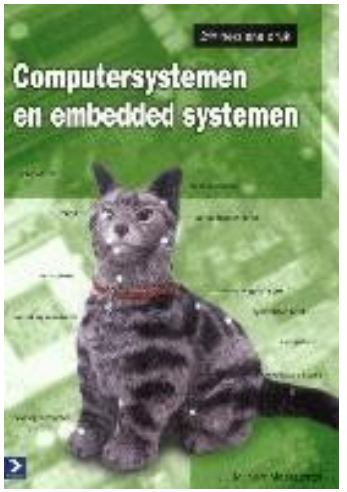
Agent Technology and IoT

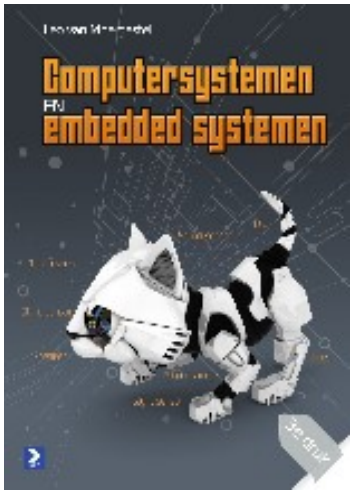
Leo van Moergestel
HU Utrecht University of Applied Sciences
Utrecht, the Netherlands

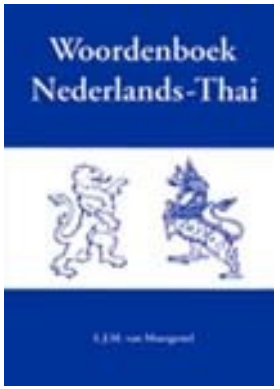
Let me introduce myself

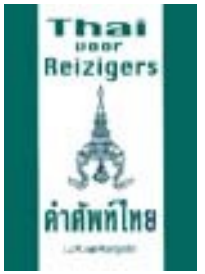
- Master degree in physics Utrecht University
- PhD in computer science
- Associate professor at the HU Utrecht University of Applied Sciences

Some of my books









Overview

- Part 1: Overview agent technology
- Part 2: Internet of things
- Part 3: Agent-based IoT

Overview Part 1

- Agents types
- Cooperation
- Negotiating
- Languages, pitfalls, standards etc.

What is an agent? (1)

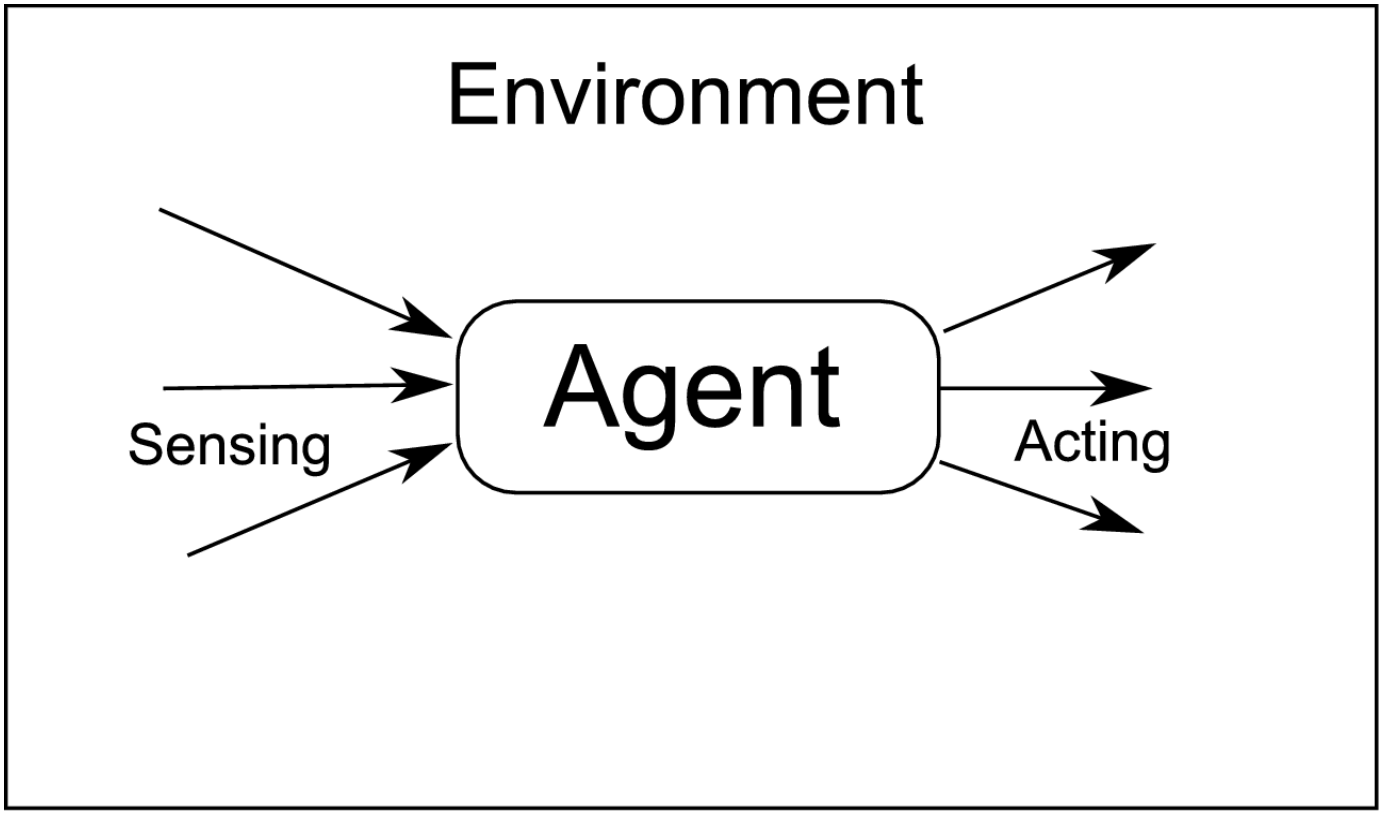
- Definition by Wooldridge and Jennings:

"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* design objectives*"*

What is an agent? (2)

- Autonomous
- Goal(s)
- Role(s)
- Plans
- Environment
- (Representative)
- Act (agere (Lat.), react)

agent



Processes and objects

- Object are passive
- An object will accomplish a method if it is called by another object. An agent checks to see if the request fits its goals
- Processes (general term for programs in execution phase)
- Process is not necessary autonomous or interacting with the environment

Applications

- Web bots
- Robotics
- Computer played character in games
- Malware
- Auctions
- Swarm intelligence
- Simulations of behaviour in groups

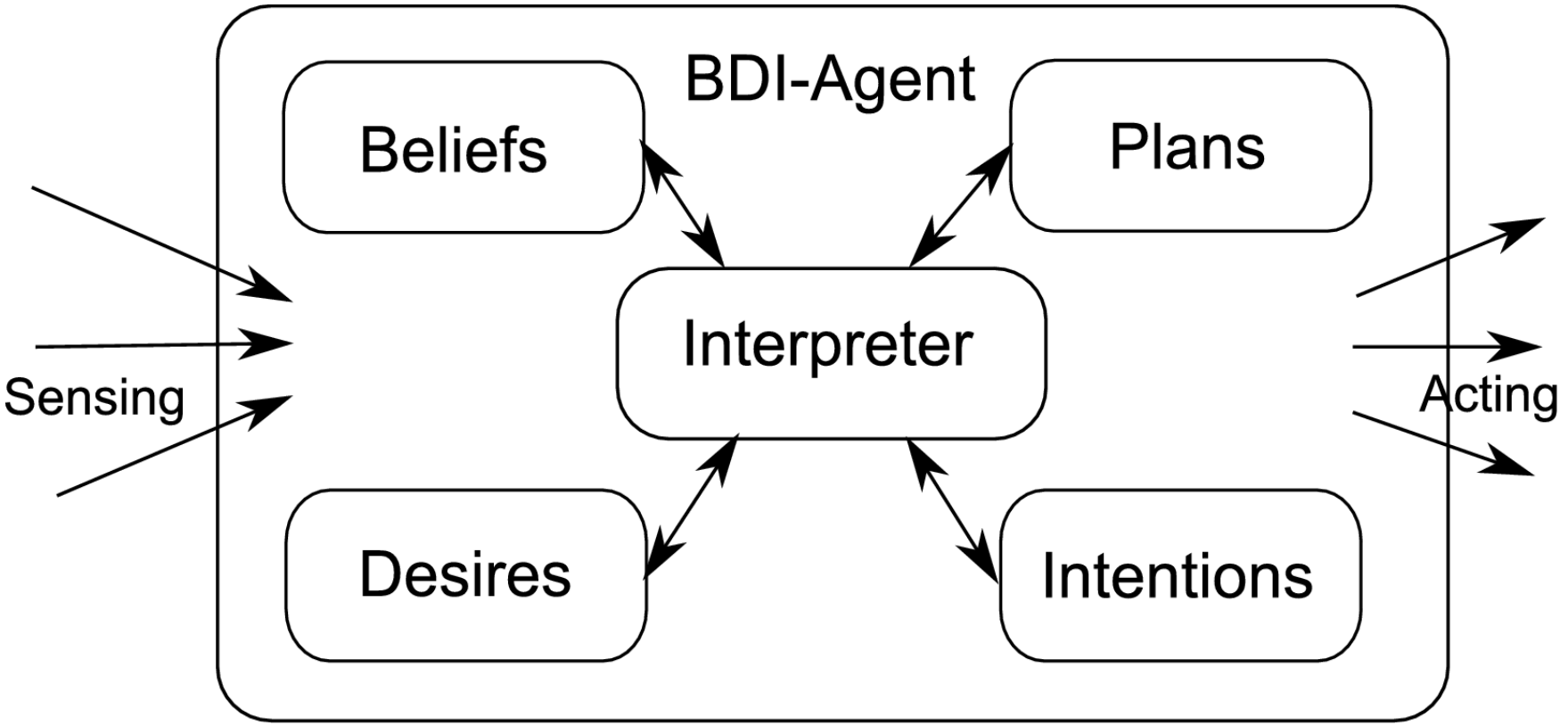
Agent types

- Reactive agents (reflex agents)
- Reactive agents with state
- Goal based agents
- Utility based agents

BDI-agent

- Belief Desire Intentions: The beliefs, goals, desires and intentions could be viewed as the mental states of a BDI-agent. (based on Michael Bratman's theory of human practical reasoning)
- From the inputs of its sensors the agent builds a set of *beliefs*. Beliefs characterize what an agent imagines its environment state to be;
- *Desires* describe agents preferences;
- *Intentions* characterize the goals or desires the agent has selected to work on.

