LES AGENTS

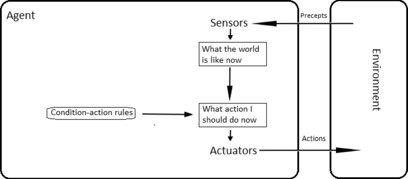
# Intelligent agent

According to Nikola Kasabov IA systems agents should exhibit the following characteristics:

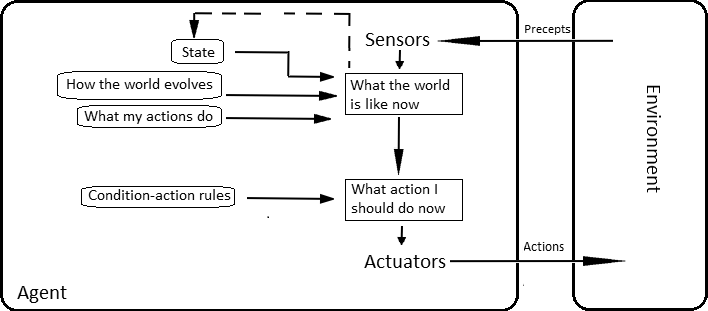
* accommodate new [problem solving](http://en.wikipedia.org/wiki/Problem_solving) rules incrementally
* adapt online and in [real time](http://en.wikipedia.org/wiki/Real-time_computing)
* be able to analyze [itself](http://en.wikipedia.org/wiki/Self-concept) in terms of behavior, error and success.
* learn and improve through interaction with the environment ([embodiment](http://en.wikipedia.org/wiki/Embodied_agent))
* learn quickly from large amounts of [data](http://en.wikipedia.org/wiki/Data_%28computing%29)
* have memory-based exemplar [storage](http://en.wikipedia.org/wiki/Computer_storage) and retrieval capacities
* Have [parameters](http://en.wikipedia.org/wiki/Parameter_%28computer_science%29) to represent short and long term memory, age, forgetting, etc.

[**Russell & Norvig** (2003](http://en.wikipedia.org/wiki/Intelligent_agent#CITEREFRussellNorvig2003)) group agents into five classes based on their degree of perceived intelligence and capability:

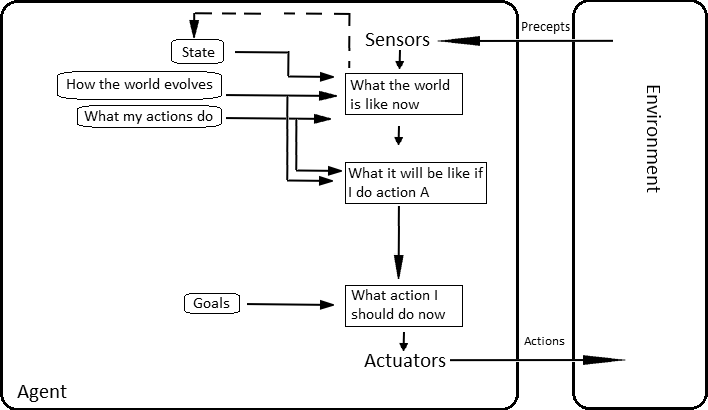
1. simple reflex agents : réagissent simplement à un stimulus environnemental



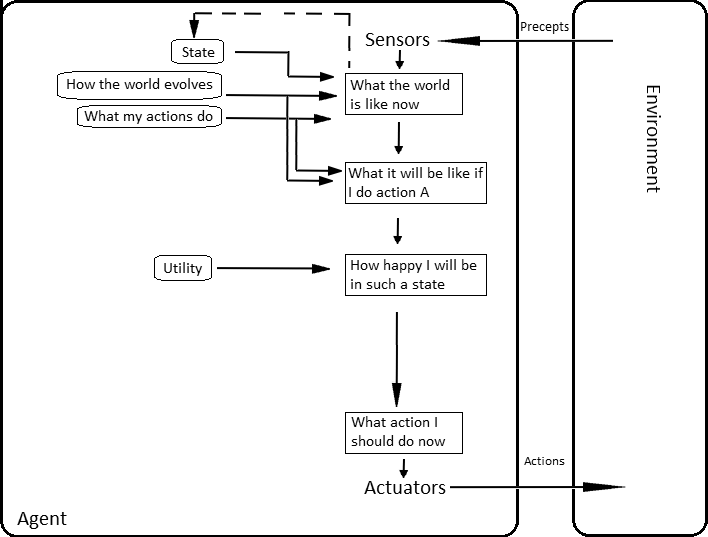
1. model-based reflex agents : comme au-dessus mais en prenant compte de son état personnel les impacts de ces actions sur le l’env et l’EV du monde



