Coordination Problem

- Managing the interdependencies between the activities of agents. e.g.
  - You and I both want to leave the room. We independently walk towards the door, which can only fit one of us. I graciously permit you to leave first.



In this case, the door is a SHARED RESOURCE

Cartoon taken from Klein, AAMAS2002.

Coordination

”The process by which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure that the community acts in a coherent manner.”

Jennings,1996

Coordination

- Preventing anarchy or chaos
- Meeting global constraints
- Distributed expertise, resources or information
- Dependencies between agents' actions
- Efficiency

We need some rules to prevent chaos: without rules each agent would do what they want

Sources will be consumed and not overspent

Coordination

To ensure:

- coverage
- connectivity
- rationality
- capability

Activities:

- supplying timely information to needy agents
- ensuring synchronization
- avoiding redundant problem solving

How to reach coherent behavior?

- complete knowledge
- centralized control
- distributed control

We can adjust our behaviour if you know everything about me and I know everything about you

There is an agent that controls the situation

Fundamental coordination processes

- mutual adjustment
- direct supervision
- standardization

Agents adjust their behaviour according to others

When you have to adjust to legal actions

Basic steps in coordination problem

- analysis of a perspective for coordination
  - external observation may be not enough
    - coordinated but incoherent
    - coherent but non-coordinated
  - examining an internal goals and motivations is necessary
- applying appropriate coordination model

We try to catch the information looking at the external facts, but it may not be enough

For coordination we need information about others

What is an abstract model for coordination?

Control decisions and search

View to coordination as a matter of effective control of distributed search.

What is a control?

- control decisions are decisions about what actions to take next
- control decisions are choices
- control knowledge is any knowledge that informs control decisions
- each control choice is the outcome of an overall control regime which includes knowledge about:
  - what are the control alternatives?
  - what are the decision criteria?
  - what is the decision procedure?

Control in cooperative distributed problem solving

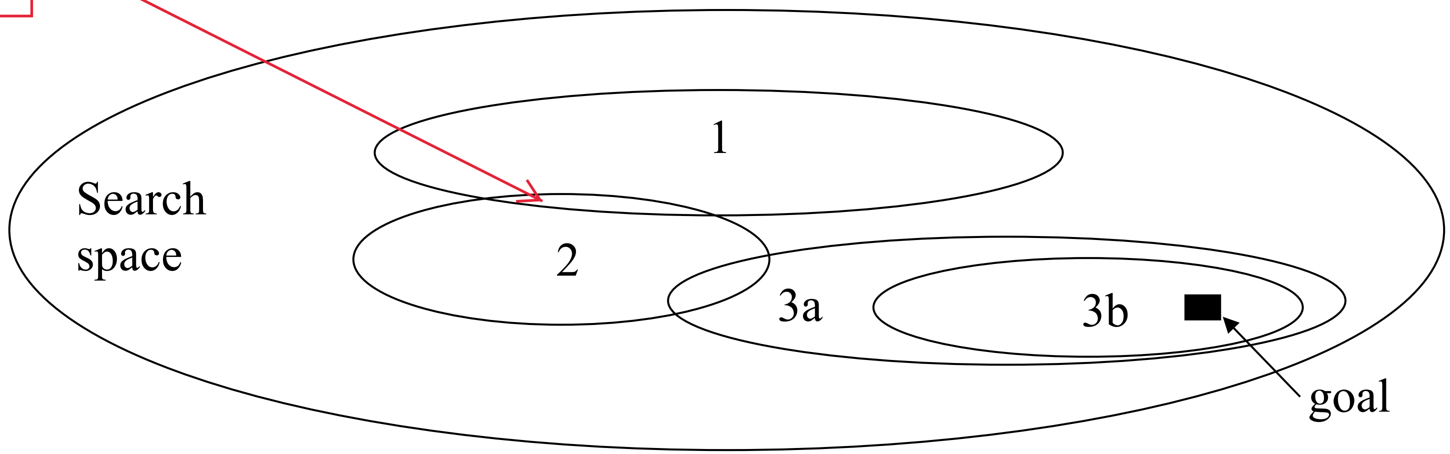
Control choice becomes more complicated to the degree that control decisions are more

- numerous
- asynchronous
- decentralized

DAI and distributed search

- the space of alternative problem states (goals) can be seen as a large search space investigated by a number of agents
- each agent has to make local control decisions
- these local control decisions have impacts on the overall efforts done by the collection of problem solvers

how to control in here?



Each agent has its own search space. It make plans and so on in there. If the spaces do not have intersections, then we have no coordination

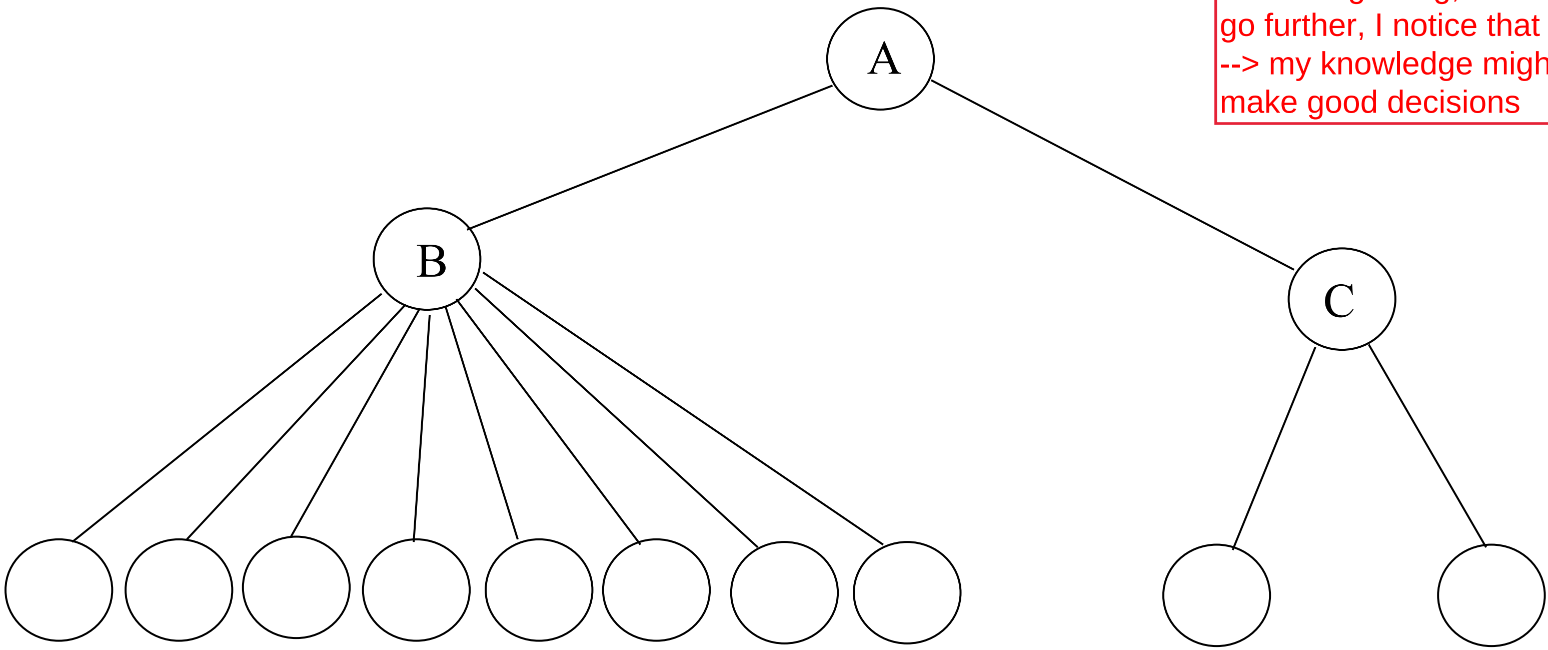
In a particular space, some elements can be taken also by another agent

--> that's how the control changes in the distributed setting

Control uncertainty

Control decision uncertainty - ambiguity in the next action choice (often is characterized as the size of the set of next-states)

I am in A.
What to choose, B or C?
At the beginning, it's the same for me. But, if I go further, I notice that the tree of B is bigger
--> my knowledge might be not enogh to make good decisions



Two kinds of control in DAI

- network control or cooperative control - comprises decision procedures that lead to good overall performance

- local control - refers to decision procedures that lead to good local decisions, and that are based on local information only

Network control

You have information about the whole network

Sets contexts for agent's individual control decisions based on network level information

Network-level information is:

- aggregated from more than one agent or abstracted from data from more than one agent
- information that concerns the relationships among a collection of agents

Network-level information can be utilized to influence:

- set of action alternatives to consider a control decisions
- the decision criteria applied to choose one of them
- the control decision procedure

Network control

One type of network control (or coordination) is allocation of search-space regions to agents

- search space - alternative problem states (goals)
- search process explores that space by generating a tree of possibilities
- since a search process explores a tree of possibilities (and any tree is recursively composed of subtrees) it follows that any region of the search space can be characterized by a set of subtree roots
- allocation of search-space regions to agents takes place by allocating collections of search-subtree roots to agents
- dynamic nature of region allocation

Local control

Concerns the status and process of a single node in its own local environment and its own local search-space region

Interaction with network control:

- network may increase or decrease local control uncertainty
- better local control may more efficiently uncover information that can focus network control decision.

Our task:

Reducing uncertainty

Reducing degree and/or reducing impact of uncertainty

- Impact - arbitrariness of control decisions
- Impact of control uncertainty can be reduced by reducing common dependencies that agents share

Two kinds of dependencies:

