Agents as Intentional Systems

Multiagent Systems LS

Sistemi Multiagente LS

Andrea Omicini & Michele Piunti
{andrea.omicini, michele.piunti}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2009/2010

Intelligent Agents

- According to a classical definition, an intelligent agent is a computational system capable of *autonomous* action and *perception* in some environment

Reminder: Computational Autonomy

- Agents are autonomous as they encapsulate (the thread of) control
- Control does not pass through agent boundaries
  - only data (knowledge, information) crosses agent boundaries
- Agents have no interface, cannot be controlled, nor can they be invoked
- Looking at agents, MAS can be conceived as an aggregation of multiple distinct loci of control interacting with each other by exchanging information

Intelligent Agents

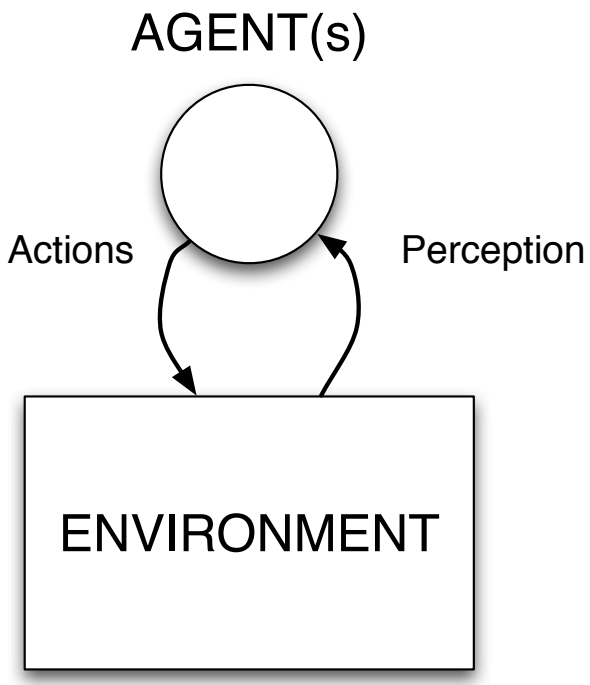
- According to a classical definition, an intelligent agent is a computational system capable of *autonomous* action and *perception* in some environment

Reminder: Computational Autonomy

- Agents are autonomous as they encapsulate (the thread of) control
- Control does not pass through agent boundaries
  - only data (knowledge, information) crosses agent boundaries
- Agents have no interface, cannot be controlled, nor can they be invoked
- Looking at agents, MAS can be conceived as an aggregation of multiple distinct loci of control interacting with each other by exchanging information

Actions and Perception

- Actions are supposed to change the environment in order to meet agents design objectives
- Perception is a process by which the agent recognises the state of the environment, so as to be able to adapt its behaviour to it



[Russell and Norvig, 2002]

- The perception function is the agent's ability to observe its environment
  - Output of the perception function is a percept
    $Perception : E \rightarrow Per$
    which maps environment states to percepts
- The Action function represents the agent's decision making process
  - Output of the action function is an action
    $Action : Per^* \rightarrow A_c$
    which maps sequences of percepts to actions

Autonomy

- As we know, the main point about agents is they are *autonomous*: capable of acting independently, encapsulating control over their internal state.
  - Thus, an agent is a computer system capable of autonomous actions in some environment
- What's about agent's behaviour? The most typically-mentioned features are
  - Reactivity
  - Situatedness
  - Pro-activeness
  - Social ability

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. $stimolous \longrightarrow response$ rules
  - $action : E \longrightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. $stimolous \longrightarrow response$ rules
  - $action : E \longrightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. $stimolous \longrightarrow response$ rules
  - $action : E \longrightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. *stimolous* $\rightarrow$ *response* rules
  - $action : E \rightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. *stimolous* $\rightarrow$ *response* rules
  - $action : E \rightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. *stimolous* $\rightarrow$ *response* rules
  - $action : E \rightarrow A_c$
  - A thermostat is a purely reactive agent

Reactivity

- Application in real world domains are characterised by highly dynamic conditions: situations change, information is incomplete, resources are scarce, the actions performed are not deterministic in their effects
- A *reactive* system is one that maintains an ongoing interaction with its environment, and responds to changes that occour in it—in time for the response to be useful
- Purely reactive agents decide what to do without reference to their history
- Reaction is entirely based on the present, with no reference to past states
  - e.g. *stimolous* $\rightarrow$ *response* rules
  - $action : E \rightarrow A_c$
  - A thermostat is a purely reactive agent