BDI-agent


BDI-Agent
Beliefs
Plans
Interpreter
Desires
Intentions
Sensing
Acting

Computing perspective

- Computational power
- Interconnection
- Distributed (in case of Multiagent Systems)
- Intelligence
- Delegation of control
- Human-orientation

Software engineering perspective

- Procedural methodology
- Object-oriented methodology
- Agent-oriented methodology

- Multiagent system (MAS) is a new software engineering paradigm

Artificial intelligence perspective

MAS and AI are NOT the same

Understand and model social intelligence and emergent behavior

- AI
  - Planning, Learning, Vision, Language understanding
- MAS
  - Interaction, Communication, Obligation, Norms, Responsibilities, Co-ordination, Co-operation

Characteristics of MAS

- MAS consists of a number of interacting autonomous agents
- MAS are designed to achieve some global goal
- MAS are specified in terms of high-level abstract concepts such as role, permission, responsibily and interaction
- MAS can be used to implement distributed systems

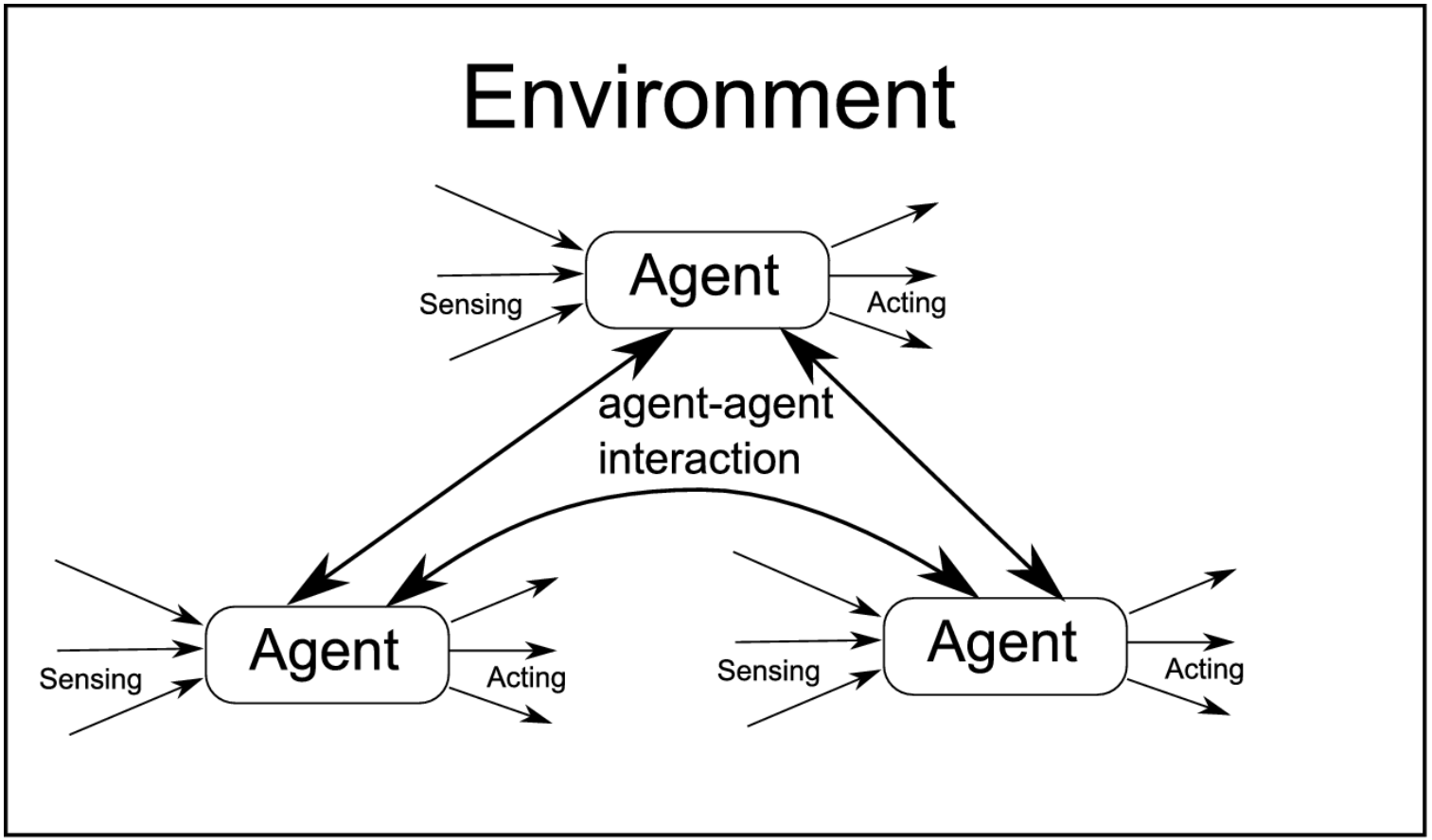
Some concepts used by agents

- Utility
- Reasoning
- Environment
- Communication
- Learning
  - Supervised learning, training
  - Re-inforcement learning

Multi-agent systems (1)

- A multi-agent system (MAS) consists of two or more interacting autonomous agents.
- Such a system is designed to achieve some global goal.
- The agents in a multi-agent system should cooperate, coordinate and negotiate to achieve their objectives.

Multi-agent systems (2)



Multi-agent Systems(3)

- *Role:* what is the role of a certain agent in a multi-agent system. Perhaps an agent has more than one role;
- *Permissions:* what are the constraints the agent is tied to;
- *Responsibility:* i.e. the responsibility an agent has in achieving the global goal;
- *Interaction:* agents interact with each other and the environment

Cooperation

- Pay-off matrix
- Nash equilibrium
- Pareto efficiency / optimum
- Maximizing social welfare
- Prisoners dilemma

Game Theory

- Two persons zero sum game
  - Chess (1-0, 1/2-1/2, 0-1)
  - Minimax theorema (von Neumann, Morgenstern)
- Other types of games
  - John Forbes Nash

Pay-off Matrix

- Consider two agents i and j
- Agent i can choose to do either A or B and agent j can chose between C or D
- Build a matrix where all combinations or outcomes with their utilities for both agents are available

- In many situations agents can choose to defect or to cooperate (prisoners dilemma)

Pay-off Matrix(2)

	i A	i B
j C	$u_i(A,C)$ $u_j(A,C)$	$u_i(B,C)$ $u_j(B,C)$
j D	$u_i(A,D)$ $u_j(A,D)$	$u_i(B,D)$ $u_j(B,D)$

