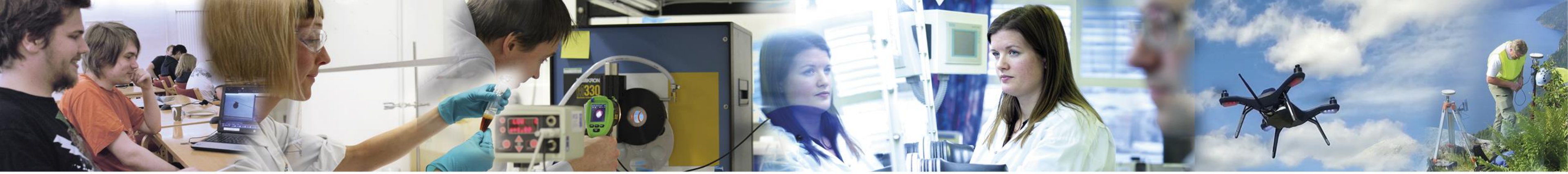


UiO : Department of Technology Systems
University of Oslo

TEK5010/9010 - Multiagent systems 2020
Lecture 2

Agents, communication and cooperation

Jonas Moen

Corona restrictions at UiO

Remember to keep everyone safe by:

1. Washing hands
2. Keeping your distance (1 metre)
3. Staying home if you are sick

https://www.uio.no/english/about/hse/corona/index.html

Highlights lecture 2 – Agents, communication and cooperation*

- Defintion of an intelligent agent
  - Agents as intentional systems and utility maximizers
  - Types of agents: Reasoning, reactive and hybrid agents
- Communication fundamentals
  - Reproducing data vs. conveying meaning
  - Ontology, knowledgebase and speech acts
- Types of cooperation and evaluation of success in MAS
  - Benevolence assumption in PPS/CDPS/MAS
  - Task sharing and result sharing

*Wooldridge, 2009: chapter 2-8

The agent

Definition of an agent:

«An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives», [Wooldridge & Jennings, 1995]

The agent

Definition of an agent:

- Objective/goal is to affect the environment in some desirable way
- Autonomy is the only generally accepted requirement
- Acting on behalf of someone

The intelligent agent

The capabilities of an intelligent agent,
[Wooldridge & Jennings, 1995]

1. Reactivity – respond to changes in the environment
2. Proactiveness – initiate goal-directed behaviour on their own
3. Social ability – interact with other agents in order to satisfy goals

Reactivity

A reactive system is one that maintains an ongoing interaction with its environment, and responds to changes that occur in it (in time for the response to be useful).

Proactiveness

Proactiveness = generating and attempting to achieve goals, not driven solely by events, but taking the initiative.

Recognizing opportunities.

Social ability

The real world is a multi-agent environment: we cannot go around attempting to achieve goals without taking others into account. Some goals can only be achieved by interacting with others.

Social ability in agents is the ability to interact with other agents (and possibly humans) via cooperation, coordination, and negotiation.

Social ability

- Cooperation is working together as a team to achieve a shared goal.
- Coordination is managing the interdependencies between activities.
- Negotiation is the ability to reach agreements on matters of common interest.

The intelligent agent

Notes:

- When the environment is fully known, a functional system can fulfill the goal of the programmer.
- When the environment is not fully known, due to changes, other agents and other uncertainties, a reactive system is more appropriate.
- Social ability is more than info exchange. It is more like negotiations, cooperation and coordination between agents.

Agents as intentional systems

The intentional stance:

‘Endow’ agents with mental states; in particular desires & beliefs, but also goals, wishes, hopes and so on.

For many researchers the idea of programming computer systems in terms of mentalistic notions is key component of agent-based systems.

Agents as intentional systems

The intentional stance:

- An abstraction tool for managing complexity in the social sciences, humanities, and biology. Seems to coordinate our activities in some way.
- Computer science is steadily moving towards higher level of abstraction in programming, from the early machine code paradigm via OO-design to agent-based intentional systems.

The intelligent agent

How would we tell an agent what to do?

1. We could tell it explicitly what to do in all cases, or
2. We could use some kind of performance measure, typically associate environment states with utility.

Task is to achieve optimal utility.

The intelligent agent

Maximizing expected utility:

$$Ag_{opt} = \max_{Ag \in AG_m} \sum_{r \in R(Ag,Env)}^{k} u(r) P(r|Ag, Env)$$

where Ag is expected utility of agent Ag in AG_m in Env

AG_m is the set of feasible agents (bounded optimality)

$u(r)$ is utility of run r for Ag, $u: R \rightarrow \mathbb{R}$,$\mathbb{R}$ is real number

$\sum P(*) = 1$ is the density function of all runs

The intelligent agent

Maximizing expected utility:

$$Ag_{opt} = \max_{Ag \in AG_m} \sum_{r \in R(Ag,Env)}^{k} u(r) P(r | Ag, Env)$$

This is decision theory, generally hard to define the utility function [Russell & Norvig, 1995]

Example of calculating max utility

Consider agent 1 and agent 2 in an environment having available to them some actions.

What is the expected utility of each agent?

And which agent is optimal in this problem?

