Multi-Agent Programming

– Agent –

O. Boissier

Univ. Clermont Auvergne, IMT Mines Saint-Etienne, LIMOS UMR CNRS 6158, France

CPS2 M1 – Fall 2021

Multi-Agent Oriented Programming

Agent concepts and approaches

Outline

Agent Basic Concepts

Panorama of Agent Models

Literature

Agent Basic Concepts

Books: [Bordini et al., 2005], [Bordini et al., 2009]

Proceedings: ProMAS, DALT, LADS, EMAS, AGERE, ...

Surveys: [Bordini et al., 2006], [Fisher et al., 2007] ...

Languages of historical importance: Agent0 [Shoham, 1993], AgentSpeak(L) [Rao, 1996], MetateM [Fisher, 2005], 3APL [Hindriks et al., 1997], Golog [Giacomo et al., 2000]

Other prominent languages: *Jason* [Bordini et al., 2007], Jadex [Pokahr et al., 2005], 2APL [Dastani, 2008], GOAL [Hindriks, 2009], JACK [Winikoff, 2005], JIAC, AgentFactory

But many other languages and platforms...

Some Languages and Platforms

Jason (Hübner, Bordini, ...); 3APL and 2APL (Dastani, van Riemsdijk, Meyer, Hindriks, ...); Jadex (Braubach, Pokahr); MetateM (Fisher, Guidini, Hirsch, ...); ConGoLog (Lesperance, Levesque, ... / Boutilier – DTGolog); Teamcore/ MTDP (Milind Tambe, ...); IMPACT (Subrahmanian, Kraus, Dix, Eiter); CLAIM (Amal El Fallah-Seghrouchni, ...); GOAL (Hindriks); BRAHMS (Sierhuis, ...); SemantiCore (Blois, ...); STAPLE (Kumar, Cohen, Huber); Go! (Clark, McCabe); Bach (John Lloyd, ...); MINERVA (Leite, ...); SOCS (Torroni, Stathis, Toni, ...); FLUX (Thielscher); JIAC (Hirsch, ...); JADE (Agostino Poggi, ...); JACK (AOS); Agentis (Agentis Software); Jackdaw (Calico Jack); *simpAL*, *ALOO* (Ricci, ...);

...

Theories, Models, Architectures

Agent Basic Concepts

- Agents are used to solve problems (e.g. to find solutions, to take decisions, to act on the environment)
- The characteristics of the problem influence the way the agents are built
  - $\rightsquigarrow$ we then talk about **agent architectures**
- It may be the case that some architectures are designed using general principles
  - $\rightsquigarrow$ we then talk about **agent models**
- Some of these models have a theory associated with them that allows the verification of some properties
  - $\rightsquigarrow$ we then talk about **agent theories**

Several agent architectures, models and theories exist in the literature!!!

Outline

Analysis Grid

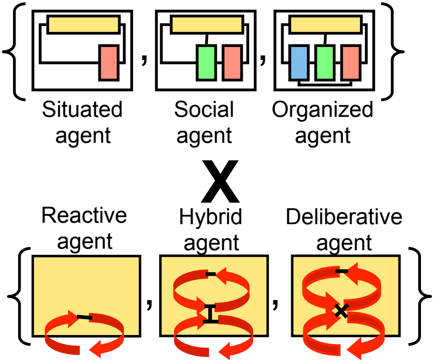
Panorama of Agent Models

Agent models depend on:

- the type of inputs that they reason from (external factor)
- the control cycle connecting inputs to actions (coupling)

External factor

Coupling



External Factor

Panorama of Agent Models

- **Situated** 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 **organizations** (e.g. social structures, norms) enforcing these interactions

Coupling

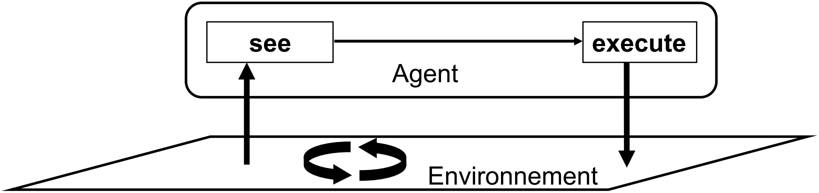
Panorama of Agent Models

- **Reactive** Agent
  - tight coupling between perception of the external factors with action
- **Deliberative** Agent
  - loose coupling between perception and actions: agents deliberate on the actions to execute from their perception of the external factors and from their goals
- **Hybrid** Agent
  - agents that are mixing reactivity and deliberation

Reactive Agent Models

Panorama of Agent Models/ Coupling Dimension

- The process cycle of an agent is a closed loop between "execute" and "see" (Stimulus/Response)
- reaction to the evolution of the environment
- No explicit representation of the environment, of the other agents, of its skills,
- Decisions are done without reference to the past (no history), to the futur (no planning)



Reactive Agent models

Panorama of Agent Models/ Coupling Dimension

Reactive approach arises in opposition to the symbolic reasoning model (AI). Several approaches that are based on :