Head-tail game

	i head	i tail
j head	1 -1	-1 1
j tail	-1 1	1 -1

two coins
if same
    i wins
else
    j wins

Rock paper scissor

	i rock	i paper	i scissor
j rock	0 0	1 -1	-1 1
j paper	-1 1	0 0	1 -1
j scissor	1 -1	-1 1	0 0

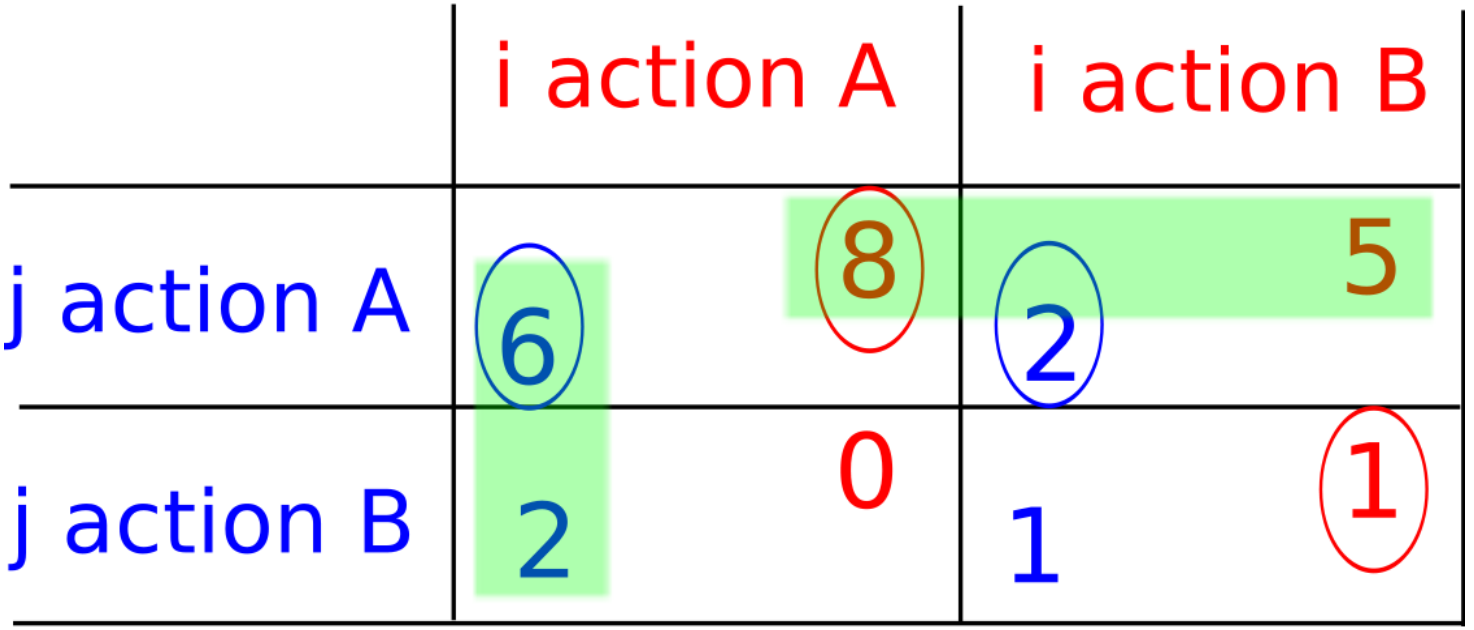
Nash equilibrium

- A nash equilibrium is a place in the pay-off matrix where in the same row there is no higher utility for agent i and in the same column no higher utility for agent j (but there are more situations for a nash equilibrium).
- Find the equilibrium by placing circles around the maxima. If there is a place in the matrix with circles around both utilities then this is the nash equilibrium.

Nash equilibrium (2)

	i action A	i action B
j action A	6, 8	2, 5
j action B	2, 0	1, 1

Nash equilibrium (3)



Video Intermezzo

- TED presentation Nash Equilibrium

  https://www.youtube.com/watch?v=jILgxeNBK_8
- Part from “a beautiful mind”

  https://www.youtube.com/watch?v=2d_dtTZQyUM&spfreload=10

Prisonners Dilemma

- Two prisonners, separately kept in jail.
- If both deny: small penalty (6 months).
- If both confess: penalty 5 years in prison for both prisoners.
- If i confesses but j denies: i is free because of cooperation, but j severely punished (10 years).
- If j confesses but i denies: j is free because of cooperation, but i severely punished (10 years).
- What would you do?

