LECTURE 5: REACTIVE AND HYBRID ARCHITECTURES

An Introduction to Multiagent Systems

http://www.csc.liv.ac.uk/~mjw/pubs/imas/
There are many unsolved (some would say insoluble) problems associated with symbolic AI.

0.1 Reactive Architectures

These problems have led some researchers to question the viability of the whole paradigm, and to the development of alternative architectures.

Although united by a belief that the assumptions underpinning mainstream AI are in some sense wrong, reactive agent researchers use many different techniques.

If this presentation, we start by reviewing the work of one of the most vocal critics of mainstream AI: Rodney Brooks.

There are many unsolved (some would say insoluble) problems associated with symbolic AI.

...
Brooks has put forward three theses:

1. Intelligent behaviour can be generated without explicit representations of the kind that symbolic AI proposes.
2. Intelligent behaviour can be generated without explicit abstract reasoning of the kind that symbolic AI proposes.
3. Intelligence is an emergent property of certain complex systems.

Brooks—behaviour languages
He identifies two key ideas that have informed his research:

1. Situatedness and embodiment: "Real" intelligence is situated in the world, not in disembodied systems. Also, "real" intelligence is situated in the eye of the beholder, it is not an innate property.

2. Intelligence and emergence: Intelligent behaviour arises as a result of an agent's interaction with its environment. Also, intelligence is 'in the eye of the beholder'; it is not an innate property.
To illustrate his ideas, Brooks built some based on his subsumption architecture by symbolic AI systems.

- Some of the robots do tasks that would be impressive if they were accomplished by symbolic AI systems.
- They do, extremely simple.
- The resulting systems are, in terms of the amount of computation they do, extremely simple.

A subsumption architecture is a hierarchy of task-accomplishing behaviors. Lower layers represent more primitive kinds of behavior, such as avoiding obstacles, and have precedence over layers further up the hierarchy. Each behavior competes with others to exercise control over the agent.

- Each behavior is a rather simple rule-like structure.
- Each behavior competes with others to exercise control over the agent.

- The resulting systems are, in terms of the amount of computation they do, extremely simple.
- Some of the robots do tasks that would be impressive if they were accomplished by symbolic AI systems.
The objective is to explore a distant planet, and in particular, to simulate rock gathering on Mars' domain. Stiebels' Mars explorer system, using the subsumption architecture, achieves near-optimal cooperative performance in collecting samples of a precious rock. The location of the samples is not known in advance, but it is known that they tend to be clustered.

An Introduction to Multiagent Systems
For individual (non-cooperative) agents, the lowest-level behavior, (and hence the highest priority), is obstacle avoidance: (and hence the behavior with the highest priority is)
Agents will collect samples they find:

(4) If detect a sample then pick sample up.

(5) If true then move randomly.

An agent with "nothing better to do" will explore randomly:

- An agent with "nothing better to do" will explore randomly.

Agents will collect samples they find:
0.3 Situated Automata

A sophisticated approach is that of Rosenschein and Kaelbling. In their situated automata paradigm, an agent is specified in a rule-like (declarative) language, and this specification is then compiled down to a digital machine, which satisfies the declarative specification. This digital machine can operate in a provable time bound. Reasoning is done offline, at compile time, rather than online at run time.
The theoretical limitations of the approach are not well understood.

Compilation (with propositional specifications) is equivalent to an NP-complete problem.

The more expressive the agent specification language, the harder it is to compile it.

(There are some deep theoretical results which say that after a certain expressiveness, the compilation simply can't be done.)
Many researchers have argued that neither a completely deliberative nor a completely reactive approach is suitable for building agents.

They have suggested using hybrid systems, which attempt to marry classical and alternative approaches. An obvious approach is to build an agent out of two (or more) subsystems:

- a deliberative one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI; and
- a reactive one, which is capable of reacting to events without complex reasoning.

A hybrid architecture can attempt to marry classical and alternative approaches.
Often, the reactive component is given some kind of precedence.

This kind of structuring leads naturally to the idea of a layered architecture, of which TURINGMACHINES and INTERRAP are examples.

In such an architecture, an agent's control subsystems are arranged into a hierarchy, with higher layers dealing with information at increasing levels of abstraction.

Often, the reactive component is given some kind of precedence.

An Introduction to Multiagent Systems
A key problem in such architectures is what kind control framework to embed the agent's subsystems in, to manage the interactions between the various layers.

In effect, each layer itself acts like an agent, producing suggestions as to what action to perform.

Horizontal layering:

- Layer(s) are each directly connected to the sensory input and action output.

Vertical layering:

- There are no layers between the various layers.
- Sensory input and action output are each dealt with by at most one layer each.
(b) Vertical layering

(One pass control)

(c) Vertical layering

(a) Horizontal layering

Layer 1
Layer 2
...
Layer n

Layer 1
Layer 2
...
Layer n

Layer 1
Layer 2
...
Layer n

Input
Output
Perceptual
Action

Layer 1
Layer 2
...
Layer n

Layer 1
Layer 2
...
Layer n

Layer 1
Layer 2
...
Layer n

Layer 1
Layer 2
...
Layer n

Input
Perceptual
Action
Output
The Turing Machines architecture consists of perception and action subsystems, which interface directly with the agent's environment, and three control layers, embedded in a control framework, which mediates between the layers.
execute in order to achieve the agent’s goals.

The planning layer constructs plans and selects actions to achieve the agent’s goals.

Example:

```
rule-1: kerb-avoidance
  if is-in-front(Kerb, Observer) and speed(Observer) > 0 and separation(Kerb, Observer) > KerbThreshold then change-orientation(KerbAvoidanceAngle)
```

The reactive layer is implemented as a set of situation-action rules, a la subsumption architecture.
Lecture 5: An Introduction to Multiagent Systems

The modelling layer contains symbolic representations of the 'cognitive state' of other entities in the agent's environment.

The three layers communicate with each other and are embedded in a control framework, which use control rules.

Example:

```
<table>
<thead>
<tr>
<th>censor-rule-1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>if entity(obstacle-6) in perception-buffer</td>
</tr>
<tr>
<td>then remove-sensory-record(layer-R, entity(obstacle-6))</td>
</tr>
</tbody>
</table>
```

- The modelling layer contains symbolic representations of the 'cognitive state' of other entities in the agent's environment.
- The three layers communicate with each other and are embedded in a control framework, which use control rules.