MASTER WEB INTELLIGENCE

Multi-Agent Systems

Agent Architectures

Cosmin Carabelea

Cosmin.Carabelea@emse.fr

SMA/G2I

ENS Mines Saint-Etienne

November 22nd, 2004

Motivation

- **Agents** are used to solve problems.
- The characteristics of the problem influence the way the agents used to solve it are built (their **architecture**).
- It may be the case that some architectures are designed using general principles → we then talk about agent **models.**
- Some of these models have a **theory** associated with them that allows the verification of some properties.

This course...

- ...is about some of the most well-known agent architectures proposed until now.
- We will use a case study to illustrate the differences between these architectures:
  - agents must execute tasks
  - each task has associated a duration of execution, a deadline and a reward
  - an agent needs an amount of time before beginning to execute a task.
- This is a generic problem that will be detailed progressively

Structure

- Individual Agents
  - agents that reason about themselves and about their environment
- Social Agents
  - agents that reason about themselves, about their environment and about the interactions with others
- Organized Agents
  - agents that reason about themselves, about their environment and about the interactions with others and about the social structures enforcing these interactions

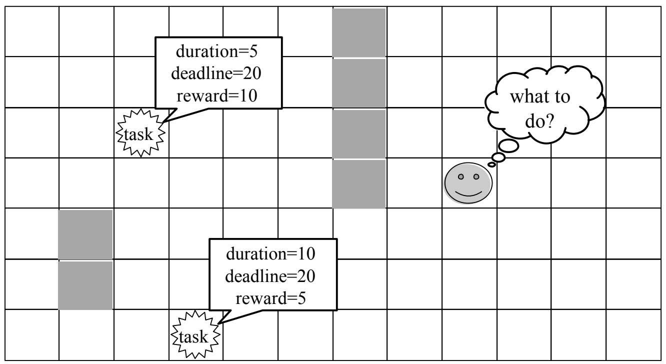
Outline

- Individual Agents
  - reactive agents: the subsumption architecture
  - deliberative agents: the BDI model and the PRS architecture
  - hybrid agents: Touring Machines
- Social Agents
- Organized Agents

Individual agents

- Reason about themselves and about their environment
- We need to model the environment (subject of the Environment course)
- Our case study:
  - the agents move on a 2D grid
  - there are obstacles blocking their movements
  - an agent should find a path to a task, to execute it, and then to move on to another task
- Note: movement on a grid stands for real movement (e.g., robots) or virtual movement (e.g., searching on Internet)

Case study – individual agents



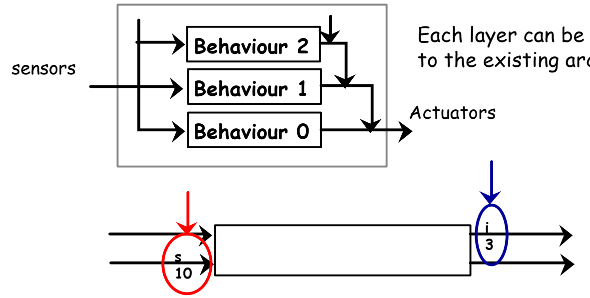
Reactive agents – the subsumption architecture

[Brooks 86]

Subsumption architecture

- Agent's decision making is realised through a set of tasks accomplishing behaviours.
- A behaviour continually takes perceptual inputs and maps them to an action to perform (finite state machines, no symbolic reasoning, no symbolic representation)
- Many behaviours can fire simultaneously. In order to choose between them, use of a subsumption hierarchy, with the behaviours arranged into layers.
  A higher layer has priority on lower layers (inhibition)

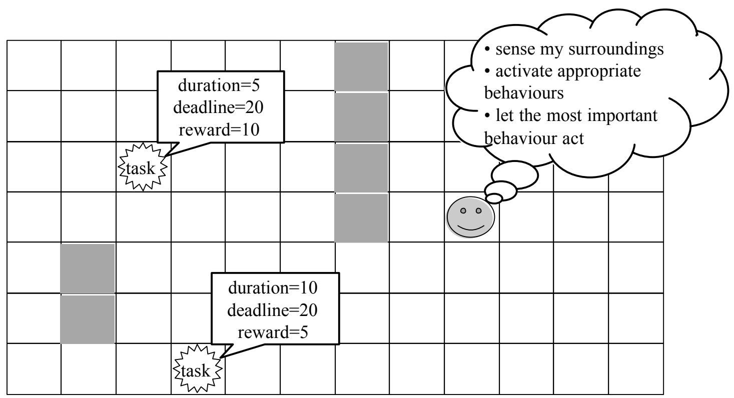
Subsumption architecture (2)



Each layer can be incrementally added to the existing architecture.