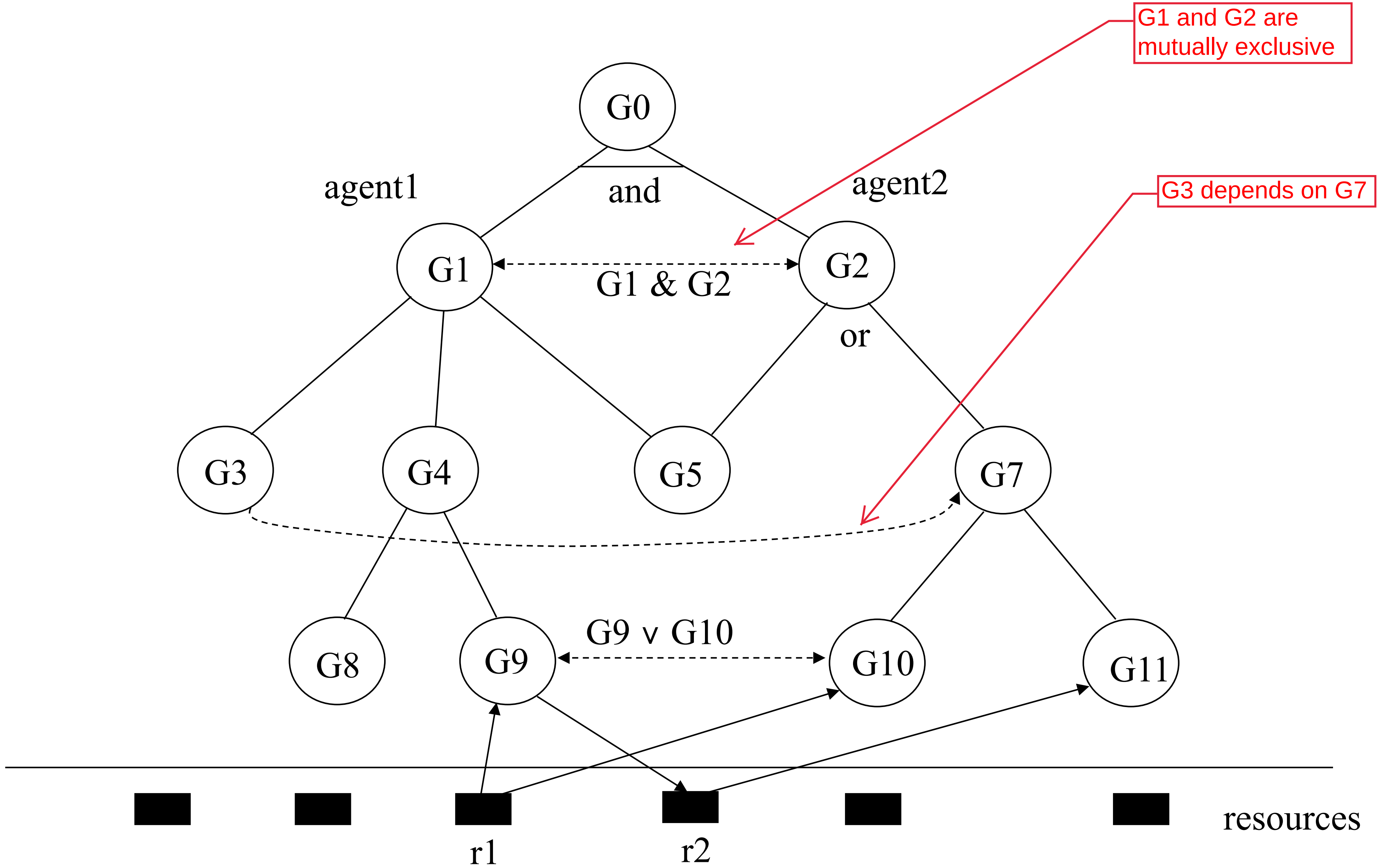
- logical dependencies
- resource dependencies

one action depends on other actions

A very important part of coordination is analysis of interdependencies

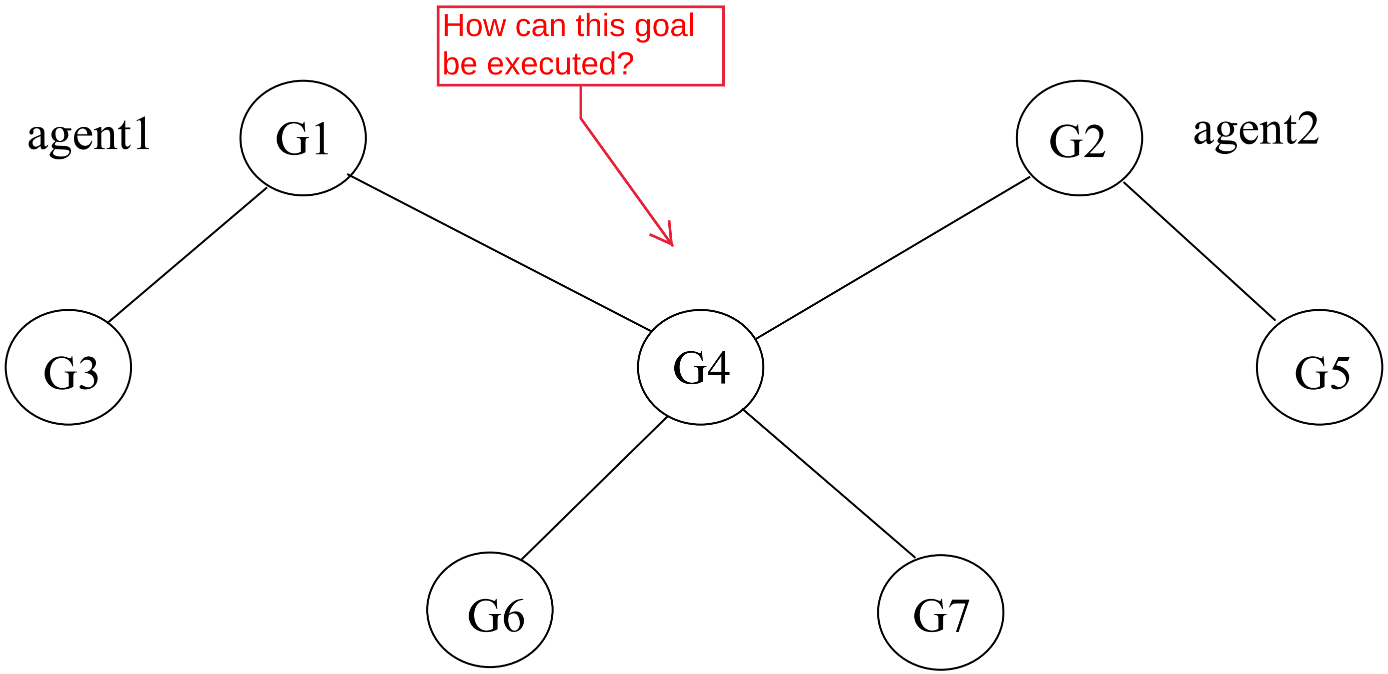
Each agent has a set of possibilities. It has information about what to do next. But in coordination there might be some dependences, some constraints that must be taken into account.
For instance in the search tree there are dependences between goals that make agents interact.

Search tree (example)



Joint goals

- team members are mutually responsible to one another
- the team members have a joint commitment to the joint activity
- the team members are committed to be mutually supportive of one another



Activities in coordination

- defining the goal graph
- assigning regions of search space
- controlling decisions about which areas of the graph to explore
- traversing the goal structure satisfying dependencies
- ensuring report of the successful traversal

Determining the approach for each of the phases is a matter of system design and it depends upon:

- the nature of the domain
- the type of agents included into community
- the desired solution characteristics

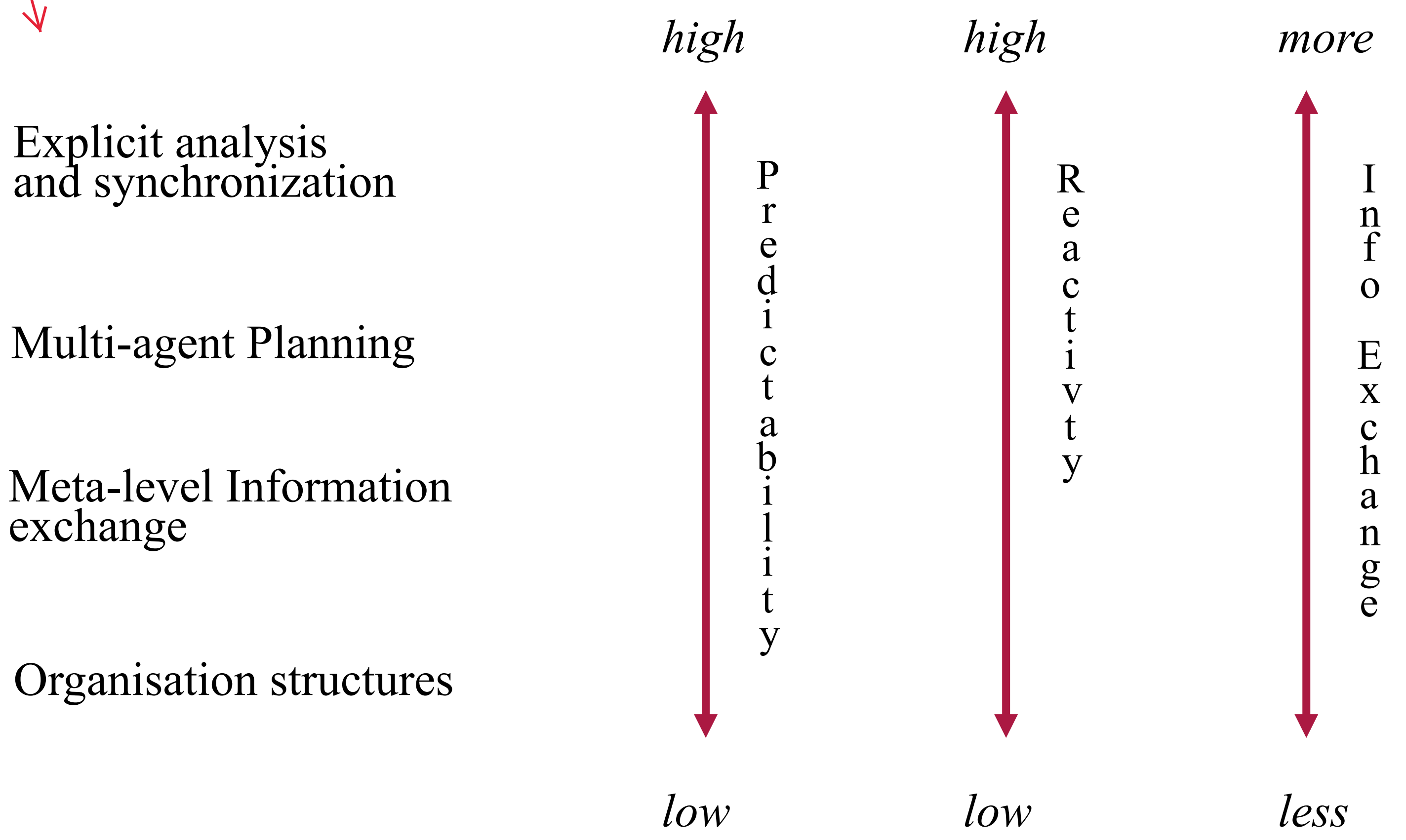
Coordination Techniques

- Organisational Structures
- Meta-level Information Exchange
  - e.g. Partial Global Planning (PGP), *(Durfee)*
- Multi-agent Planning
- Explicit Analysis and Syncronization
- Norms and social laws
- Coordination Models based on human teamwork:
  - Joint commitments *(Jennings)*
  - Mutual Modelling

4 different approaches. How well can we predict next action in these approaches? What about reactivity?

--> it depends on the information that we have

Comparing Common Coordination Techniques



Organizational structures

- the simplest coordination approach
- implicit coordination
- provides framework for defining
  - roles
  - communication paths
  - authority relationships
- pre-defined, long-term relations
- specifies the distribution of specializations among agents
- a precise way of dividing the problem space without specifying particular problems
- agents are associated with problem types and problem instances circulate to the agents which are responsible for instances of that type

Organizational Structures

- Organisational structures may be:
  - Functional
  - Spatial
  - Product-Oriented

Organizational Structure Models

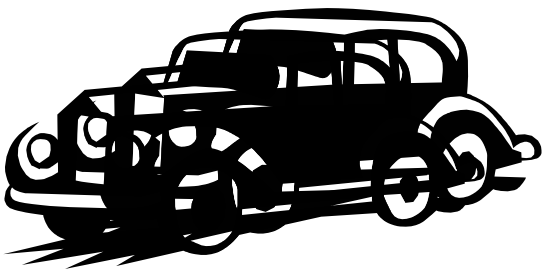
- A pattern for decision-making and communication among a set of agents who perform tasks in order to achieve goals.

  e.g.