Example of calculating max utility

Consider an environment $Env_1 = \langle E, e_0, \tau \rangle$ defined as follows:

$E \ = \ \{e_0, e_1, e_2, e_3, e_4, e_5\}$ is the environment states

$\tau \left(e_0 \overset{\alpha_0}{\rightarrow} \right) = \{e_1, e_2\ \}$ is state transform of action α_0

$\tau \left(e_0 \overset{\alpha_1}{\rightarrow} \right) = \{e_3, e_4, e_5\}$ is state transform of action α_1

There are two agents possible in this environment:

$Ag_1(e_0) = \alpha_0$ is describing agent 1's action set

$Ag_2(e_0) = \alpha_1$ is describing agent 2's action set

Example of calculating max utility

The probabilities of the various runs are as follows:

$$P\left(e_0 \overset{\alpha_0}{\rightarrow} e_1 \middle| Ag_1, Env_1\right) = 0.4$$

$$P\left(e_0 \overset{\alpha_0}{\rightarrow} e_2 \middle| Ag_1, Env_1\right) = 0.6$$

$$P\left(e_0 \overset{\alpha_1}{\rightarrow} e_3 \middle| Ag_2, Env_1\right) = 0.1$$

$$P\left(e_0 \overset{\alpha_1}{\rightarrow} e_4 \middle| Ag_2, Env_1\right) = 0.2$$

$$P\left(e_0 \overset{\alpha_1}{\rightarrow} e_5 \middle| Ag_2, Env_1\right) = 0.7$$

Example of calculating max utility

Assume the utility u of the different states is defined as:

$$u\left(e_0 \overset{\alpha_0}{\rightarrow} e_1\right) = 8$$

$$u\left(e_0 \overset{\alpha_0}{\rightarrow} e_2\right) = 11$$

$$u\left(e_0 \overset{\alpha_1}{\rightarrow} e_3\right) = 70$$

$$u\left(e_0 \overset{\alpha_1}{\rightarrow} e_4\right) = 9$$

$$u\left(e_0 \overset{\alpha_1}{\rightarrow} e_5\right) = 10$$

Example of calculating max utility

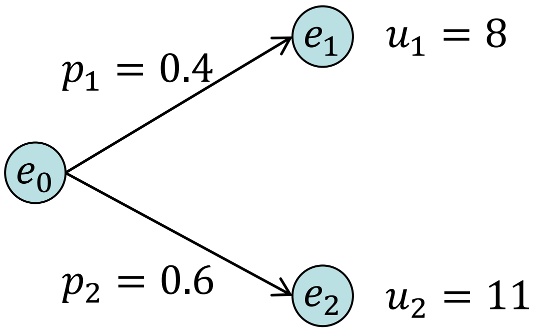
What is the expected utility of each agent? And which agent is optimal in this problem?

$$\hat{u}(Ag, Env) = \sum_{r \in R(Ag, Env)} u(r) P(r | Ag, Env)$$

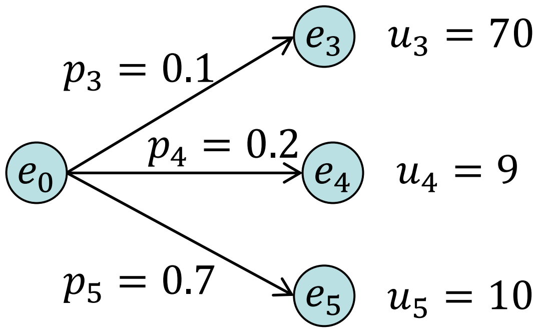
Example of calculating max utility

What is the expected utility of each agent? And which agent is optimal in this problem?

$Ag_1\left(e_0 \xrightarrow{\alpha_0}\right)$



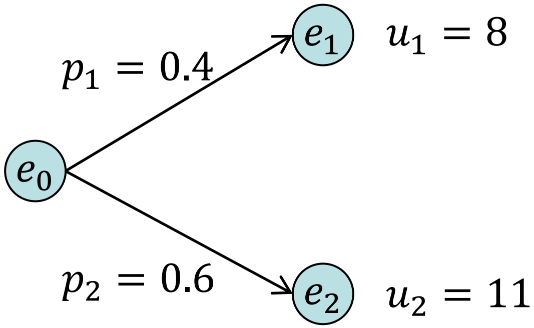
$Ag_2\left(e_0 \xrightarrow{\alpha_1}\right)$



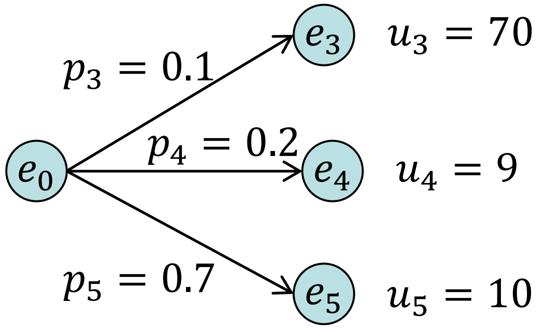
Example of calculating max utility

What is the expected utility of each agent? And which agent is optimal in this problem?

