Agent Communication

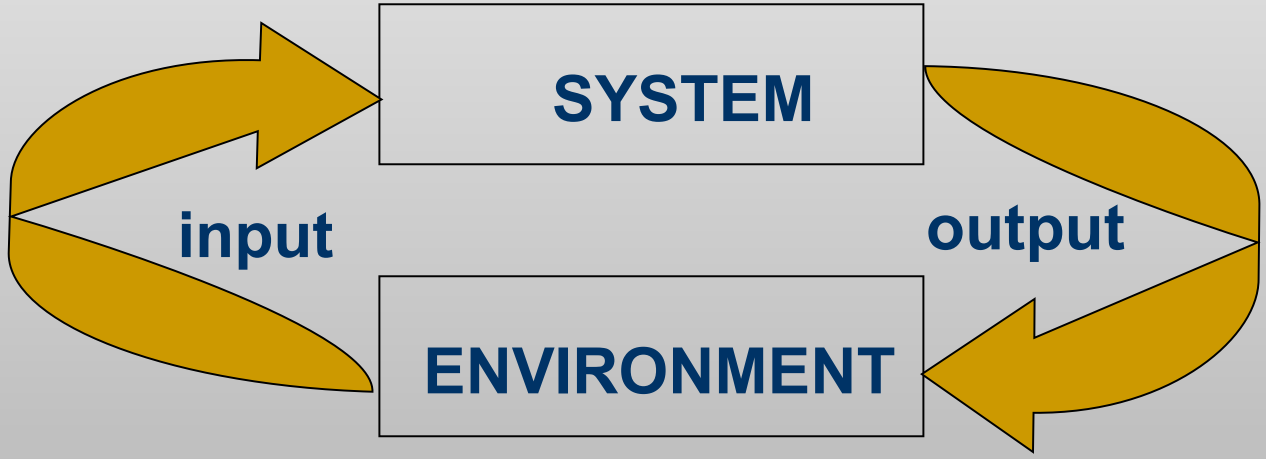
Based on, and inspired by slides from:
Michael Wooldridge, Jeff Rosenshein, Jean-Paul Sansonnet

Philippe Pasquier
Office 565 (floor 14)
pasquier@sfu.ca

SFU

Autonomous agents

- **An agent is a computer system capable of autonomous action *in some environment* in order to meet its design objectives**



BDI agents – basic algorithm

BDI-interpreter

- (B,D,I):= Initialize-state();
- While true do
  - Update(B,D, I); // perceptions may update beliefs, desires and intentions (ex. Once fulfilled, an intention is dropped)
  - Options:= option-generator(B,D,I);
  - Selected-options:=deliberate(B,D,I);
  - Update-intentions(Selected-options,I);
  - Plan:=Planing(I,B);
  - Execute(Plan);
  - Get-new-perceptions();
- End While

Summary

- An agent is a computer system capable of **flexible autonomous** action in some **environment**.
- **Situatedness**: peceiving the environment via sensors and being abble to affect the environment via effectors
- **Autonomy**: capability of action without intervention, and control over internal state
- **Flexibility**:
  - Responsiveness: respond in a timely fashion to change in the environment
  - Pro-activity: actions which go beyond simple response to stimulus
  - **Sociability**: ability to interact with other agents and humans for mutual benefit

Agent Communication

- Designing MAS:
  - Agent design
  - Sociaety design
- In this lecture, we cover *macro-aspects* of intelligent agent technology: issues relating to the agent *society*, rather than the individual
- Adress the sociability of the agent

Outline of the presentation

- Introduction:
  - What is cooperation?
  - Cooperative *versus* non-cooperative encounters
- Early systems:
  - Methode invocation
  - The blackboard architecture
- *Agent Communication Languages (ACL):*
  - KQML & KIF;
  - FIPA ACL
- Protocols and agent conversations:
  - The contract net protocol
  - Argumentation
  - Negotiation protocols: Bargaining and Interest based negotiation (IBN)

Working Together

- **Cooperation is:** the practice of working in common with mutually agreed-upon goals and possibly methods, instead of working separately in competition, and in which the success of one is dependent and contingent upon the success of another.
  - E.g. I can't play a quintet alone!
- **When agents are working together, it is important to make a distinction between:**
  - ***Benevolent agents***
  - ***Self-interested agents:*** ***it does not mean that they want to cause harm to other agents or that they care only about themselves. It means that it follows its interest as represented by a utility function (representing the agent preferences)***

Benevolent Agents

- **If we "own" the whole system, or are in a cooperative environment, we can design agents that help each other whenever asked (if possible)**
- **In this case, we can assume agents are *benevolent*: others best interest is their best interest**
- **Problem-solving in benevolent systems is called *cooperative distributed problem solving* (CDPS)**
- ***Benevolence simplifies the system design task enormously!***

Self-Interested Agents

- **If agents represent individuals or organizations, (the more general case), then we cannot make the benevolence assumption**
- **Agents will be assumed to act to further their own interests, possibly at expense of others**
- **Potential for *conflict.***
- ***E.g. Competitive environment (sport, war, ...)***
- **May complicate the design task enormously**

Task Sharing and Result Sharing

- **Two main modes of cooperative problem solving:**
  - ***Task sharing*: components of a task are distributed to various agents**
  - ***Result sharing*: information (partial results, etc.) is distributed**
- **Both benevolent and self-interested agents have to work together and need to cooperate.**
- **Cooperation requires coordination**
- **Coordination of multiple independent autonomous agent require communication (of some sort).**
- *E.g. Lifting a table*
- *E.g. I can't play a quintet with the others if I can't ear them!*

Outline of the presentation

- Introduction:
  - What is cooperation?
  - Cooperative *versus* non-cooperative encounters
- Early systems:
  - Methode invocation
  - The blackboard architecture
- Agent Communication Languages (ACL):
  - KQML & KIF;
  - FIPA ACL
- Protocols and agent conversations:
  - The contract net protocol
  - Argumentation
  - Negotiation protocols: Bargaining and Interest based negotiation (IBN)

Outline of the presentation

- Introduction:
  - What is cooperation?
  - Cooperative *versus* non-cooperative encounters
- Early systems:
  - Methode invocation
  - The blackboard architecture
- Agent Communication Languages (ACL):
  - KQML & KIF;
  - FIPA ACL
- Protocols and agent conversations:
  - The contract net protocol
  - Argumentation
  - Negotiation protocols: Bargaining and Interest based negotiation (IBN)

Blackboard Systems

- One of the first scheme introduced for cooperative problem solving
- Introduce in a system called HEARSAY-II (1975, Carnegie Mellon) as results sharing system via shared data structure (BB)
- Multiple agents can read and write to BB
- Agents write partial solutions to BB
- BB can be structured into hierarchy
- Problems:
  - Mutual exclusion of access to the BB required $\Rightarrow$ bottleneck
  - Not concurrent activity
- Compare: JavaSpaces (http://java.sun.com/developer/technicalArticles/tools/JavaSpaces/)