Situatedness

- Reactive models and state-less agency are not enough for entities engaged in dynamic environments
- Such entities continually face with external events requiring adequate services and behavioural responses
- Any "ground" model of action is strictly coupled with the context where the action takes place
- An agent comes with its own model of action
- Any agent is then strictly coupled with the environment where it lives and (inter)acts by the very actions it is capable of

Proactiveness

- We want agents able to intelligently adapt to their environment
  - autonomously (re)adapting to changes
  - recognising opportunities
  - goal-oriented behaviour
- *Proactiveness* is a generative approach
  - agents generate and attempt to achieve their objectives
  - encapsulate control, and the rule to govern it
  - not driven solely by stimuli
  - take the initiative and make something happen, rather than waiting for something to happen

Social Ability

- MAS can be conceived as an articulated world
  - where we cannot go around attempting to achieve goals without taking other entities into account
- Some goals can only be achieved with the cooperation of others
- Thus, *Social ability* in agents is the ability to interact with other agents (and possibly humans) via some kind of agent-communication means
  - Speech-Acts
  - Artifact based Interactions
  - Signals
  - Environment traces
  - ...
- *Coordination* is social: cooperation, collaboration, but also competition with others

Agents with State

- To face these growing complexities, stateful agents can be conceived
- These agents have some internal data structure, which is typically used to record information about the environment state and history
- Let I be the set of all internal states of the agent. The *perception* function is the same:
  $Perception : E \rightarrow Per$
  The action-selection function action is now defined as a mapping:
  $Action : I \rightarrow A_c$
  from internal states to actions.
- An additional function *next* is introduced, which maps an internal state and percept to an internal state: $Next : I \times Per \rightarrow I$

Implementing an Agent with Internal State

Agent Control Loop Version 1

```
1: while true
2:    observe the environment state 'e' and generates a perception(e);
3:    update internal state model:
         i ::= next(i, perception(e));
4:    select an action to execute:
         action(i);
5: end while
```

- We now have the problem to define such agent states
- How to build an effective *internal state model*?
- How to make it run? How to update it?

Intentional Systems

- An idea is to refer to human attitudes as *intentional notions*
- When explaining human activity, it is often useful to make statements such as the following:
  - Felipe got rain tires because he believed it was going to rain
  - Lewis is working hard because he wants to win his first world championship
- These statements can be read in terms of *folk psychology*, by which human behaviour can be explained and can be *predicted* through the attribution of mental attitudes, such as believing and wanting (as in the above examples), hoping, fearing, and so on.

The Intentional Stance

- The philosopher – cognitive scientist – Daniel Dennett coined the term *Intentional System* to describe entities *'whose behaviour can be predicted by the method of attributing to it belief, desires and rational acumen'* [Dennett, 1971].
- Dennett identifies several grades of intentional systems:
  1. A first-order intentional system has beliefs, desires, etc.
     - Felipe believes P
  2. A second-order intentional system has beliefs, desires, etc. about beliefs, desires, etc. both its own and of others
     - Felipe believes that Lewis believes P
  3. A third-order intentional system is then something like
     - Felipe believes that Lewis believes that Felipe believes P
  4. ...

The Intentional Stance in Computing

What entities can be described in terms of intentional stance?

- Human beings are prone to provide an intentional stance to almost anything
  - sacrifices for ingratiating gods benevolence
  - animism
- 'Ascribing mental qualities like beliefs, intentions and wants to a machine is sometimes correct if done conservatively and is sometimes necessary to express what is known about its state [...] it is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it' [McCarthy, 1979]

Agents as Intentional Systems

- As computer systems become ever more complex, we need more powerful abstractions and metaphors to explain their operation
- With complexity growing, mechanicistic / low level explanations become impractical
- The intentional notions can be adopted as abstraction tools, providing us with a convenient and familiar way of describing, explaining, and predicting the behaviour of complex systems
- The idea is to use the intentional stance as an abstraction in computing in order to explain, understand, drive the behaviour—then, crucially, program computer systems

Agents as Intentional Systems

Strong Notion of Agency

- Early agent theorists start from a (strong) notion of agents as intentional systems
- Agents were explained in terms of mental attitudes, or mental states
- In their social abilities, agents simplest consistent description implied the intentional stance
- Agents contain an explicitly-represented – *symbolic* – model of the world (written somewhere in the working memory)
- Agents make decision on what action to take in order to achieve their goals via *symbolic* reasoning

When?