- behaviours
  - [Brooks, 1986], (Steels 89), (robotic)
  - (Drogoul 93) (ethology)
- interactions
  - (Demazeau 93) (image analysis, cartography, ...)
  - (Bura 91) (games)
- situations
  - (Agre 87) (games)
  - (Wavish 90) (design, manufacturing)

Reactive Agent models

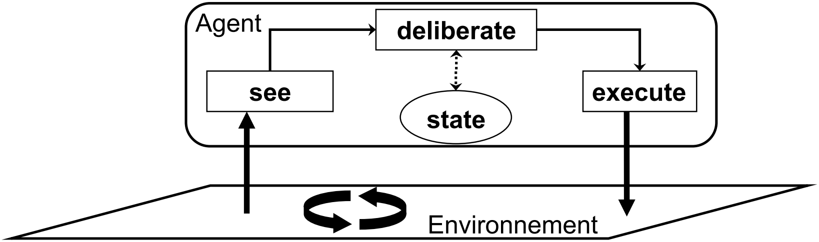
Panorama of Agent Models/ Coupling Dimension

- Example of control cycle of a reactive agent (implemented as a set of condition/action rules):

```
condition-action rules
set of percepts
do {
    percepts := see();
    state := interpret-inputs(percepts);
    rule := match(state,rules);
    execute(rule[action]);
} while (true);
```

Deliberative Agent models

- The process cycle of an agent introduces a "deliberate" function between "see" and "execute" in order to choose the "right" action
- Explicit Representation of the environment, of the other agents, of its skills, ...
- History management, ...



Deliberative Agent models

Panorama of Agent Models/ Coupling Dimension

- Goal-based Agents
  - Rich internal state
  - Can anticipate the effects of their actions (e.g. Planning)
  - Take those actions expected to lead toward achievement of goals
  - Capable of reasoning and deducing properties of the world (Knowledge representation)
- Utility-based Agent
  - Decision Theory + Probabilities
  - Use of utility function that maps state (or state sequences) into real numbers
  - Permits more fine-grained reasoning about what can be achieved, what are the trade-offs, conflicting goals, etc

Hybrid Agent Models

Hybrid Agent's Model: Reactive and Deliberative Agent

- 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
- The reactive and deliberative behaviors are organized in layers
- Examples: Touring Machines [Ferguson, 1995], InterRaP[Müller and Pischel, 1994],

Outline

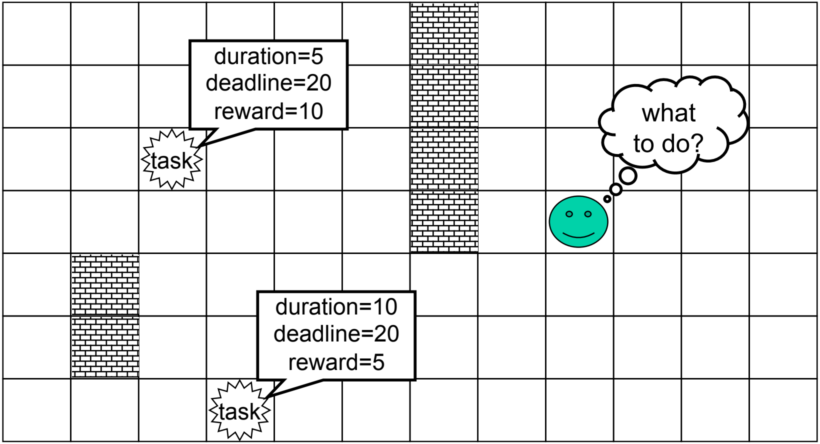
Situated Agent

Panorama of Agent Models

- **Reactive** agents: the subsumption architecture [Brooks, 1986]
- **Deliberative** agents: the BDI model and the PRS architecture
- **Hybrid** agents: Touring Machines [Ferguson, 1995]
- Reason about themselves and about their environment
- We need to model the environment (subject of the **Agent working 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

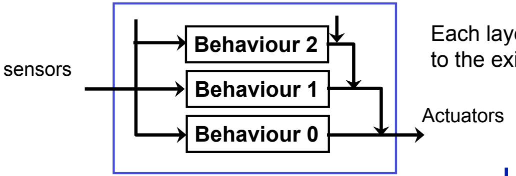
Panorama of Agent Models/ Situated Agent



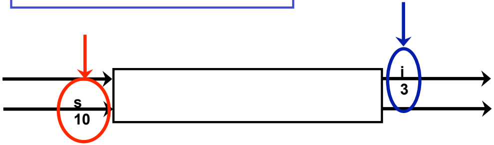
The Subsumption Architecture

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

The Subsumption Architecture



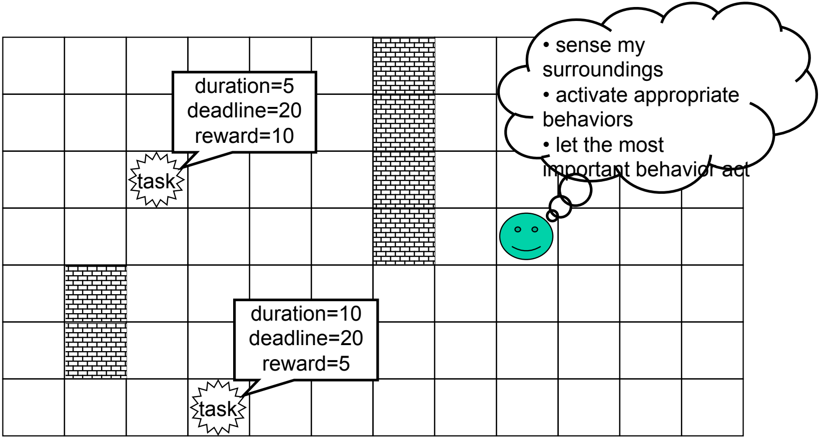
Each layer can be incrementally added to the existing architecture.



