



Escola Politécnica da USP - Depto. de Enga. Mecatrônica

PMR-3510 Inteligência Artificial

Aula 2 - Automação e Agentes Inteligentes

Prof. José Reinaldo Silva

reinaldo@usp.br





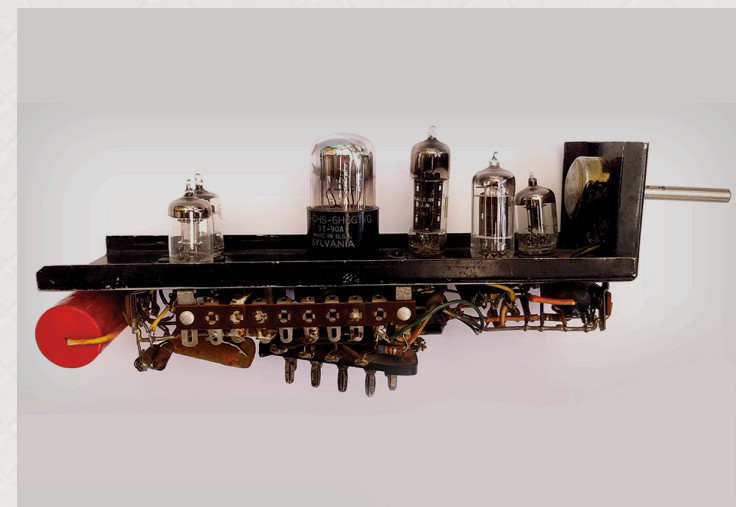
Na aula passada fizemos uma resumida digressão histórica que nos levou a concluir que a moderna inteligência artificial nasceu baseada em dois pilares:

- ✦ a reprodução de processos racionais, como a capacidade de jogar xadrez, fazer deduções, planejar, aprender e "tomar decisões";
- ✦ o uso de autômatos, ou máquinas, capazes de abrigar os processos acima, física ou virtualmente.



Sobre as teorias de aprendizado, Warren McCulloch e Walter Pitts publicaram um trabalho em 1943 propondo um modelo funcional inspirado nos neurônios humanos e na Lógica Proposicional de Bertrand Russell e Alfred Whitehead.

- ✦ em 1950, dois alunos de graduação da Universidade de Harvard, Marvin Minsky e Dean Edmonds, apresentaram um engenho chamado SNARC que emulava uma rede de neurônios seguindo este modelo;
- ✦ o mesmo Minsky apresentou sua tese de doutorado quatro anos mais tarde com um modelo computacional (forma) para a inferência e aprendizado por uma rede de neurônios.



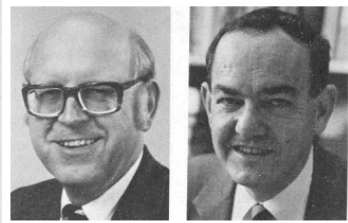


Arthur Samuel, IBM, e o seu Checkers Player, lançado em 1956. O primeiro programa que “aprendia” as melhores jogadas.



Após a Conferência do Dartmouth College, a orientação era direcionar a pesquisa na área de IA para processos formais para transcender o processo direto de programação procedural.

Allen Newell and Herbert A. Simon



Logic Theorist

General Problem Solving (GPS)

Knowledge-based Systems (KBS)