  – Automobile industry

    We want to product different models of cars

    - Has a set of goals: To produce different lines of cars
    - Has a set of agents to perform the tasks: designers, engineers, salesmen
    - Costs considered
      – production costs
      – coordination costs
      – vulnerability costs



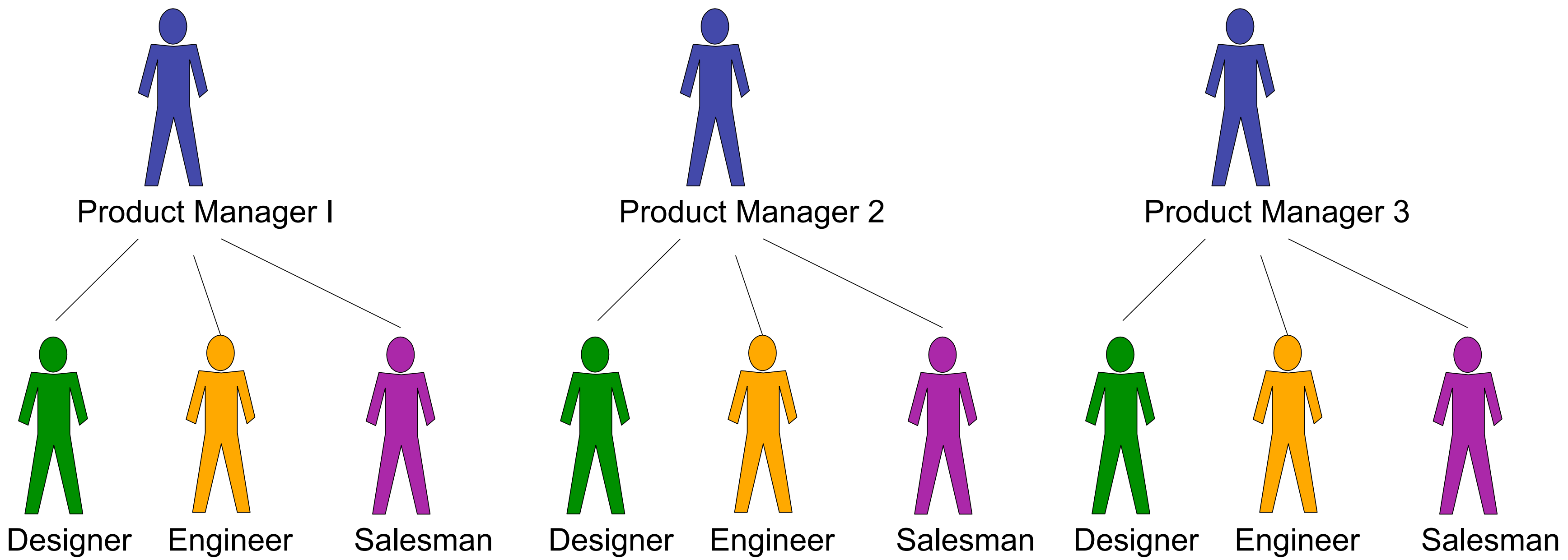
Reference: Malone 1987

For instance, the cost of changing the sequence (painting,...)

Alternative Coordination Structures 1

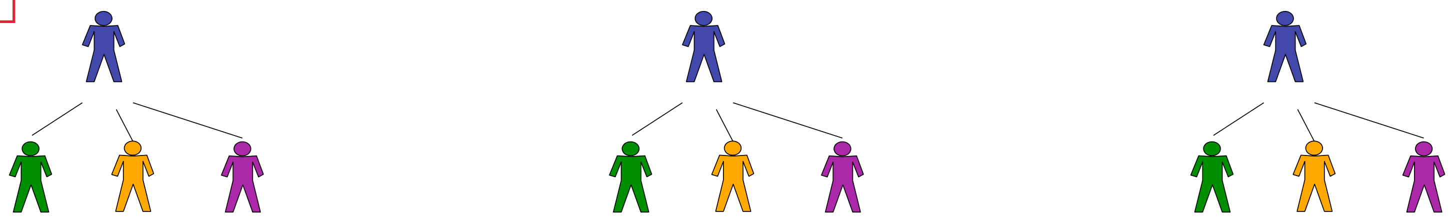
Product Hierarchy

One tree for each car model:



In this case coordination is associating each agent to a role and build the hierarchy --> static! Once defined, never changes

Product hierarchy

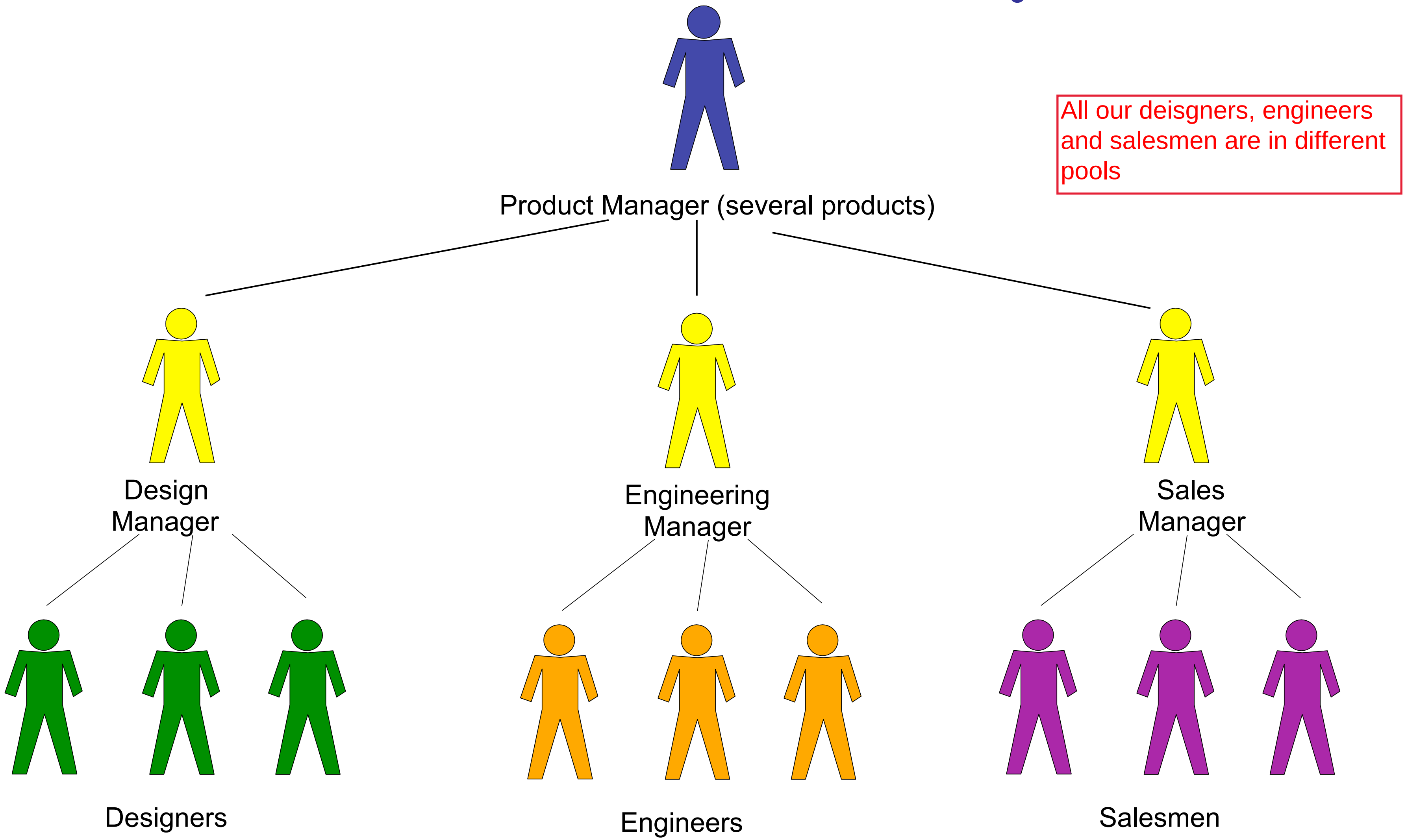


- several divisions for different product lines
- each division has a product manager
- each division has its own separate departments (agents) for different functions
- product manager assigns tasks to agents in its department
- communication links in product structure
- fails inside department do not affect other products
- one message to assign task and one message to notify result
- also model for set of separate companies

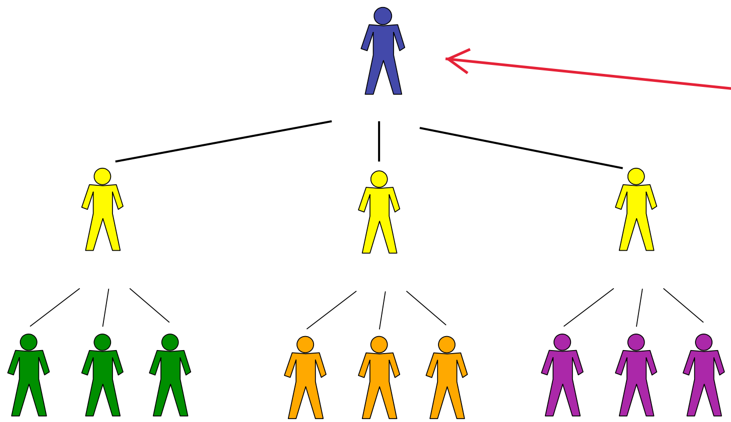
OSS: a bad sales manager, the whole product will fail