Each layer is a set of modules (FSM) which sends messages to each other without central control.
Inputs to modules can be suppressed and Outputs can be inhibited by wires terminating from other modules for a determined time. (subsumption)

Case study – subsumption arch.



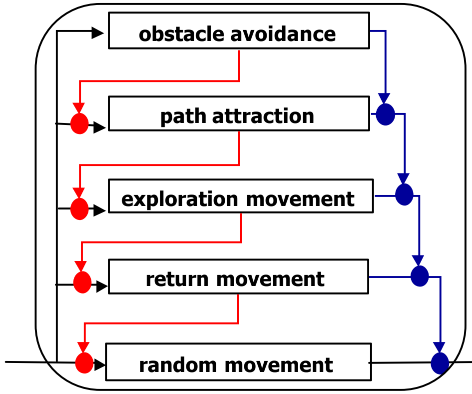
Subsumption architecture (cont.)

- Does it work? The agents are very simple, there is no symbolic reasoning or representation of their environment...
- It works if there are many agents: "the intelligence is in the system, not in the entities composing it".
- [Steels 89] used this architecture in a scenario very similar with our case study:
  - robots have to collect samples of precious rock (unknown location) and bring it back to a mothership spacecraft.
  - cooperation without direct communication : through the environment.
  - ➔ gradient field with a signal generated by the mothership
  - ➔ radioactive crumbs are picked up, dropped and detected by robots.

Example: Distributed Robots

- **Two sets of behaviours running in parallel:**
  - Handling behaviour
    - If I sense a sample and I don't carry one, I pick it up.
    - If I sense the vehicle-platform and I carry a sample, I drop it.
    - If I carry a sample, I drop 2 crumbs.
    - If I carry no sample and crumbs are detected, I pick up one crumb.
  - Movement behaviours organized along a subsumption hierarchy

Example: Distributed Robots (2)



If I sense an obstacle in front, I make a random turn.

If I am not carrying a sample and I sense crumbs, I move towards the highest concentration of crumbs.

If I am in exploration mode, I choose the direction of lowest gradient.

If I am in return mode, I choose the direction of highest gradient.

Choose randomly a direction to move. Move in that direction.

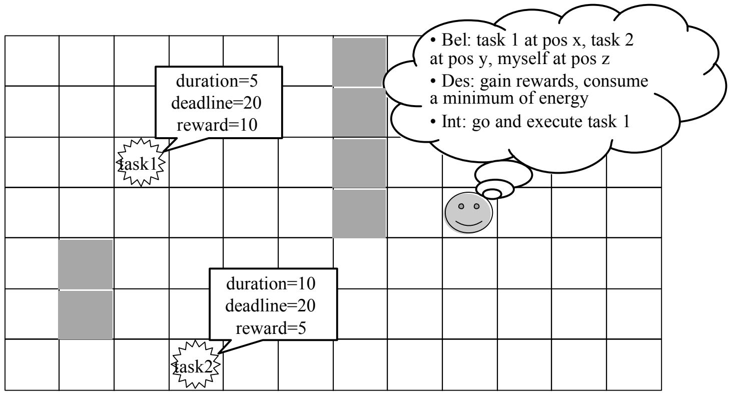
Deliberative (BDI) agents – the PRS architecture

[Georgeff, Lansky 87]

BDI Model

- [Georgeff 83], [Bratman 90] – the use of intentions in agent's design.
- [Rao, Georgeff 91] – the BDI model: an agent contains:
  - a set of beliefs about itself and the world;
  - a set of (possibly conflicting) desires
  - a set of non-conflicting intentions
  - reasoning mechanisms to update its beliefs, choose the desire(s) to pursue and generate new intentions