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)

The Subsumption Architecture

Panorama of Agent Models/ Situated Agent/ Reactive agents



The Subsumption Architecture

- 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 them 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.

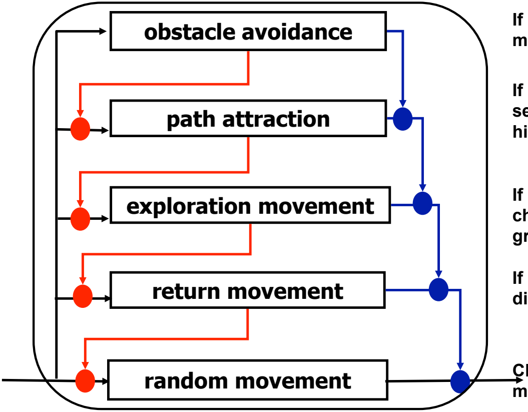
The Subsumption Architecture

Two sets of behaviors running in parallel:

- Handling behavior
  - 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 behaviors organized along a subsumption hierarchy

The Subsumption Architecture

Panorama of Agent Models/ Situated Agent/ Reactive agents



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.

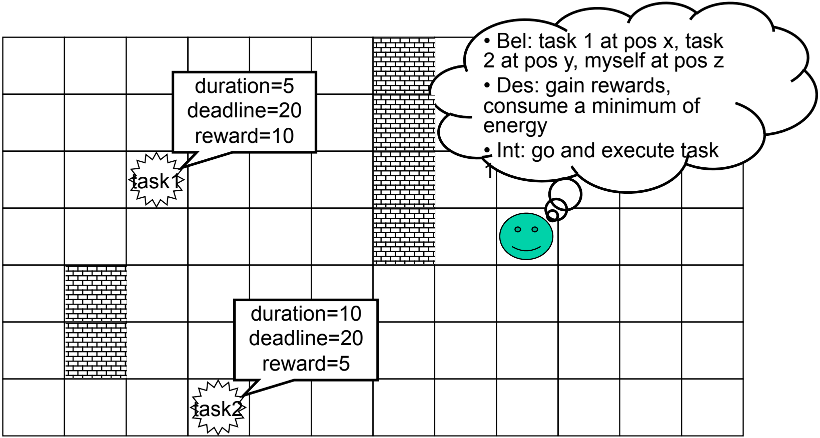
Choose randomly a direction to move. Move in that direction.

PRS Architecture

- the use of intentions in agent's design [Georgeff and Lansky, 1987, Bratman, 1990]
- the BDI model: an agent contains [Rao et al., 1995]
  - 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

PRS Architecture

Panorama of Agent Models/ Situated Agent/ Deliberative agents



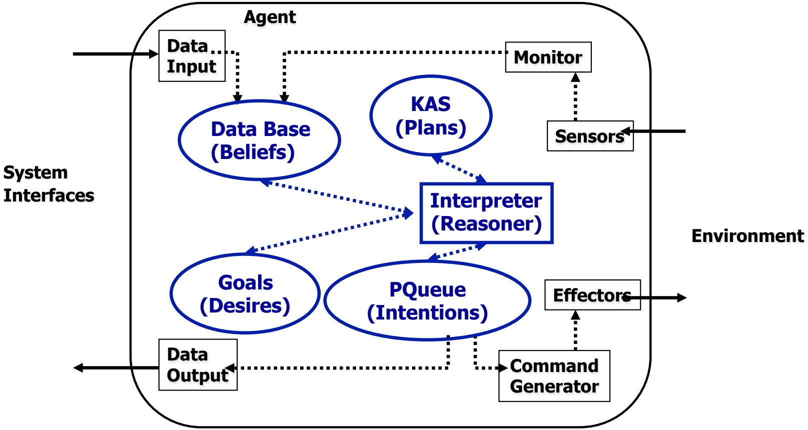
PRS Architecture

BDI Implementations:

- Procedural Reasoning System uses and supports the BDI model [Georgeff and Lansky, 1987]
- BDI-logics - modal operators for Beliefs, Desires and Intentions [Rao et al., 1995]
- BDI applications: Space Shuttle (Diagnosis), Sydney Airport (air traffic control).
- BDI Agents Platform: JACK, Zeus, Jadex, Jason.