Alternative Coordination Structures 2

Functional Hierarchy



Functional hierarchy



this person is a
bottle neck: if it
doesn't work
very well,
verything will fail

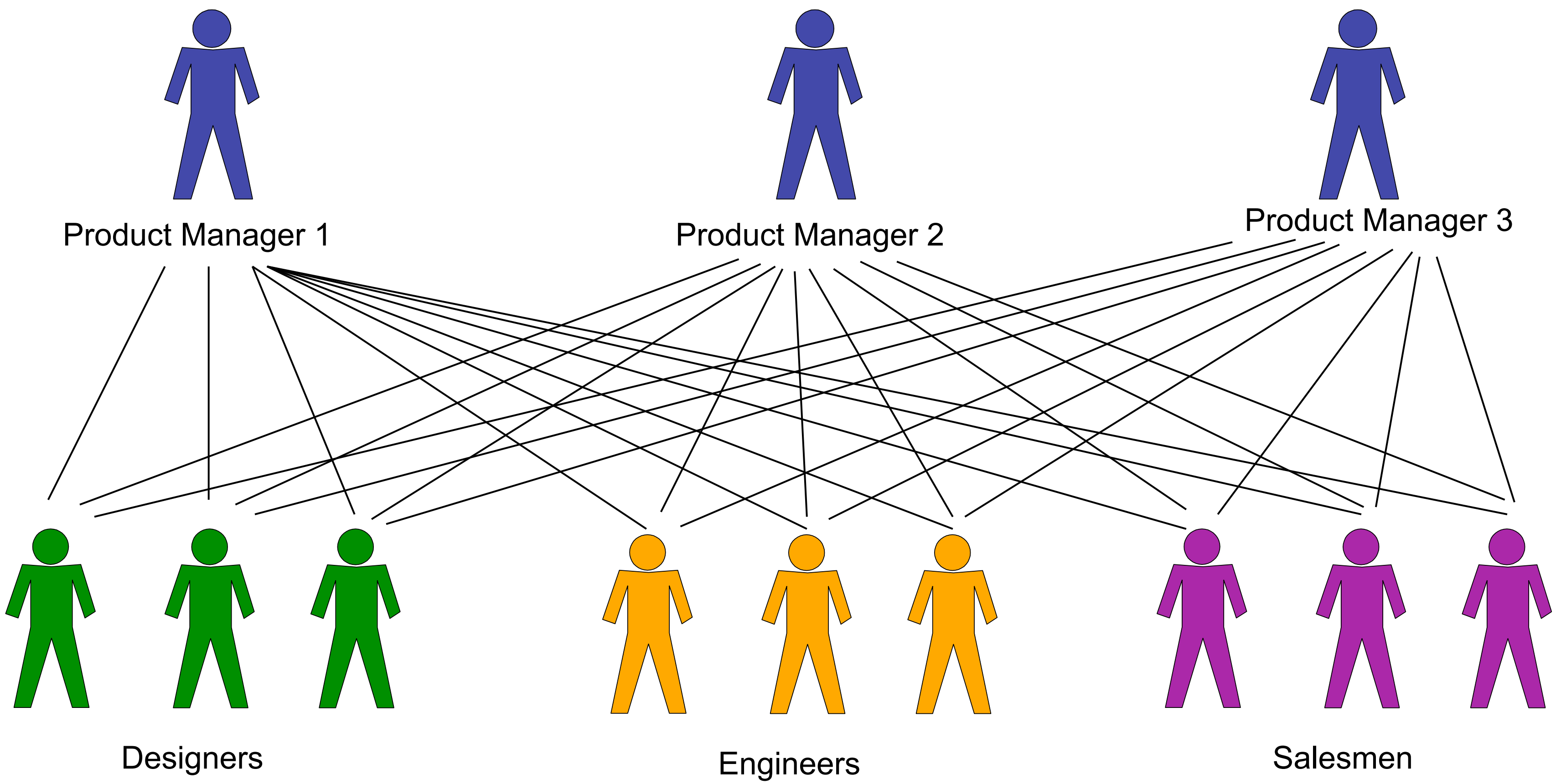
- a number of agents of similar types are pooled into functional departments
- each department has a functional manager
- reduce duplication of efforts
- executive office - production manager for all products hierarchical task allocation
- 2 messages to assign task and 2 messages to notify about results
- failure of task agent - delay and task reallocation
- managers are critical

Markets

- not necessary to make all components by itself
- subcontracting

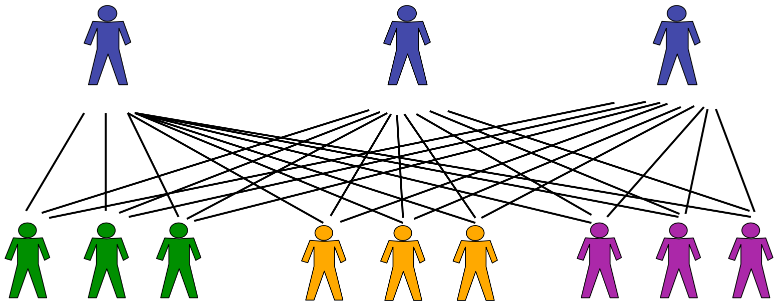
Alternative Coordination Structures 4

Decentralised Market



Availability of all salesmen, engineers and designers for all product manager
--> extremely flexible but we need a lot of coordination

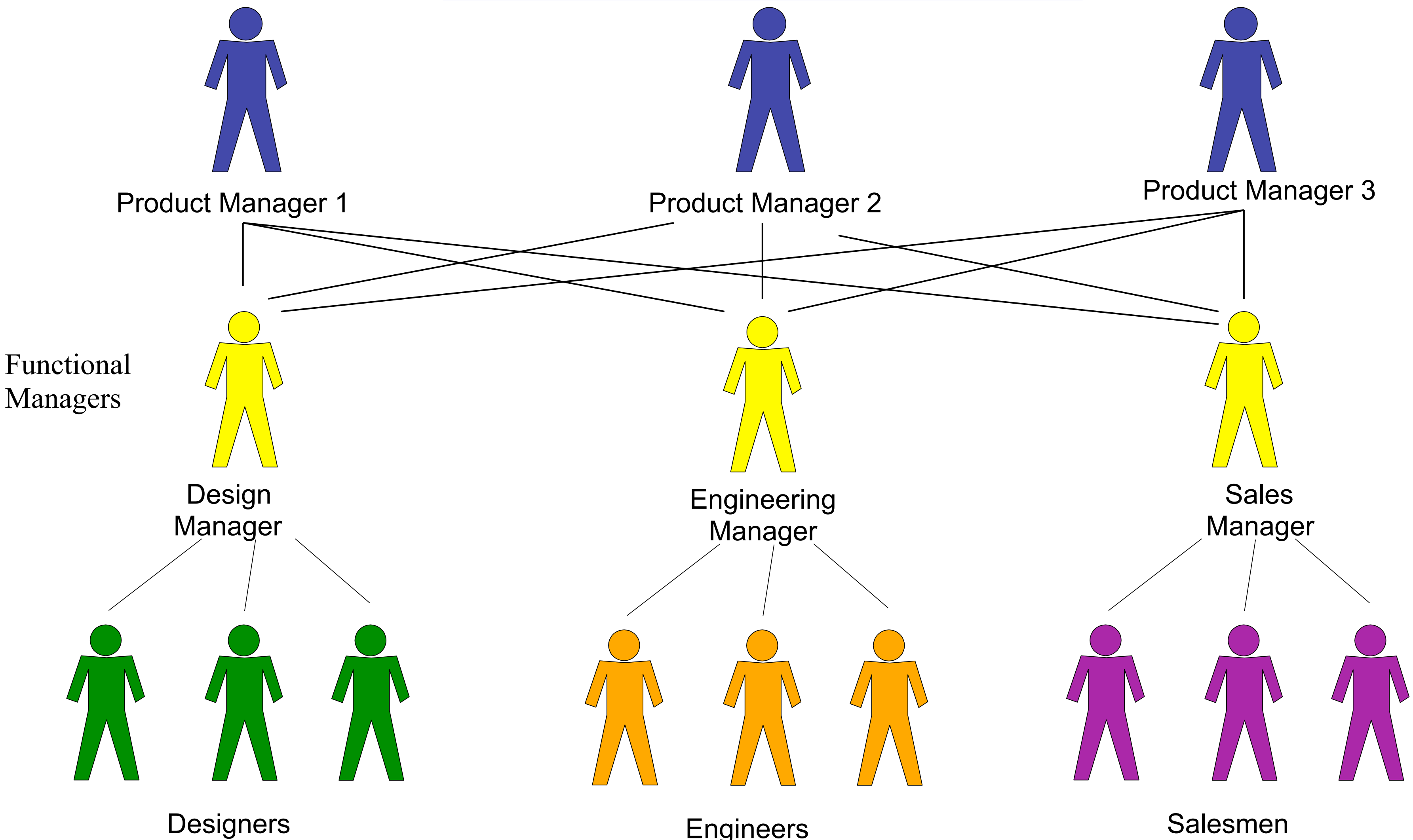
Decentralized market



- all buyers are in contact with all possible suppliers
- buyers play a role of product managers
- each has a communication link with each supplier
- buyers can choose the best supplier
- m suppliers - 2m+2 messages
- processor fails the task is reassigned
- if manager fails – product is failed

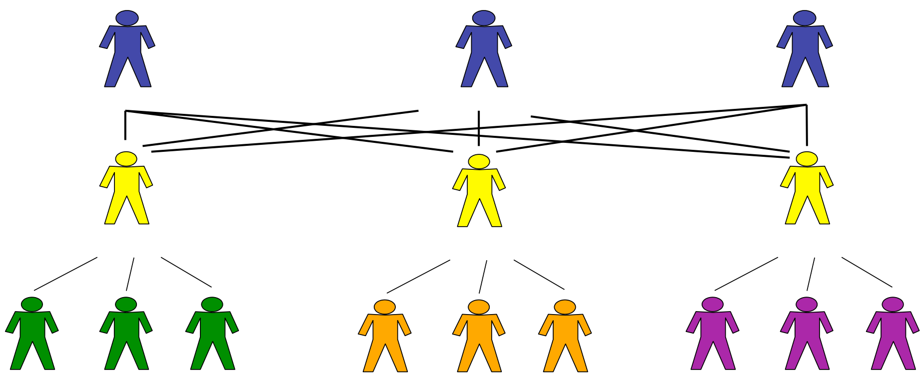
Alternative Coordination Structures 3

Centralised Market



A product manager doesn't communicate with each designer, each engineer and each salesman, but a broker does this operation
--> less communication

Centralized market



- brokers are in contact with possible sellers
- fewer connections and communications are required
- brokers play a role of functional managers
- broker chooses the best supplier
- 4 messages
- similar to functional model
- failure of one product manager does not affect others

Summary of organization structures

	Processors shared among products	Messages required to assign task	Result of task processor failure	Result of functional manager failure	Result of product manager failure
Product hierarchy	no	2	1 product disrupted	-	1 prod. disrupted
Decentralised market	yes	2m+2	Task reassigned	-	1 prod. disrupted
Centralised market	yes	4	Task reassigned	All prod. disrupted	1 prod. disrupted
Funtional hierarchy	yes	4	Task reassigned	All prod. disrupted	All prod. disrupted

Comparison of Organization Structures

	Production cost	Coordination cost	Vulnerability cost
Product hierarchy	H	L	H
Functional hierarchy	L	M-	H+
Centralised market	L	M+	H-
Decentralised market	L	H	L

Changes in costs as size of structure increases

	Production cost	Coordination cost	Vulnerability cost
Product hierarchy	+	+	+++
Funtional hierarchy	+	++	+
Centralised market	+	+++	++
Decentralised market	+	++++	++

Organizational Structures - Critique

- Useful when there are master/slave relationships in the MAS.
- Control over the slaves actions – mitigates against benefits of DAI such as reliability, concurrency.
- Presumes that atleast one agent has global overview – an unrealistic assumption in MAS.