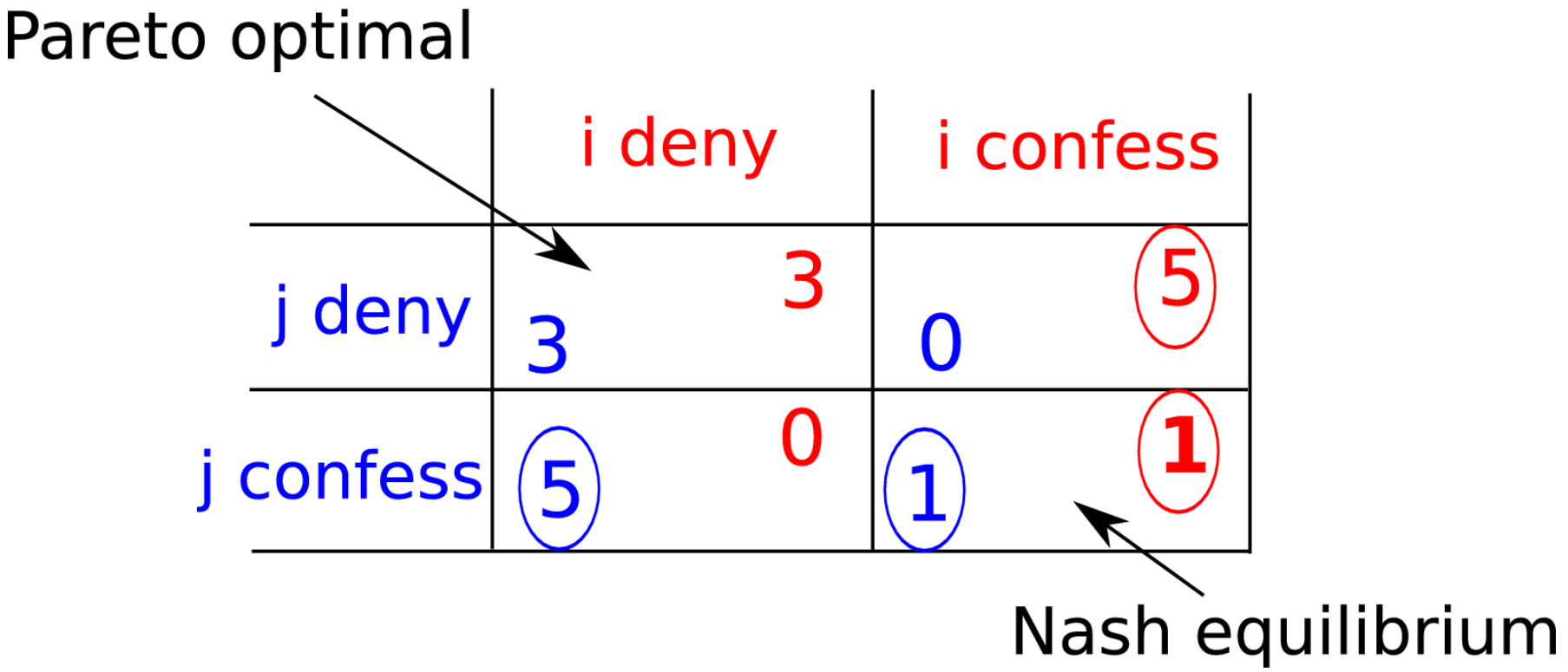
Prisonners Dilemma (2)

	i deny	i confess
j deny	3 3	5 0
j confess	0 5	1 1

Prisonners Dilemma (3)

	i deny	i confess
j deny	3 3	0 5
j confess	5 0	1 1

Prisonners Dilemma (4)



Negotiation

- Auctions
  - English auction
  - Dutch auction
  - First-price sealed-bid auction
  - Vickrey auction
  - Chinese auction
- Types
  - Sealed bid / Open cry
  - One round / multiple round (one shot / ascending, descending)

Negotiation (2)

- Common value (“real value”)
- Private value
- Correlated value

- Winner determination
  - First price auction
  - Second price auction

Negotiation (3)

- English auction
- First price, open cry, ascending auction
  - Start with reservation price (may be 0)
  - Bids are invited from agents
  - Bid should be more than the current highest bid
  - When no agents is willing to raise a bid, the good is allocated tot the agent with the current highest bid. This agents pays the price of the highest bid
- Winners curse and bogus bidders (shills)

Negotiation (4)

- Dutch auction
- First price, open cry, descending auction
  - Start with artificially high price
  - Auctioneer continually lowers the price until some agent makes a bid
  - Bid is the current price
  - The good is allocated tot the agent with the current bid. This agents pays the price of the current bid
- Winners curse

Negotiation (5)

- First-price sealed-bid auctions
- One shot auction, one round, alle agents interested in the good submit a bid to the auctioneer
- Good is allocated to the highest bidder
- Difference between the highest price and the second highest price is in effect wasted money for the winner

Negotiation (6)

- Vickrey auction
- Second price sealed bid
- Good is allocated to the highest bidder
- Highest bidder pays the price of the second highest bid
- Bidders dominant strategy is to bid the true value (i.e. private value)

Negotiation (7)

- Chinese auction
- Auctioneer sells tickets for a lottery
- Amount of tickets is fixed as well as the price
- When all tickets are sold, a ticket number is drawn
- Good is alocated to the buyer of the winning ticket
- Stategy is to buy many tickets if the good has a high private value

Babylonian Wife Auction

Description originates from Herodotus (Greek historian 484-425 B.C.)

Annually held marriage auctions

English auction type, however …..

Agent programming languages

- No current standard
- Single or multiagent environment
- Mostly Java based
  - Platform independence
  - Installed base
- Software design tools and methods (AOP, Gaia, Prometheus)

Agent Programming Languages

- Jade
  - MAS
  - Multiplatform
- 2APL (BDI, Jade based)
- Jadex (BDI extensions to Jade)
- Jason (BDI language derived form AgentSpeak)