Mental states are a worth abstraction for developing agents to effectively act in a class of application domains characterized by various practical limitations and requirements [Rao and Georgeff, 1995]

- At any instant of time there are many different ways in which an environment may evolve (the environment is not deterministic)
- At any instant of time there are many actions or procedures the agent may execute (the agent is not deterministic, too)
- At any instant of time the agent may want to achieve several objectives
- The actions or procedures that (best) achieve the various objectives are dependent on the state of the environment (i.e., on the particular situation, context)
- The environment can only be sensed locally
- The rate at which computations and actions can be carried out is within reasonable bounds to the rate at which the environment evolves

Goal-Oriented & Goal-Directed Systems

- There are two main families of architectures for agents with mental states

  Teleo-Reactive / Goal-Oriented Agents are based on their internal design model and their control mechanism. The Goal is not figured within the internal state but it is an 'end state' for agents internal state machine

  Deliberative / Goal-Directed Agents are based on a further symbolyc reasoning about Goals, which are explicitely represented and processed aside the control loop

Conceiving Agents with Mental States

Modelling agents based on mental states. . .

- . . . eases the development of agents exhibiting complex behaviour
- . . . provides us with a familiar, non-techical way of understanding and explaining agents
- . . . allows the developer to build MAS by taking the perspective of a cognitive entity engaged in complex tasks – what would *I* do in the same situation?
- . . . simplifies the construction, maintainance and verification of agent-based applications
- . . . is useful when the agent has to comunicate and interact with users or other system entities

Conceiving agents with mental states

- "The intentional stance is the strategy of interpreting the behaviour of an entity (person, animal, artifact, whatever) by treating it as if it were a rational agent who governed its 'choice' of 'action' by a 'consideration' of its 'beliefs' and 'desires' ".
- "The scare-quotes around all these terms draw attention to the fact that some of their standard connotations may be set aside in the interests of exploiting their central features: *their role in practical reasoning*, and hence in the prediction of the behaviour of practical reasoners" [Dennett, 2007].

Agents with Mental States

Agents with Mental States

- Agents governing their behaviour on the basis of internal states that mimic cognitive mental states

  Epistemic States representing agents knowledge (of their world)
  - i.e., Percepts, Beliefs

  Motivational States representing agents objectives
  - i.e., Goals, Desires

- The process of selecting an action to execute based on the actual mental states (i.e., $action(next(i, perception(e)))$) is called *Practical Reasoning*

Practical and Epistemic Reasoning

What Practical Reasoning is

- Practical reasoning is reasoning directed towards actions—the process of figuring out what to do in order to achieve what is desired
- "Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes." [Bratman, 1987]

What Epistemic Reasoning is

- Epistemic reasoning is reasoning directed towards knowledge—the process of updating information, replacing old information (no longer consistent with the world state) with new information

Practical and Epistemic Reasoning

What Practical Reasoning is

- Practical reasoning is reasoning directed towards actions—the process of figuring out what to do in order to achieve what is desired
- "Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes." [Bratman, 1987]

What Epistemic Reasoning is

- Epistemic reasoning is reasoning directed towards knowledge—the process of updating information, replacing old information (no longer consistent with the world state) with new information