Typically the structure of the organizaion doesn't change

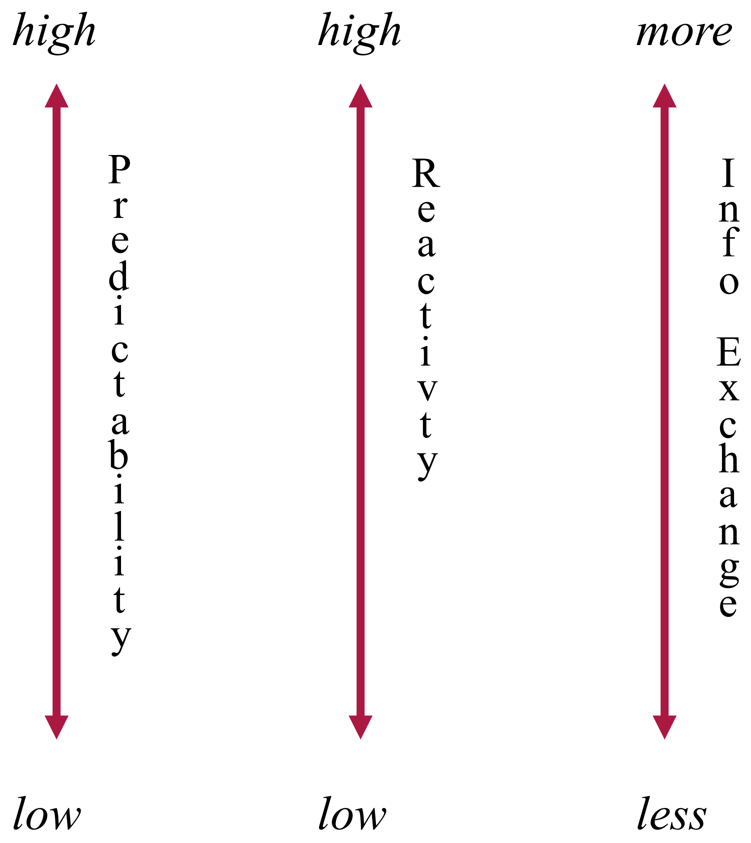
Comparing Common Coordination Techniques

Explicit analysis and synchronization

Multi-agent Planning

Meta-level Information exchange

Organisation structures

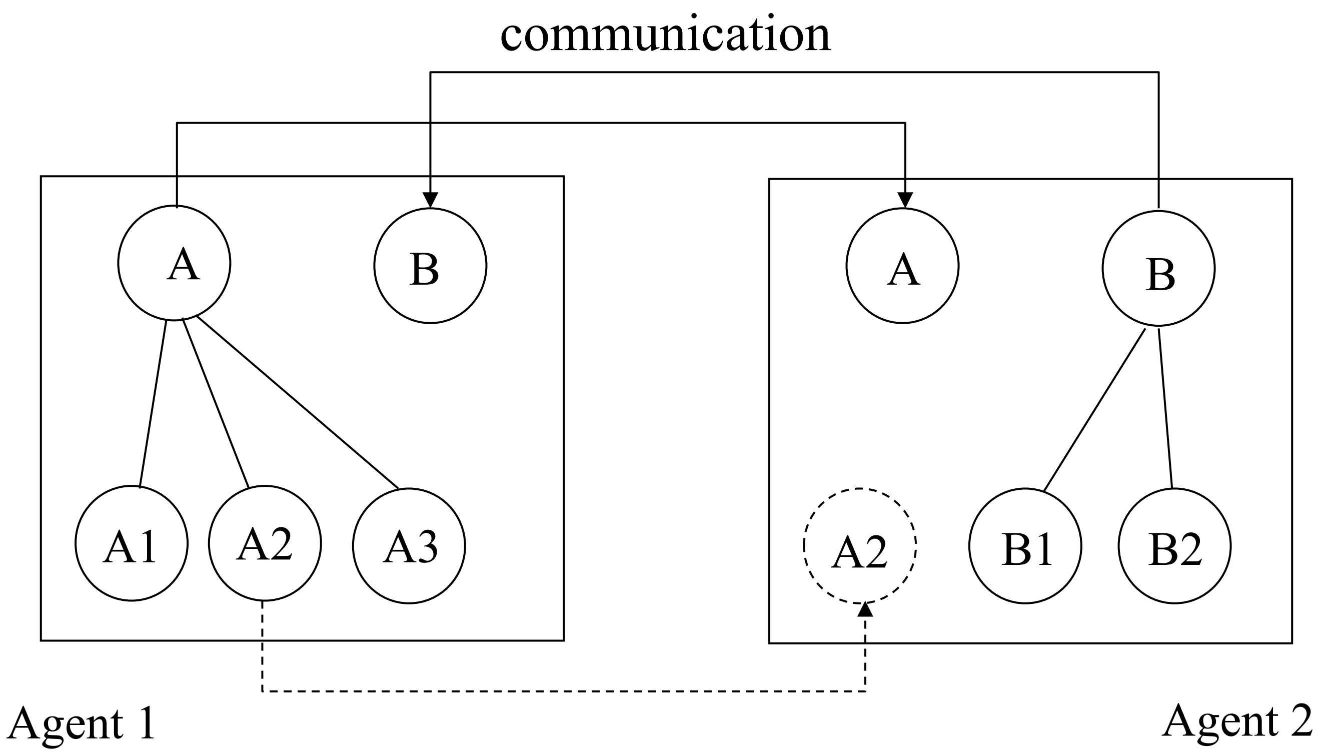


Meta-level Information Exchange

- Exchange control level information about current priorities and focus.
- Control level information
  - May change
  - Influence the decisions of agents
- Does not specify which goals an agent will or will not consider. More flexible
- Imprecise
- Medium term – can only commit to goals for a limited amount of time.

Partial Global Planning (PGP)
Durfee

- The basic condition - requirement that several distributed agents work on solution of the same problem
- <u>An agent can observe actions and relations between groups of other agents</u>



Agent1 starts from A, Agent2 starts from goal B. They also exchange information about what they do. If, for instance, goal A2 is important for B, then the agents may reallocate the goal A2

The main principle of partial global planning is that cooperating agents exchange information in order to reach common conclusions about the problemsolving process. Planning is partial because the system does not (indeed cannot) generate a plan for the entire problem. It is global because agents form non-local plans by exchanging local plans and cooperating to achieve a non-local view of problem solving.

Partial Global Planning (PGP)

- PGP involves 3 iterated stages:

1. Each agent decides what its own goals are and generates short-term plans in order to achieve them.
2. Agents exchange information to determine where plans and goals interact.
3. Agents alter local plans in order to better coordinate their own activities.

Partial Global Planning (PGP)

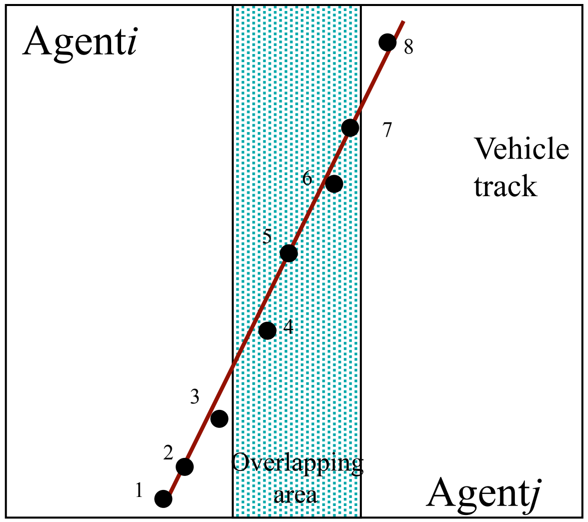
- Task decomposition
- Local plan formulation
- Local plan abstraction
- Communication of Local plan abstraction
- Partial global goal identification
- PGP construction and modification
- Communication planning
- Acting on PGP
- Ongoing modifications
- Task reallocation

They identify if they are working on the same goals

Suppose to have a big geographical area. This area is divided into squares and each square is assigned to an agent. Vehicle pass through the whole area, so that agents can record some information about the vehicle when it passes though their area. Each agent works locally and independently of the other agents.
Coordination: coordination may happen when the squares are overlapping.

Partial Global Planning (PGP) 1

- A DAI testbed – Distributed Vehicle Monitoring Testbed (DVMT) – to successfully track a number of vehicles that pass within the range of a set of distributed sensors (agents).



- Geographically distributed agents
- Each agent monitors sensing a portion of the overal area
- Discrete time-interval
- By analyses of sound – type of vehicle and movement map
- Provides information to another agent
- There should be overlapping areas

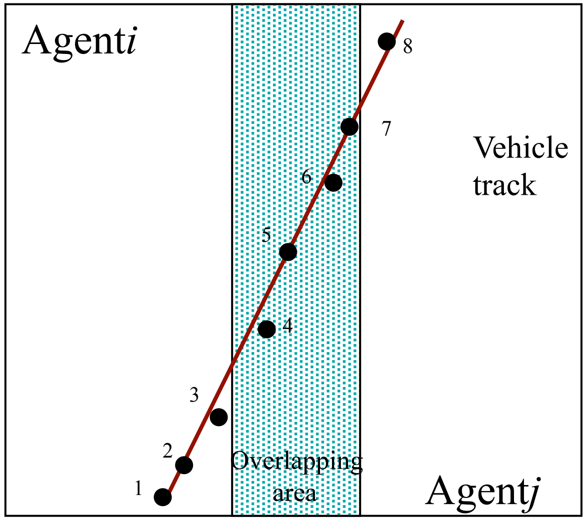
Partial Global Planning (PGP)

- Main principle: cooperating agents exchange information in order to reach common conclusions about the problem solving process.
- Why is planning partial?
  - A agent does not generate a plan for the entire problem.
- Why is planning global?
  - Agents form non-local plans by exchanging local plans and cooperating to achieve a non-local view of problem solving.

First step:

Task decomposition (PGP)

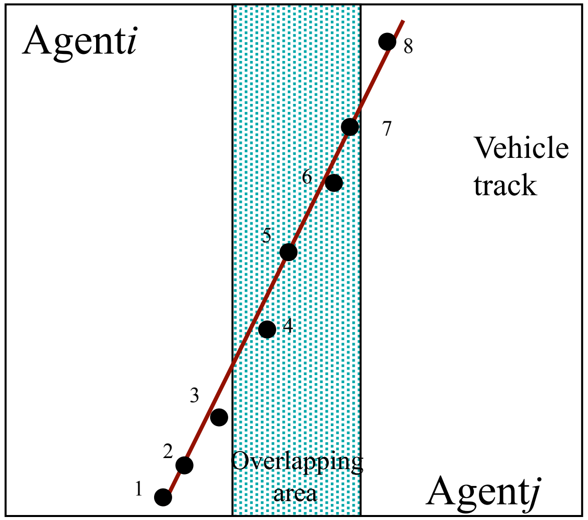
- Starts with the premise that tasks are inherently decomposed.
- Assumes that an agent with a task to plan for might be unaware of what tasks other agents might be planning for and how those tasks are related to its own.



- No individual agent might be aware of the global tasks or states.
- Purpose of coordination is to develop sufficient awareness.