PRS Architecture

Panorama of Agent Models/ Situated Agent/ Deliberative agents



- 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)

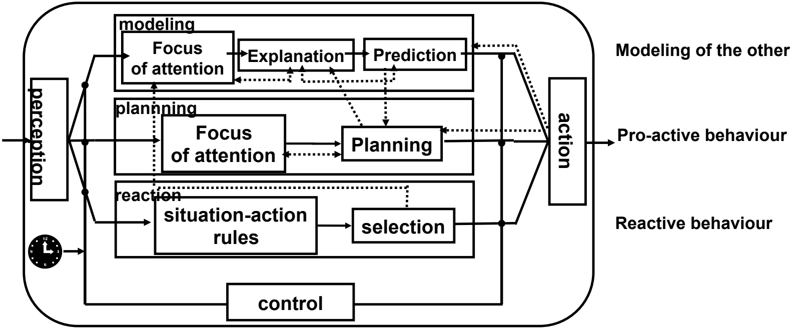
Hybrid agents

- 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 behaviors are organized in layers $\rightsquigarrow$ layered architectures.

Touring Machines

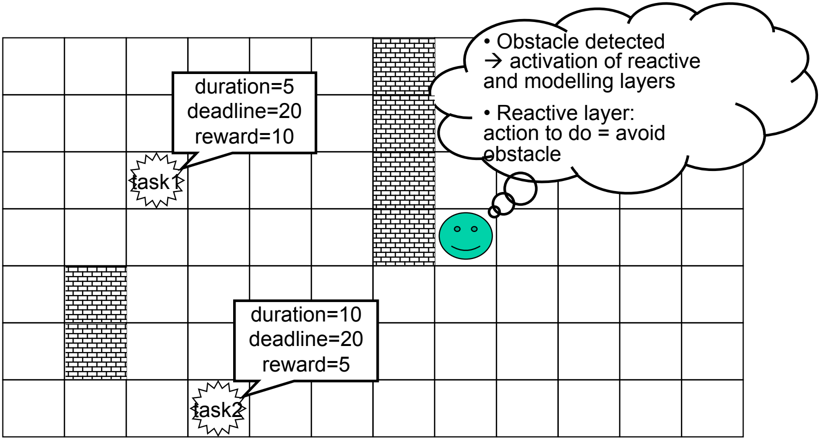
- Constrained navigation in dynamic environments
- Consists of three activity producing layers : each layer produces suggestions for the actions to perform.
  - Reactive layer: reactive behaviour
  - Planning Layer: proactive behaviour
  - Modeling Layer: world updates, beliefs; it predicts conflicts between agents and it changes the plans/goals
- Control-subsystem: chooses the active layer: certain observations should never reach certain layers.