Practical Reasoning

What Practical Reasoning is

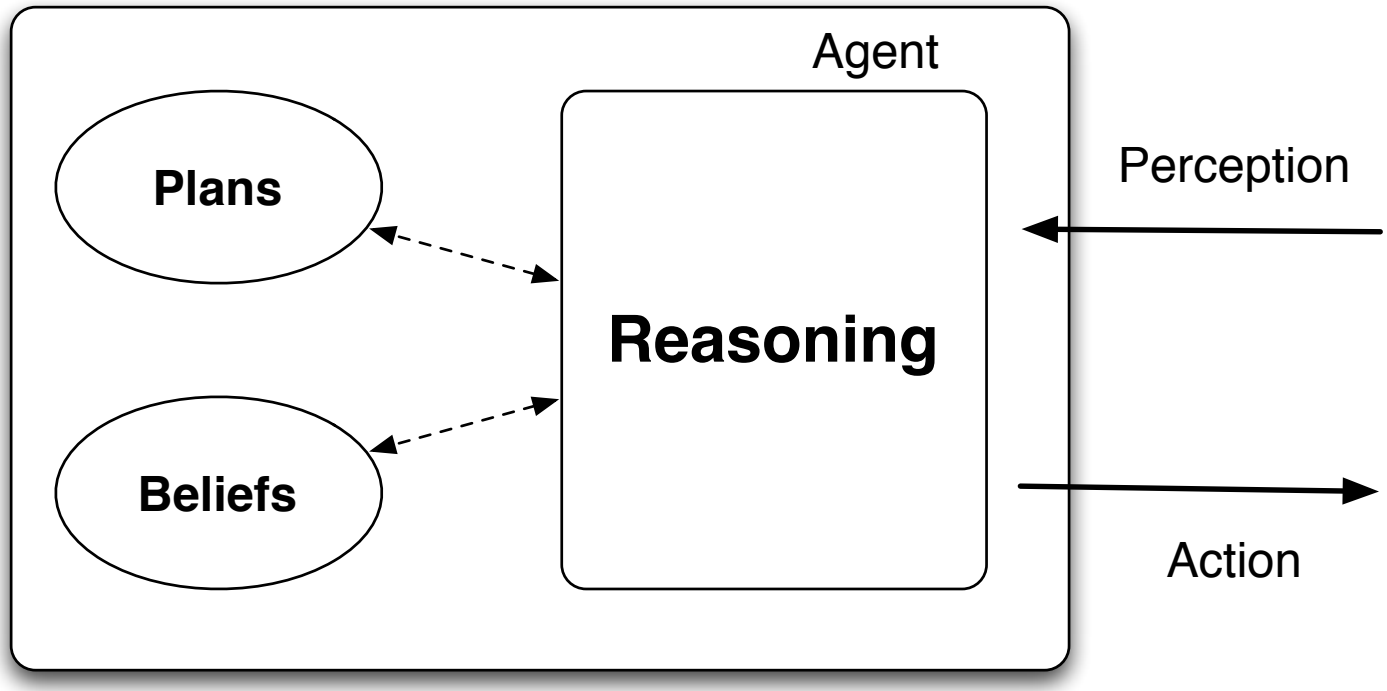
- Cognitive practical reasoning consists of two main activities

  Deliberation when the agent makes decision on what state of affairs the agent desire to achieve

  Means-Ends Reasoning when the agent makes decisions on how to achieve these state of affairs

- The output of the Deliberation phase are the *Intentions*—what agent desires to achieve, or what he desires to do
- The output of Means-Ends is in selecting a given course of actions—the workflow of actions the agent need to do to achieve the Goals

Basic Architecture of a Mentalistic Agent



Intentions and Practical Reasoning I

1. Intentions pose problems for agents who need to determine ways of achieving them. If I have an intention to ϕ, you would expect me to devote resources to deciding how to bring about ϕ.
2. Intentions provide a *filter* for adopting other intentions, which must not conflict. If I have an intention to ϕ, you would not expect me to adopt an intention ψ such that ϕ and ψ are mutually exclusive.
3. Agents track the success of their intentions, and are inclined to try again if their attempts fail. If an agent's first attempt to achieve ϕ fails, then all other things being equal, it will try an alternative plan to achieve ϕ.
4. Agents believe their intentions are possible. That is, they believe that there is at least some way that the intentions could be brought about.

Intentions and Practical Reasoning II

5. Agents do not believe they will not bring about their intentions. It would not be rational for me to adopt an intention to ϕ if I believed ϕ was not possible.
6. Under certain circumstances, agents believe they will bring about their intentions. It would not normally be rational of me to believe that I would bring my intentions about; intentions can fail. Moreover, it does not make sense that if I believe ϕ is inevitable that I would adopt it as an intention.
7. Agents need not intend all the expected side effects of their intentions. If I believe $\phi \rightarrow \psi$ and I intend that ϕ, I do not necessarily intend ψ also. (Intentions are not closed under implication.)
   - This last problem is known as the side effect or package deal problem. I may believe that going to the dentist involves pain, and I may also intend to go to the dentist – but this does not imply that I intend to suffer pain!

Intentions vs. Desires

- The adoption of an intention follows the rise af a given Desire (i.e. follows the adoption of a given Goal);
- Desires and Intentions are different concepts:
  - "My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires [...] before it is settled what I will do".
  - "In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions." [Bratman, 1990]