Outline of the presentation

- Introduction:
  - What is cooperation?
  - Cooperative *versus* non-cooperative encounters
- Early systems:
  - Methode invocation
  - The blackboard architecture
- Agent Communication Languages (ACL):
  - KQML & KIF;
  - FIPA ACL
- Protocols and agent conversations:
  - The contract net protocol
  - Argumentation
  - Negotiation protocols: Bargaining and Interest based negotiation (IBN)

Introduction to agent communication

- Here, we will focuss on message-based communication between cognitive agents (e.g. BDI). We will see next week that other type of communication (low level, signal oriented) are possible
- Like for the agent architecture, a strong anthropomorphism will be noticed.
- We have presented ideas that holds for human communication and are used for artificial agents
- This is usefull since agents and humans are meant to communicate:
  - Human-human communication
  - Human-machine communication
  - Machine-machine communication

Based on Speech Act Theory

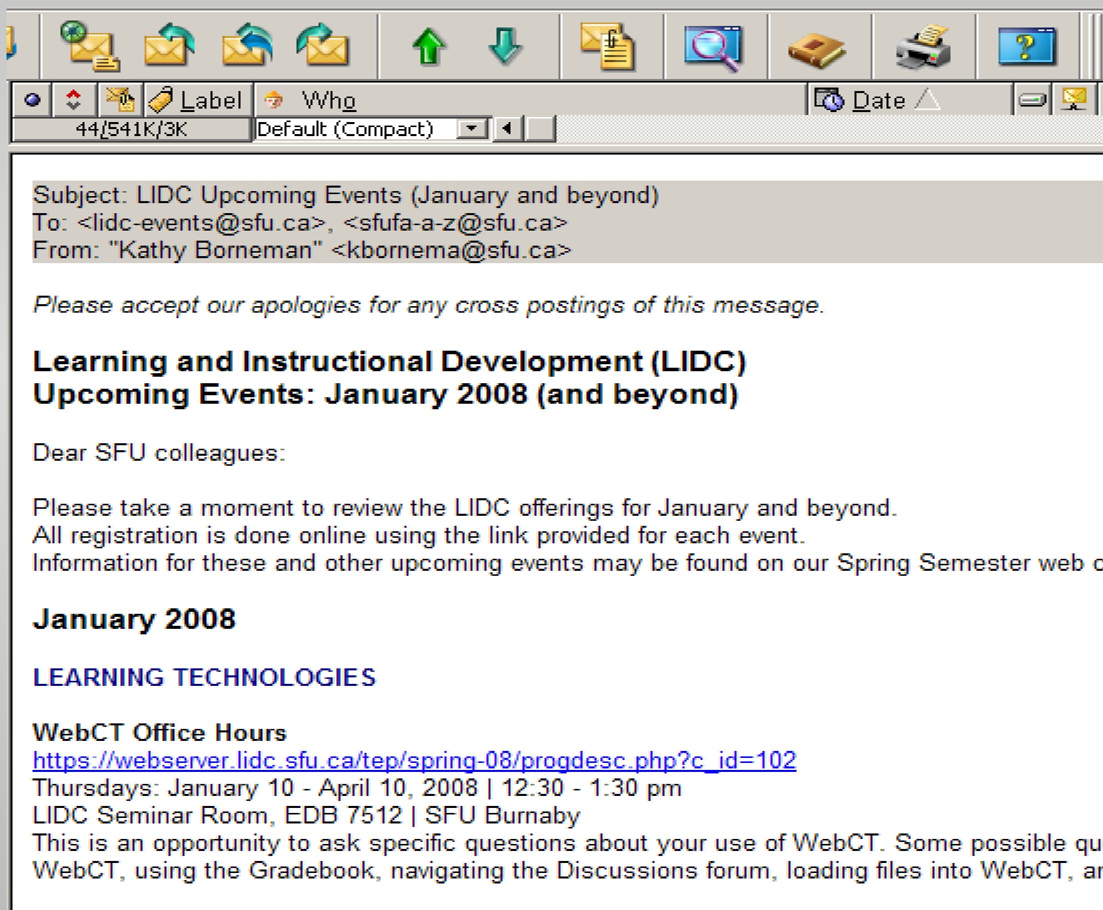
- Most treatments of communication in cognitive (multi)agent systems borrow their inspiration from the *speech act theory*
- Utterances are treated like 'physical actions' that aim (goal, intention) to *change the state of the world*
- The speech act theory is a theory of how utterances are used to achieve intentions
- Agent communication languages only consider two direction of fit:
  - The direction from the words to the world (assertive)
  - The direction from world to the words (directive)

Speech Acts

- In general, a speech act can be seen to have two components:
  - a *performative verb*: request, inform, inquire, ...
  - *propositional content*: e.g., “the video is played”
- Examples:
  - performative = request
content = “the door is closed”
speech act = “please close the door”
  - performative = inform
content = “the door is closed”
speech act = “the door is closed!”
  - performative = inquire
content = “the door is closed”
speech act = “is the door closed?”

ACLs: Agent Communication Languages

- **The e-mail metaphor**



Underlying Communication Infrastructure

Message

Header

Message Content

KQML and KIF

- We now consider *agent communication languages* (ACLs) — standard formats for the exchange of messages
- The best known ACL is KQML, developed by the ARPA knowledge sharing initiative
  KQML is comprised of two parts:
  - Message format: the knowledge query and manipulation language (KQML)
  - Content: the knowledge interchange format (KIF)

KQML and KIF

- KQML is an 'outer' language, that defines various acceptable 'communicative verbs', or *performatives*
  Example performatives:
  - `ask-if` ('is it true that. . . ')
  - `perform` ('please perform the following action. . . ')
  - `tell` ('it is true that. . . ')
  - `reply` ('the answer is . . . ')
- KIF is a language for expressing message *content*

KIF – Knowledge Interchange Format

Used to state and represent:

- Properties of things in a domain (e.g., "Noam is chairman")
- Relationships between things in a domain (e.g., "Amnon is Yael's boss")
- General properties of a domain (e.g., "All students are registered for at least one course")

KIF – Knowledge Interchange Format

- “The temperature of m1 is 83 Celsius”:
```
(= (temperature m1) (scalar 83 Celsius))
```
- “An object is a bachelor if the object is a man and is not married”:
```
(defrelation bachelor (?x) :=
    (and (man ?x) (not (married ?x))))
```
- “Any individual with the property of being a person also has the property of being a mammal”:
```
(defrelation person (?x) :=> (mammal ?x))
```