$Ag_1\left(e_0 \xrightarrow{\alpha_0}\right)$



$Ag_2\left(e_0 \xrightarrow{\alpha_1}\right)$



$\hat{u}(Ag_1) = p_1 u_1 + p_2 u_2 = 0.4 \cdot 8 + 0.6 \cdot 11$=9.8

Example of calculating max utility

What is the expected utility of each agent? And which agent is optimal in this problem?

$Ag_1\left(e_0 \overset{\alpha_0}{\rightarrow}\right)$

$Ag_2\left(e_0 \overset{\alpha_1}{\rightarrow}\right)$

$$\begin{aligned}\hat{u}(Ag_1) &= p_1 u_1 + p_2 u_2 \\ &= 0.4 \cdot 8 + 0.6 \cdot 11 \\ &= 9.8\end{aligned}$$

$$\begin{aligned}\hat{u}(Ag_2) &= p_3 u_3 + p_4 u_4 + p_5 u_5 \\ &= 0.1 \cdot 70 + 0.2 \cdot 9 + 0.7 \cdot 10 \\ &= 15.8\end{aligned}$$

Agent 2 have higher expected utility than agent 1 and is thus optimal agent for this problem.

Types of Agents

- Deductive reasoning agents (1956–present)
  Propose that agents use explicit logical reasoning in order to decide what to do.
- Reactive agents (1985–present)
  Problems with symbolic reasoning led to a reaction against this - the reactive agents movement.
- Hybrid agents (1990-present)
  Hybrid architectures attempt to combine the best of symbolic and reactive architectures.

Deductive reasoning agents

Deductive reasoning agents are agents in the ‘traditional’ approach to build artificial intelligent systems called *symbolic AI*.

- Logic-centred view of AI and the agent concept
- Logical approach to building agents and how agents should execute their behaviour
- Symbolic AI was the dominant approach to AI until 1980s

Symbolic AI

Intelligent behaviour can be generated in a system by giving that system a symbolic representation of its environment and its desires, and syntactically manipulate this representation.

- Symbols are logical formula
- Syntactic manipulation is logical deduction or theorem proving

Symbolic AI

It is a elegant logical semantic, but not without its problems:

- The symbolic representations of many environments are not obvious.
- *Calculative rationality* is not guaranteed in environments that change faster than agents can make decisions.
- Simple procedural knowledge might be unintuitive and cumbersome to represent i traditional logic.

Reactive and hybrid agents

In mid 1980s researchers begin to investigate alternatives to symbolic AI. Themes of research are:

1. Rejection of the symbolic representation of decision-making based on syntactic manipulation of such representation.
2. Intelligent, rational behaviour seems innately linked to the environment an agent occupies.
3. Intelligent behaviour emerges from the interaction of various simpler behaviours.

Reactive agents

Agents that simply react to an environment, without reasoning about it, are called reactive agents:

1. Behavioural
   Develop and combine individual behaviour to produce complex behaviours.
2. Situated
   Agents are in some environment rather than disembodied from it.

Reactive agents

Limitations to reactive agents

1. Reactive agents make decisions based on local information. How is this representative of the environment in order to determine an acceptable action?
2. Reactive agents are the ‘short-term’ view
   How is local information combined with non-local information?

Reactive agents

Limitations to reactive agents

3. Emergens indicate that the relation between individual behaviour, environment and overall behaviour is not understandable. How do we engineer agents that fulfil specific tasks? How do we develop a methodology for building such agents?
4. Complexity of designing behaviours
   How do we design agents with layers too complex to be understood? Alife, EC and neural nets?

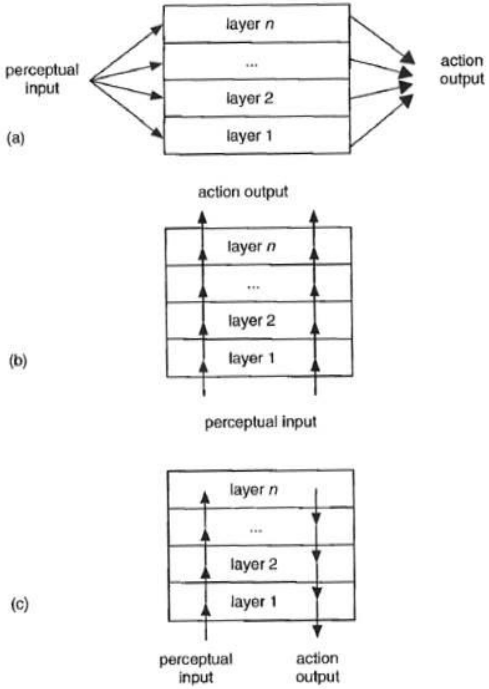
Hybrid agents

Hybrid agents combine reactive and deliberate reasoning in (usually) layered architecture:

1. Reactive layer
2. Proactive layer (deliberation)

In principle there is no limit to number of layers applied

Hybrid agents



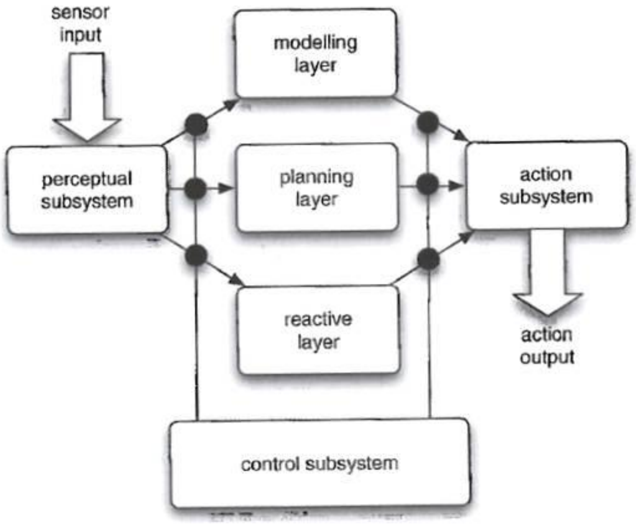


Image: Figure 5.2 and 5.3, Wooldridge 2009

Communication

Communication = the process of sharing
data/information/meaning

A topic of central importance in computer science for a long time in parallel, distributed, and agent-based systems.

Communication

How can agents understand each other?

The area of

1. Communicating data
2. Ontologies for meaning

Communication fundamentals

Claude Shannon

- ‘The father of information theory’
- Shannon founded information theory with “A Mathematical Theory of Communication”, published in 1948.

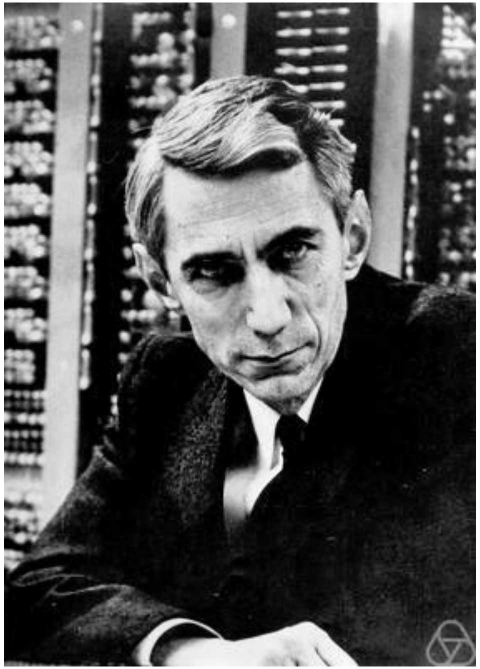


Image: Wikipedia

Communication fundamentals

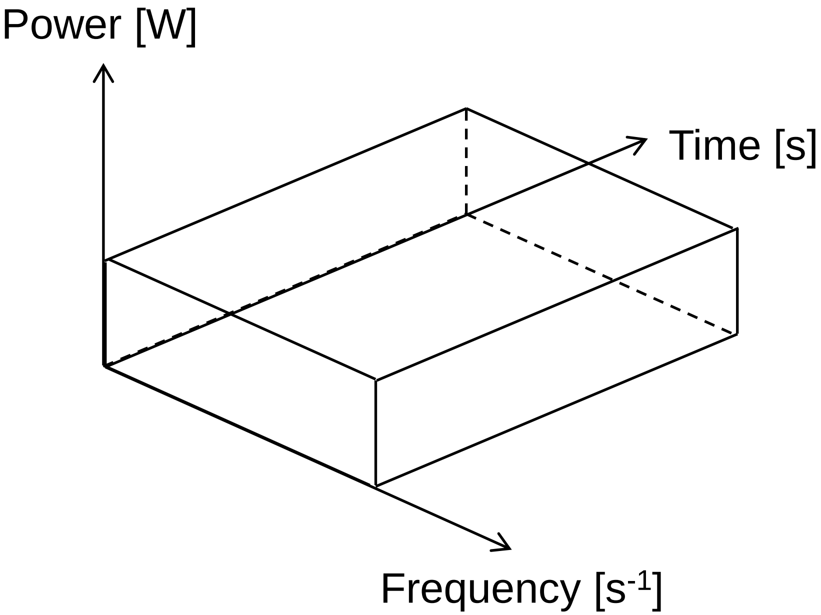
Communication, in terms of Shannon, is not concerned by meaning, but the reproduction of bit-strings from one place to another in a noisy environment. We talk about