What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

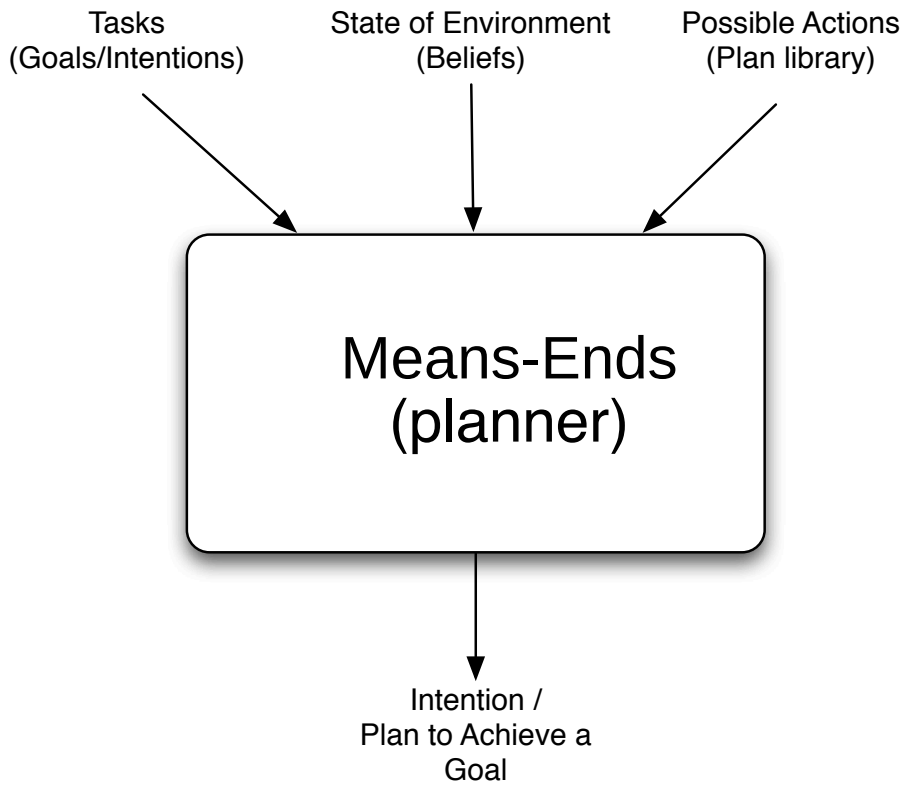
What Means-Ends Reasoning is

Means-Ends Reasoning

- The basic idea is to provide agents with three kinds of Representations:
  - Representation of Goal/Intention to achieve
  - Representation of Actions – Plans – in repertoire
  - Representation of Environment
- Given the environmental conditions, means-ends reasoning will find a Plan to achieve the adopted Goal

Mean-Ends Reasoning

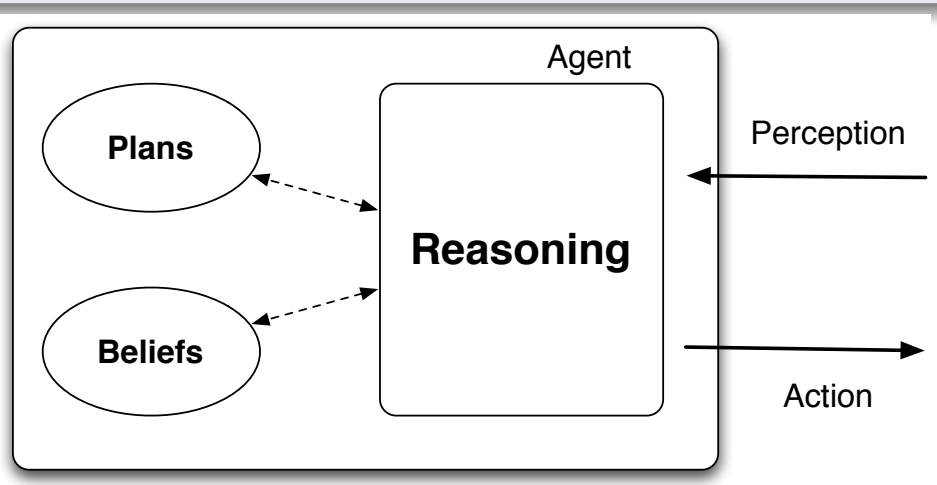
- The selected intention is an emergent property, reified at runtime by selecting a given plan for achieving a given goal.