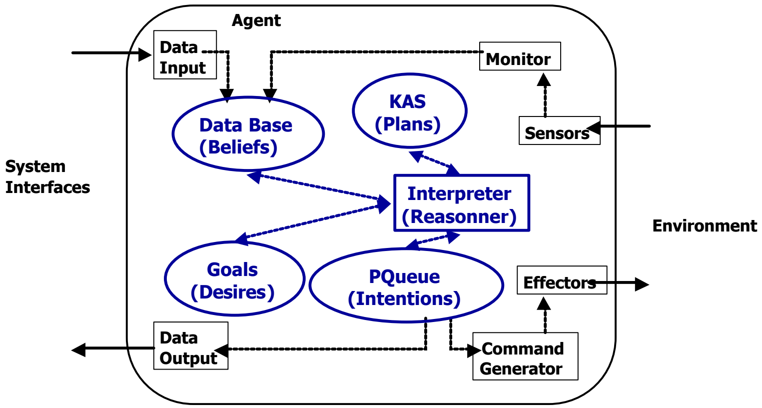
Case study – BDI agents



BDI Implementations

- [Georgeff, Lansky 87]: **P**rocedural **R**easoning **S**ystem uses and supports the BDI model.
- [Rao 95]: BDI-logics – modal operators for Beliefs, Desires and Intentions.
- BDI applications: Space Shuttle (Diagnosis), Sydney Airport (air traffic control).
- BDI Agents Platform: JACK, Zeus.

PRS



PRS (2)

- The plan-recipes library (KAS) builds the procedural knowledge to satisfy the intentions.
- A plan-recipe (KA) is defined by :
  - a body
  - triggering condition to activate a plan (➔ Desire)
  - a pre-condition (feasability)

Hybrid agents – the Touring Machines

[Ferguson 94]

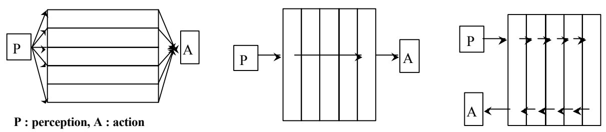
Hybrid agents

- Reactive agents are too simple – they work well in some scenarios, but they fail to solve complex problems
- Deliberative agents are too complex – they need too much time to deliberate, they fail in very dynamic environments

- Solution: hybrid agents that are both reactive and deliberative, depending on the situation.
- The reactive and deliberative behaviours are organized in layers → layered architectures.

Layered Architectures

- Modules' Organization:



P : perception, A : action

a) horizontal architecture

b) modular vertical architecture one path

c) layered vertical architecture two paths

- Control flow : one / several
- Data flow : broadcast, traduction
- Control structure : inhibition, hierarchy, ...

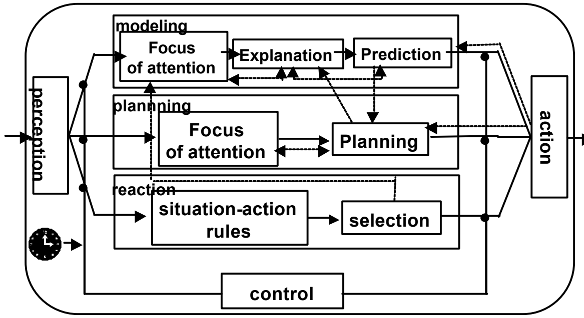
Touring Machines

- Constrained navigation in dynamic environments

- Consists of three activity producing layers : each layer produces suggestions for the actions to perform.
  - **Reactive layer :** reactive behaviour
  - **Planning-Layer:** proactive behaviour
  - **Modeling Layer:** world updates, beliefs; it predicts conflicts between agents and it changes the plans/goals

- **Control-subsystem:** chooses the active layer: certain observations should never reach certain layers.

Touring Machines (2)