Local plan formulation (PGP)

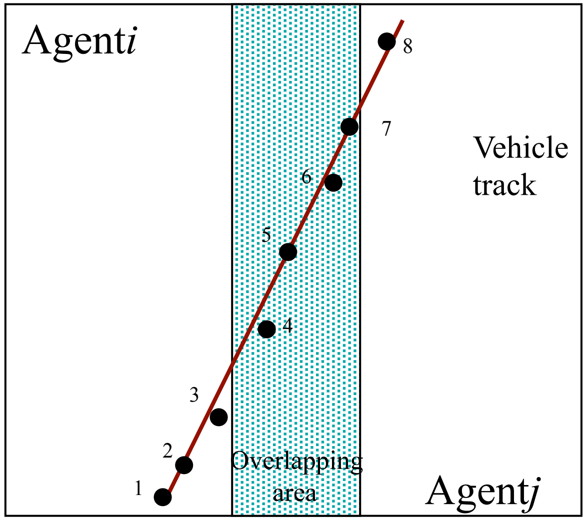
- before agent communicates with other it must first develop understanding of what goals it is trying to achieve and which actions to perform
- local plans will most often be uncertain, involving branches of alternative actions depending on results of previous actions and changes in environments



Local plan abstraction (PGP)

- alternative courses of action for achieving the same goal are important for an agent

- however, the detail of alternatives might be unnecessary considering the agent's ability to coordinate with others



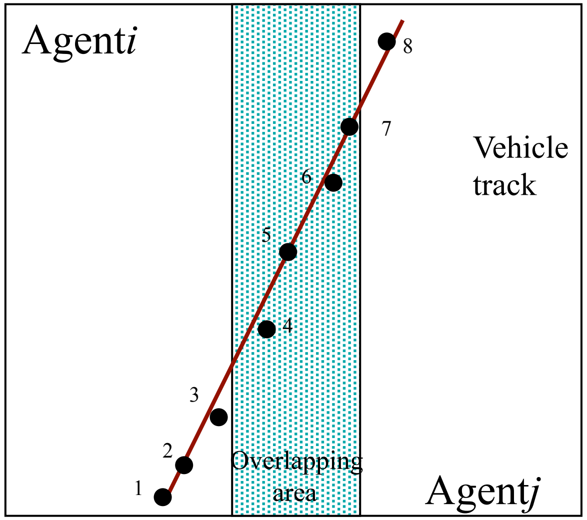
- an agent might have to commit to activities at one level of details without committing to activities at more detailed levels

- agents are designed to identify their major plan steps that could be of interest to other agents

Communication (PGP)

- agents must communicate about abstract local plans in order to build models of joint activity
- in PGP, the knowledge to guide this communication is contained in the meta-level-organization

Partial global goal identification (PGP)



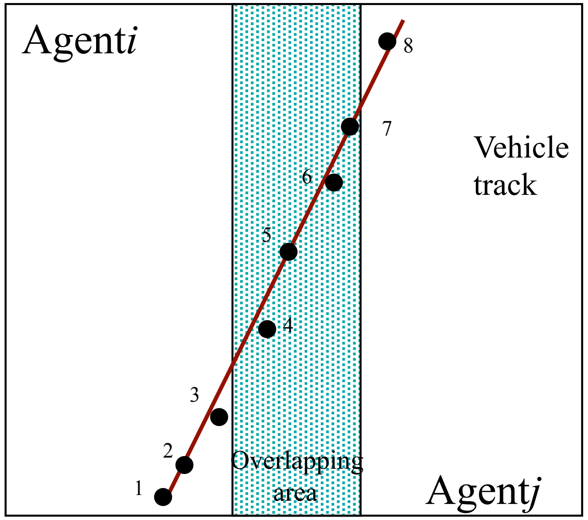
- the exchange of local plans gives an opportunity to identify when the goal of one or more agents could be subgoals of a single global goal

- only portions of the global goal might be known to agents

- partial global goal

PGP construction and modification

- local plans can be integrated into PGP
- PGP can identify opportunities for improved coordination (performing related tasks earlier, avoiding redundant task achievement)
- reordering of actions (usage of action rates)

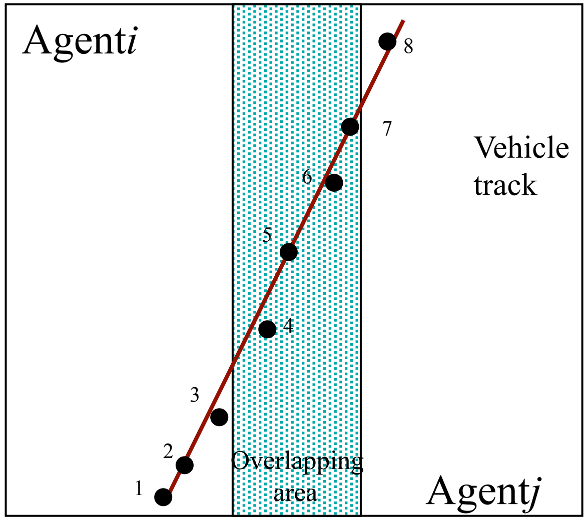


4,5,6,7 should not be duplicated, but splitted
{1,2,3,4,5}
{6,7,8}

PGP construction and modification

Rates:

- whether the task is unlikely to have been accomplished already by another agent
- how long it is expected to take
- how useful its results will be to others in performing their tasks

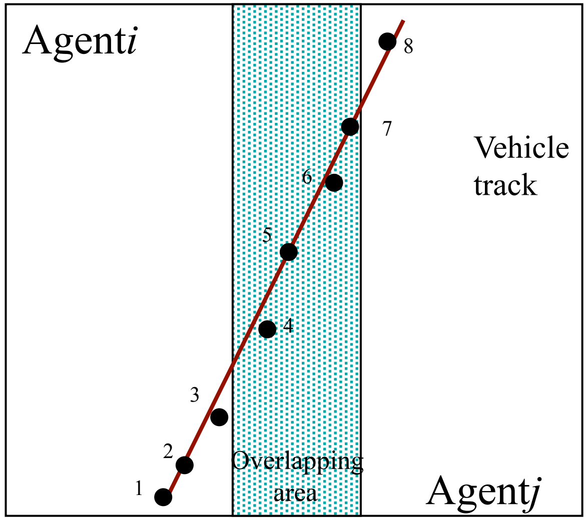


Algorithm

1. For the current ordering, rate the individual actions and sum the ratings
2. For each action, examine the later actions for the same agent and find the most highly-rated one. If it is higher rated, then swap the actions
3. If the new ordering is more highly rated than the current one, then replace the current ordering with the new one and go to step 2.
4. Return the current ordering

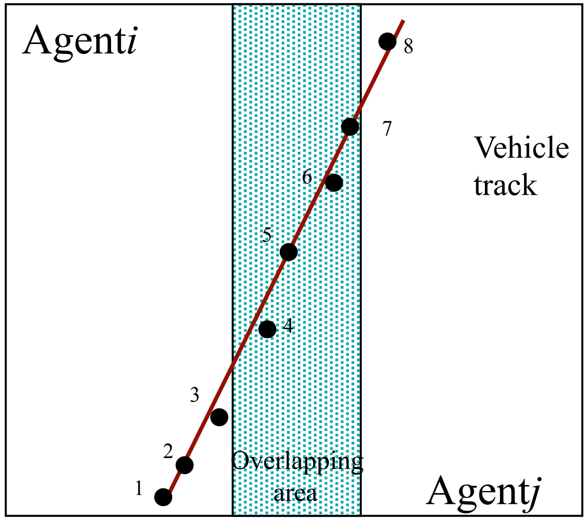
Communication planning (PGP)

- agent must next consider what interactions should take place between agents
- interactions in form of communicating the results of tasks are planned
- by examining PGP an agent can determine when a task will be completed by one agent that could be of interest to another agent and can explicitly plan the communication action to transmit the result



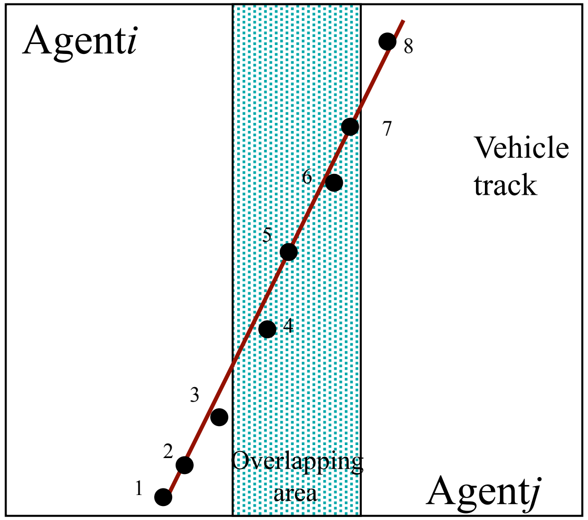
Communication planning (PGP)

1. Initialize the set of partial task results to integrate
2. While the set contains more than one element:
   (a) For each pair of elements: find the earliest time and agent at which they can be combined
   (b) For the pair that can be combined earliest: add a new element to the set of partial results for the combination and remove the two elements that were combined
3. Return the single element in the set



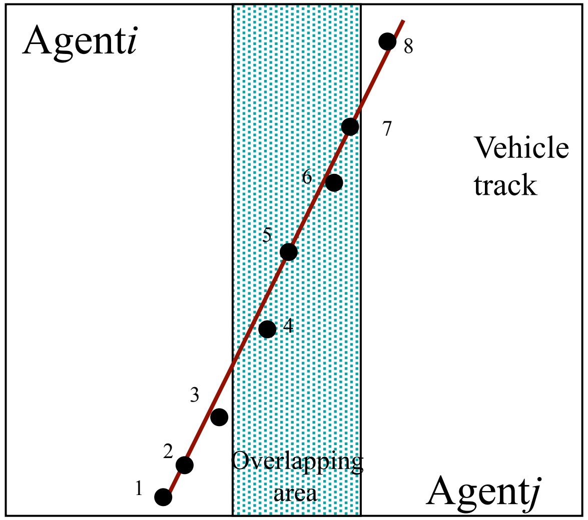
Acting on PGP

- once a PGP has been constructed and the concurrent local and communication actions have been ordered, the collective activities of the agents have been planned
- these activities must be translated back to the local level
- an agent responds to a change in its PGP by modifying the abstract representation of its local plans
- modified representation is used by agent when choosing its next local action



If agent1 is very quick in performing his tasks, then he can suggest to modify the division --> changes in coordination

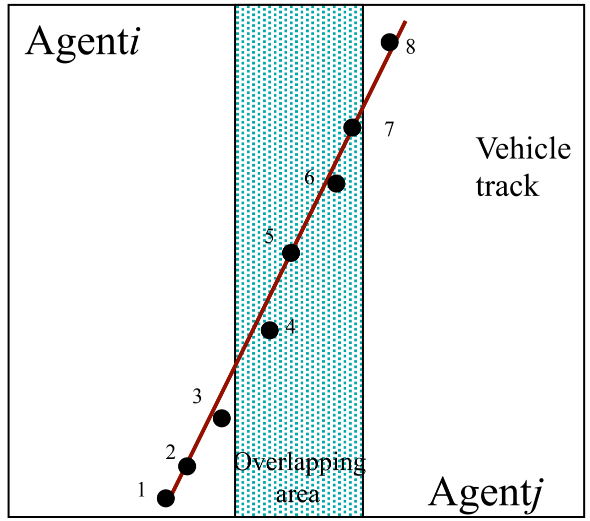
Ongoing modifications (PGP)



- as agents pursue their plans, their actions or events in environment might lead to changes in tasks
- a change in coordination is deciding when the changes in local plans are significant enough
- defining a threshold (significant temporal deviations)

Task reallocation (PGP)

- disproportional task load
- through PGP exchange of abstract models of agent's activities and detection whether they are overburdened
- possible task reallocation



Partial Global Planning (PGP)

- Partial Global Plan: a cooperatively generated datastructure containing the actions and interactions of a group of agents.
- Contains:
  - Objective – the larger goal of the system.
  - Activity map – what agents are actually doing and the results generated by the activities.
  - Solution construction graph – a representation of how the agents ought to interact in order to successfully generate a solution.