Touring Machines

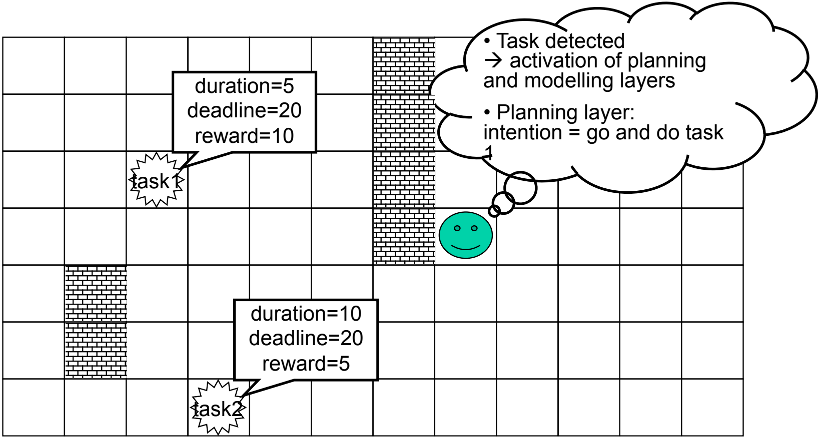


Touring Machines

Panorama of Agent Models/ Situated Agent/ Hybrid agents

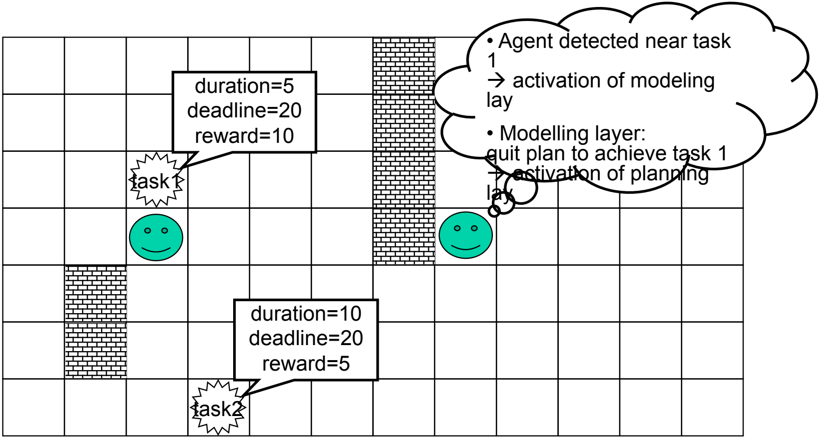


Touring Machines



Touring Machines

Panorama of Agent Models/ Situated Agent/ Hybrid agents



Outline

Agent Basic Concepts

Panorama of Agent Models

- Situated Agent
- Social Agents
- Organized agents
- Agent Architectures

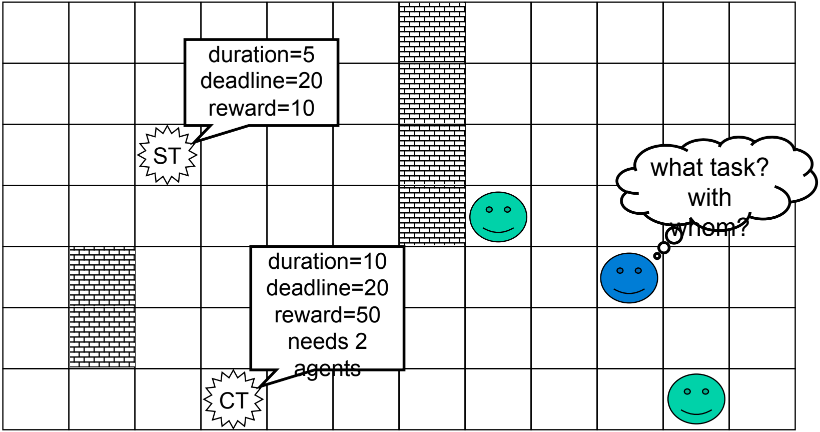
Social Agents

Panorama of Agent Models

- AOP/AgentO [Shoham, 1993]
- The InterRaP Architecture [Müller and Pischel, 1994]
- Reason about themselves, their environment and about the interactions with other agents
- We need to model these interactions (subject of the **Agent and Agent Working Environment** courses)
  - 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.

Case Study

Panorama of Agent Models/ Social Agents



Agent0

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, 1993]

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

Agent0

Panorama of Agent Models/ Social Agents/ Deliberative Agent

- 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

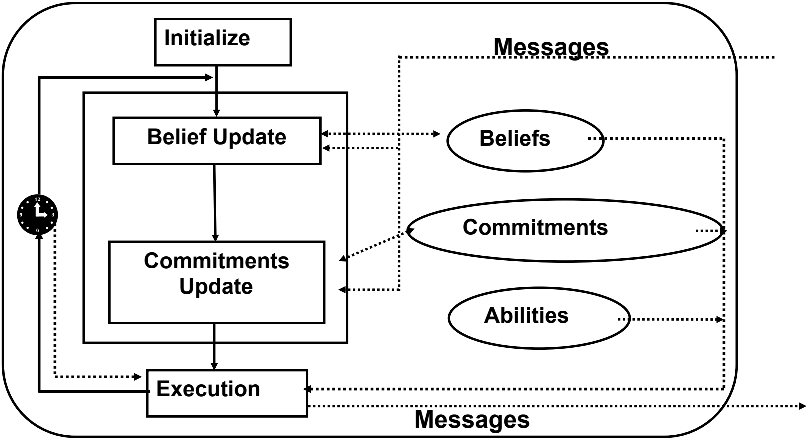
Agent0

- 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

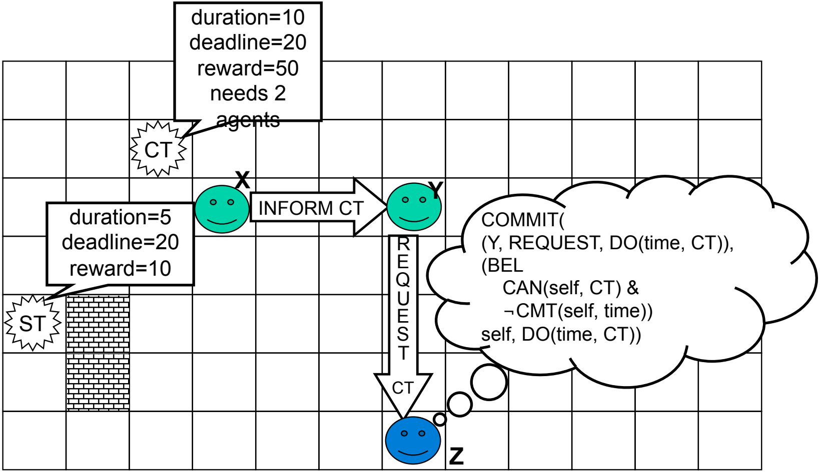
  Request and unrequest messages typically result in a modification of agent's commitments.
  Inform messages result in a change to the agent's beliefs.