1. Bits per second transferred
2. Probability of error per bit transferred

i.e. we do not talk about the meaning of the bit-string

Communication fundamentals

$$Energy = \int Pt\, df$$



Communication fundamentals

Communication system

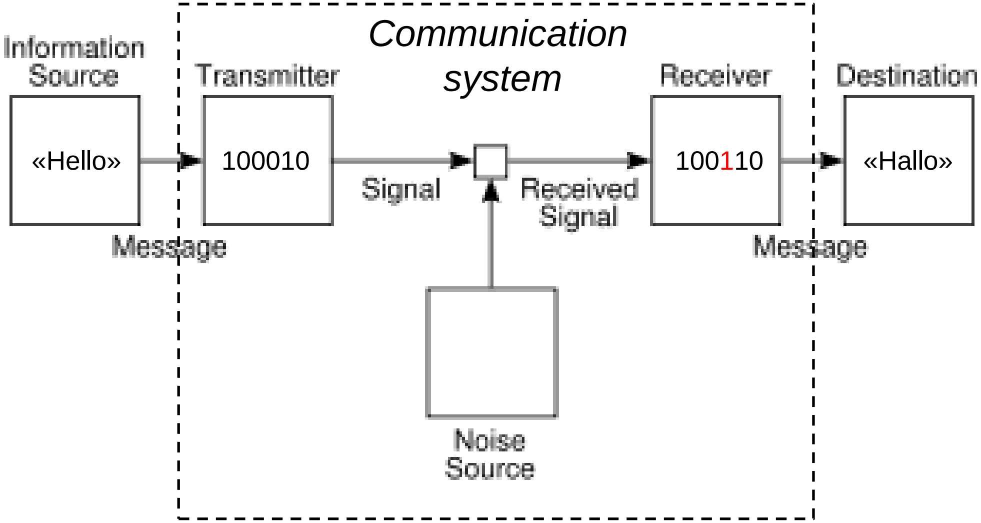
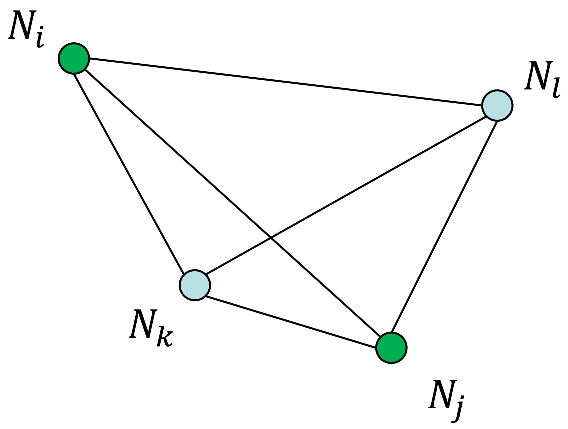


Image: Wikipedia

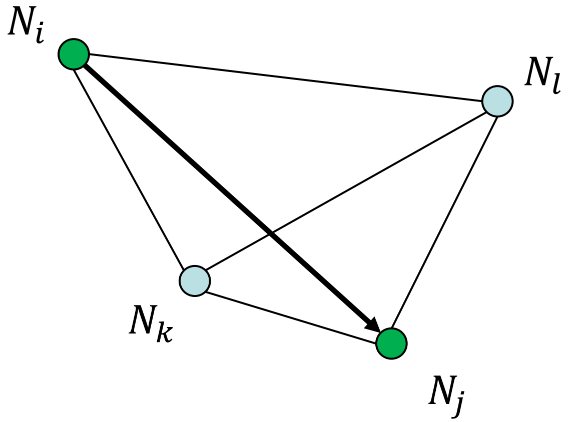
Communication fundamentals

How do we get a binary string from node N_i to node N_j as efficiently as possible?



Communication fundamentals

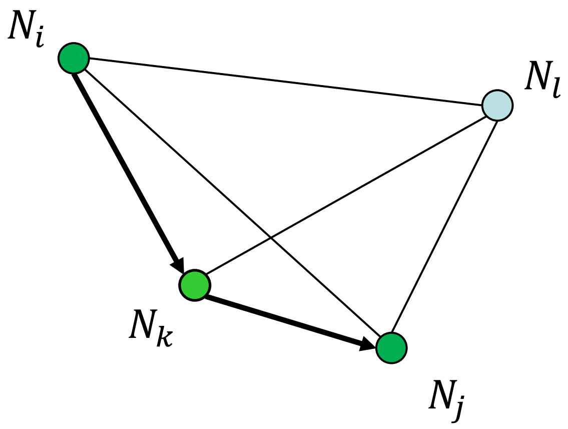
How do we get a binary string from node N_i to node N_j as efficiently as possible?



Direct communication between N_i and N_j could give low bit-rates due to distance.

Communication fundamentals

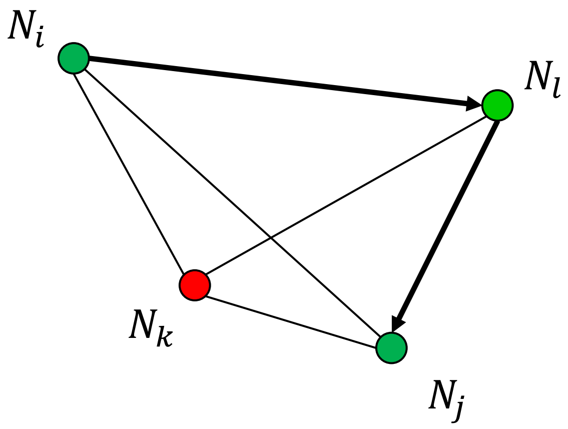
How do we get a binary string from node N_i to node N_j as efficiently as possible?



Ad hoc networking via node N_k could boost bit-rates.

Communication fundamentals

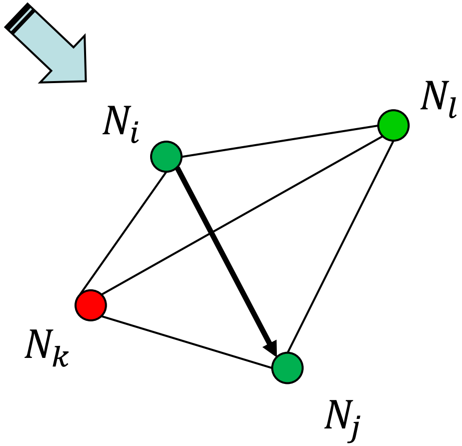
How do we get a binary string from node N_i to node N_j as efficiently as possible?