Learning

Machine Learning



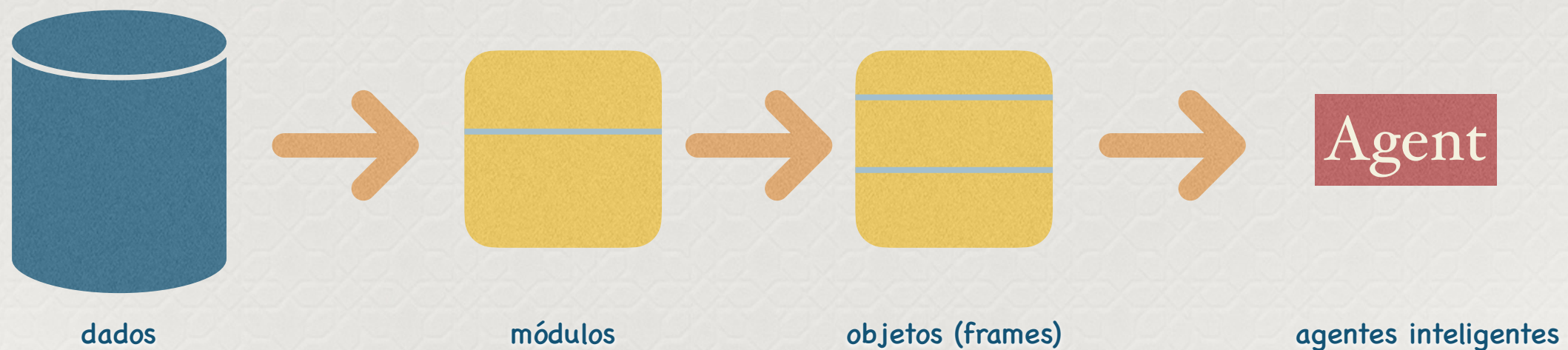
Na aula passada discutimos a tendência inicial pela “reprodução da inteligência” por mecanismos, ou, genericamente, por “máquinas” desde, no mínimo, o século XIII. Discutimos brevemente a relação entre conhecimento e “dados”, e a “individualização” dos elementos artificiais de manipulação: os “agentes”.

- ✦ Seguindo o modelo “simbolista” de Newell e Simon, o conhecimento deve estar na “máquina” (em alguma forma de representação), sempre supondo que este é todo o conhecimento necessário para “resolver os problemas”;
- ✦ A “máquina” deve ter um mecanismo de inferência para, usando o a base de conhecimento, gerar as devidas respostas às perguntas (queries).



A evolução desde dados estáticos até os agentes inteligentes

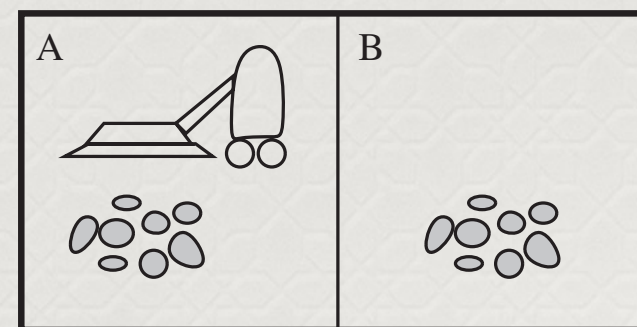
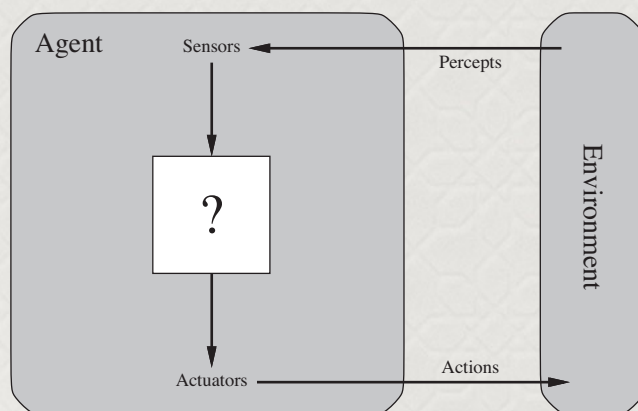
Portanto o objetivo é representar conhecimento e processos racionais de forma computável. O ponto de partida será a forma mais simples de conhecimento: dados – especialmente aqueles considerados confiáveis e verdadeiros. Mesmo a programação procedural tem dedicado muita atenção à representação de dados, que podem ser usados de forma estática ou interagindo com processos dinâmicos. A linha de evolução nesta área pode ser sintetizada como:





Evoluindo dos dados até os agentes inteligentes

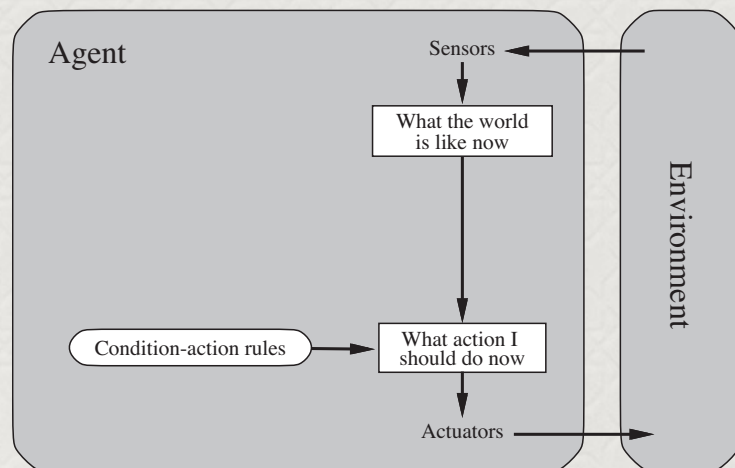
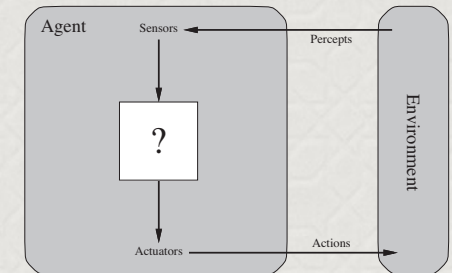
Um agente inteligente é capaz de perceber o que acontece no seu domínio de trabalho, ou environment, e definir que ação deve tomar, atuando sobre este mesmo environment. A ação deve ser "inteligente". Um exemplo simples seria um aspirador (inteligente) que detecta e localiza o pó a ser removido, traça um caminho (ótimo) até ele, evitando obstáculos, e aciona o aspirador.



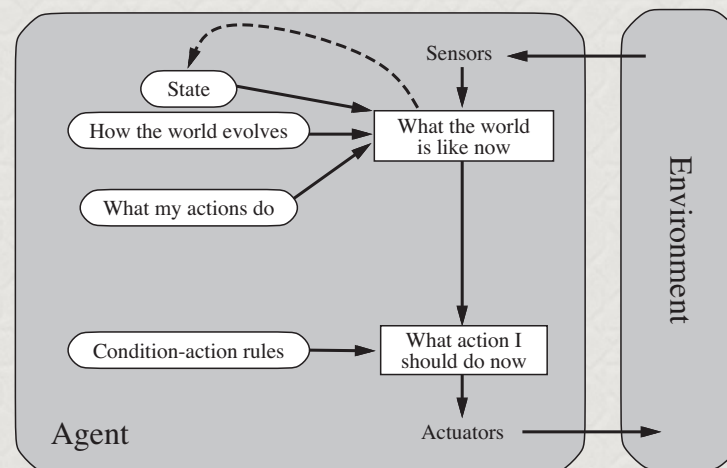


Introdução

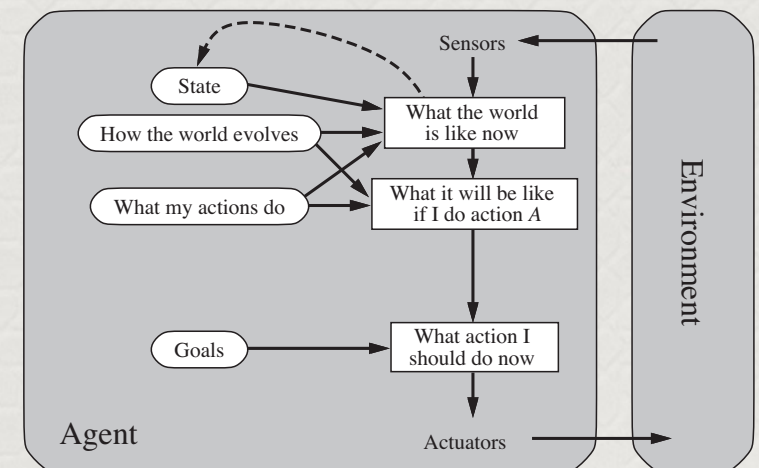
From Data to Intelligent Agents Agentes inteligentes autônomos