KQML and KIF

- In order to be able to communicate, agents must have agreed on a common set of terms
- A formal specification of a set of terms is known as an *ontology*
- The knowledge sharing effort has associated with it a large effort at defining common ontologies — software tools like Ontolingua for this purpose
- Example KQML/KIF dialogue…

```
A to B: (ask-if (> (size chip1) (size chip2)))
B to A: (reply true)
B to A: (inform (= (size chip1) 20))
B to A: (inform (= (size chip2) 18))
```

Semantics of the Performative

Plan Based Semantics

- Action: <precond> Body <Postcond>
- Cohen & Perrault (1979) defined semantics of speech acts using the *precondition-delete-add* list formalism of planning research
- Note that a speaker cannot (generally) *force* a hearer to accept some desired mental state
- In other words, there is a separation between the *illocutionary act* and the *perlocutionary act*

Plan-Based Semantics

- **Here is their semantics for *request*:**
  $request(s, h, \phi)$
  pre:
  - s **believe** h **can do** ϕ
    (you don't ask someone to do something unless you think they can do it)
  - s **believe** h **believe** h **can do** ϕ
    (you don't ask someone unless *they* believe they can do it)
  - s **believe** s **want** ϕ
    (you don't ask someone unless you want it!)

  post:
  - h **believe** s **believe** s **want** ϕ
    (the effect is to make them aware of your desire)

FIPA-ACL

- **More recently, the Foundation for Intelligent Physical Agents (FIPA) started work on a program of agent standards — the centerpiece is an ACL**
- **Basic structure is quite similar to KQML:**
  - ***performative***
    20 performatives in FIPA-ACL
  - ***housekeeping***
    e.g., sender, content language, etc.
  - ***content***
    the actual content of the message

FIPA

- **Example:**

```
(inform
    :sender      agent1
    :receiver    agent5
    :content     (price good200
150)
    :language    sl
    :ontology    hpl-auction
)
```

FIPA

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

“Inform” and “Request”

- “Inform” and “Request” are the two basic performatives in FIPA. All others are *macro* definitions, defined in terms of these.
- The meaning of inform and request is defined in two parts:
  - pre-condition
    what must be true in order for the speech act to succeed
  - “rational effect”
    what the sender of the message hopes to bring about

“Inform” and “Request”

- For the “inform” performative…
  The content is a *statement*.
  Pre-condition is that sender:
  - believe that the content is true
  - intends that the recipient believe the content
  - does not already believe that the recipient is aware of whether the content is true or not

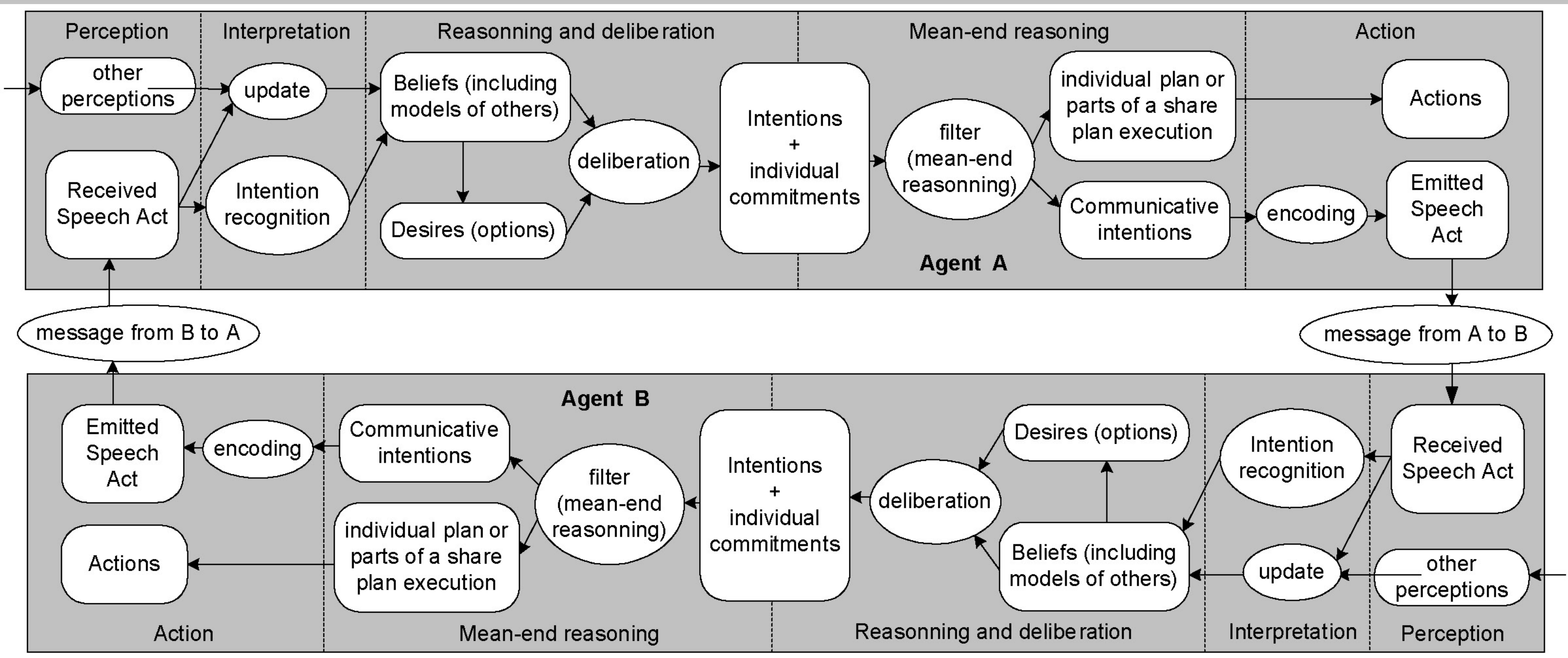
“Inform” and “Request”

- For the “request” performative…
  The content is an *action*.
  Pre-condition is that sender:
  - intends action content to be performed
  - believes recipient is capable of performing this action
  - does not believe that receiver already intends to perform action

Using ACLs in Open Systems

- Open systems are made of heterogeneous agents (hence the use of keeping the content language flexible and handle the not-understood performative)
- Agents can get in and out at any time:
  - White pages: register agents physical address
  - Yellow pages: register agents capabilitites

Application: artificial agents, ...