Implementing a Pratical Reasoning Agent

Agent Control Loop Version 2

```
1: while true
2:     observe the world;
3:     update internal world model (beliefs);
4:     deliberate which intention to adopt next;
5:     use means-ends reasoning to get a plan for the given intention;
6:     execute the plan;
7: end while;
```



Implementing a Pratical Reasoning Agent

- Problem: Agents have Bounded Resources (bounded rationality)
  - Deliberation and Means-Ends processes are not for free: they have computational costs
  - The time taken to reason and the time taken to act are potentially unbounded, thereby destroying the reactivity and the promptness that is essential to agent survival (fitness)
- Suppose the agent
  - starts deliberating at t_0
  - begins means-ends at t_1
  - begins executing a plan at t_2
  - ends executing a plan at t_3
- Time for Deliberation is

  $t_{deliberation} = t_1 - t_0$
- Time for Means-Ends reasoning is

  $t_{meansend} = t_2 - t_1$

- Agents environments are supposed to be highly dynamic
  - Many concurrent changhes may occur during agent decision-making as well as during the execution of plans
- The deliberated intention is worth when it is calculated—say, at t_0
  - At time t_1, the agent will select a goal/intention that would have been optimal if it had been achieved at t_0
- The agent runs the risk that the intention selected is no longer optimal – or no longer achievable – by the time the agent has committed to it

So, this agent will have overall optimal behaviour in the following circumstances:

1. When deliberation and means-ends reasoning take a vanishingly small amount of time
2. When the world is guaranteed to remain (essentially) static while the agent is deliberating and performing means-ends reasoning, so that the assumptions upon which the choice of intention to achieve and plan to achieve the intention remain valid until the agent has completed both deliberation and means-ends reasoning
3. When an intention that is optimal when achieved at t_0 (the time at which the world is observed) is guaranteed to remain optimal until t_2 (the time at which the agent has found a course of action to achieve the intention)

BDI Agents

The abstract architecture proposed by [Rao and Georgeff, 1992] comprise three dynamic and global structures representing agent's *Beliefs*, *Desires* and *Intentions* (BDI), along with an input queue of events.

- Update (write) and Query (read) operations are possible upon the three structures
  - Update operation are subject to compatibility requirements
  - Formalised constraints hold upon the mental attitudes
- The events that the system is able to recognise could be either external (i.e., coming from the environment) or internal ones (i.e., coming from some reflexive action)
  - Events are assumed to be atomic and can be recognized after they have occurred

Implementing a BDI Agent

Agent Control Loop Version 3 [Rao and Georgeff, 1995]

```
1.  initialize-state();
2.  while true do
3.        options := option-generator(event-queue);
4.        selected-options := deliberate(options);
5.        update-intentions(selected-options);
6.        execute();
7.        get-new-external-events();
8.        drop-successful-attitudes();
9.        drop-impossible-attitudes();
10. end-while
```

1. The agent initializes the internal states
2. The agent enters the main loop
3. The option generator reads the event queue and returns a list of options
4. The deliberator selects a subset of options to be adopted and adds these to the intention structure
5. The intentions to be adopted are filtered from the selected ones
6. If there is an intention to perform an atomic action at this point in time the agent executes it
7. Any external events that have occurred during the interpreter cycle are then added to the event queue (the same for internal events)
8. The agent modifies the intention and the desire structures by dropping successful ones
9. Finally, impossible desires and intentions are dropped too

More formally:

Agent Control Loop Version 4

1. $B := B_0$;
2. $I := I_0$;
3. *while true do*
4. *get new percept* Γ;
5. $B := brf(\Gamma, B)$;
6. $I := deliberate(B)$;
8. $\pi := plan(B, I)$;
9. $execute(\pi)$;
10. *end−while*

How does a BDI Agent Deliberate?

Till now, we gave no indication on how reasoning procedures of deliberation and option generation can be made sufficiently fast to satisfy the real time demands placed upon the cognitive system.
Deliberation can be decomposed in two pahses:

Option Generation understand what are the available alternatives

Deliberation choose (and filter) between the adoptable goals/intentions

Chosen options are then intentions, so the agents commit to the selected ones—and executes them.

Refining Deliberation Function

Option Generation the agent generates a set of possible alternatives; represents option generation via a function, options, which takes agent's current beliefs and current intentions, and from them determines a set of options (= desires).

Deliberation the agent chooses between *competing* alternatives, and commits to the intention to achieving them. In order to select between competing options, an agent uses a *filter* function.