Agente reflexo



Agente baseado em modelo



Agente baseado em objetivos



Sobre os dados e como seriam inseridos em sistemas inteligentes

A forma canônica de representação de dados é atribuindo tipos e relações.



Vamos tomar como base a programação lógica e a linguagem Prolog para os nossos exemplos. No final do curso discutiremos como fazer o mesmo usando outras linguagens, tanto declarativas, como o LISP quanto semi-procedurais, como Python, ou completamente procederias, como o C. Discutiremos também como fazer o link com sistemas de dados de maior porte, como bancos de dados relacionais e não-relacionais.



A linguagem Prolog (de Programação Lógica) foi criada para expressar de forma declarativa (não procedural) ações cognitivas, satisfazendo a um sub-conjunto da lógica de primeira ordem. Foi criada por Alain Colmerauer e Robert Kowalski e lançada em 1972.



Alain Colmerauer (1941-2017)



Robert Kowalski (1945-2017)



```
male(mike).  
male(tom).  
female(trade).
```

```
mother_child(trude, sally).
```

```
father_child(tom, sally).  
father_child(tom, erica).  
father_child(mike, tom).
```

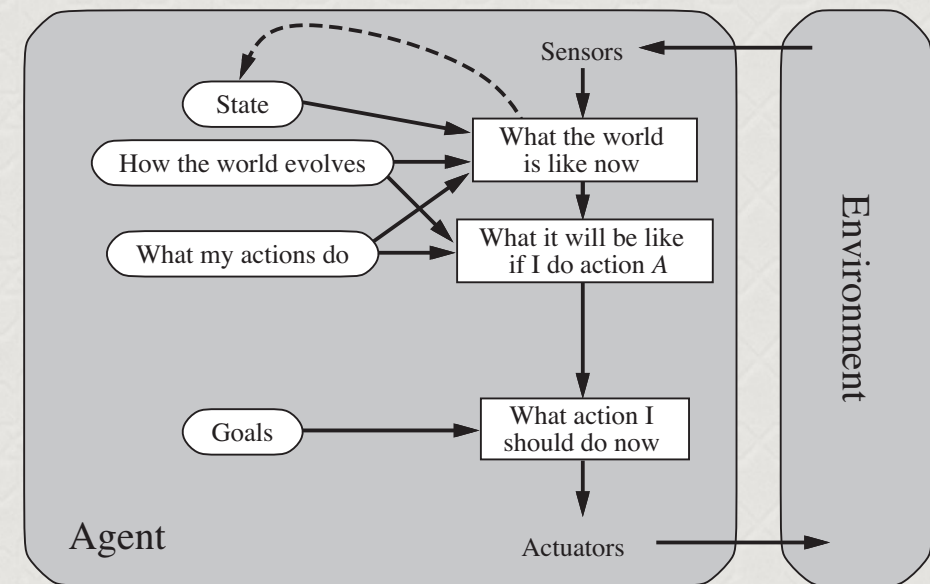
```
sibling(X, Y):- parent_child(Z, X),  
parent_child(Z, Y).
```