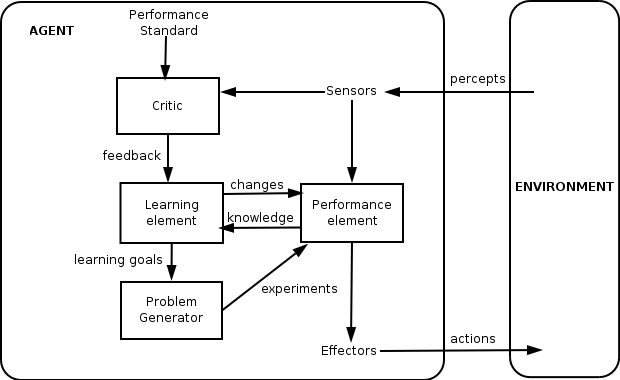
1. goal-based agents similaire a 2 mais en prenant en compte un état optimal final a obtenir (peu ne pas agir pour ne pas retarder cet état final)



1. utility-based agents : qd différentes possibilités a une action laquelle est la meilleure ? sois + ou – mais si plus subtil nécessaire apparition d’une échelle de satisfaction est un entier réel, qui indique la probabilité de réussite ou d’utilité de l’action. Permet de balancer entre plusieurs choix finaux positif, ou d’évaluer un gain par rapport a la probabilité de réussite si elle n’est pas acquise



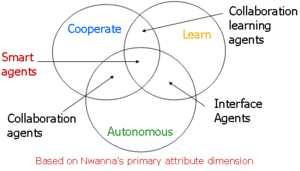
1. Learning agents :a chaque nouvelle actions analyse son impact pour opti sur prochaine appel. Est il égal ce qui était prévu + ? - ? opti ? sub opti ?etc… va changer son behavior en fct des retours précédents. Tout en conservant une perf standart minimum a atteindre. De plus utilise un générateur de problème qui a pour but de proposer de nouvelles expériences pouvant fournir un nouveau résultat (TGN ??)



# Software agents

The basic attributes of a software agent are that agents

* are not strictly invoked for a task, but activate themselves,
* may reside in wait status on a host, perceiving context,
* may get to run status on a host upon starting conditions,
* do not require interaction of user,
* may invoke other tasks including communication.



Other authors have proposed different definitions of agents; these commonly include concepts such as

* *Persistence* (code is not executed on demand but runs continuously and decides for itself when it should perform some activity)
* *Autonomy* (agents have capabilities of task selection, prioritization, goal-directed behavior, decision-making without human intervention)
* *Social ability* (agents are able to engage other components through some sort of communication and coordination, they may collaborate on a task)
* *Reactivity* (agents perceive the context in which they operate and react to it appropriately).

# Definition ACSEL

Un agent est un logiciel qui est Persistant, qui perçoit a son environnement, autonome.

**Autonomie** : service indépendant et pouvant opérer sans support constant dois réagir en fct de son env (et de son expérience pour les agents d’apprentissage)

Ces actions sont fruit de comportement prédéfini lors de son design.

**Perception** : il peu lire des sockets des fichiers des perfs des intreactions utilisateurs provenant de son opérateur ou de périphérique

Toute information perçue a un moment t est sa perception.

Toute info perçue entre t et t+x est sa séquence de perception.

**Autonomie/action :** il peut agir par lui-même en écrivant sur un socket, lancer un software, interagir avec une gui ou du hardware etc…

# Utilisation principale

Monitoring, systéme adaptatif, assistant personnels etc…

Divisé en 5 catégorie cf definition de Russell et Norvig

# Les P.E.A.S

Performance, environment, actuators sensors

(mon but, mon monde, mes outils pour y agir, ce que je vois)

Perf : quel est le resultat minimum que je dois atteindre comme faire plus vite mieux ????????

Env : connu/inconnu, observable intégralement ou non, deterministe/stochastique (== aleatoire) mono ou polyagent statique ou dynamique etc…

Actuators : mes outils : output vers user, hardware de sortie, activation d’autre software etc…

Sensors comment je recois le monde input de l’user, monitoring de mon hardware ou de mes soft tournant etc…