An introduction to INTELLIGENT AGENTS

INTELEINCI ARTIFICIAL II
INSTITUTO UNIVERSITARIO GASTÓN DACHARY

CARLOS MANUEL TOLEDO - HÉCTOR RUIDIAS

“La inteligencia artificial nunca superará la estupidez natural”
INTRODUCTION

- The computers are obedient, literal, unimaginative.
- Sometimes, we require systems that can decide for themselves what they need to do in order to satisfy their design objectives. These systems are called agents.
- It must operate in rapidly changing, unpredictable, or open environments.
A definition:

- **Agent:**
  - An autonomous entity that can interact with its environment

- **Software agent:**
  - An autonomous software entity that can interact with its environment.
WHAT ARE AGENTS?

- **Autonomy** is central to the notion of agency.
- “An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives” [Wooldridge, 1995]
- The definition refers to “agents” and not “intelligent agents”
- It does no say anything about what type of environment an agent occupies.
- **Autonomy** is a somewhat tricky concept to tie down precisely. (Agents are able to act without the intervention of humans or other systems, they have control both over their own internal state, and over their behavior [Wooldridge])
WHAT ARE AGENTS?

Diagram:
- Environment
- Percepts
- Actions
- Actuators
- Sensors
- Agent

CARLOS MANUEL TOLEDO
cmtoledo@dachary.edu.ar
AGENT = ARCHITECTURE + PROGRAM
AGENTS’ FEATURE

- An agent will have a repertoire of actions available to it.
- This set represents the agent's effectoric capability (its ability to modify its environments).
- The actions have preconditions associated with them, which define the possible situations in which they can be applied.
- The key problem facing an agent is that of deciding which of its actions it should perform in order to best satisfy its design objectives.
ENVIRONMENT PROPERTIES

AGENT

ENVIRONMENT

Sensor input

Action output
ENVIRONMENT PROPERTIES

- Accessible vs inaccessible
- Deterministic vs non-deterministic
- Episodic vs non-episodic
- Static vs dynamic
- Discrete vs continuous
When do we consider an agent to be intelligent?

- When it is capable of **FLEXIBLE AUTONOMOUS** action in order to meet its design objectives.
- Flexible is:
  - Reactivity (perceive their environment, and respond)
  - Pro-activeness (initiative)
  - Social ability (interacting with other agents or humans)
Reactive behaviour

- If a program’s environment is guaranteed to be fixed, the program need never worry about its own success or failure – program just executes blindly.
- The real world is not like that: things change, information is incomplete. Many interesting environments are dynamic.
- Software is hard to build for dynamic domains: program must take into account possibility of failure.
- A reactive system is one that maintains an ongoing interaction with its environment, and responds to changes that occur in it.
Proactiveness behaviour

- Reacting to an environment is easy (e.g., stimulus → response rules)
- But we generally want agents to do things for us.
- *Hence goal directed behavior*
- Pro-activeness = generating and attempting to achieve goals; not driven solely by events; taking the initiative.
- Recognizing opportunities.
Balancing Reactive and Goal-Oriented Behavior

- Building purely goal directed systems is not hard.
- Is hard building a system that achieves an affective balance between goal-directed and reactive behaviour.
- We want agents that will attempt to achieve their goals systematically, but we don’t want our agents to continue blindly executing actions in an attempt to achieve a goal when it is clear that these actions will not work, or when the goal is for some reason no longer valid.
- In such circumstances, we want our agent to be able to react to the new situation.
However, we do not want our agent to be continually reacting and hence never focusing on a goal long enough to actually archive it.
Social ability

- The ability to exchange bit streams is not really social ability.
- The humans to achieve their goals, they must negotiate and cooperate with others. They may be required to understand and reason about the goals of others, and to perform actions that we would otherwise choose to perform.
Social ability

The problem:

- Mixture of different agents.
- Mixture of simple and sophisticated agents.
Social ability

- The main topics of social ability are:
  - Negotiation
  - Cooperation
Other properties discussed in the context of agency:

- **mobility**: the ability of an agent to move around an electronic network
- **veracity**: an agent will not knowingly communicate false information
- **benevolence**: agents do not have conflicting goals. **rationality**: agent will act in order to achieve its goals, and will not act in such a way as to prevent its goals being achieved — at least insofar as its beliefs permit
- **learning/adaption**: agents improve performance over time
AGENTS AND OBJECTS

- Are agents just objects by another name?
- Object:
  - encapsulates some state
  - communicates via message passing
  - has methods, corresponding to operations that may be performed on this state
Main differences:

- **Agents are autonomous:**
  Agents embody stronger notion of autonomy than objects, and in particular, they decide for themselves whether or not to perform an action on request from another agent.

- **Agents are smart:**
  Capable of flexible (reactive, pro-active, social) behavior, and the standard object model has nothing to say about such types of behavior.