Summary (PGP)

- PGP focuses on dynamically revising plans in cost-effective ways given an uncertain world
- PGP particularly suited to applications where some uncoordinated activity can be tolerated and overcome
- PGP works well for many tasks, but could be inappropriate for domains such as air-traffic control where guarantees about coordination must be made prior to any execution

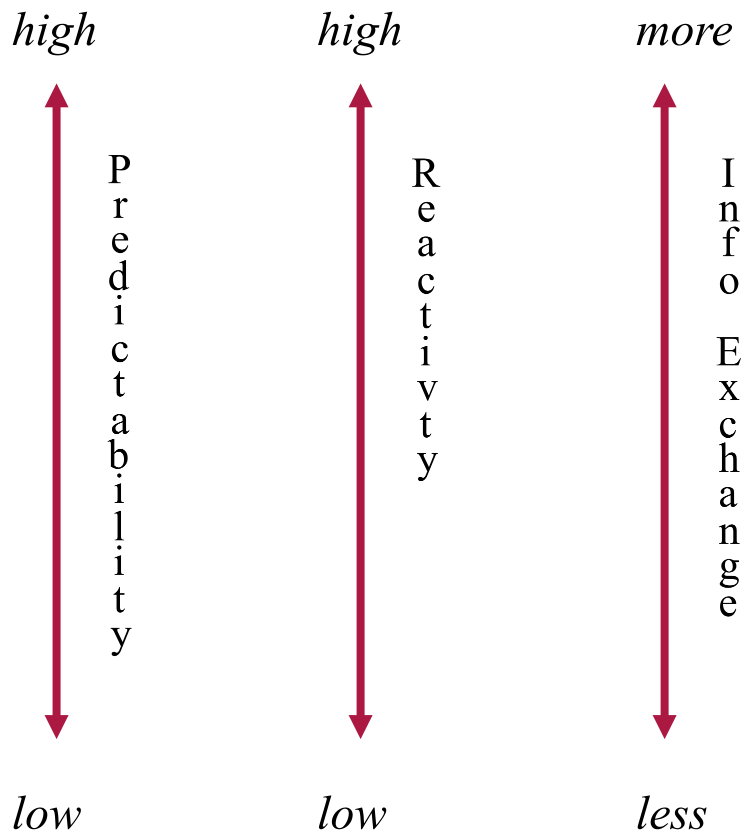
Comparing Common Coordination Techniques

Explicit analysis and synchronization

Multi-agent Planning

Meta-level Information exchange

Organisation structures



Multiagent planning must take into consideration the fact that the activities of agents can interfere with one another

Multi-agent Planning

- Agents generate, exchange and synchronize explicit plans of actions to coordinate their joint activity.
- They arrange apriori precisely which tasks each agent will take on.
- Plans specify a sequence of actions for each agent.
- Consideration of uncertainty is moved to the planning activity
- More specific than organizational structures or PGP and shorter time-horizon

Multi-agent Planning

A centralized planning system develops a plan for a group of agents, in which the division and ordering of labour is defined. This 'master' agent then distributes the plan to the 'slaves', who then execute their part of the plan.

- Two basic approaches:
  1. Centralised – a central coordinator develops, decomposes and allocates plans to individual agents.
  2. Distributed – a group of agents cooperate to form:
     1. Centralised plan
     2. Distributed plan

A group of agents cooperate to form a centralized plan. Typically, the component agents will be 'specialists' in different aspects of the overall plan, and will contribute to a part of it. However, the agents that form the plan will not be the ones to execute it; their role is merely to generate the plan.

A group of agents cooperate to form individual plans of action, dynamically coordinating their activities along the way. The agents may be self-interested, and so, when potential coordination problems arise, they may need to be resolved by negotiation

In general, centralized planning will be simpler than decentralized planning, because the 'master' can take an overall view, and can dictate coordination relationships as required. The most difficult case to consider is the third. In this case, there may never be a 'global' plan. Individual agents may only ever have pieces of the plan which they are interested in.

Multi-agent Planning

Centralized planning for distributed plans

- Given a goal description, a set of operators and an initial state description, generate a plan
- Decompose the plan into subproblems such that ordering relationships between steps tend to be minimized across subproblems
- Insert synchronization actions into subplans
- Allocate subplans to agents using task-passing mechanisms
- Initiate plan execution and optionally monitor progress

Multi-agent Planning

- Distributed Planning for centralised plans:
  - e.g. Air traffic control domain (Cammarata)
    - Aim: Enable each aircraft to maintain a flight plan that will maintain a safe distance with all aircrafts in its vicinity.
    - Each aircraft sends a central coordinator information about its intended actions. The coordinator builds a plan which specifies all of the agents' actions including the ones that they should take to avoid collision.
  - e.g. A general technological process (manufacturing)
    - Centralized controll and plan is centralized but each step can be planned in parallel
    - Disitrbuted expertize but control is centralized

Multi-agent Planning

- Distributed Planning for distributed plans:
  - Individual plans of agents, coordinated dynamically.
  - No individual with a complete view of all the agents' actions.
  - More difficult to detect and resolve undesirable interactions.

Centralized multi-agent plans (plan coordination) - an example (Georgeff)

- robots world
- actions - sequences of states
- each action has pre- and post-conditions
- plans are generated separately

This approach has several steps

- interaction analysis
- safety analysis
- interaction resolution

Centralized multi-agent plans

An action a is a sequence of states p :

$a = \{p_1, ..., p_i, ..., p_n\}$, where p_1 is action's initial state, p_n is action's final state and $p_2, \ldots, p_{n-1}$ are intermediate states

A planning problem consists of

- set of states S
- a designated set of initial states I
- a set of actions A
- a set of goal states G

A single-agent plan P is a description of sequence of actions $a_1, a_2, \ldots, a_n$ from A such that:

- a_1 is applicable to all initial states (a_1 initial state is in I)
- for all i, $1 < i \leq n$, the action a_i is applicable to all states in the range of a_{i-1}
- a_n achieves the goal G (final state of a_n is in G)

Interaction analysis involves generating a description of how single agent plans interact with one another. Some of these interactions will be harmless; others will not.

Interaction analysis

If two actions may be executed in parallel, then they are said to be commutative. It follows that if two actions are commutative, then either they do not interact, or any interactions are harmless

- establishing incompatible situations
- <p, q> - the situation where both p and q hold (it can be unsatisfiable if contradictory)
- possible alternatives for safe action execution
  - actions can be executed in parallel (actions said to be commuted)
  - suspending one of the action
- precedence may exist of one over another (post-condition of one is a pre-condition for another)
  - both actions may have precedence over each other
  - neither actions may have precedence
- plans are generated separately

Precedence describes the sequence in which actions may be executed; if action a1 has precedence over action a2, then the preconditions of a2 are met by the postconditions of a1.

Safety analysis involves two stages.
1. All actions which are harmless (i.e. where there is no interaction, or the actions commute) are removed from the plan. This is known as simplification.

2. the set of all harmful interactions is generated

Safety analysis

Determining unsafe situations for each pair of actions in two plans

- if a_i and b_j do not commute, then <begin(a_i),begin(b_j)> is unsafe
- if a_i doesn't have precedence over b_j, then <begin(a_i),end(b_{j-1})> is unsafe

The set of all such unsafe situations is called interaction set

Rules for safety governing

- if $s = \langle begin(a_i), begin(b_j) \rangle$, then s is unsafe if either of its successor situations are unsafe
- if $s = \langle begin(a_i), end(b_j) \rangle$, then s is unsafe if $\langle end(a_i), end(b_j) \rangle$ is unsafe
- if $s = \langle end(a_i), end(b_j) \rangle$, then s is unsafe if both successor situations are unsafe
- together with those situations occurring in the interaction set, these are all the unsafe situations

Interaction resolution

<u>The set of unsafe situations is next analyzed to identify contiguous sequences of unsafe situations</u>

Handling synchronization

Unsafe plan interactions are called critical sections.
To resolve the conflicts, mutual exclusion of the critical sections must be guaranteed

- critical regions
- communication commands
  - P!s - send s to process P
  - P?s - receive s from P

Centralized multi-agent plans (plan coordination)

- drop commuting actions
- exponential search of interleavings
- introducing synchronization

Example - 2 robots

The actions are informally as follows:

- a1m: agent 1 moves to the lathe
- a2m: agent 2 moves to the lathe
- a1p: agent 1 places metal stock in lathe
- a2p: agent 2 places metal stock in lathe
- a1b: agent 1 makes a bolt
- a2n: agent 2 makes a nut
- a1f: agent 1 moves to end
- a2f: agent 2 moves to end

Preconditions for a1b and a2n - the lathe must be in possession of the appropriate agent, as well as postconditions for a1p and a2p

Example (cont.)

Assume the simple planner produces the following single-agent plans:

a1m -> a1p -> a1b -> a1f

a2m -> a2p -> a2n -> a2f

The following properties can be established

- actions a1b and a2n conflict with one another
- actions a1p and a2p each have precedence over each other but do not commute
- action a1b has precedence over a2p, but not vice versa
- action a2n has precedence over a1p, but not vice versa

Example (cont.)

Interaction set is determined to be:

<begin(a1b),begin(a2n)>, <begin(a1b),end(a2p)>,

<end(a1p),begin(a2n)>, <begin(a1p),begin(a2p)>,

<begin(a1b),begin(a2p)>, <end(a1p),begin(a2p)>,

<begin(a1p),begin(a2n)>, <begin(a1p),end(a2p)>

We next form the simplified solutions:

begin(a1p) -> end(a1p) -> begin(a1b) -> end(a1b)

begin(a2p) -> end(a2p) -> begin(a2n) -> end(a2n)

Example (cont.)

Safety analysis - we get 2 critical regions

begin(a1p) -> end(a1b)

begin(a2p) -> end(a2n)

Next we insert synchronization commands into initial plans:

a1m -> S!begin(a1p) -> a1p -> a1b -> S!end(a1b) -> a1f

a2m -> S!begin(a2p) -> a2p -> a2n -> S!end(a2n) -> a2f

Synchronizer (very generally):