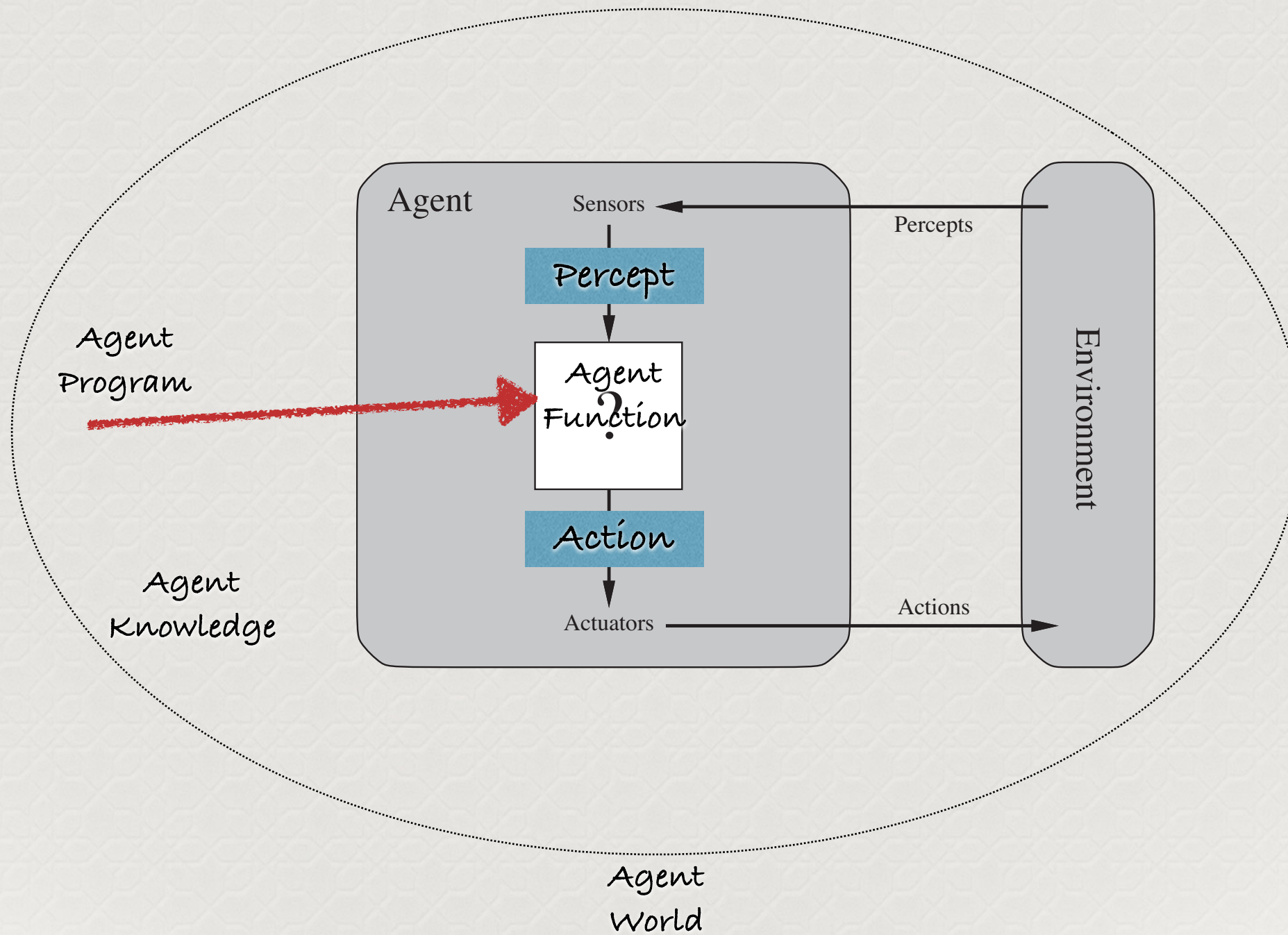
```
parent_child(X, Y) :- father_child(X, Y).  
parent_child(X, Y) :- mother_child(X, Y).
```