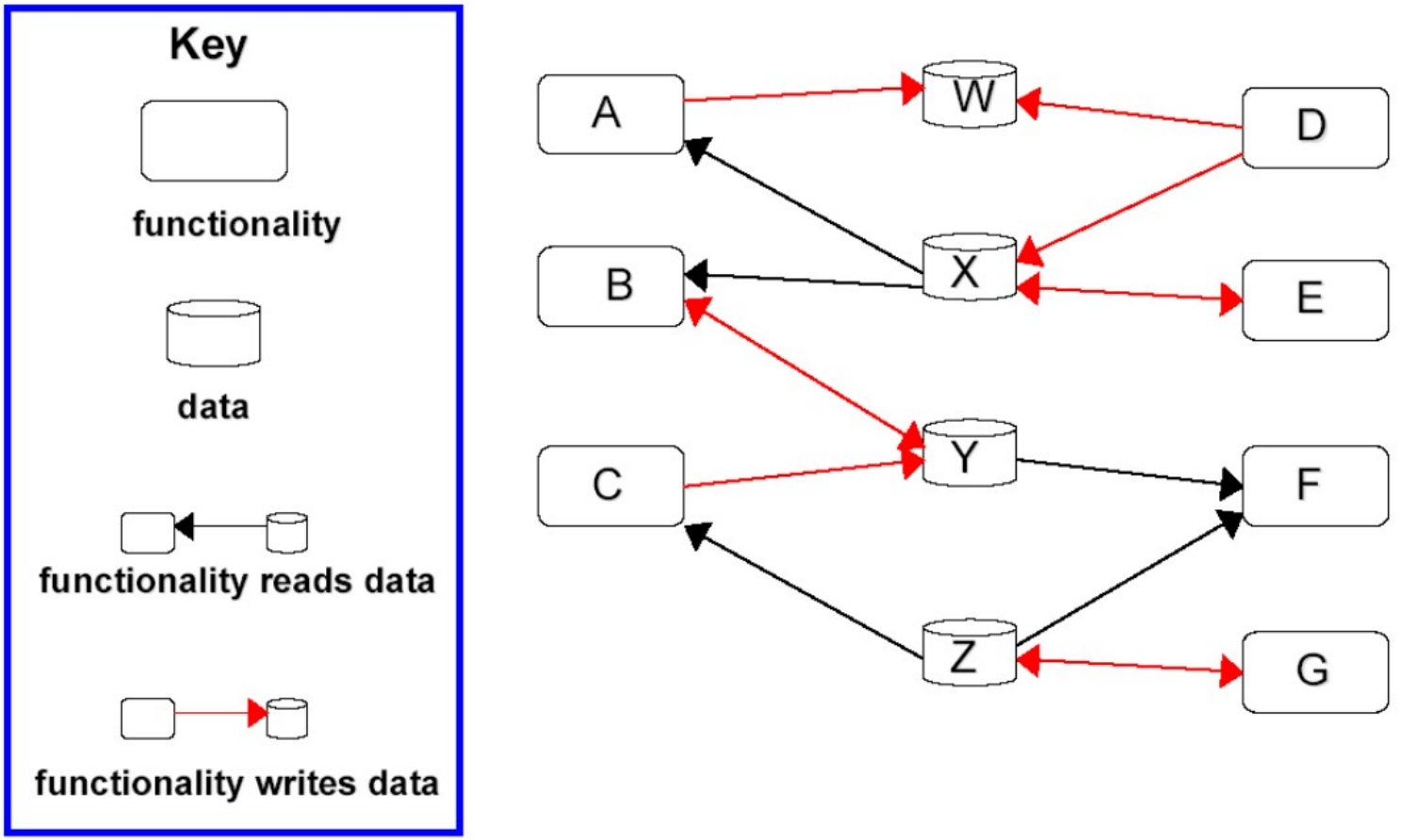
Pitfalls of agent technology

- Overselling, religious attitude towards agents
- Don't know what, why ...
- Confuse prototypes with real systems
- It is distributed software development
- You spend all your time implementing infrastructure
- You ignore de facto standards