- the strategy for deliberating between goals typically is in the hands of the agent developer
- most BDI programming platforms provide mechanisms to describe under which conditions some goal should inhibit the others (Goal Formulae)
- typically, such Goal Formulae are first-order logic predicates indicating contexts and trigger conditions
- Game Theory can enter the picture: i.e., Maximizing 'Expected Utilities'

BDI Agent Control Loop

Agent Control Loop Version 5

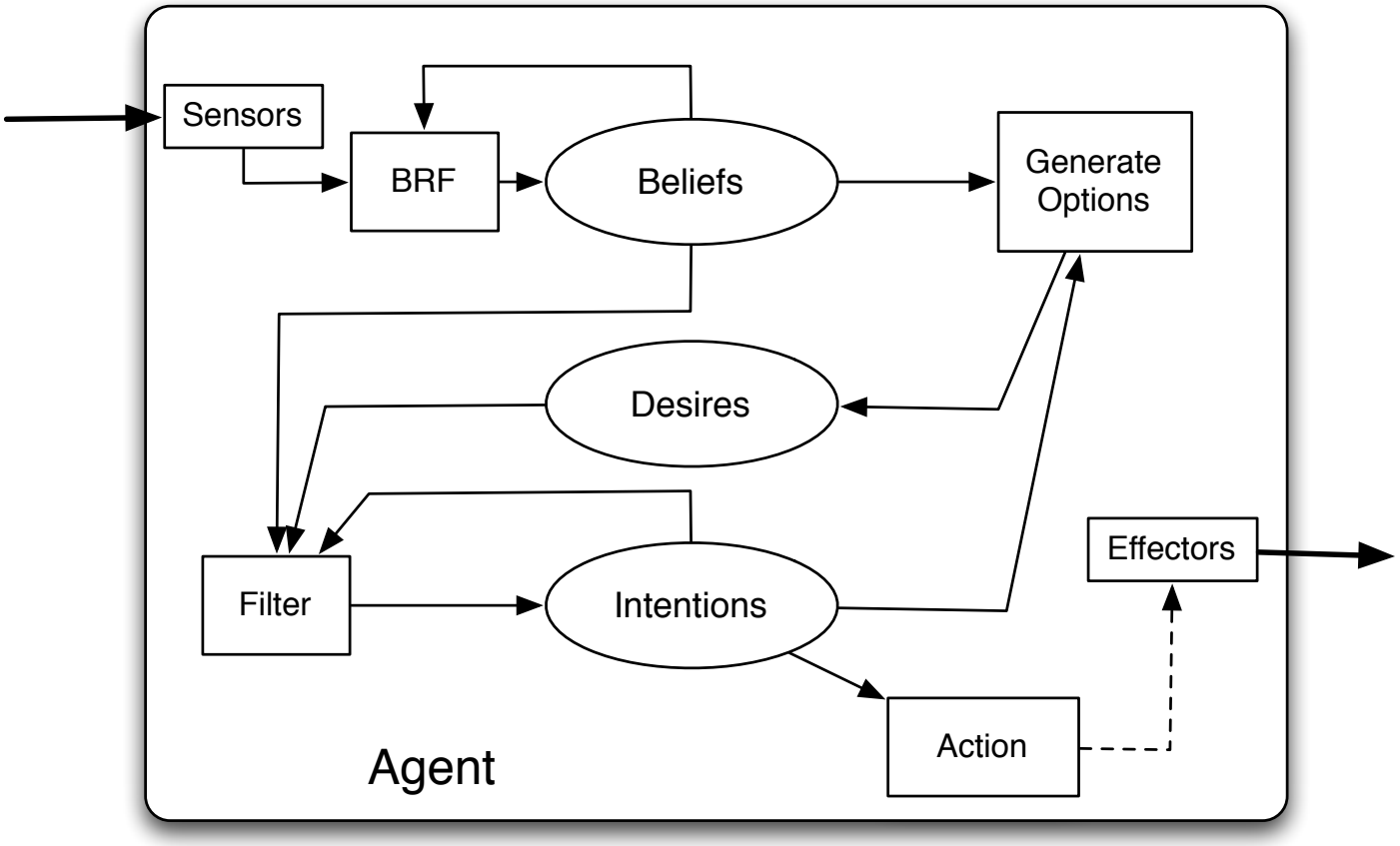
1. $B := B_0$;
2. $I := I_0$;
3. *while true do*
4. *get new percept* Γ;
5. $B := brf(\Gamma, B)$;
6. $D := options(B, I)$;
7. $I := filter(B, D, I)$;
8. $\pi := plan(B, I)$;
9. $execute(\pi)$;
10. *end−while*

Structure of BDI Systems

BDI architectures are based on the following constructs

- A set of Beliefs
- A set of Desires (or Goals)
- A set of Intentions
  - or better, a subset of the goals with an associated stack of plans for achieving them. These are the intended actions
- A set of Internal Events
  - elicited by a belief change (i.e., updates, addition, deletion) or by goal events (i.e. a goal achievement, or a new goal adoption)
- A set of External Events
  - perceptive events coming form the interaction with external entities (i.e. message arrival, signals, etc.)
- A Plan library (repertoire of actions) as a further (static) component

