Reactive and Hybrid Agents

Marco Alberti

Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

Multi-Agent Systems, 2011/2012
Slides updated on April 2, 2012 at 14:53
1 Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2 Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Reactive architectures

Rejection of the symbolic approach

Criticism of the symbolic AI approach

- Mid to late 80s
- The symbolic approach is not viable due to
  - unrealistic representation;
  - even when possible, complex representation;
  - computational complexity of symbolic manipulation algorithms for decision making.
- Minor changes to the approach (such as weakening the logical representation language) are not sufficient for agents to operate in time-constrained environments.
Diverse approaches, which share some points:

- The symbolic approach is not fruitful in real applications
- Intelligence cannot be isolated from interaction
- Intelligence emerges
These approaches are usually referred to by the following terms:

- **behavioral**: developing and combining individual behaviors
- **situated**: agents are defined as situated in some environment, rather than disembodied from it
- **reactive**: agents’ main task is to react to an environment rather than reasoning on it.
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
The nature of intelligence

Theses proposed by Brooks in the early 90s [Brooks, 1991a, Brooks, 1991b]:

- Intelligent behavior can be generated *without* explicit representation of the kind that symbolic AI proposes;
- Intelligent behavior can be generated *without* explicit abstract reasoning of the kind that symbolic AI proposes;
- Intelligence is an *emergent* property of certain complex systems.
Real intelligence is situated in the world, not in disembodied systems such as theorem provers or expert systems.
Intelligence and emergence

- Intelligent behavior arises as a result of an agent’s interaction with its environment
- Intelligence is “in the eye of the beholder”.
The subsumption architecture

Defining characteristics
- task accomplishing behaviors
- subsumption hierarchy
Task accomplishing behaviors

- Define the agent’s decision making.
- Each can be thought as an individual action function, which maps percepts to actions.
- In Brook’s implementation, behavior modules are finite state machines.
- No symbolic representation, no symbolic reasoning
- In many implementations, rules of the form situation $\rightarrow$ action
Subsumption hierarchy

- Many behaviors can fire simultaneously
- How to choose among different actions?
- *Subsumption hierarchy*, with behaviors arranged into *layers*.
- Lower layers *inhibit* higher layers; i.e., lower layer, higher priority.
see function unchanged

*action* function defined by
- a set of behaviors
- an inhibition binary relation between behaviors
Behaviors

Definition
Given a set $Per$ of percepts and a set $Ac$ of actions, a behavior is a pair

$$\langle c, \alpha \rangle$$

where $c \subseteq Per$ is a set of percepts called the condition, and $\alpha \in Ac$.

Definition
A behavior $\langle c, \alpha \rangle$ fires in a state $e \in E$ if and only if $\text{see}(e) \in c$

Definition
The set of all possible behaviors is

$$Beh = \{\langle c, \alpha \rangle \mid c \subseteq Per \land \alpha \in Ac\}$$
Inhibition relation

Definition

Given a set of behaviors \( R \subseteq \text{Beh} \), the associated inhibition relation is a total order over \( R \):

- \( \prec \subseteq R \times R \)
- \( \forall b \in R \; b \not\prec b \) (irreflexive)
- \( \forall b, b' \in R \) it can’t be the case that \( b \prec b' \land b' \prec b \) (antisymmetric)
- \( \forall b, b', b'' \in R \; b \prec b' \land b' \prec b'' \rightarrow b \prec b'' \) (transitive)
- \( \forall b, b' \in R \; b \prec b' \lor b' \prec b \lor b' = b \) (total)
Action selection algorithm

**input**: A percept $p \in Per$

**output**: An action $\alpha \in Ac$

**begin**

```plaintext
fired ← \{⟨c, \alpha⟩ ∈ R | p ∈ c\};

foreach ⟨c, \alpha⟩ ∈ fired do
  if ¬∃⟨c′, \alpha′⟩ ∈ fired | ⟨c′, \alpha′⟩ < ⟨c, \alpha⟩ then
    return \alpha;
  end
end

return null;
end
```
Overall time complexity of the subsumption action function is $O(n^2)$

where $n$ is the larger of
- number of behaviors
- number of percepts

But it would be possible to encode decision making logic in hardware, with constant, micro-second decision time.
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Mars explorer

[Steels, 1990]

- Objective: to collect rock samples on a planet
- Location of rocks not known, but they are typically clustered in certain spots
- Samples are collected by a number of autonomous vehicles
- Samples have to be delivered to the mother ship
- Hills and obstacles prevent direct communications between vehicles
Solution

- Logic-based solution “entirely unrealistic” [Steels, 1990].
- Solution based on the subsumption architecture
- Localization: the mothership emits a field whose value decreases with the distance $\Rightarrow$ field gradient points towards the mothership
- Communication: radioactive crumbs
Individually operating agents

Behaviors

1. detect an obstacle \(\implies\) change direction
2. carrying samples \& at the base \(\implies\) drop samples
3. carrying samples \& not at the base \(\implies\) travel up gradient
4. detect a sample \(\implies\) pick sample up
5. true \(\implies\) move randomly

Subsumption hierarchy:

\[1 < 2 < 3 < 4 < 5\]
The subsumption architecture ensures that:

- An agent will *always* change direction if an obstacle is detected.
- An agent will drop a sample if it is near the mother ship, *unless* that would make the agent crash.
- The random walk will only be carried out if the agent has nothing more urgent to do.