Agent0

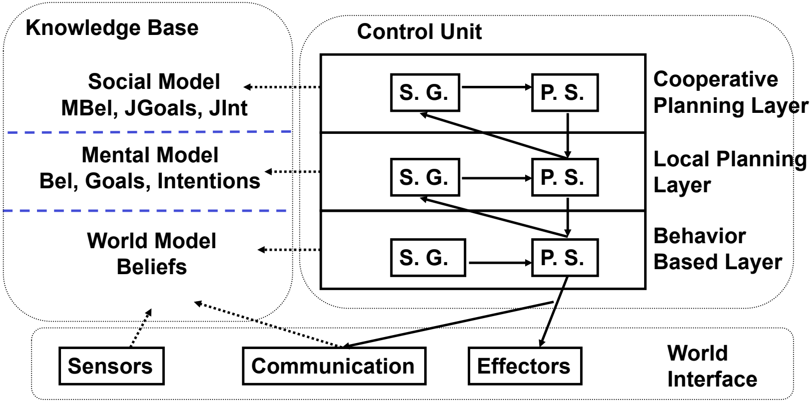
Panorama of Agent Models/ Social Agents/ Deliberative Agent



Agent0


duration=10
deadline=20
reward=50
needs 2
agents
CT
X
INFORM CT
Y
duration=5
deadline=20
reward=10
ST
REQUEST
CT
Z
COMMIT(
(Y, REQUEST, DO(time, CT)),
(BEL
CAN(self, CT) &
¬CMT(self, time))
self, DO(time, CT))

The InterRaP architecture



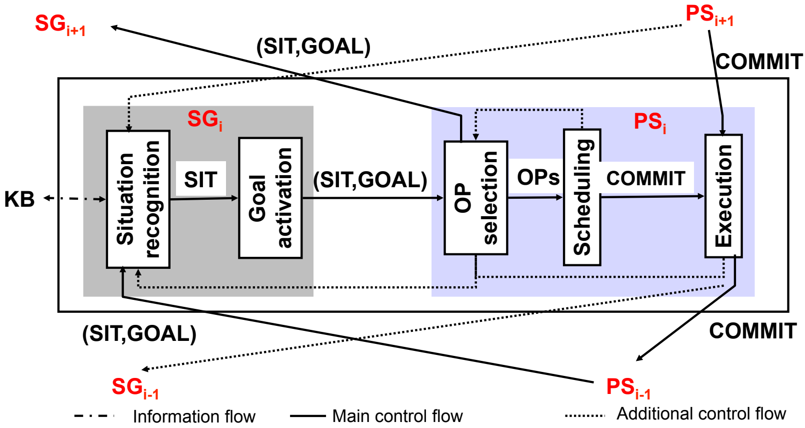
The InterRaP architecture

Panorama of Agent Models/ Social Agents/ Hybrid Agent

	BB Layer	LP Layer	CP Layer
Belief Revision	Generation and revision of beliefs (world model)	Abstraction of local beliefs (mental model)	Maintaining models of other agents (social model)
Situation recognition **G**oal activation	Activation of reactor patterns	Recognition of situations requiring local planning	Recognition of situations requiring cooperative planning
Planning **S**cheduling	Reactor: direct link from situations to action sequences	Modifying local intentions; local planning	Modifying joint intentions; cooperative planning.