If node N_k is unavailable node N_l could be used instead.

Communication fundamentals

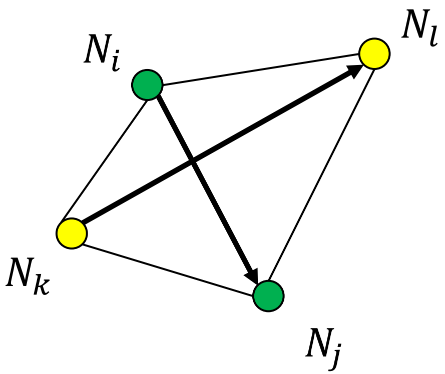
How do we get a binary string from node N_i to node N_j as efficiently as possible?



What if the position of all the nodes are dynamic?

Communication fundamentals

How do we get a binary string from node N_i to node N_j as efficiently as possible?



And what if node N_k simultaneously wants to talk to node N_l?

Communication fundamentals

1. Direct communication between N_i and N_j could give low bit-rates due to distance.
2. Ad hoc networking via node N_k could boost bit-rates
3. If node N_k is unavailable node N_l could be used instead
4. What if the position of all the nodes are dynamic and they cross-talk?

Network protocol for routing is part of communication and very important in real multiagent systems.

Ontology

An ontology is a specification of a terminology (a set of terms) intended to provide a common basis of understanding about some domain.

Ontology

«An ontology is a formal definition of a body of knowledge. The most typical type of ontology used in building agents involves a structural component. Essentially a taxonomy of classes and subclass relations coupled with definitions of the relationships between these things», [Hendler]

Ontology

In terms of computer science:

- The development and deployment of ontologies have their origins in the semantic web research
- The semantic web is the main motivation for research into ontologies

Ontology fundamentals

Ontologies describe relations between objects/concepts (as in object-oriented design)

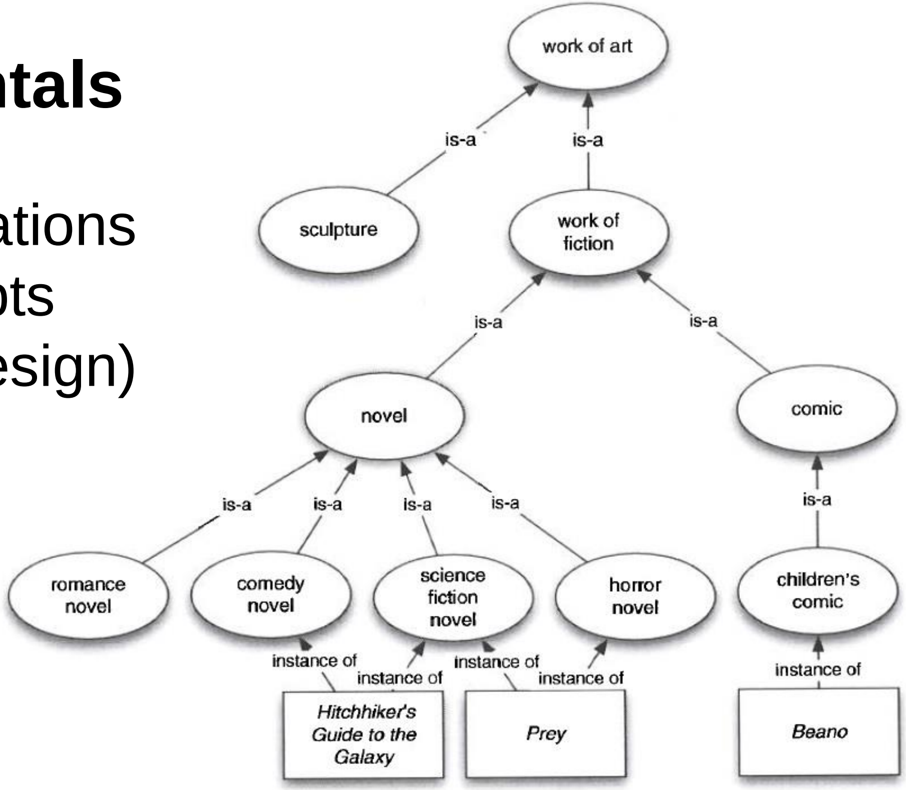


Image: Figure 6.1, Wooldridge 2009

Ontology fundamentals

- Defining new terms in relation to old ones
- Classes and instances
- Class properties
- Subclasses, superclasses and inheritance

Ontology fundamentals

Ontology vs. knowledgebase

1. Ontology is the structural part of a class diagram
2. Knowledgebase is an ontology together with a set of instances of classes.

Agent communication languages

Many formalisms exist for communication in concurrent systems:

1. Synchronization - type ‘lost update’.
2. Object-oriented programming
   External objects invoke/execute methods in other objects
3. Agent-based setting
   Agents determine on their own to act or not based on information. The sender must try to manipulate desires and beliefs in receiver in order to obtain desired action.