- **Agents are active:**
  A multi-agent system is inherently multi-threaded, in that each agent is assumed to have at least one thread of active control.
OBJECTS DO IT FOR FREE,
AGENTS DO IT FOR MONEY
AGENTS AND EXPERT SYSTEMS

- An expert system is one that is capable of solving problems or giving advice in some knowledge-rich domain.
- The most important distinction between agents and expert systems is that expert systems are inherently disembodied (don’t interact directly with any environment, they get their information not via sensors, but through a user acting as middle man).

THEY DO NOT ACT IN ANY ENVIRONMENT AND, THEY GENERALLY DON’T BE CAPABLE OF CO-OPERATING WITH OTHER AGENTS
ABSTRACT ARCHITECTURES
ABSTRACT ARCHITECTURES

- Environment state
  \[ S = \{ s_1, s_2, \ldots \} \]

- The effectoric capability (actions)
  \[ A = \{ a_1, a_2, \ldots \} \]

- Abstract agent
  \[
  \text{action: } S^* \rightarrow A \\
  \text{Maps sequences of environment states to actions}
  \]

- The non-deterministic behaviour of an environment
  \[ \text{env: } S \times A \rightarrow \delta(S) \]
- **History**

A history \( h \) is a sequence

\[
\begin{align*}
h &: s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \ldots
\end{align*}
\]

- \( \text{hist}(\text{agent}, \text{environment}) \rightarrow \) set of all histories of \textit{agent} in \textit{environment}.

- Iff \( \text{hist}(ag1, env) = \text{hist}(ag2, env) \) they are behaviourally equivalent with respect to all environments
PURELY REACTIVE AGENTS

ABSTRACT ARCHITECTURES
PURELY REACTIVE AGENTS

- They base their decision making entirely on the present, with no reference at all to the past.
- They simply respond directly to their environment.
  \[ \text{action } S \rightarrow A \]
- There is an equivalent standard agent.
PURELY REACTIVE AGENTS

AGENT

see → action

ENVIRONMENT
PURELY REACTIVE AGENTS

- Exists a separation of an agent’s decision function into perception an action subsystems.
- The function *see* capture the agent’s ability to observe its environment, whereas the *action* function represents the agent’s decision making process.

\[
\text{see: } S \rightarrow P \\
\text{action: } P^* \rightarrow A
\]
AGENT WITH STATE

ABSTRACT ARCHITECTURES
These agents have some internal data structure, which is typically used to record information about the environment state and history.
AGENTS WITH STATE

AGENT 

see 

next 

state 

ENVIRONMENT 

action
LOGIC-BASE ARCHITECTURES

- Traditional approach
- Intelligent behaviour can be generate in a system by giving that system a symbolic representation of its environment and its desired behaviour, and syntactically manipulating this representation.
- These representation are logical formulae, and the syntactic manipulation corresponds to logical deduction, or theorem proving.
Let $L$ be the set of sentences of classical first-order logic, and let $D = \wp(L)$ be the set of $L$ databases, i.e., the set of sets of $L$-formulae. The internal state of an agent is then an element of $D$. We write $\Delta, \Delta_1, \ldots$ for members of $D$. The internal state of an agent is then simply a member of the set $D$. An agent’s decision making process is modelled through a set of deduction rules, $\rho$. These are simply rules of inference for the logic. We write $\Delta \vdash_\rho \phi$ if the formula $\phi$ can be proved from the database $\Delta$ using only the deduction rules $\rho$. An agent’s perception function $\text{see}$ remains unchanged:

$\text{see} : S \rightarrow P.$

Similarly, our $\text{next}$ function has the form

$\text{next} : D \times P \rightarrow D$

It thus maps a database and a percept to a new database. However, an agent’s action selection function, which has the signature

$\text{action} : D \rightarrow A$
function action(Δ : D) : A
begin
    for each a ∈ A do
        if Δ ⊧ ρ Do(a) then
            return a
        end-if
    end-for
    for each a ∈ A do
        if Δ ⊨ ρ ¬Do(a) then
            return a
        end-if
    end-for
    return null
end function action
In the mid-to-late 1980s, the researchers began to investigate alternatives to the symbolic AI paradigm, due intractable problems with symbolic/logical approaches to building agents.

**Characteristics for a new approach:**
- Rejection of symbolic representations, and of decision making based on syntactic manipulations of such representations.
- The idea that intelligent, rational behaviour is seen as innately linked to the environment an agent occupies – intelligent behaviour is not disembodied, but is a product of the interaction the agent maintains with its environment.
- The idea that intelligent behaviour emerges from the interaction of various simpler behaviours.
Agency as referred to as behavioural (combining individual behaviours) situated (in some environment) as reactive (it perceived as simply reacting to an environment, without reasoning about it).