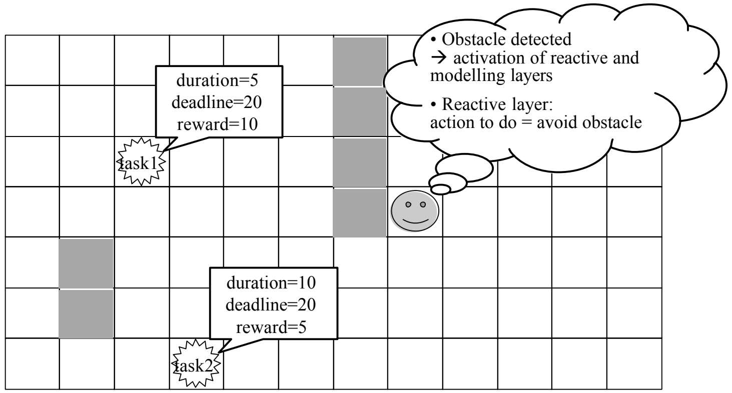


Modeling of the other

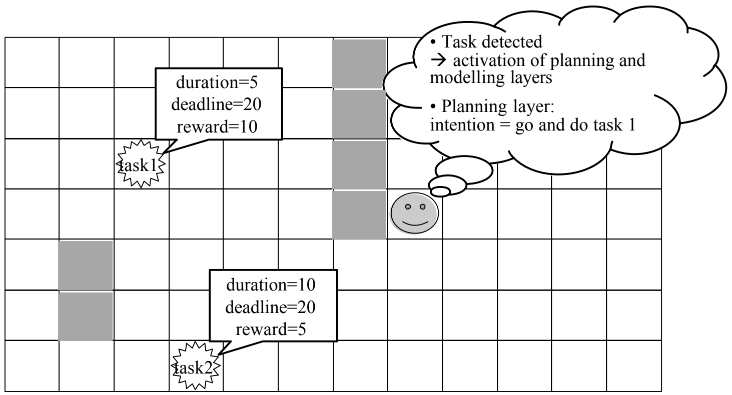
Pro-active behaviour

Reactive behaviour

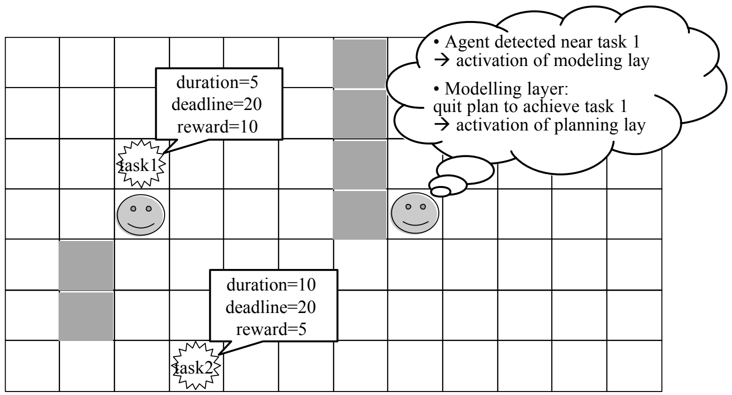
Case study – Touring Machines



Case study – Touring Machines (2)



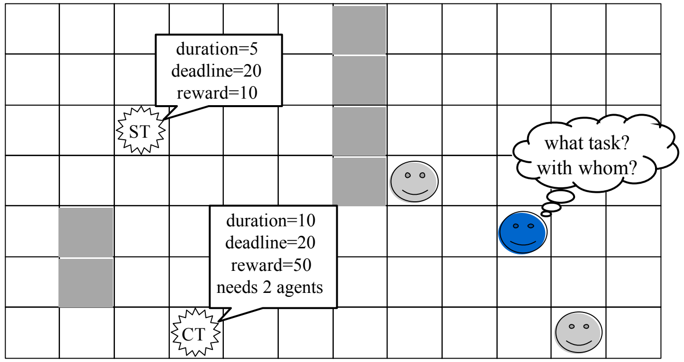
Case study – Touring Machines (3)



Outline

- Individual Agents
  - reactive agents: the subsumption architecture
  - deliberative agents: the BDI model and the PRS architecture
  - hybrid agents: Touring Machines
- Social Agents
  - Agent Oriented Programming
  - the InterRaP architecture
- Organized Agents

Case study – social agents



Social agents

- Reason about themselves, their environment and about the interactions with other agents
- We need to model these interactions (subject of the Interaction course)
  - agent interaction is generally done by means of communication via exchanged messages (e.g., request, inform, etc.)
  - how these messages modify the internal state of an agent?
- Our case study:
  - SingleTasks (ST) and CooperativeTasks (CT) that need several agents to execute them and to divide their rewards
  - agents communicate to inform each other about task positions and to form agreements on CT execution.