Outline of the presentation

- Introduction:
  - What is cooperation?
  - Cooperative *versus* non-cooperative encounters
- Early systems:
  - Methode invocation
  - The blackboard architecture
- Agent Communication Languages (ACL):
  - KQML & KIF;
  - FIPA ACL
- Protocols and agent conversations:
  - The contract net protocol
  - Argumentation
  - Negotiation protocols: Bargaining and Interest based negotiation (IBN)

Communication Protocols

- Protocols are structured ways to encode a type of conversation.
- Often formalised as finite state machines (state, transitions) with an initial and a final state
- Protocols simplify the computation by restricting the agent to a limited set of performative at any given point.
- Protocols also allow a build-in turn-taking mechanism
- Protocols capture the conventional aspects of conversations

The Contract Net Protocol

- A well known task-sharing protocol for *task allocation* is the *contract net*:
  1. Recognition
  2. Announcement
  3. Bidding
  4. Awarding
  5. Expediting

Recognition

- In this stage, an agent recognizes it has a problem it wants help with.
- The agent has a goal, and either:
  - Realizes it cannot achieve the goal in isolation — does not have capability (no plan for it)
  - Realizes it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc.)

Announcement

- In this stage, the agent with the task sends out an *announcement* of the task which includes a *specification* of the task to be achieved
- Specification must encode:
  – description of task itself (maybe executable)
  – any constraints (e.g., deadlines, quality constraints)
  – meta-task information (e.g., “bids must be submitted by…”)
- The announcement is then *broadcasted*

Bidding

- Agents that receive the announcement decide for themselves whether they wish to *bid* for the task
- Factors:
  - Agent must decide whether it is capable of expediting task
  - Agent must determine quality constraints & price information (if relevant)
- If they do choose to bid, then they submit a *tender* *(completely specified offer)*

Awarding & Expediting

- The agent that sent task announcement must choose between bids & decide who to "award the contract" to
- The result of this process is communicated to all the agents that submitted a bid
- The successful *contractor* then expedites the task
- May involve *sub-contracting* *(eventually using contract net)*

Implementing Contract Net

As simple as it looks, implementing the contract net raises a number of issues:

- **How to…**
  - **…specify *tasks*?**
  - **…specify *quality of service*?**
  - **…select between competing offers?**
  - **…differentiate between offers based on multiple criteria?**

The Contract Net

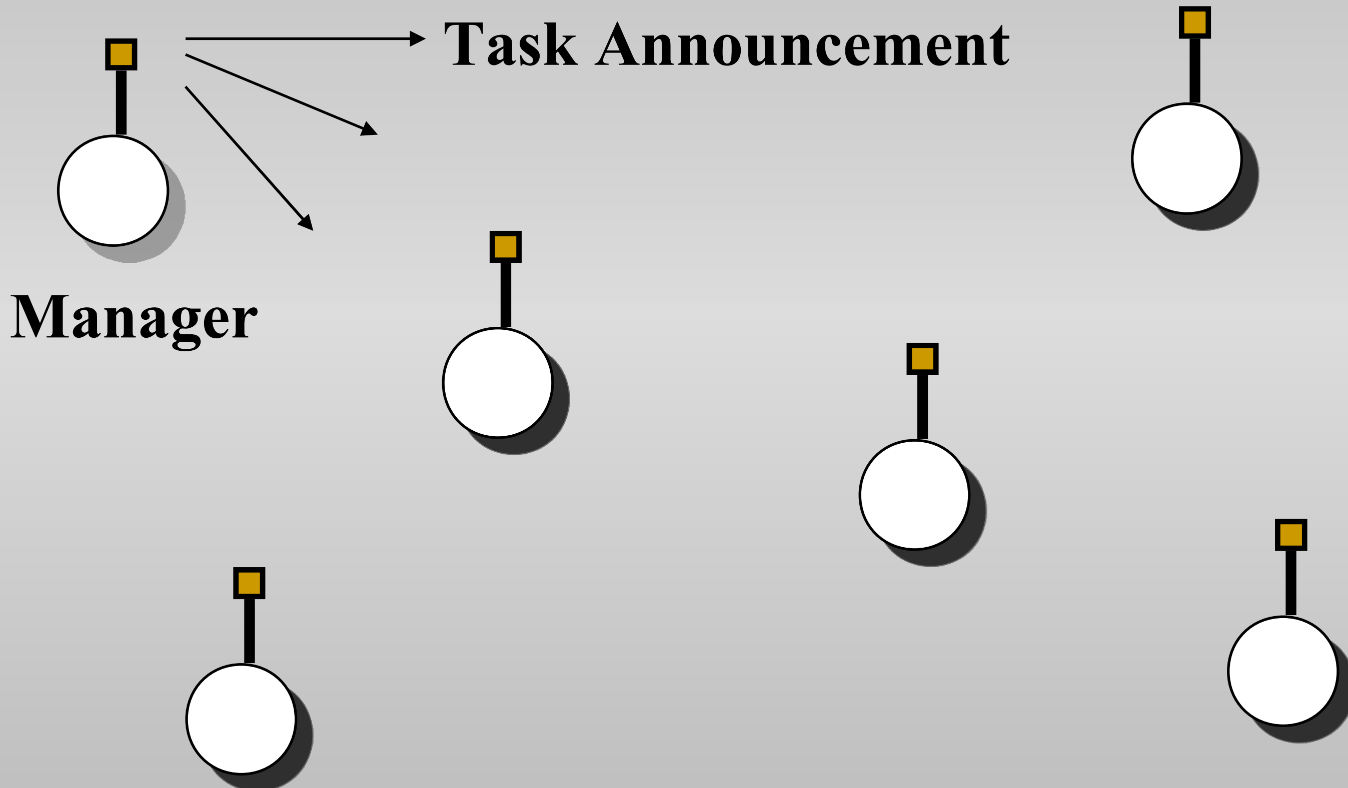
- An approach to *distributed problem solving*, focusing on task distribution
- Task distribution viewed as a kind of **contract negotiation**

Four Phases to Solution in cooperative Problem solving, as Seen in Contract Net:

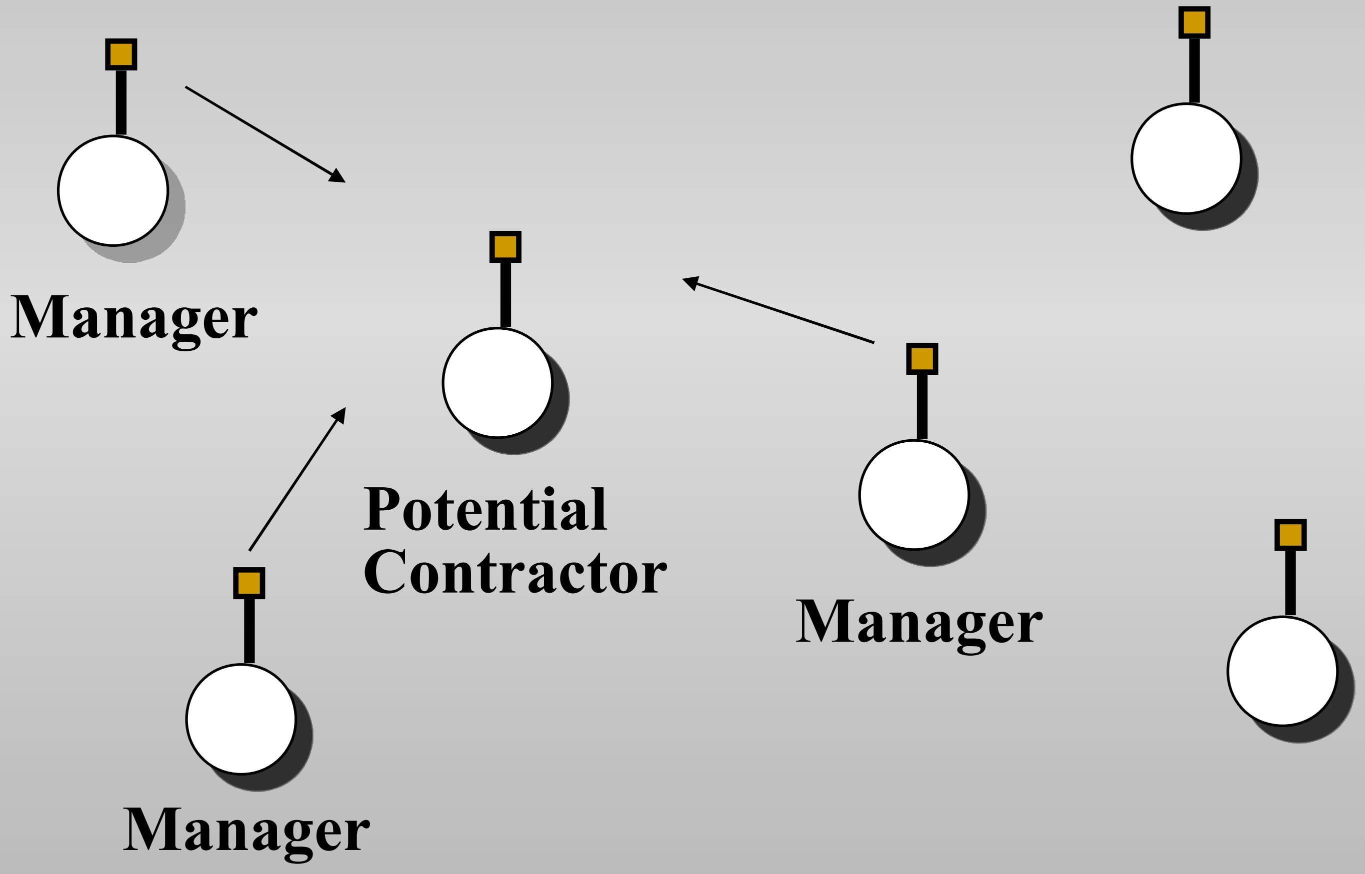
1. Problem Decomposition
2. Sub-problem distribution
3. Sub-problem solution
4. Answer synthesis

The contract net protocol deals with phase 2.