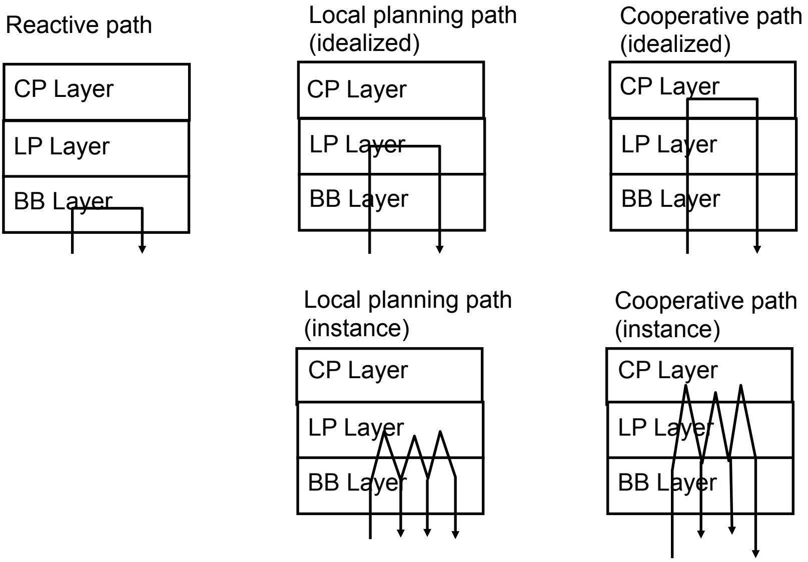
The InterRaP architecture

Panorama of Agent Models/ Social Agents/ Hybrid Agent



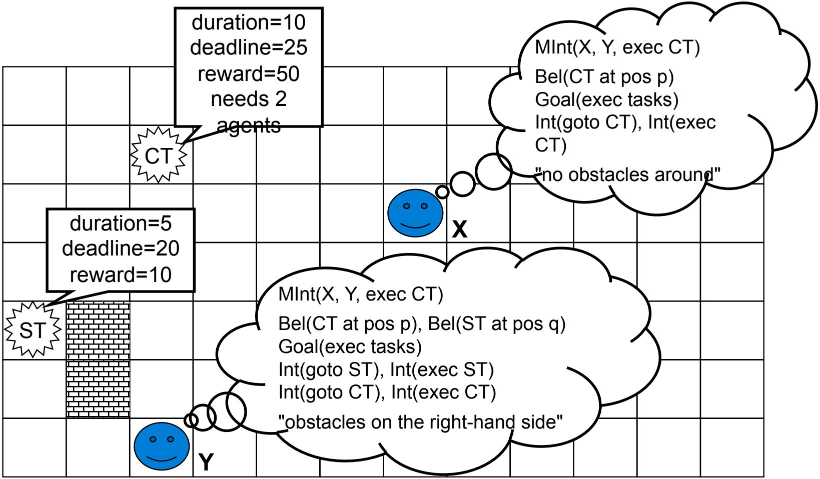
The InterRaP architecture

Panorama of Agent Models/ Social Agents/ Hybrid Agent



The InterRaP architecture

Panorama of Agent Models/ Social Agents/ Hybrid Agent



Outline

Agent Basic Concepts

Panorama of Agent Models

- Situated Agent
- Social Agents
- Organized agents
- Agent Architectures

Organized agents

Panorama of Agent Models

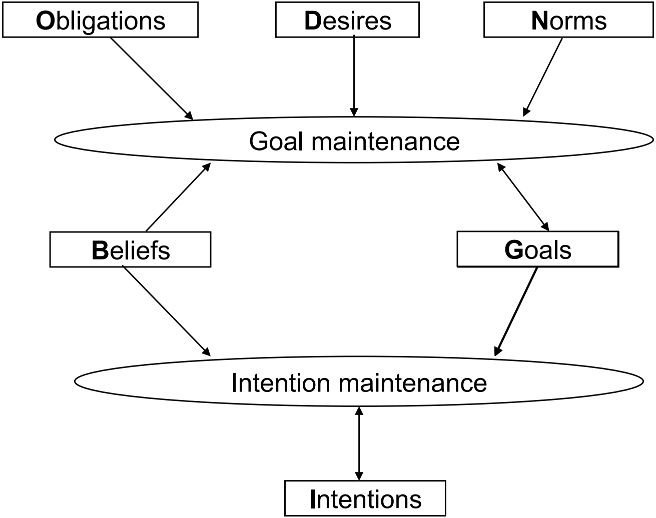
- Reason about themselves, their environment, the interactions with other agents and the organizational structures enforcing these interactions
- We need to model these organizational structures (subject of the **Agent organization** course)
  - many notions are used: groups, roles, norms, etc.
  - e.g., 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

B-DOING

- B-DOING (Dignum 01) 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 architecture

Panorama of Agent Models/ Organized agents/ Deliberative Agent

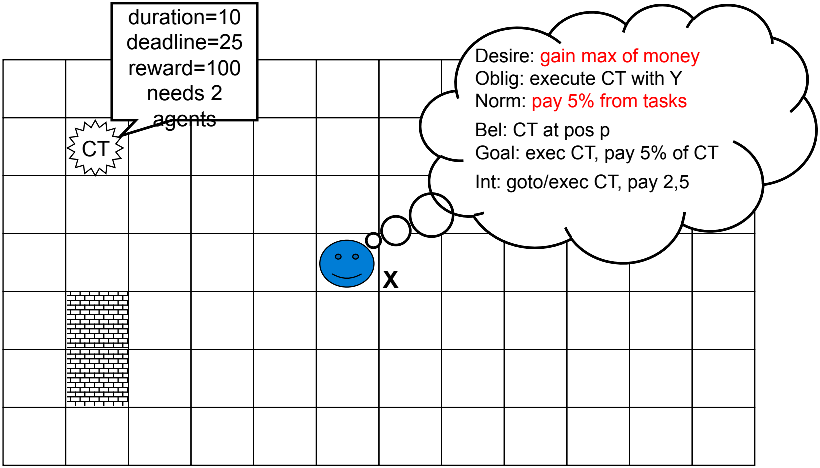


B-DOING architecture

- Example of a control cycle of a BDOING agent
  - *b* : beliefs, *g* : desires, *i* : intentions, *eq* : event queue

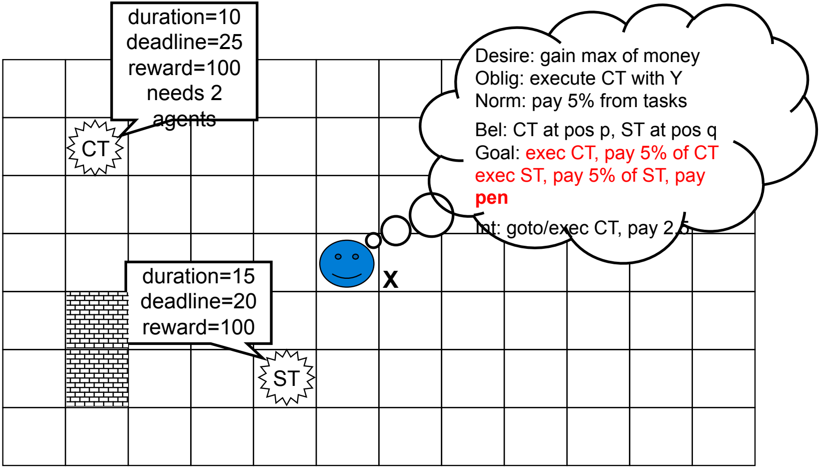
```
(b,g,i) := initialize();
repeat
    options := option_generator(eq,b,g,i, oblEvents);
    selected := deliberate(options, b,g,i, oblEvents);
    i := selected ∪ i;
    execute(i);
    eq := see();
    b := update_beliefs(b,eq);
    (g,i) := drop_successful_attitudes(b,g,i);
    (g,i) := drop_impossible_attitudes(b,g,i);
forever
```