N, M := false

not N; P?begin(a1p) -> M := true

not M; Q?begin(a2p) -> N := true

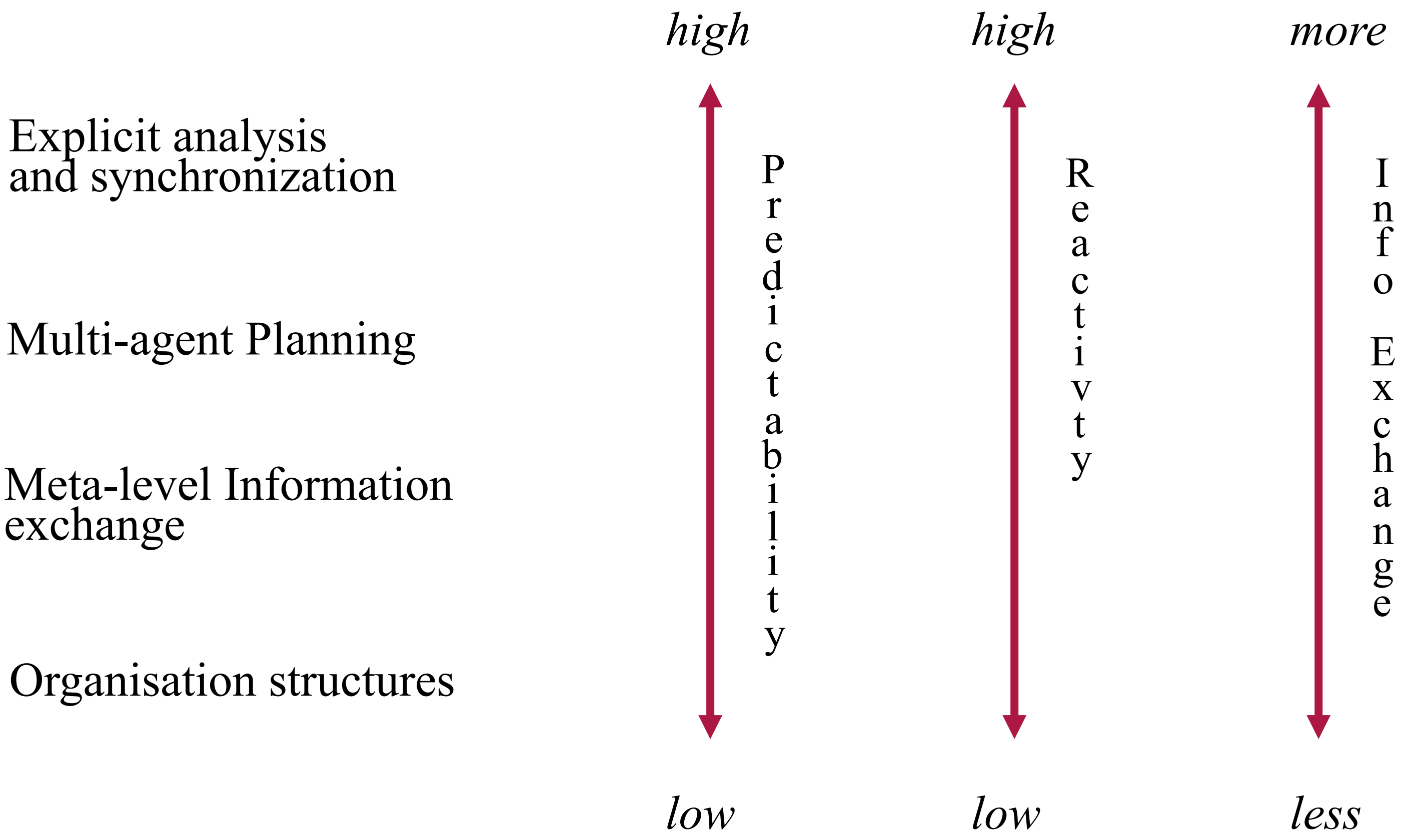
true; P?end(a1b) -> M := false

true; Q?end(a2n) -> N := false

Multi-agent Planning

- Critique:
  - Agents share and process a huge amount of information.
  - Requires more computing and communication resources.
- Difference between multi-agent planning and PGP:
  - PGP does not require agents to reach mutual agreements before they start acting.

Comparing Common Coordination Techniques



Explicit analysis and synchronization

- analysis of a situation in each decision-making step (possible-next-action-set)
- interacting with other agents
  - exchange among all interdependent agents
  - locks action
  - uses reply information to prune its next-action set
  - send synchronization unlocking messages
- a lot of time for communication during planning
- if the level of dependency is low and the granularity of actions is high - can provide useful coordination
- short time-horizon

Coordination Techniques

- Organisational Structures
- Meta-level Information Exchange
  - e.g. Partial Global Planning (PGP), *(Durfee)*
- Multi-agent Planning
- ➢ Norms and social laws
- Coordination Models based on human teamwork:
  - Mutual Modelling
  - Joint commitments

Social Norms and Laws

This norm is not enforced in any way: it is simply expected behaviour: diverging from this norm will (usually) cause nothing more than icy looks from others on the bus. This norm provides a template that can be used by all those around to regulate their own behaviour.

- **Norm**: an established, expected pattern of behaviour.
  - e.g. To queue when waiting for the bus
- **Social laws**: similar to Norms, but carry some authority.
  - e.g. Traffic rules.
- Social laws in an agent system can be defined as a set of constraints:
  - Constraint => $\langle E', \alpha \rangle$,
    - $E' \subseteq E$ is a set of environment states
    - $\alpha \in Ac$ *is an action,* (Ac is the finite set of actions possible for an agent)

➢ if the environment is in some state $e \in E'$, then the action α is forbidden.

An agent (or plan) is said to be legal wrt a social law if it never attempts to perform an action that is forbidden by some constraint in the social law

Coordination Techniques

- Organisational Structures
- Meta-level Information Exchange
  - e.g. Partial Global Planning (PGP), *(Durfee)*
- Multi-agent Planning
- Norms and social laws
- Coordination Models based on human teamwork:
  - Mutual Modelling
  - Joint commitments

Mutual Modelling

- Build a model of the other agents – their beliefs and intentions.
  - *Put ourselves in the place of the other*
- Coordinate own activities based on this model
- Cooperation without communication

You and I are both walking to the door, and there is not enough room for both of us - a collision is imminent. What should we do? One option is for both of us to simply stop walking. This possibility guarantees that no collision will occur, but it is in some sense sub-optimal: while we stand and wait, there is an unused resource (the door), which could fruitfully have been exploited by one of us. Another possibility is for both of us to put ourselves in the place of the other. In this case, you might believe that I am eager to please you, and therefore that I will likely allow you to pass through the door first; on this basis, you can continue to walk to the door.

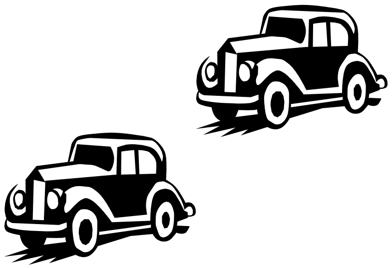
Coordination, Cooperation & Teamwork

- Can we have coherent behaviour without cooperation?

  ”A group of people are sitting in a park. As a result of a sudden downpour, all of them run to a tree in the middle of the park because it is the only source of shelter.”

The idea was that if you assume that both you and the other agents with which you interact share a common view of the scenario (in game-theory terms, you all know the payoff matrix), then you can do a gametheoretic analysis to determine what is the rational thing for each player to do.

Teamwork Example



Two vehicles travelling in a convoy:

> Consider two agents Bob and Alice. Bobs wants to drive home, but does not know his way. He knows that Alice is going near there and that she does know the way. Bob talks to Alice and they both agree that he follows her through traffic and that they drive ***together***.

Ref: Cohen & Levesque, 1991

Teamwork

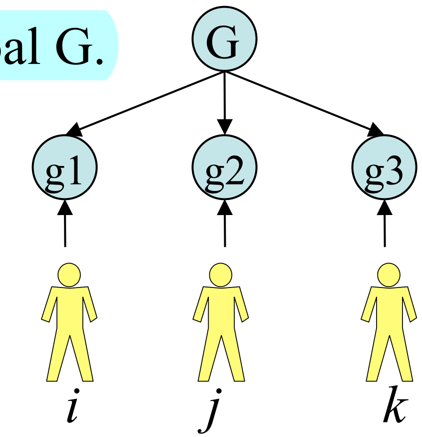
- Important distinction:
  - Coherent behaviour that is not cooperative, e.g
    - Individual drivers in traffic following traffic rules
  - Coordinated cooperative action, e.g
    - A convoy of drivers

Teamwork Definition

- American Heritage Dictionary
  - ***Cooperative effort*** *by the members of a team to achieve a* ***common goal***.

Teamwork

- How does an individual intention towards a particular goal differ from being a part of a team with a collective intention towards a goal?
  - **Responsibility** towards the other members of the team.
    - e.g. You and I are lifting a heavy object
  - Agents *i*, *j* and *k* are a **team** and have a common goal G.



Joint Intentions model

- Based on human teamwork models

  ”When a group of agents are engaged in a cooperative activity, they must have a joint commitment to the overall aim as well as their individual commitments.”

Commitments

- Commitment – a pledge or promise (e.g. to lift the heavy object).
  - Commitment persists – if an agent adopts a commitment, it is not dropped until for some reason it becomes redundant.
  - Commitments may change over time, e.g. due to a change in the environment
  - Main problem with joint commitment:
    - Hard to be aware of each others states at all times

Conventions

- Convention – means of monitoring a commitment
  - e.g. specifies under what circumstances a commitment can be abandoned.
- Need <u>conventions</u> to describe when to change a commitment:
  1. When to keep a commitment (retain)
  2. When to revise a commitment (rectify)
  3. When to remove a commitment (abandon)

Convention - Example

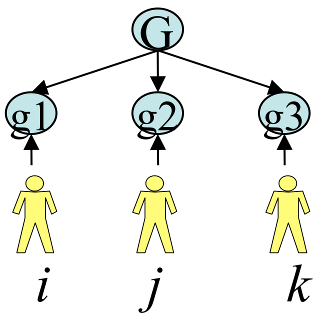
- Reasons for terminating a Commitment:
  - Commitment Satisfied
  - Commitment Unattainable
  - Motivation for commitment no longer present

- Rule R1:

```
If Commitment Satisfied OR
   Commitment Unattainable OR
   Motivation for Commitment no longer present
then
   terminate Commitment.
```

Joint Commitments

- Joint action by a team involves more than just the union of simultaneous individual actions.
  - When a group of agents are engaged in a cooperative activity, they must have:
    - **Joint commitment** to the overall activity
    - **Individual commitment** to the specific task that they have been assigned to

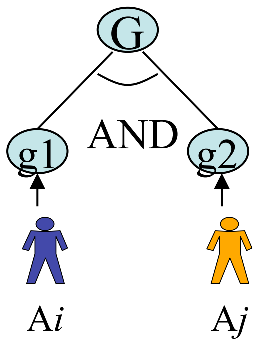


Social Conventions

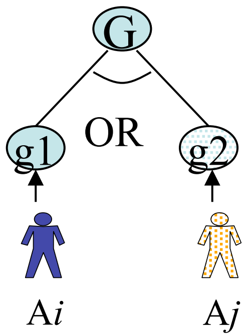
- Individual Conventions describe how an agent should monitor its commitments, but not how it should behave towards other agents.
  – Asocial
  – Sufficient for goals that are independent.
- For inter-dependent goals:
  – Need ***social* conventions**
    - Specify how to behave with respect to the other members of the **team**.

Social Conventions

- Team members must be aware of the convention that govern their interactions. e.g.



- Both A*i* and A*j* must fulfill their commitments to achieve G.



- Either A*i* or A*j* must fulfill their commitment.

➢ ***There is a need for all agents in a team to inform other members of the status of their commitments!***

Coordination Summary

- Coordination – ensuring coherent behaviour.
- Can be considered as a control over distributed search
- Coordination Techniques:
  - Organisational structures
  - Meta-level information exchange
  - Multi-agent Planning
  - Social norms and laws
  - Mutual Modelling
  - Joint Intentions

References – Recommended Reading for Teamwork

- Not curriculum:
  - Cohen, P. R. and Levesque, H. J., ”Teamwork”, *Nous*, 25, 1991.
  - Tambe, M., ”Towards Flexible Teamwork”, *Journal of Artificial Intelligence Research,* Volume 7, 1997, pp. 83-124.

Next Topic: MAS Architectures

Wooldridge: ”Introduction to MultiAgent Systems”
Appendix A