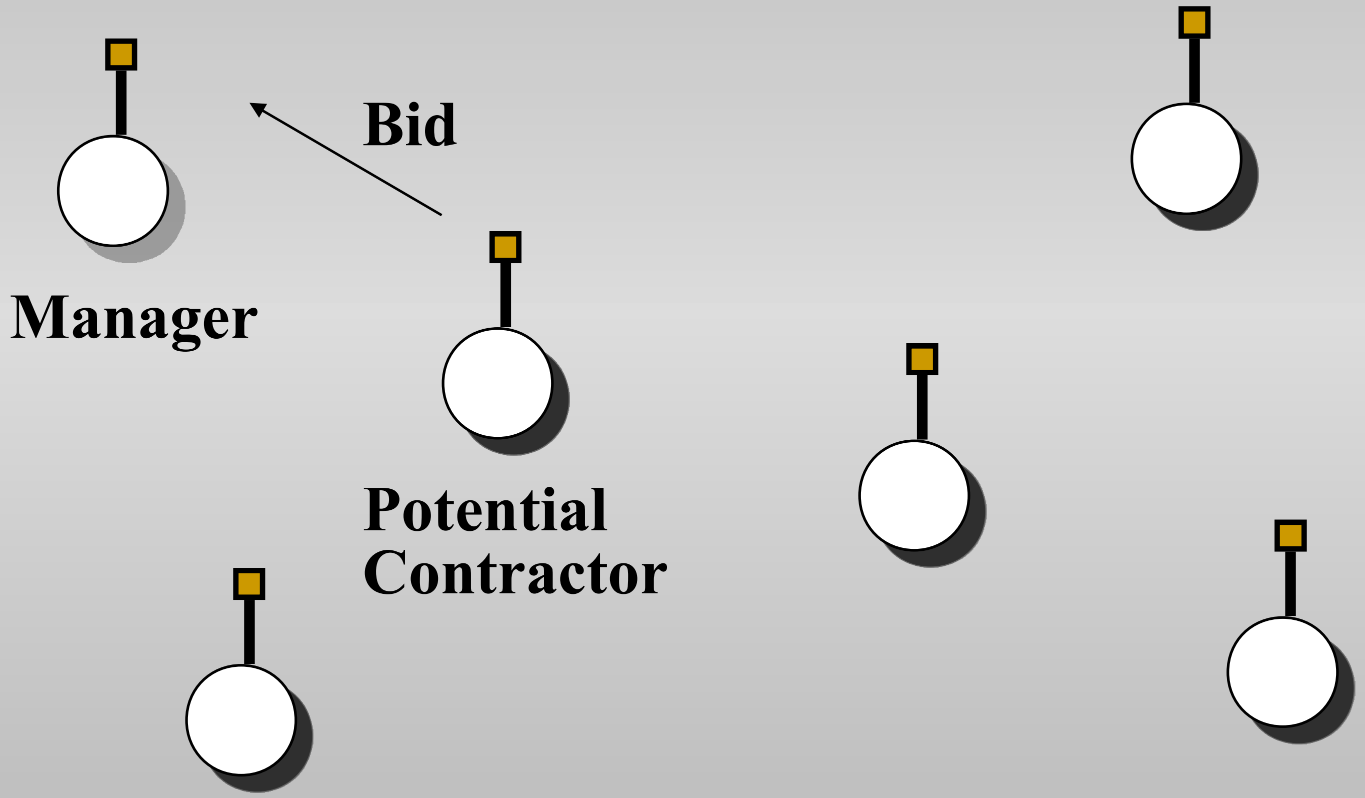
Manager Issues Task Announcement



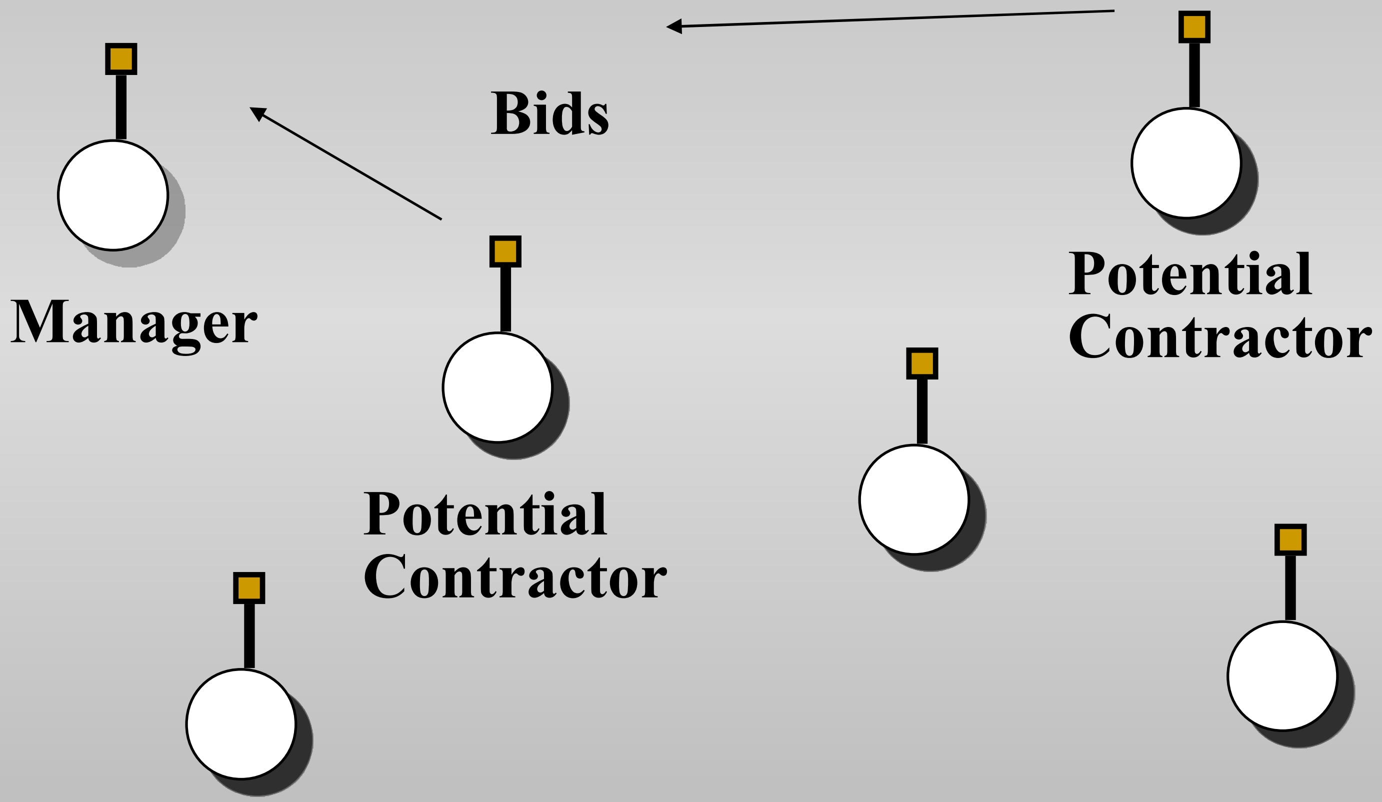
Idle Agents Listening to Task Announcements