Agent communication languages

FIPA – the Foundation of Intelligent Physical Agents [FIPA, 1999]

1. IEEE standardization for agent systems.
2. The ACL – Agent Communication Language
3. Conformance testing. Does a particular agent program respect the semantics of the language?

Agent communication languages

JADE – Java Agent Development Environment [Bellifemine et *al.*, 2007]

1. An infrastructure for deploying FIPA agent systems.
2. De facto standard in agent-based communication on the net.

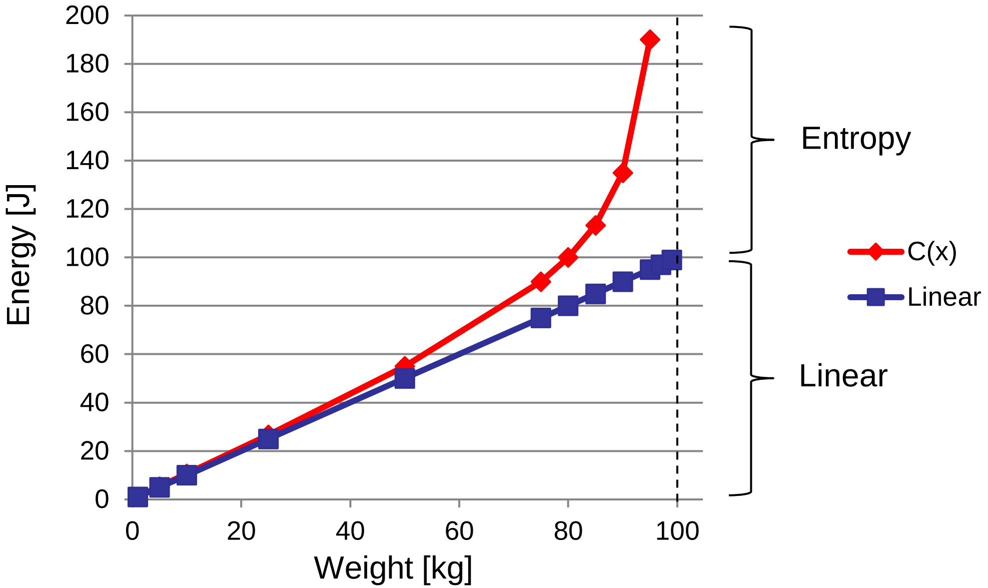
Types of cooperation

1. Cooperative distribution of free resources/utility
   - Ultimatum game

2. Cooperative production of resources/utility
   - Direct cooperation, e.g. lifting a heavy rock
   - Indirect cooperation, i.e. specialization (division of labour and reciprocal cooperation)
   - Prone to cheating/defecting

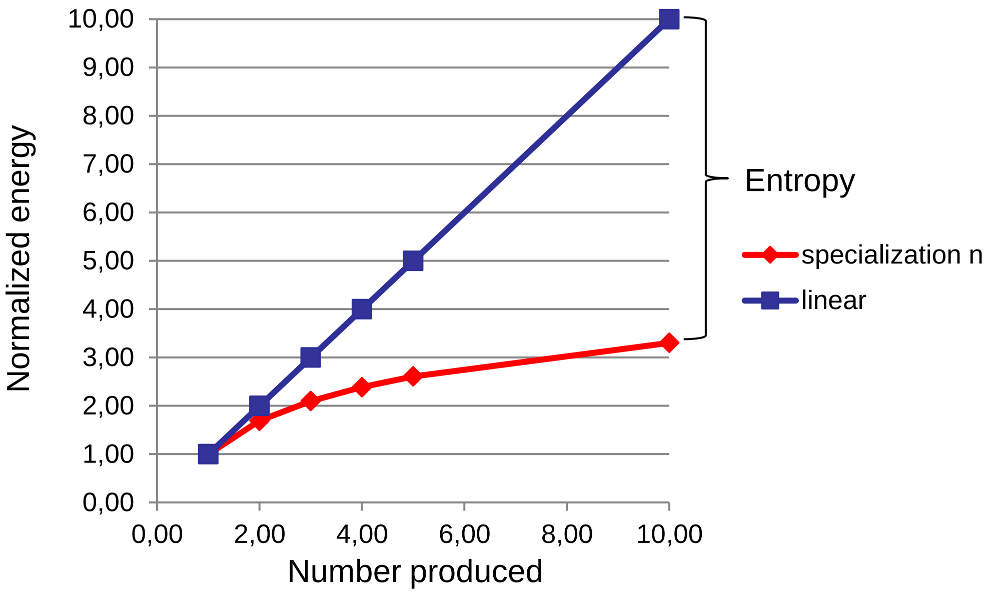
Why is it beneficial to cooperate?

Direct cooperation: example lifting a heavy rock



Why is it beneficial to cooperate?

Indirect cooperation: example of division of labour



Working together

Main distinction between multiagent systems and ‘traditional’ distributed or concurrent systems

1. Agent systems may have agents with different desires (goals/intentions). The dynamics resembles that of games and strategic interaction.
2. Agent systems are autonomous with actions not hardwired at design time.

Cooperative Distributed Problem Solving (CDPS)

«CDPS studies how loosely coupled networks of problem solvers can work together to solve problems that are beyond their individual capability», [Durfee et *al.*, 1989]

The benevolence assumption is common in CDPS.