Agent Oriented Programming

[Shoham 93]

AOP

- AOP : Agent Oriented Programming
- Three main components :
  - a formal language with a syntax and a semantic to describe mental states,
  - an interpreted programming language to program agents
  - agentification process to convert native applications

➔ **Agent** : an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments, (…) What makes any hardware or software component an agent is precisely the fact that one has chosen to analyse and control it in these mental terms. [Shoham 93]

Master Web Intelligence Systèmes Multi-Agents 2004

AOP : Agent0

- Agent specified in terms of:
  - a set of capabilities (things it can do)
  - a set of initial beliefs
  - a set of initial commitments (like intentions in BDI)
  - a set of commitment rules
- Key component, which determines how the agent acts, is the set of commitment rules. Each rule contains:
  - a message condition
  - a mental condition
  - an action

Master Web Intelligence Systèmes Multi-Agents 2004

AOP : Agent0 (2)

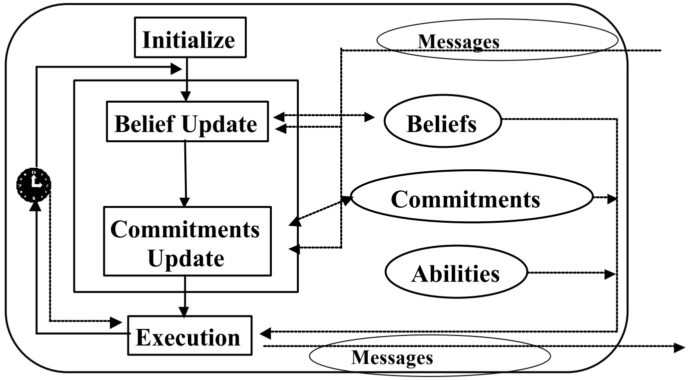
- If the message condition matches a message the agent has received and the mental condition matches the beliefs of the agent, the rule fires.
- When a rule fires, the agent becomes committed to the action.
- The operation of an agent is simply:
  - (1) read all current messages, update beliefs and commitments
  - (2) execute all commitments where capable of action
  - (3) goto 1

Master Web Intelligence Systèmes Multi-Agents 2004

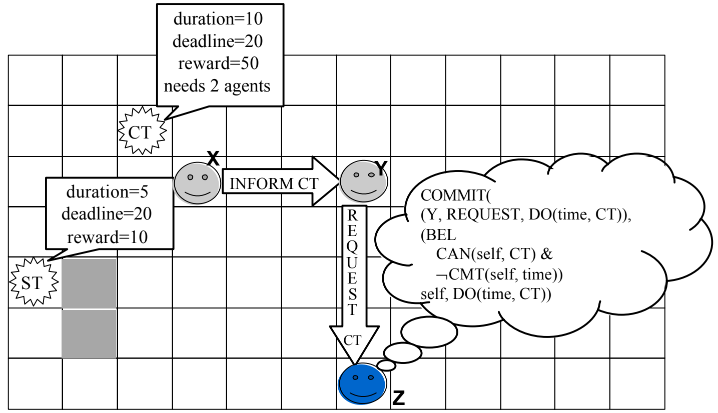
AOP : Agent0 (3)

- Each action is either:
  - private : an internal subroutine, or
  - communicative : a message sent to other agents
- Messages are constrained to be one of three types:
  - request : perform an action
  - unrequest : refrain from performing an action
  - inform : pass an information
- Request and unrequest messages typically result in a modification of agent's commitments.
- Inform messages result in a change to the agent's beliefs.

AOP : Agent0 (4)



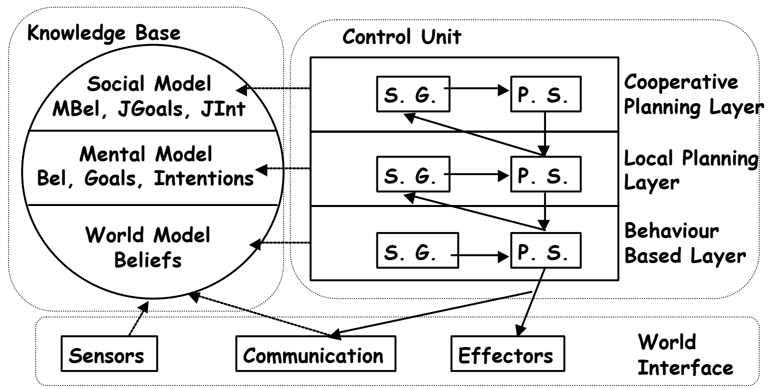
Case study – social agents



The InterRaP architecture

[Muller 95]