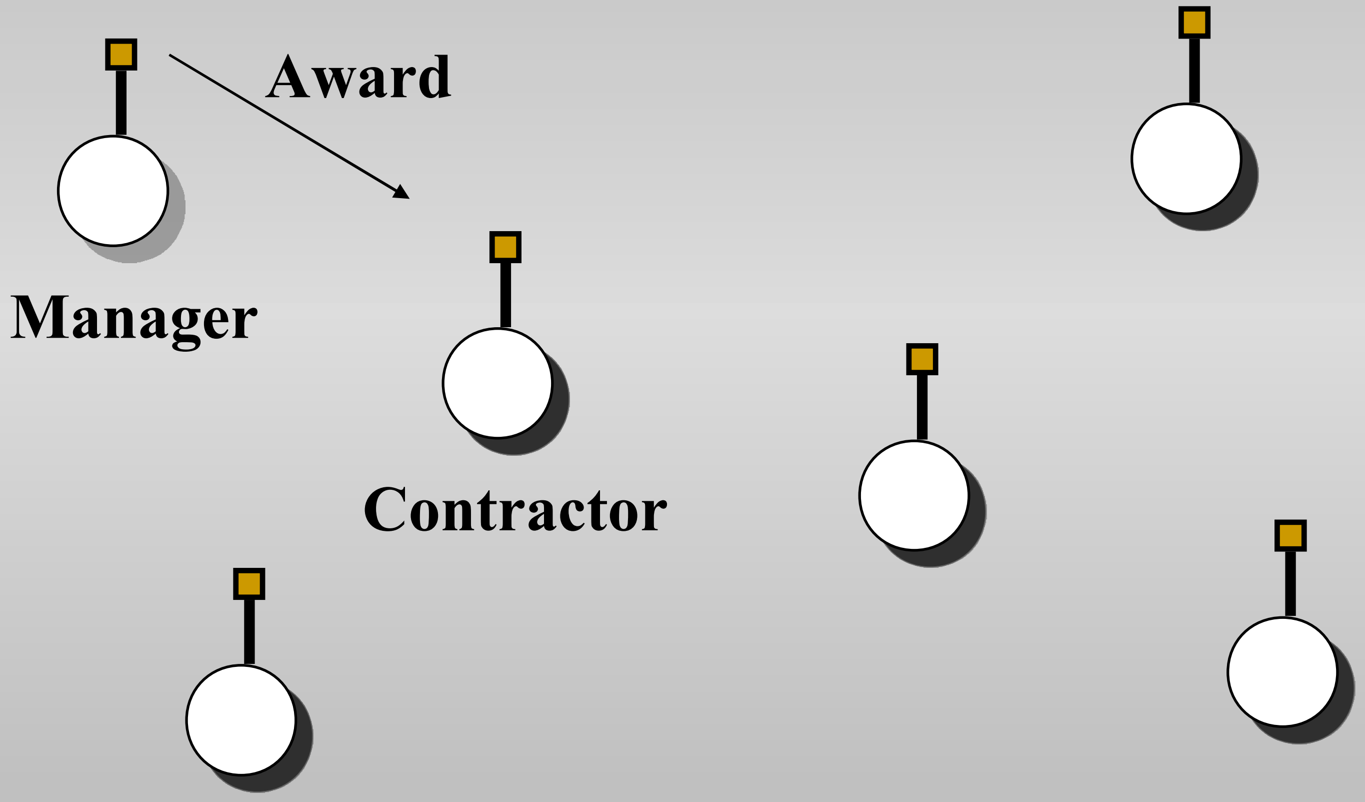
Contractor Submitting a Bid



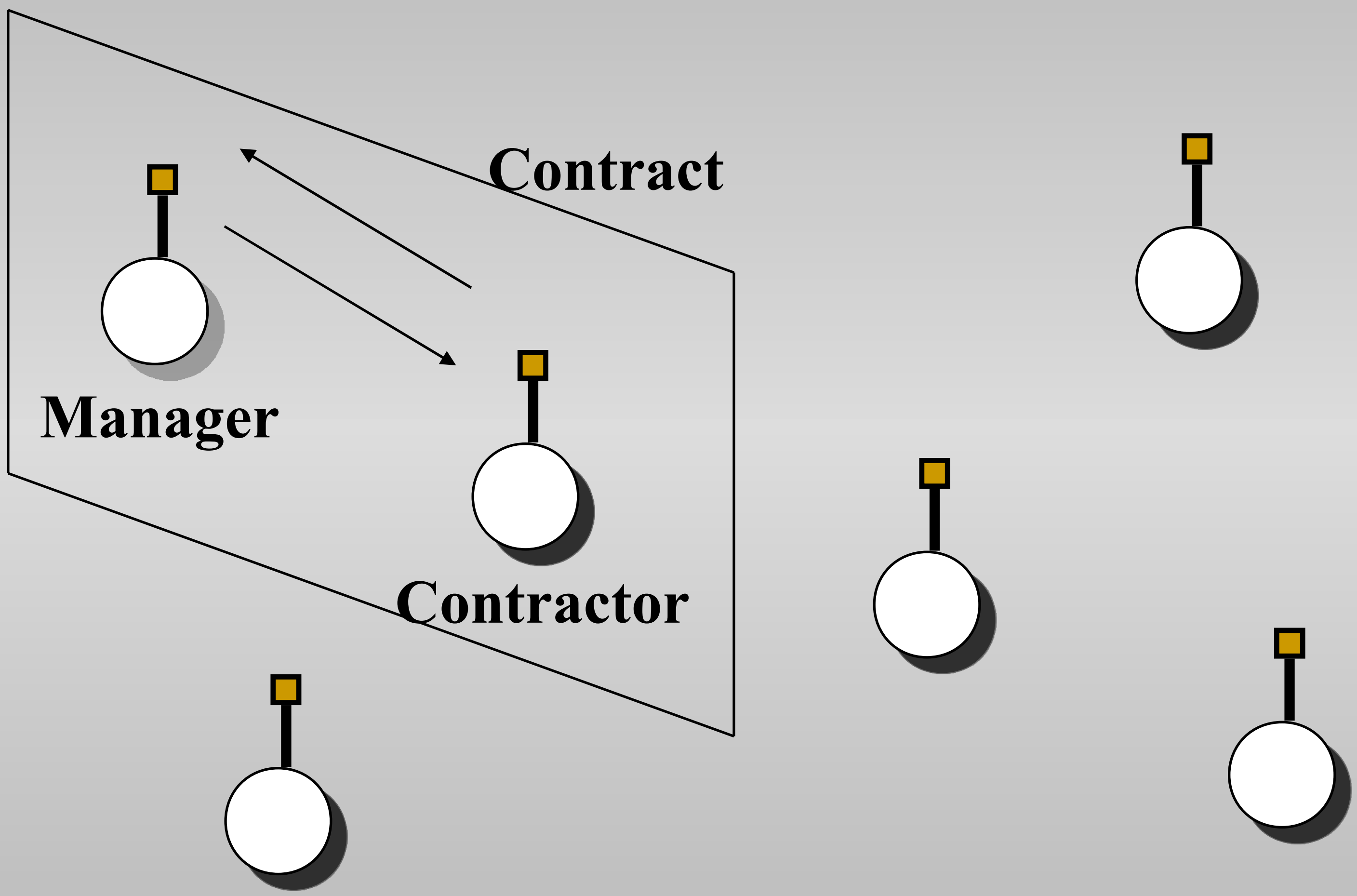
Manager listening to Bids



Manager Making an Award



Contract Established



Types of Messages

- Task announcement messages, with the following slots:
  - Eligibility specification
  - Task abstraction
  - Bid specification
  - Expiration time
- Bid (folowing the Bid specification)
- Award
- Interim report (on progress)
- Final report (including result description)
- Termination message (if manager wants to terminate contract)

Efficiency Modifications

- Depending on the MAS and the problem, many variations are possible:
  - Focused addressing — when general broadcast isn't required
  - Directed contracts — when manager already knows which node is appropriate
  - Request-response mechanism — for simple transfer of information without overhead of contracting

Other Protocols

- There is a variety of protocols:
  - Negotiation protocols:
    - One-to-one negotiation (Monotonic Bargaining, intertest-nased negotiation),
    - Multilateral negotiation (Auctions)
  - Argumentation
- There is also hybrid approches:
  - Micro-protocols
  - Dialogue games
  - ...

Argumentation

- Argumentation can be defined as an activity aimed at convincing of the acceptability of a standpoint by putting forward propositions justifying or refuting the standpoint.
- Numerous works:
  - Dialectic: structure of argumentation (acceptable arguments vs. fallacies) } **Syntax, structure and semantics**
  - Social psychology: attitude change and persuasion } **Pragmatics**
  - Artificial Intelligence **formalize** those approaches in the aim to:
    - Provide formal theoretical results about particular models
    - Automate agent or multi-agent argumentation capabilities

Argumentation

- Argument: Reasons / justifications supporting a conclusion
- Represented as: support ➔ conclusion
  - Informational arguments: Beliefs ➔ Belief
    e.g. If it is cloudy, it might rain.
  - Motivational args: Beliefs, Desires ➔ Desire
    e.g. If it is cloudy and you want to get out then you don't want to get wet.
  - Practical arguments: Belief, Sub-Goals ➔ Goal
    e.g. If it is cloudy and you own a raincoat then put the raincoat.
  - Social arguments: Social commitments ➔ Goal, Desire e.g. I will stop at the corner because the law say so.
    e.g I can't do that, I promise to my mother that I won't.