Basic Architecture of a BDI Agent [Wooldridge, 2002]



Post-Declarative Systems

Someone said that this approach leads to a kind of *post-declarative* programming

- In procedural programming, we say exactly what a system should do
- In declarative programming, we state something that we want to achieve, give the system general info about the relationships between objects, and let a built-in control mechanism (e.g., goal-directed theorem proving) figure out what to do
- with intentional agents, we give a very abstract specification of the system, and let the control mechanism figure out what to do, knowing that it will act in accordance with the built-in theory of agency

Actually, the BDI framework combines in an excellent way both (post)declarative structure and procedural knowledge in terms of plans.

Beliefs

Beliefs

Agents knowledge is structured in Beliefs about the current state of the world

- they are informational units, typically implemented as ground sets of literals, possibly with no disjunctions or implications
- they should reflect only the information which is currently held (i.e. situated)
- they are expected to change in the future, i.e., as well as the environment changes

Plans

Plans

Plans represent the means the agent has to change the world, and to bring it closer to his desires

- they are language constructs, typically implemented in the form of procedural structures
- plans have a 'body', describing the workflow of activities (actions) that have to be executed for plan execution to be succesful
- the conditions under which a plan can be chosen as an option are specified in an *invocation condition* (triggering event) and a *pre- or context- condition* (situation that must hold for the plan to be executable)

Intentions

Intentions

Intentions are emergent properties reified at runtime by selecting a given plan for achieving a given goal

- Represented 'on-line' using a run-time stack of hierarchically plans related to the ongoing adopted goals
- Similarly to how Prolog interpreter handle clauses
- Multiple intention stacks can coexist, either running in parallel, suspended until some condition occurs, or ordered for execution in some way

BDI Agents Programming Platforms

JASON (Brasil) `http://jason.sourceforge.net/` Agent platform and language for BDI agents based on AgentSpeak(L)

JADEX (Germany) `http://jadex.informatik.uni-hamburg.de/` Agent platform for BDI and Goal-Directed Agents

2APL (Netherlands) `http://www.cs.uu.nl/2apl/` Agent platform providing programming constructs to implement cognitive agents based on the BDI architecture

3APL (Netherlands) `http://www.cs.uu.nl/3apl/` A programming language for implementing cognitive agents

PRACTIONIST (Italy) `http://www.practionist.org/` Framework built on the Bratman's theory of practical reasoning to support the development of BDI agents

Bibliography I

Bratman, M. E. (1987).
Intention, Plans, and Practical Reason.
Harvard University Press.

Bratman, M. E. (1990).
What is intention?
In Cohen, P. R., Morgan, J. L., and Pollack, M. E., editors, *Intentions in Communication*, pages 15–32. The MIT Press, Cambridge, MA.

Dennett, D. (1971).
Intentional systems.
Journal of Philosophy, 68:87–106.

Dennett, D. (2007).
Intentional systems theory.
In Beckermann, A. and Walter, S., editors, *Oxford Handbook of the Philosophy of Mind*, chapter 19. Oxford University Press.

McCarthy, J. (1979).
Ascribing mental qualities to machines.
In Ringle, M., editor, *Philosophical Perspectives in Artificial Intelligence*, Harvester Studies in Cognitive Science, pages 161–195. Harvester Press, Brighton.

Bibliography II

Rao, A. S. and Georgeff, M. P. (1992).
An abstract architecture for rational agents.
In Nebel, B., Rich, C., and Swartout, W. R., editors, *3rd International Conference on Principles of Knowledge Representation and Reasoning (KR '92)*, pages 439–449, Cambridge, MA, USA. Morgan Kaufmann.
Proceedings.

Rao, A. S. and Georgeff, M. P. (1995).
BDI agents: From theory to practice.
In Lesser, V. R. and Gasser, L., editors, *1st International Conference on Multi Agent Systems (ICMAS 1995)*, pages 312–319, San Francisco, CA, USA. The MIT Press.

Russell, S. J. and Norvig, P. (2002).
Artificial Intelligence: A Modern Approach.
Prentice Hall / Pearson Education International, Englewood Cliffs, NJ, USA, 2nd edition.

Wooldridge, M. J. (2002).
An Introduction to MultiAgent Systems.
John Wiley & Sons Ltd., Chichester, UK.

Agents as Intentional Systems

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Michele Piunti
{andrea.omicini, michele.piunti}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2009/2010