EFREI 2011/2012
L3

Aide à la Décision
Agents

Support de cours

Cours « Aide à la Décision »
EFREI – 2011/2012 – L3 / L3A

« Agent » : definition?

Born from Artificial Intelligence and Networks

Distributed systems
Cooperating systems

⇒ Intelligent Systems

built from multiple « agents »

 2

« Agent » : definition?

- No unique and standard definition
- Simplest definition of an agent :

  « a small software component specialized in one specific task »

  … « a component of a distributed application / system » !

 3

Goals

- Multidisciplinary problems
  - Domains, constraints, …

  Need to have « specialized » systems for problem résolution : specific technics (algorithms), specific knowledge (what, how)
- Taking advantages from problem specificity

  Want to use the « best » system
- Cooperation

  Need inter-operation mechanisms between heterogeneous systems
- Heterogeneous data

  Required for each domain, shared between cooperating parts

…

 4

Goals

…

- React to undefined situations

  Ability to adapt, to learn
- Use mobility to choose execution host
  - for example according to available ressources

  but how to maintain accessibility from others?

⇒ Intelligent agent = software & hardware

 5

Quelques définitions

- « Un agent est une entité autonome, réelle ou abstraite,
  -qui est capable d'agir sur elle-même et sur son environnement,
  -qui, dans un univers multi-agent, peut communiquer avec d'autres agents,
  -et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents »

  Ferber

What are agents ?

- Autonomous problem-solving entities
  - complex, dynamic environments (physical or software)
  - no permanent guidance from the user
- Intelligent Agents
  - Perceive and interpret 'sensor'-data
  - Reflect events in their environment
  - Take actions to achieve given goals

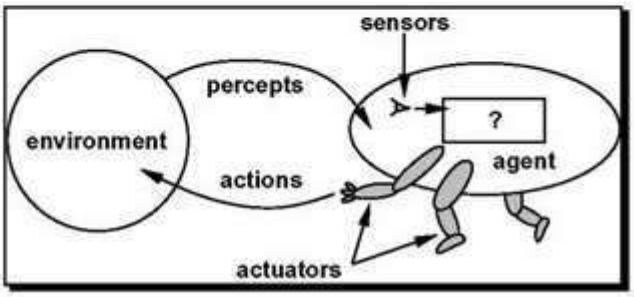
Source : FIPA (Foundation of Intelligent Physical Agents)

Agent = « Rational » entity

- Overall goal : **maximize its performance**
  - « Performance » is :
    - External vision
    - Established as part of the objectives of the agent
    - Dependant of its specific tasks
- « Rational » means :
  - Able to explore different situations and alternatives
  - Able to learn
  - Be autonomous
- « Rational » does not mean :
  - All-knowing (omniscient)
  - Perceptive (clairvoyant)

Interactions

- Interactions with the environment
  - Perception
    - Captors / Sensors
  - Action
    - Effectors



Interaction means (examples)

	Perception	*Action*
Human	Eye - Skin - Ear – Nose	Hand - Leg – Mouth
Robot	Video camera – IrdA – Control board (human)	Wheels – Arm – Pliers
Software	Data – character stream	Data – Character stream

Types of interaction

- Agent to Software
  - Software interaction
    - Collaborative aspect
    - Concurrent aspect
- Agent to Humans
  - Humans interaction
    - Authentication
    - Profiling
    - Type of interaction / ease of use
    - ...
- Agent to agent
  - Agent to agent interaction
    - Collaboration
    - Delegation
    - Negociation
    - Concurrent access
    - Information transfert
    - ...

And ...

- Decomposition of complex situations / problems
- Collaborative behavior between agents with different roles & skills
- Distribution, modularity, flexibility

Architecture of an agent

- React to the environment :

  series of percepts lead to actions

  Percept 1, Percept 2, ... ⇒ Action

  $f : P^* \rightarrow A$

« P* » is the chronology of what has been observed

Some agents may take decision bases on the latest event, without memory of the previous ones.

« A » is the action that the agent performs

 12

Architecture of an agent

- Example :

```
function thermostat
              ( temp )
if temp < desired -
              epsilon
  action= open tap

if temp > desired +
              epsilon
  action= close tap

else
  action = nothing
```

- General structure :

```
function agent
              ( percept )
local state is L
L= update_local_state
        ( L , percept )
action= choose_best
                    ( L )

L= update_local_state
          ( L , action )
```

 13

Types of agents

- **Table-driven**

  Has a table of possible actions

  Sequences of percepts X Actions

- **Reflex**

  Has a set of rules

  Condition X Action

  No memory of what happened or has been decided

- **Reflex with states**

  Knowledge of the past states

 14

Types of agents

- **Based on objectives / goals**
  - Knows what is to be reached
    - The « ideal » situation
  - Can envision future actions / events and states
- **With usefullness function**
  - Has the knowledge of how usefull would be such action / such state