Argumentation

- Interactions (binary or collective) between arguments:
  - Conflict (defeat): e.g. attacks
    - Rebut (symmetrical):
      - support1 ➔ conclusion1
        - » e.g. Tweety is a bird ➔ tweety flies
      - support2 ➔ (not) conclusion1
        - » e.g. Tweety is a small bird ➔ tweety does not fly
    - Undercut (asymmetrical): defeat the assumptions or their link to the conclusion
      - support2 ➔ (not) support1
        - » e.g. no Tweety is not a bird, it is just a cartoon
  - Support-type interactions

Computational Models of Argumentation

- Given the definition of arguments over a content language (and its logic), the models allow to:
  - Compute interactions between arguments: attacks, defeat, support, …
  - Valuation of arguments: assign weights to arguments in order to compare them.
    - Intrinsic value of an argument
    - Interaction-based value of an argument
  - Selection of acceptable argument (conclusion)
    - Individual acceptability
    - Collective acceptability

} Computing the status of arguments according to various semantics

Applied Models of Argumentation

- Automating the argumentation processes and their effects
- Number of formal characterizations (along with their implementation) has been proposed for:
  - Inference (non monotonic reasoning): OSCAR, IACAS, BDKT, Nathan, DeLP, ABEL, …
  - Practical reasoning and decision making: PROforma, gIBIS, SIBYL, ZENO, HERMES
  - Argumentation dialogues in MAS: Artikis, Homey, PARMA, …
  - Argumentation and Machines learning: HYPO, IBP, …

Negotiation

The chalenge of negotiation:

⇨ **How to allocate *scarce* resources among agents representing *self-interested parties*?**

- Resources: bandwidth, commodities, money, processing power, …
- Scarce: competing claims cannot be simultaneously satisfied

Models of Negotiation

- Various models of negotiation have been proposed, based on:
  - Heuristic approaches (domain dependent formalization by experts)
  - Game theoretic approaches

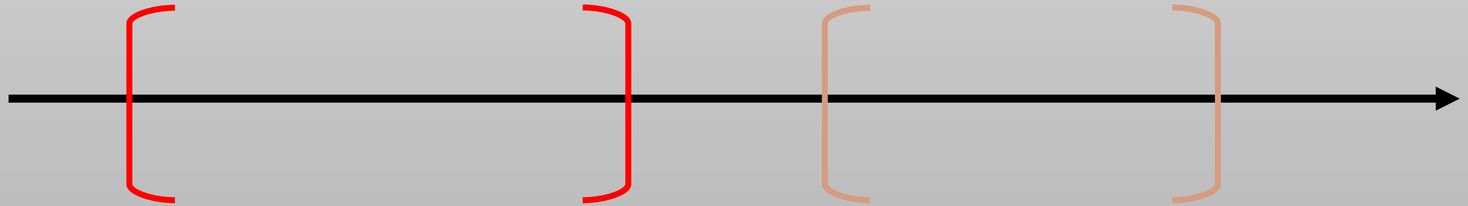
⇨ Accommodate the agents preferences

  - Argumentation based approaches:
    - Argue about negotiation-related issues (beliefs, goals, social aspects, …)
    - Interest Based Negotiation: argue about the underlying interests (making underlying goals explicit and discussing them)

⇨ Agent preferences may change

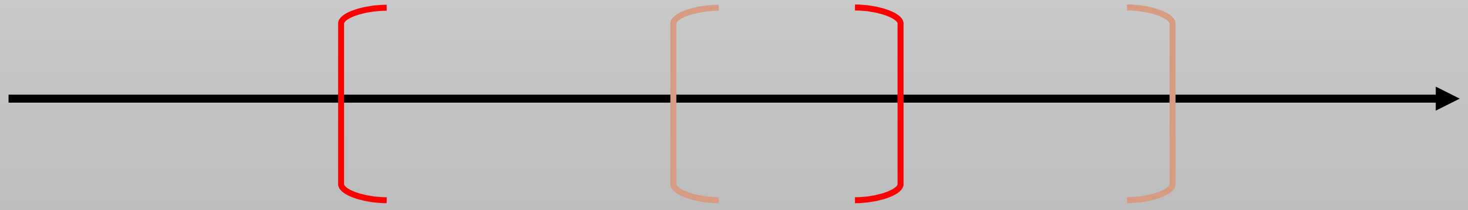
Example: non-IBN 1

- No deal!
  - B: I would like to rent a car for 4 days please.
  - S: I offer you one for $400.
  - B: I reject! How about $200?
  - S: I reject!



Example: non-IBN 2

- Deal after price concession
  - B: I would like to rent a car for 4 days please.
  - S: I offer you one for $400.
  - B: I reject! How about $200?
  - S: I reject! How about $300 then?
  - B: I guess that's the best I can do! I accept!



Example: IBN

- Deal after discussion of interests
  - B: I would like to rent a car for 4 days please.
  - S: I offer you one for $400.
  - B: I reject! How about $200?
  - S: I reject! Why do you need the car?
  - B: I want to drive to Sydney to attend a conference.
  - S: You can also fly to Sydney! I can book you a ticket with Qantas airlines for $200.
  - B: I didn't know flights were so cheap! I accept!

Auctions

- Single good auctions:
  - English auctions: auctioner set starting price, agents annouce raising bids. Auctions ends after a fixed time or a fixed period without bids.
  - Japanese auctions: ascending auction in which the agents decide to stay in of not at each step. The last agent in gets the good.
  - Dutch auctions: descending auction. The auction ends when an agents stops the auctioner
  - Sealed-bid auctions: unlike open-outcry auctions, agent submit secret bid to the auctioner. The agent with the highest bid can pushase the good (for the anounced price, first price auction or second price auction – Vickrey auction)

?

“Inform all the troops that communications have completely broken down”

Ashleigh Brilliant

Assignments

- Possible topics for final project:
  - Developing a new metacreation
  - Developing a metacreative module in an existing application
  - Conducting experiments for the validation of an existing system
  - Theoretical topic
  - State of the art (including classification and review of existing work and underlying technics used):
    - Cognitive Agents in musical metacreation
    - Reactive Agents in visual metacreation
    - Metacreation in Dance
    - Metacreation in Architecture...

Assignments: for next Week

- **Readings:**
  - Brooks, R. A. "A Robust Layered Control System for a Mobile Robot" , IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986, pp. 14–23;
  - OPTIONAL, Brooks, R. A., "Elephants Don't Play Chess" , Robotics and Autonomous Systems (6), 1990, pp. 3–15. (Available from the authors Web Page).
  - OPTIONAL, Luc Steels: Fifty Years of AI: From Symbols to Embodiment - and Back. In 50 Years of Artificial Intelligence, Lecture Notes in Computer Science, Volume 4850, 2006: 18-28
  - Hutzler, G., Gortais, B. From Computer Art to Ambient Displays. Machine GRAPHICS & VISION (MGV), Volume 13, Number 1/2, page 181--191 - 2004 (available online).
- Thoughs about the final project:
  - One page about your final project (PDF)
  - We will use the ACM format.

?

“Before we work on artificial intelligence why don't we do something about natural stupidity?”

Steve Polyak