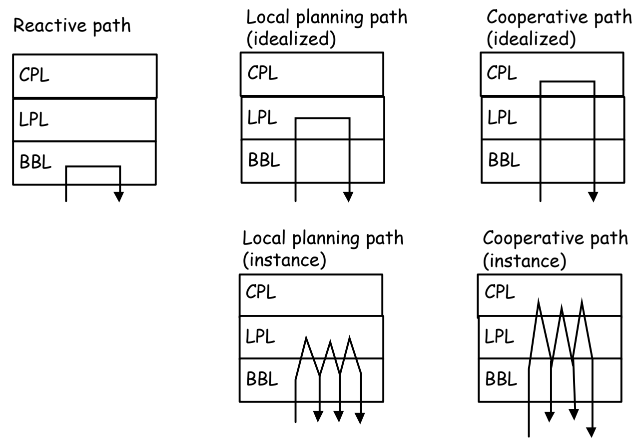
InterRaP



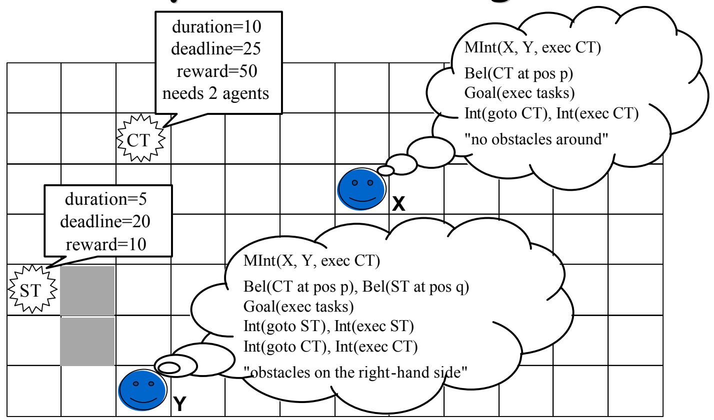
InteRRaP (2)

	BBL	LPL	CPL
Belief Revision	Generation and revision of beliefs (world model)	Abstraction of local beliefs (mental model)	Maintaining models of other agents (social model)
Situation recognition Goal activation	Activation of reactor patterns	Recognition of situations requiring local planning	Recognition of situations requiring cooperative planning
Planning Scheduling	Reactor: direct link from situations to action sequences	Modifying local intentions; local planning	Modifying joint intentions ; cooperative planning.

InteRRaP (3)



Case study – InterRaP agents



Outline

- Individual Agents
  - reactive agents: the subsumption architecture
  - deliberative agents: the BDI model and the PRS architecture
  - hybrid agents: Touring Machines
- Social Agents
  - Agent Oriented Programming
  - the InterRaP architecture
- Organized Agents
  - the B-DOING model

Organized agents

- Reason about themselves, their environment, the interactions with other agents and the organizational structures enforcing these interactions
- We need to model these organizational structures (subject of the Organization course)
  - many notions are used: groups, roles, norms, etc.
  - e.g., a norm saying that a car must stop at the red light
  - agents that violate a norm pay penalities
- Our case study:
  - a norm saying that an agent is forbidden to violate a commitment towards another to cooperatively execute a CT
  - a norm saying that a tax on the reward gained is to be payed

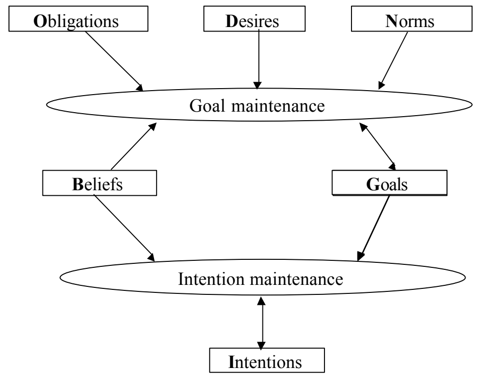
The B-DOING model

[Dignum et al., 2001]

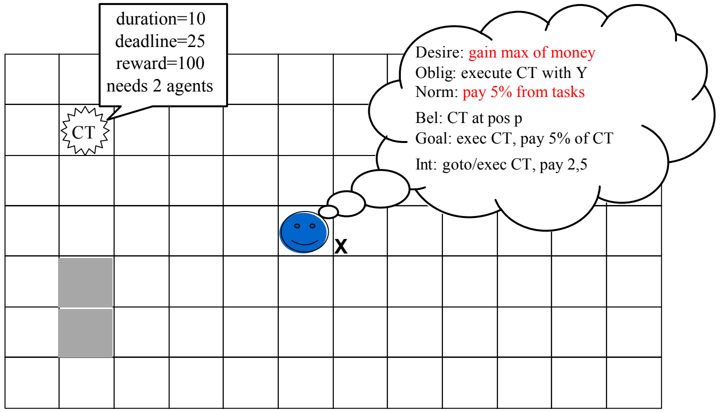
B-DOING

- Extends the BDI model.
- The agent's intentions are generated based on its current beliefs and a set of possibly conflicting goals.
- The goals are generated from:
  - a set of desires: what the agent wants;
  - a set of obligations: what other agents want;
  - a set of norms: what is good for the society.
- B-DOING logic: an extention of BDI-logic with three new modal operators.