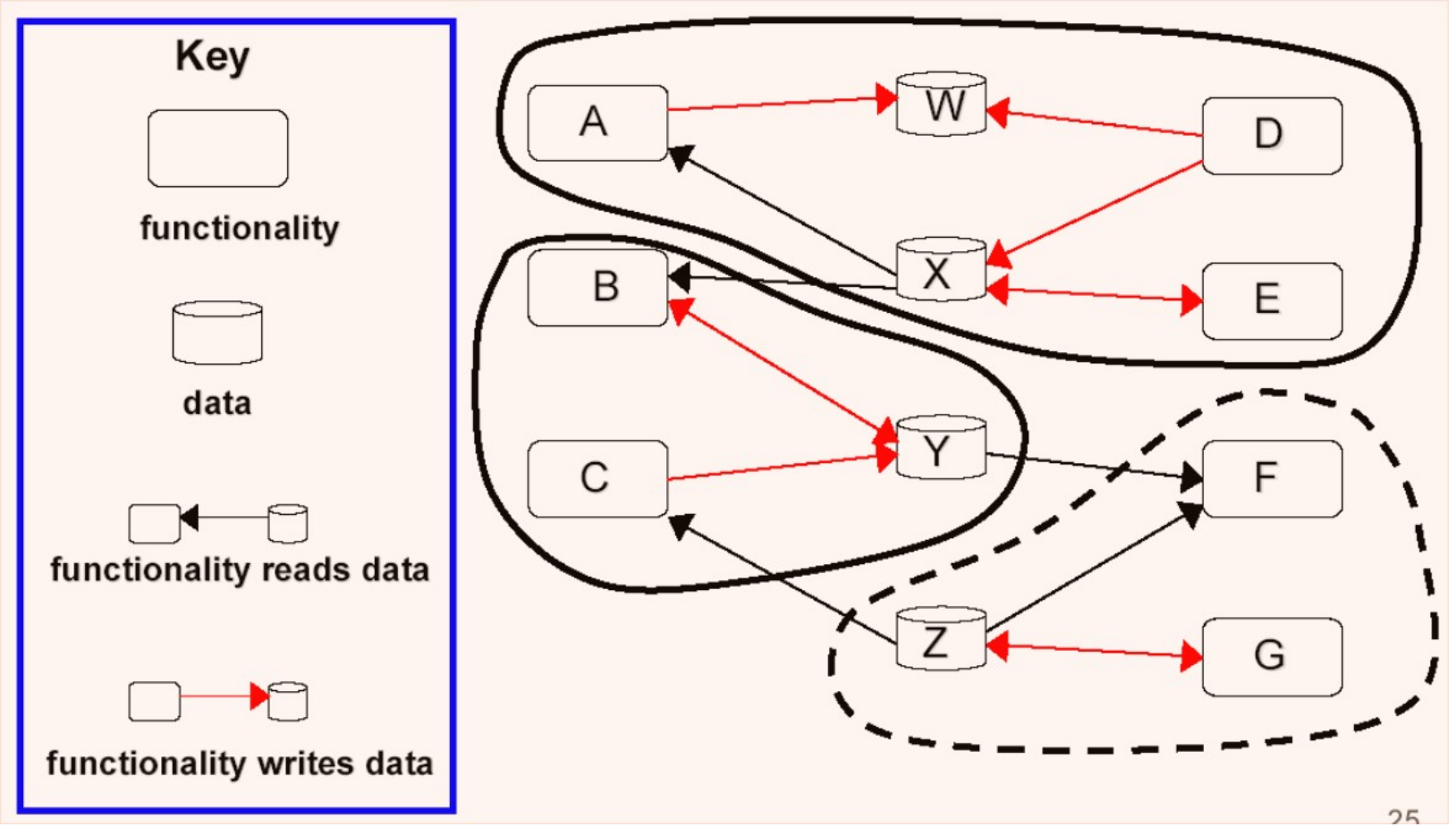
Data sharing

- Peer to peer communication
- Blackboard (mutual exclusion during updates)
- Subscribe/Notify Pattern

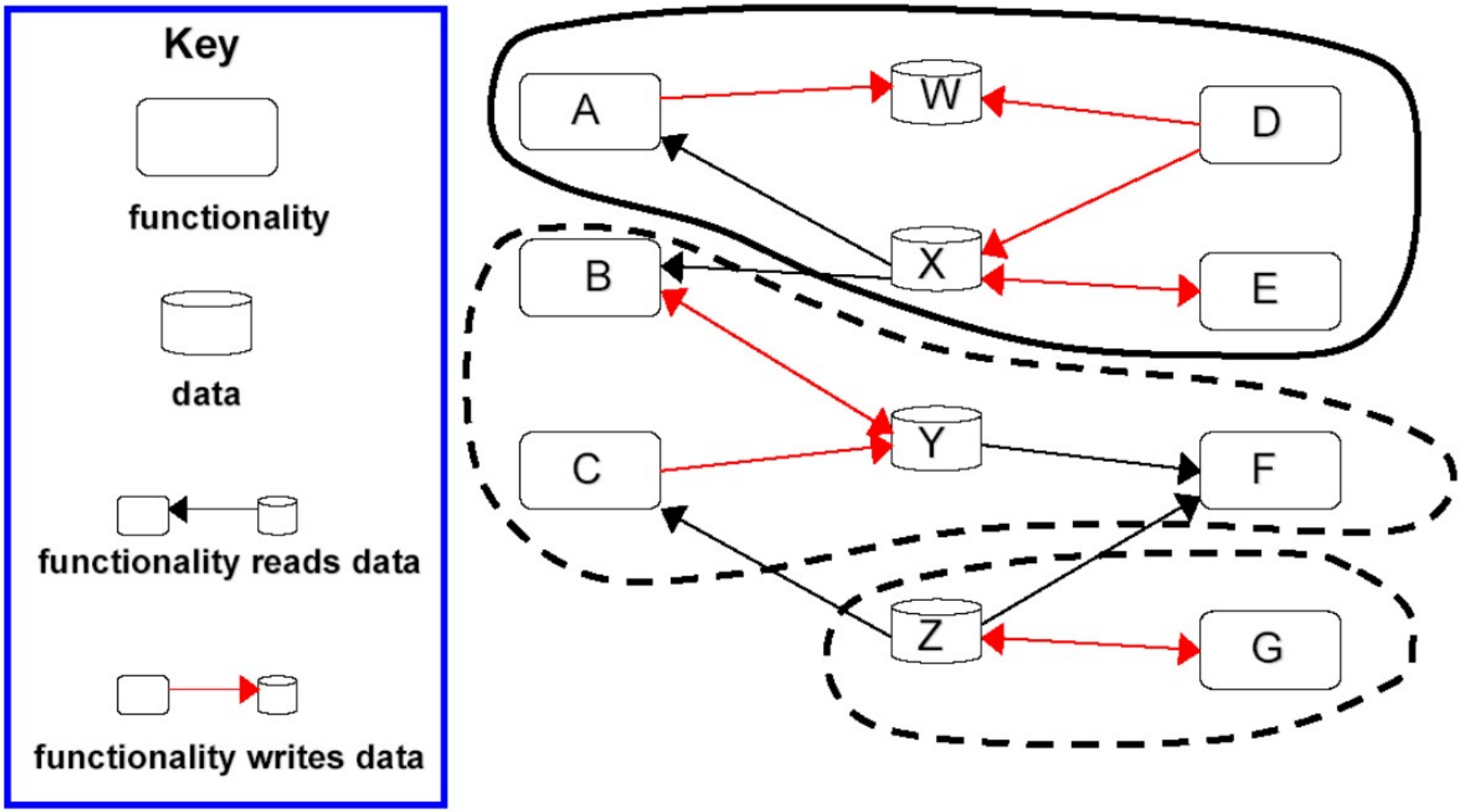
Data coupling diagram



Data coupling diagram



Data coupling diagram



FIPA

- Foundation of Intelligent Physical Agents
- Structure of messages in Agent Communication Language
  - Performative (see next slide)
  - Meta information (sender, receiver, language..)
  - Actual content