 15

Table-driven agent

```
P* is a sequence of percepts,
   initially empty
T is a table indexed by percept sequences,
  initially fully specified

append percept to the end of P*
action = lookup ( P* , T )
```

- Problems ?
  - Size of the table
    - Chess game = 35^{100} entries
  - How / how long to build the table
  - Autonomy : none if the table can not change

 16

Reflex agent

```
R is a set of condition-action rules,
  initially fully specified

state = interpret_input ( percept )
rule = rule_match ( state , R )
action = rule_action ( rule )
```

- No memory : only the latest « percept »
- Human examples :
  - Good : green light = go ; red light = stop
  - Bad : car drivers can cross a road after having a look on BOTH sides (...memory needed, even for a short time)

 17

Reflex agent with state

```
state is the description of the
                current world state
R is a set of condition-action rules
```

```
state= update_state ( state, percept )
rule= rule_match ( state, R )
action= rule_action ( rule )
state= update_state ( state, action )
```

- « Update-state » : how to implement ?
- How many states ?

Agent based on its objectives

- Need to envision possibles actions and their results
  - Evolution of the world : current state + action = new state
- Be able to determine which new state is the « closest » to the objective
  - Need of domain expertise
  - Need of heuristics
- A* ?...

Agent based on usefullness

- « Goal » ?
  - Binary..... « ok » or « nok »
- « Usefullness » ?
  - A scale from less usefull to most usefull

    u : state --> value
- « usefullness » instead of « goal » :
  - Conflict : speed / security
  - Several goals

Learning agents : principles

- Why « learning » ?
  Improve for decision making
  - Increase accuracy of the « percept / state X action » decisions
- How to decide what is « better » ?
  - Need to have feedback
  - Feedback needs « performance standard »

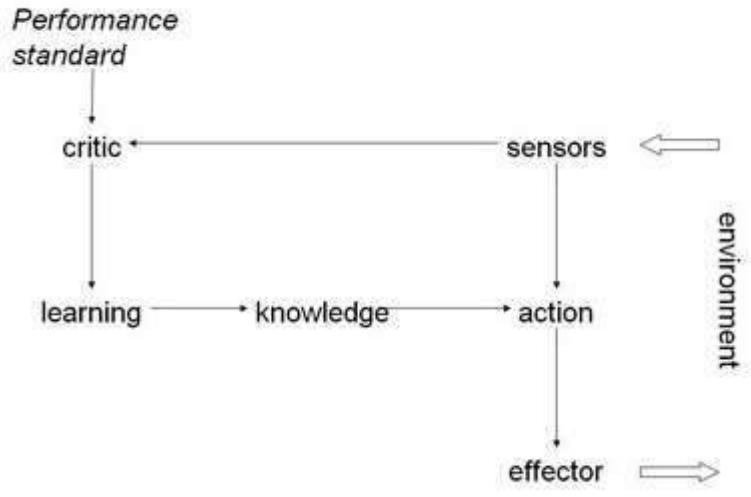
 21

Learning - General architecture

- From « normal » agent :
  - Sensors – decision / action element – effectors
  - Based on knowledge
- Critic
  - Evaluation of performances
  - Based on external measurement standards
    - Determines what is « good », « bad » or « somewhere in between »
- Learning element
  - Implements the learning mechanism
  - Improves the efficiency of the decision / action element through the knowledge base

 22

General architecture



 23

Performance

"For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has"

Russell, Norvig (1995), page 33.

 24

Performance

- How an agent should act?
  - Considérer toutes les options et faire le meilleur choix pour maximiser les chances de succès
  - Mesure de performance :
    - Réussir la tâche ?
    - Quantité de ressources consommées ?
    - Temps mis pour réaliser l'action ?
  - Exemples :
    - Véhicule automatique : destination atteinte, conduite sûre, parcours le plus court, …
    - Aspirateur robotisé : quantité de poussière récoltée, quantité d'électricité consommée, bruit généré, propreté du sol, …

 25

Performance

- Mesure de la performance d'un agent
  - Comment ?
    - Par mesures externes ou par auto-évaluation .
  - Quand ?
    - De manière continue, périodique, ou une seule fois ?

 26

Performance

- Mesures de performance extérieures
  - Les critères de performance sont analysés
  - L'environnement fournit un feedback à l'agent

$$V : S^* \rightarrow \mathcal{R}$$

  - Feedback
    après chaque action ou
    périodiquement ou
    en fin de traitement

 27

Performance

- Exemple du thermostat

- la température ambiante est échantillonnée à intervalles réguliers.

$V(s_t)$ = 100 si $température(s_t) \approx$ DÉSIRÉE
= -100 sinon

- objectif: maximiser $V(S_0) + \gamma V(S_1) + \ldots + \gamma^n V(S_n) + \ldots$
  avec le facteur $0 < \gamma \leq 1$

 28