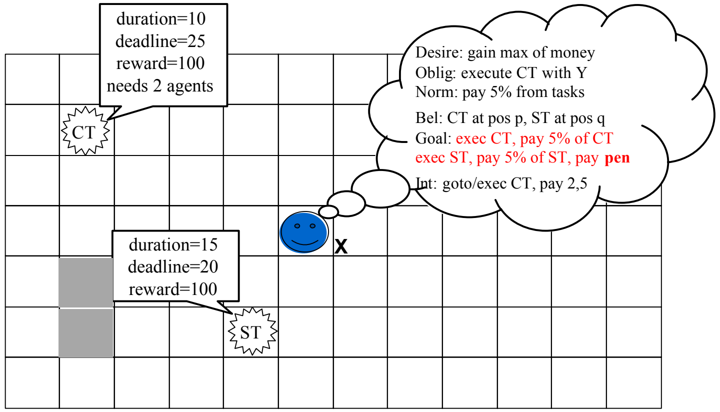
B-DOING (2)



Case study – B-DOING agents



Case study – B-DOING agents



Conclusions and references

Conclusions

- We saw several agent architectures, each one adapted to a different kind of problem.
- Is there a generic architecture?
- The choice of the environment, interaction or organizational model influences the agent architecture.
- The utilization of a MAS platform also influences the agent architecture.
- What about a component-based agent architecture? Given the application, different components are used...

Bibliography

- M.E. Bratman: *What is Intention?*. Intentions in Communication, MIT Press, 1990, p. 15-32.
- R. Brooks, J.H. Connel: *Asynchronous Distributed Control System for a mobile robot*, SPIE 727 Mobile Robots, 1986.
- Y.Demazeau: *From interactions to collective behaviour in agent-based systems*. In Proc. of the 1st European Conf. on Cognitive Science, Saint Malo, France, April, 1995, p. 117-132.
- F.Dignum, D.Kinny and E.Sonenberg: *Motivational Attitudes of Agents: On Desires Obligations and Norms*. In Proc. of the 2nd Int. Workshop of Central and Eastern Europe on MAS, 2001, p.61-70.
- I.Ferguson: *Integrating Models and Behaviors in Autonomous Agents: Some Lessons Learned on Action Control*. Intelligent Agents - Proc. of the Workshop on Agent Theories, Architectures, and Languages (ATAL'94). LNCS Series, Springer-Verlag, Berlin. 1995.
- M.P. Georgeff: *Communication and Interaction in Multi-Agent Planning*. AAAI, 1983, p. 125-129.

Bibliography (2)

- M.Georgeff, A. Lansky: *Reactive Reasoning and Planning*. In Proc. of the Conf. of the American Association of Artificial Intelligence, Seattle, WA, 1987, p. 677–682.
- J.P.Muller: *The Design of Intelligent Agents: A Layered Approach*. Springer-Verlag, 1996.
- A.S.Rao, M. Georgeff: *Modeling Rational Agents within a BDI-Architecture*. In J. F.Allen, R. Fikes and E. Sandewall (eds.), Proc. of the Int. Conf. on Knowledge Representation and Reasoning, Morgan Kaufmann, Cambridge, MA, 1991, p. 473–484.
- A.S.Rao: *Decision Procedures for Propositional Linear-Time Belief-Desire-Intention Logics*. In M. Wooldridge et al. (eds.), Intelligent Agents II – Proc. of the Workshop on Agent Theories, Architectures and Languages (ATAL-95), LNAI-890, Springer-Verlag, Berlin, 1995, p.1-39.
- Y. Shoham: *Agent Oriented Programming*, Artificial Intelligence 60, 1993, p. 51-92.
- L. Steels: *Cooperation between Distributed Agents through Self-Organization*. Decentralized AI Elsevier North-Holland, 1989.
- M.J.Wooldridge: *Intelligent Agents*, In G.Weiss (ed.): Multi-agent systems: A modern approach to Distributed Artificial Intelligence, MIT Press, 1999.