# CHAPTER 1: Introduction

## -

PART 2. Intelligent autonomous agents

# CHAPTER 2: Intelligent agents

Agents

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 delegated objectives.

The agent takes sensory input from the environment, and produces, as output, actions that affect it. The interaction is usually an ongoing, non-terminating one.

Agents are often being disassembled to 3 parts: perception (receiving and pre-processing the input), decision making and effectors (responsible for influencing the environment)

Adjustable autonomy – concept that lets user to influence multi-agent system. Since all agents are acting on their own benefits it is important to make some mechanism to make them eager to respond to user’s desires. Transfer of action control can be either hard-wired to some conditions or used by a particular agent when one thinks this decision might lead to the most beneficial result.

Examples: control systems, software demons

Intelligent agents

* **Reactivity** Intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.
* **Proactiveness** Intelligent agents are able to exhibit goal-directed behaviour by taking the initiative in order to satisfy their design objectives.
* **Social ability** Intelligent agents are capable of interacting with other agents (and possibly humans) in order to satisfy their design objectives.

Intelligent agents can be described by their:

* Goals. Some particular results, that agent is trying
* Beliefs. Used to describe memory of an agent.
* Intentions. Used to describe strategies, made by applying beliefs to intentions.
* Abilities. Used to describe possible actions that influence the environment.

Abstract architectures

Let’s imagine the environment can be described with finite number of discrete states 

Now let us assume there is an agent interacting with this environment that can execute one of possible actions 

A run, r, of an agent in an environment is thus a sequence of interleaved environment states and actions: 

R – set of possible runs.

– subset of R with sequences that end with an action.

– subset of R with sequences, that end with a state.

 - state transition function

 - generic agent model

> history-dependent

>might be non-deterministic

Subclasses:

1. Purely reactive - agents with no state (therefore history-independent)
2. History dependent

Basic working algorithm:

1. Receive (and preprocess) data
2. apply to state (possibly changing it) => generate action (based on state)
3. generate action (based on current state)

Utility functions

1. where |R means rational. Sometimes may be useless to evaluate long-term strategies
2. . Generic
3. Predicates. Useful to determine whether the task was accomplished

The agent is considered optimal in non-deterministic environment, when it has the best expected utility

In other words , 

Task predicate + environment = task environment

Tasks

1. Achievement tasks – change environment state to one of the needed states.
2. Maintenance tasks – also known as avoidance tasks

Agent Synthesis (automatized algorithm of creating agents)

* In task environment algorithm can be labelled as:

**sound** if, whenever it returns an agent, this agent succeeds in the task environment that is passed as input, and

**complete** if it is guaranteed to return an agent whenever there exists an agent that will succeed in the task environment given as input.

# CHAPTER 2: deductive agents

Symbolic AI concepts:

* Intelligent agent can be constructed from symbolic description of its preferred actions and environment

Symbolic AI problems

* The transduction problem The problem of translating the real world into an accurate, adequate symbolic description of the world, in time for that description to be useful.
* The representation/reasoning problem The problem of representing information symbolically, and getting agents to manipulate/reason with it, in time for the results to be useful.

…

## 3.2 Agent oriented programming

- introduced by Yoav Shoham.

- designed to connect BDI system to programming languages.

AGENT0 language

“ In this language, an agent is specified in terms of a set of capabilities (things the agent can do), a set of initial beliefs, a set of initial commitments, and a set of commitment rules. The key component, which determines how the agent acts, is the commitment rule set. Each commitment rule contains a message condition, a mental condition, and an action. In order to determine whether such a rule fires, the message condition is matched against the messages that the agent has received; the mental condition is matched against the beliefs of the agent. If the rule fires, then the agent becomes committed to the action. “

Actions can be internal, or “private”, as well as external, or “communicative”. Communicative actions include usage of messages (to other agents).

There are 3 types of messages : ‘request’, ‘unrequest’, and ‘inform’. First two types are used to change agent commitments.

MetateM : //irrelevant

# CHAPTER 3: practical reasoning agents

…

# Part III Communication and Cooperation (106)

They need to create