Benevolence assumption

Agents share common goal in order to

1. Achieve system overall objective
2. Greatly simplifies the design task

This is in contrast to more strategic multiagent systems that often focus on self-interested agents that do not share a common goal.

Parallel Problem Solving (PPS)

1. Decomposition and exploitation of inherent parallelism in solving problems
2. Processors are often assumed to be homogenous
3. Central steering
4. PPS is often subpart of many CDPS and MAS

PPS is distinctly different from CDPS and MAS

Evaluation of success in CDPS/MAS

1. Coherence
   «How well the system behaves as a unit along some dimension of evaluation», [Bond and Gasser, 1988], e.g. solution quality, efficiency of resource usage, conceptual clarity, performance degradation under uncertainty, etc.
2. Coordination
   «The degree to which the agents can avoid extraneous activity, synchronizing and aligning their activities», [Bond and Gasser,1988].

Evaluation of success in CDPS/MAS

«Evaluation of performance in CDPS/MAS has been the focus of attention in research more than any other issue», [Wooldridge, 2009].

Evaluation of success in CDPS/MAS

The main issues of CDPS are

1. How can a problem be divided into smaller tasks for distribution among agents?
2. How can a problem solution be effectively synthesized from subproblem results?
3. How to maximize the coherence metric?
4. How to coordinate agents in order to avoid destructive interactions?

General framework of CDPS/MAS

Task sharing and result sharing

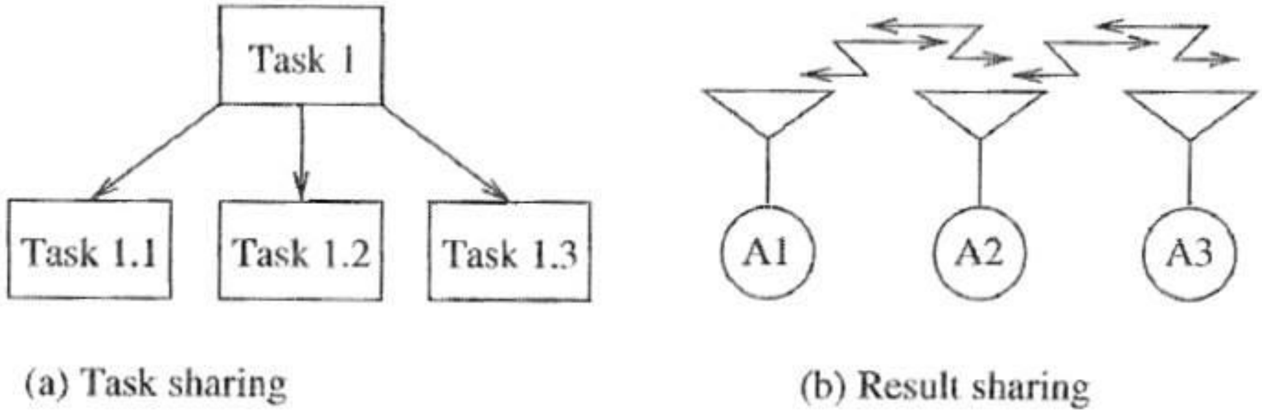


Image: Figure 8.2, Wooldridge 2009

General framework of CDPS/MAS

Task sharing

How tasks are to be allocated to individual agents

1. Homogenous agents, any task can be allocated to any agent (implicit benevolence assumption)
2. Truly autonomous (heterogenous) agents, task allocation is a matter of negotiations, agreements and bargaining, i.e. games.

General framework of CDPS/MAS

Result sharing

Agents share information relevant to their subproblem

1. Proactively, inform (help) some other agent by belief
2. Reactively, inform (help) some other agent by request or by observation of other agent

General framework of CDPS/MAS

Result sharing can improve group performance by increasing

1. Confidence
   Independently derived solutions can be cross-checked, highlighting possible errors, increase confidence in overall solution.
2. Completeness
   Agents can share their local view to achieve a better overall global view of problem.

General framework of CDPS/MAS

Result sharing can improve group performance by increasing

3. Precision
   Agents can share results to ensure that the precision of the overall solution is increased
4. Timeliness
   Even if one agent could solve a problem on its own, by sharing subsolutions among multiagents, the result could be derived more quickly.

Multiagent planning and synchronization

Multiagent planning

1. Centralized planning for distributed plans produce «master-slave» relations.
2. Distributed planning
   Agents cooperate to fom a centralized plan
3. Distributed planning for distributed plans
   Agents cooperate to form individual plans. Agents may be self-interested requiring negotiations. This instance is the most complex and demanding to implement.

Summary lecture 2 – Agents, communication and cooperation*

- Defintion of an intelligent agent
  - Agents as intentional systems and utility maximizers
  - Types of agents: Reasoning, reactive and hybrid agents
- Communication fundamentals
  - Reproducing data vs. conveying meaning
  - Ontology, knowledgebase and speech acts
- Types of cooperation and evaluation of success in MAS
  - Benevolence assumption in PPS/CDPS/MAS
  - Task sharing and result sharing

*Wooldridge, 2009: chapter 2-8