- Ex: Subsumption architecture (Rodney Brooks)
- Characteristics (I):
  - An agent’s decision-making is realized through a set of task accomplishing behaviours (each behaviour is a individual actions function, which continually takes perceptual input and maps it to an action to perform).
  - Each of these behaviour modules (include no complex symbolic representations, and are assumed to do no symbolic reasoning) is intended to achieve some particular task.

situation $\rightarrow$ action 

map perceptual input directly to actions
accomplishing: logrando
Carlos Manuel Toledo, 13/06/2007
Characteristics (II):

- Many behaviours can “fire” simultaneously. There must be a mechanism to choose between the different actions selected by these multiple actions. Brooks proposed arranging the modules into a subsumption hierarchy, with the behaviours arranged into layers. Lower layers are able to inhibit higher layers.

Ex: a robot, “avoid obstacles” it has a high priority then it is into low-level layer
The decision function “action” is realized through a set of behaviors, together with an inhibition relation holding between these behaviors.

A behavior is a pair \((c,a)\)
- \(c \subseteq P\) is a set of percepts called the condition.
- \(a \in A\) is an action

A behavior \((c,a)\) will fire when the environment is in state \(s \in S\) iff \(\text{see}(s) \in c\)

\[
\text{Beh} = \{(c,a) \mid c \subseteq P \text{ and } a \in A\}
\]

Inhibition relation
- Associated with an agent’s set of behavior rules \(R \subseteq \text{Beh}\) is a binary inhibition relation on the set of behaviors \(\subseteq R_xR\)
- It is transitive, irreflexive, and antisymmetric.
- \(b_1 \overset{\text{inhibit}}{\rightarrow} b_2\) iff \((b_1,b_2) \in \subseteq\), and read “\(b_1\) inhibits \(b_2\)”, that is, \(b_1\) is lower in the hierarchy than \(b_2\)
function action(p : P) : A
var fired : ϕ(R)
var selected : A
begin
    fired := \{(c, a) | (c, a) ∈ R and p ∈ c\}
    for each (c, a) ∈ fired do
        if ¬(∃(c', a') ∈ fired such that (c', a') < (c, a)) then
            return a
        end-if
    end-for
    return null
end function action
Example: An agent that explores a distant planet and collects samples of a particular type of precious rock.

- A number of autonomous vehicles are available. Tart can dive around the planet collecting samples and later reenter the a mothership spacecraft to go back to earth. They cooperate to collect rock samples.
- The mothership generate a radio signal (The agents seeks this signal through a “gradient”)
- An agent can communicate with other agent through “radioactive crumbs”

Behaviour:
- If detect an obstacle then change direction (1)
- If carrying samples and at the base then drop samples (2)
- If carrying samples and not at the base travel up gradient of “radio signal” (3)
- If detect a sample then pick samples up (4)
- If true then move randomly (5)
Agent cooperation:
- The samples tend to be located in clusters, then it make sense to have agents cooperate with one-another in order to find the samples.
- An agent creating a “tail” of radioactive crumbs whenever it finds a rock samples.
- Trail is only laid by agents returning to the mothership. Hence if an agent follows the trail out to the source of the nominal rock sample only to find that it contains no samples, it will reduce the trail on the way out, and will not return with samples to reinforce it.
REACTIVE ARCHITECTURES

- Behaviour modified:
  - If detect an obstacle then change direction (1)
  - If carrying samples and at the base then drop samples (2)
  - If carrying samples and not at the base then drop 2 crumbs and travel up gradient of “radio signal” (3)
  - If detect a sample then pick samples up (4)
  - If sense crumb then pick up 1 crumb and travel down gradient (5)
  - If true then move randomly (6)
BELIEF-DESIRE-INTENTION

- These have their roots in the philosophical tradition of understanding practical reasoning (The process of deciding, moment by moment, which action to perform in the furtherance of our goals).
- Practical reasoning involves two important processes:
  - Deciding what goals we want to achieve (deliberation)
  - And, how we are going to achieve these goals (means-ends reasoning).
means-ends: medios para encontrar un fin

14/06/2007
Example: when you leave university. Decision about what to do with your life.

- The decision process typically begins by trying to understand what the options available to you is (if you gain a good first degree, then one option is that of becoming an academic, else this option is not available to you). Another option is entering industry.

- After generating this set of alternatives, you must choose between them, and commit to some.

- This chosen option become intentions, which then determine the agent’s actions.
BELIEF-DESIRE-INTENTION - BDI
BELIEF-DESIRE-INTENTION - BDI

\[ \text{brf} : \wp(\text{Bel}) \times P \rightarrow \wp(\text{Bel}) \]

\[ \text{options} : \wp(\text{Bel}) \times \wp(\text{Int}) \rightarrow \wp(\text{Des}) \]

\[ \forall B \in \wp(\text{Bel}), \forall D \in \wp(\text{Des}), \forall I \in \wp(\text{Int}), \text{filter}(B, D, I) \subseteq I \cup D. \]

\[ \text{execute} : \wp(\text{Int}) \rightarrow A \]
THE END
INTELLIGENT AGENTS

References:

- Wooldridge Michael – “Intelligent Agents”