FIPA

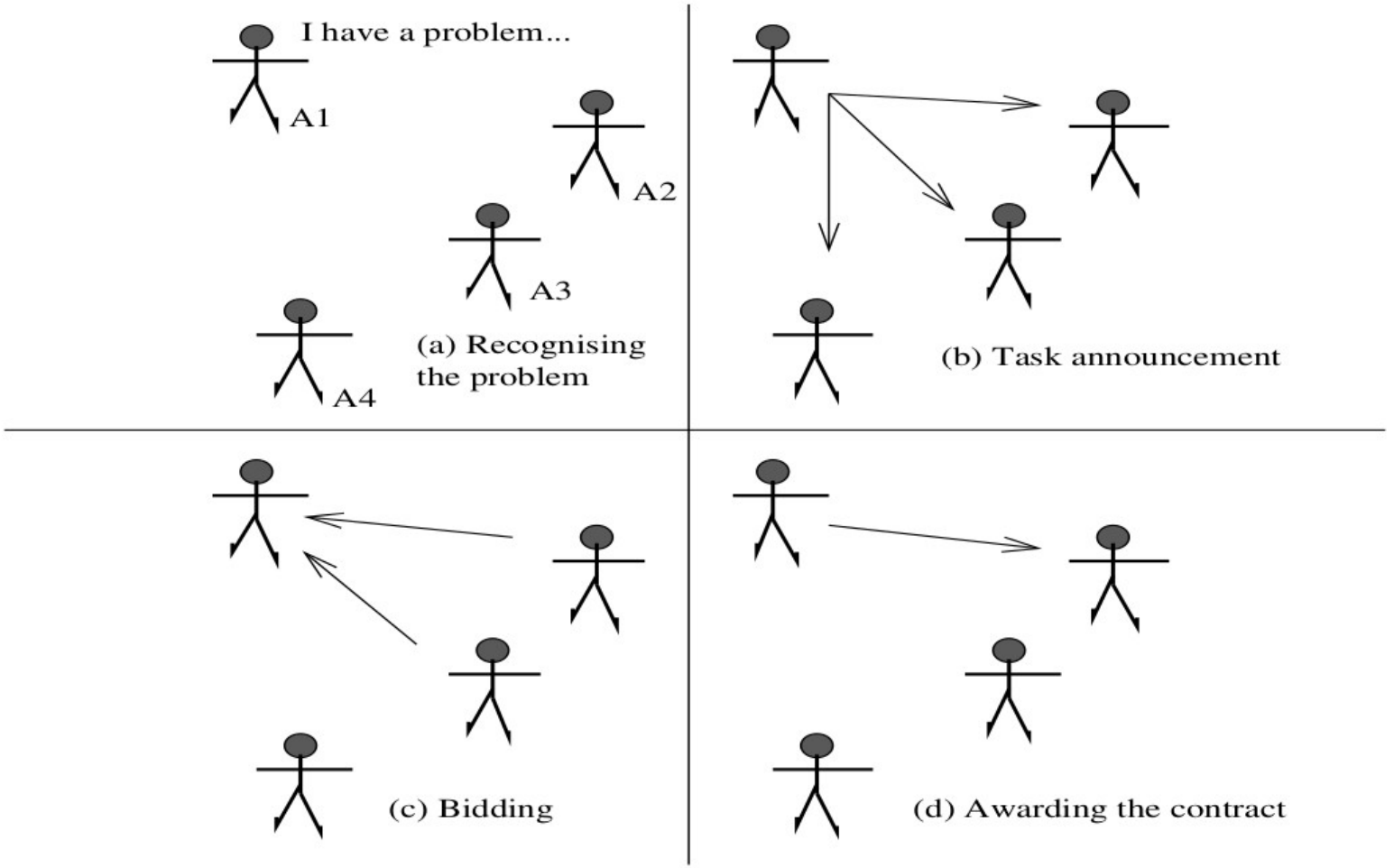
```
(inform
   :sender          agent1
   :receiver        agent5
   :content         (price good200 150)
   :language        sl
   :ontology        hpl-auction
)
```

FIPA performative

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			

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

Contract Net Protocol



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

Overview Part 2

- IoT classification
- Organisation and administration
- Type of connection
- Computing capabilities

Organisation and administration

- Static: configuration before use
- Central administration on a single administration system (sort of yellow pages)
- Central administration with a set of administration systems (local yellow pages)
- Autonomous decentralised systems

Type of connection

- Device is a node in a special type of network (wired or wireless). A gateway can be used to connect the special network to the Internet.
- Device is directly connected to the Internet, but has limited possibilities. It can only send data on request.
- Device is directly connected to the internet and can play an active role.

Computing capabilities

- The device is running a single program (a single thread, eventloop or state machine). Local storage is limited.
- The device is running an operating system. Complex software solutions are possible if the hardware can support it.

Overview Part 3

- Design phase
- Manufacturing phase
- Distribution phase
- Usage phase including maintenance and repair
- Recycling phase

General concept

- A life cycle agent is added to a product.
- The agent is the linking pin to the Internet
- In combination with the agent, the device will become a part of the IoST (Internet of Smart Things)
- In the cloud a copy of the agent is available
- Information can be generated and used in different phases

Design and manufacturing

- The life cycle agent will be a part of the design
- The role of the agent during production is guiding a product along the production cells and collecting manufacturing data.
- At the end, the agent will embed itself in the product hardware.

Distribution

- If possible, the embedded agent can support the distribution.
- All kind of environmental conditions (shock, temperature, pressure etc.) can be monitored by sensors and collected by the agent.

Usage

- The agent can be the interface to the device
- Hints, manual, status can be made available to the end user.
- Usage data can be collected.
- Redesign hints can be reported to the manufacturer
- Wearing of components can be monitored

Example: vacuum cleaner

Actual implementation

Repair

- Marketplace for spare parts
- Reuse of parts
- 3D printing info for parts

Recycling

- Economic value and life expectancy for parts
- Reuse of rare materials (rare earth elements, expensive elements). Urban ore
- Feedback to manufacturer: what are the weak parts in the design.

Urban ore

Points of concern

- Security
- Reliability of the system and data
- Privacy
- Ownership of data

Thank you for your attention
Questions?