B-DOING architecture



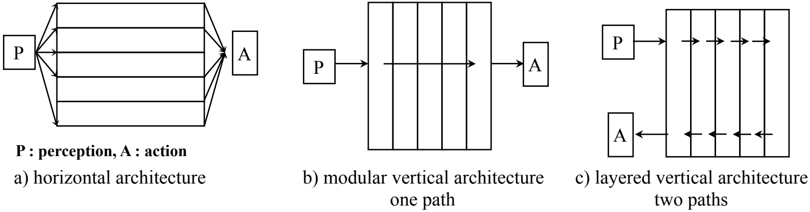
B-DOING architecture

Panorama of Agent Models/ Organized agents/ Deliberative Agent



Outline

- Modules Organisation:



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

Bibliography I

Bordini, R. H., Braubach, L., Dastani, M., Fallah-Seghrouchni, A. E., Gómez-Sanz, J. J., Leite, J., O'Hare, G. M. P., Pokahr, A., and Ricci, A. (2006).
A survey of programming languages and platforms for multi-agent systems.
Informatica (Slovenia), 30(1):33–44.

Bordini, R. H., Dastani, M., Dix, J., and Fallah-Seghrouchni, A. E., editors (2005).
Multi-Agent Programming: Languages, Platforms and Applications, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*.
Springer.

Bordini, R. H., Dastani, M., Dix, J., and Fallah-Seghrouchni, A. E., editors (2009).
Multi-Agent Programming: Languages, Tools and Applications.
Springer.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007).
Programming Multi-Agent Systems in AgentSpeak Using **Jason**.
Wiley Series in Agent Technology. John Wiley & Sons.

Bibliography II

Bratman, M. E. (1990).

What is intention.

Intentions in communication, pages 15–32.

Brooks, R. (1986).

A robust layered control system for a mobile robot.

IEEE journal on robotics and automation, 2(1):14–23.

Dastani, M. (2008).

2apl: a practical agent programming language.

Autonomous Agents and Multi-Agent Systems, 16(3):214–248.

Ferguson, I. A. (1995).

On the role of bdi modeling for integrated control and coordinated behavior in autonomous agents.

Applied Artificial Intelligence an International Journal, 9(4):421–447.

Fisher, M. (2005).

Metatem: The story so far.

In *PROMAS*, pages 3–22.

Bibliography III

Fisher, M., Bordini, R. H., Hirsch, B., and Torroni, P. (2007).
Computational logics and agents: A road map of current technologies and future trends.
Computational Intelligence, 23(1):61–91.

Georgeff, M. P. and Lansky, A. L. (1987).
Reactive reasoning and planning.
In *AAAI*, volume 87, pages 677–682.

Giacomo, G. D., Lespérance, Y., and Levesque, H. J. (2000).
Congolog, a concurrent programming language based on the situation calculus.
Artif. Intell., 121(1-2):109–169.

Hindriks, K. V. (2009).
Programming rational agents in GOAL.
In [Bordini et al., 2009], pages 119–157.

Hindriks, K. V., de Boer, F. S., van der Hoek, W., and Meyer, J.-J. C. (1997).
Formal semantics for an abstract agent programming language.
In Singh, M. P., Rao, A. S., and Wooldridge, M., editors, *ATAL*, volume 1365 of *Lecture Notes in Computer Science*, pages 215–229. Springer.

Bibliography IV

Müller, J. P. and Pischel, M. (1994).
An architecture for dynamically interacting agents.
International Journal of Intelligent and Cooperative Information Systems, 3(01):25–45.

Pokahr, A., Braubach, L., and Lamersdorf, W. (2005).
Jadex: A bdi reasoning engine.
In [Bordini et al., 2005], pages 149–174.

Rao, A. S. (1996).
Agentspeak(l): Bdi agents speak out in a logical computable language.
In de Velde, W. V. and Perram, J. W., editors, *MAAMAW*, volume 1038 of *Lecture Notes in Computer Science*, pages 42–55. Springer.

Rao, A. S., Georgeff, M. P., et al. (1995).
Bdi agents: From theory to practice.
In *ICMAS*, volume 95, pages 312–319.

Shoham, Y. (1993).
Agent-oriented programming.
Artif. Intell., 60(1):51–92.

Bibliography V

Winikoff, M. (2005).
Jack intelligent agents: An industrial strength platform.
In [Bordini et al., 2005], pages 175–193.