This solution would be appropriate if rock samples were placed randomly,

but they are clustered: near where one is found, there are probably more. Why not use this information?
Cooperative solution

- Direct communication is impossible
- An agent will create a trail of radioactive crumbs when it is carrying a sample to the mother ship
- When other agents find the trail, they will go down the gradient
- Refinements:
  - when an agent follows a trail down the gradient, it will pick up some of the crumbs, so as to weaken it
  - if agents follow a trail and do not find new samples, the trail will eventually disappear
New version

Behavior 3 is modified and 6 is added

3. carrying samples $\land$ not at the base $\implies$ drop 2 crumbs
   $\land$ travel up gradient

6. sense crumbs $\implies$ pick up one crumb $\land$ travel down gradient

Subsumption architecture

1 $<$ 2 $<$ 3 $<$ 4 $<$ 6 $<$ 5
Situated automata

[Kaelbling and Rosenschein, 1990]
[Kaelbling, 1991]

- An agent can be conceptualized in logical terms without being implemented as a theorem prover
- In the situated automata paradigm:
  - An agent is specified declaratively
  - The specification is compiled to a machine
  - The machine can operate in a provably time-bounded fashion
  - It contains no symbolic representation and does no symbol manipulation
Agent Network Architecture

[Maes, 1991]

- An agent is a set of competence modules
- Each has:
  - Preconditions
  - Postconditions
  - Activation level: a real-valued indicator of the module’s relevance in a particular situation
- Modules compiled into a spreading activation network: modules have relations depending on preconditions and postconditions.
  - E.g., if A’s postcondition coincides with B’s precondition, then B is A’s successor.
  - Other types of links: predecessor, conflicter.
- Reminds of neural networks, but here nodes have semantics.
Outline

1 Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2 Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Access to local information

- Agents do not have a model of their environment
- Perception only gives access to *local* information (in *space* and *time*)
- Does the local environment bear sufficient information?
- Necessarily short-term view
Emergent behaviour

- Selling point
- Emergent behavior is, by definition, not understandable structurally
- Impossible to design and engineer, except by trial and error
Difficulties with many-layered agents

- Approach effective when agents have few layers (less than 10)
- With more layers, the dynamics of behavior interaction becomes too complex to understand
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Layering agents

- Balancing reactive and proactive behavior
- Agents composed of two or more interacting layers
- Typically at least one reactive and one proactive layers, possibly more
- Ways to organize layers:
  - horizontal layering
  - vertical layering
Horizontal layering
Horizontal layering

- Each layer accesses the input (perception) and output (action) subsystems
- In fact, each layer is an agent, which suggests one action to perform
- The overall behavior may not be coherent
- Necessity of a mediator function to select which suggested action is to be performed
- \( n \) layers each possibly suggesting \( m \) actions cause \( m^n \) possible interactions: hard to manage
Vertical layering

- One-pass: information sequentially through each layer
- Two-pass: information flows up and control flows down
- Less flexibility: information must pass through all layers, not fault tolerant
Hybrid architectures

Layered architectures

One-pass
Two-pass

Layered architectures

Hybrid architectures

- layer $n$
- ... (omitted)
- layer 2
- layer 1

perceptual input
action output
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Hybrid architectures

TouringMachines

[Ferguson, 1992]
Hybrid architectures

TouringMachines

- Horizontally layered architecture
- Three *activity producing layers*:
  - *reactive* layer
  - *planning* layer
  - *modeling* layer
Reactive layer

- Set of situation-reaction rules

**rule-1: kerb-avoidance**

if is-in-front(Kerb, Observer) and speed(Observer) > 0 and separation(Kerb, Observer) < KerbThreshold
then change-orientation(KerbAvoidanceAngle)
Planning layer

- Decides what the agent does under normal circumstances
- Does not plan: employs a library of plan skeletons called *schemas*.
- To achieve a goal, the planning layer looks in the library for a schema that matches the goal
- The schema may contain sub-goals.
Modeling layer

- Represents the various entities in the world (including the agent itself)
- Predicts conflicts between agents
- Generates goals to resolve the conflicts
- Passes the goals down to the planning layer.
Control subsystem

- Decides which layer should actually have control
- Implemented as a set of control rules.
- Control rules can suppress sensor information or censor action outputs.

**censor-rule-1:**

if entity(obstacle-6) in perception-buffer
then remove-sensory-record(layer-R, entity(obstacle-6))
Outline

1. Reactive architectures
   - Rejection of the symbolic approach
   - The subsumption architecture
   - Example reactive agent architectures

2. Hybrid architectures
   - Limitations of reactive agents
   - Layered architectures
   - TouringMachines
   - InteRRaP
Hybrid architectures

[ Müller, 1997]

cooperation layer

plan layer

behavior layer

social knowledge

planning knowledge

world model

world interface

perceptual input

action output

InteRRaP
InteRRaP architecture

- Three layers:
  - *behavior* layer: deals with reactive behavior
  - *plan* layer: deals with achieving the agent’s goals
  - *cooperative* layer: deals with social interactions

- Each layer maintains a *knowledge base*: a representation of the world at an appropriate level of abstraction.

- Each layer implements two general functionalities:
  - *situation recognition and goal activation*: maps current situation and goals to new goals
  - *planning and scheduling*: selects which plan to execute to achieve the current goal
Layer interactions

- Vertically layered, 2-pass architecture
- Differently from TouringMachines, layers interact with each other rather than directly with the input and output subsystems.
- Two types of interaction:
  - *bottom-up activation*: a layer passes control to a higher layer when it is not *competent* to manage the current situation
  - *top-down execution*: a layer passes control to a lower layer to achieve one of its goals
Layered architectures

- Most popular agent architecture
- Easy to understand: behaviour specification is clearly decomposed in its
  - reactive,
  - proactive,
  - and social
  parts.
- Hard to give formal semantics.