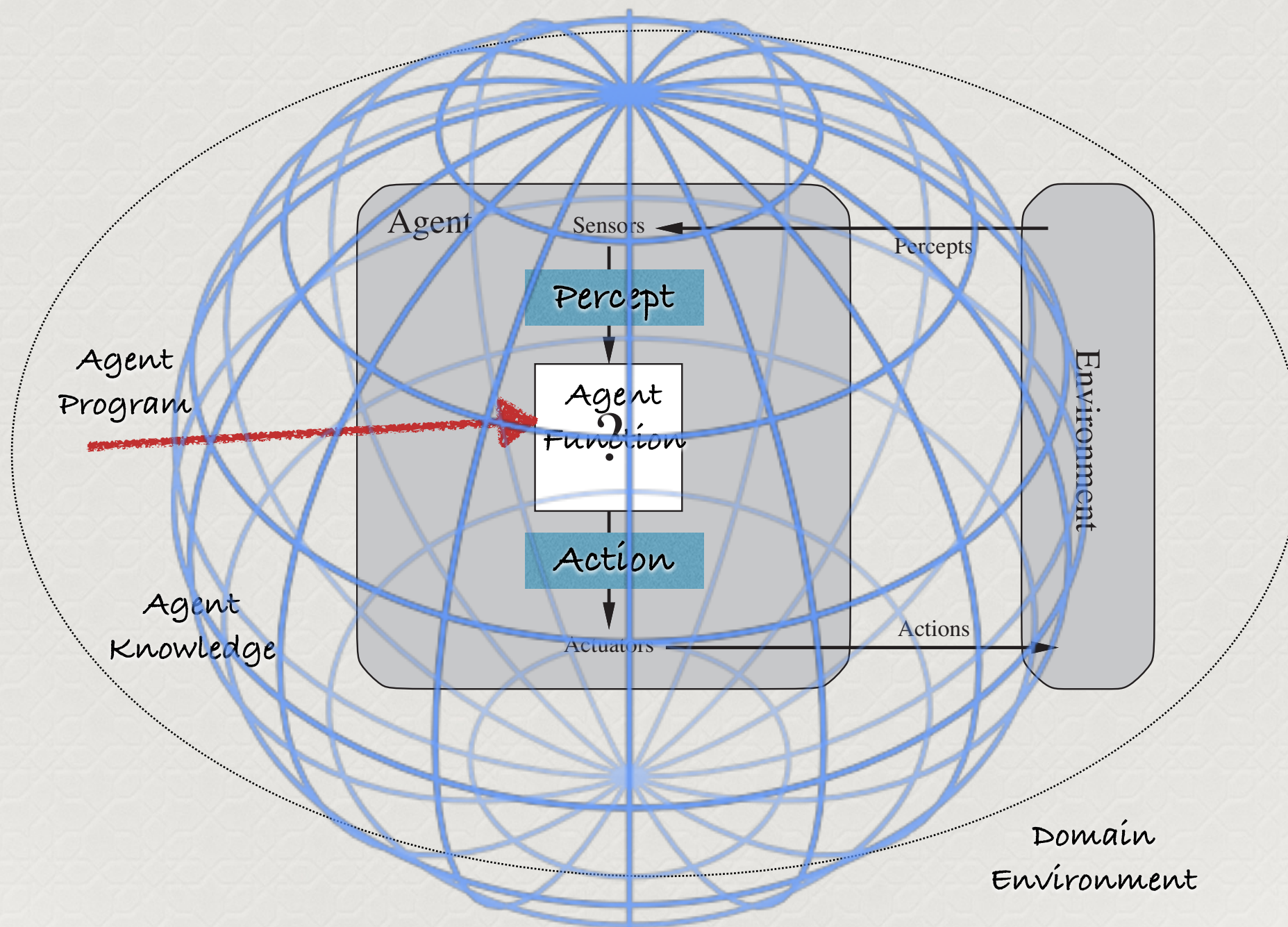



Sobre os agentes

Agentes podem fazer uma associação direta entre a percepção do ambiente (environment), o domínio de aplicação (o ambiente influenciável pelo agente) e a “resposta”, ou seja, o acionamento de algum dispositivo ou mensagem. Esta relação é feita de forma “racional”, ou simbólica.





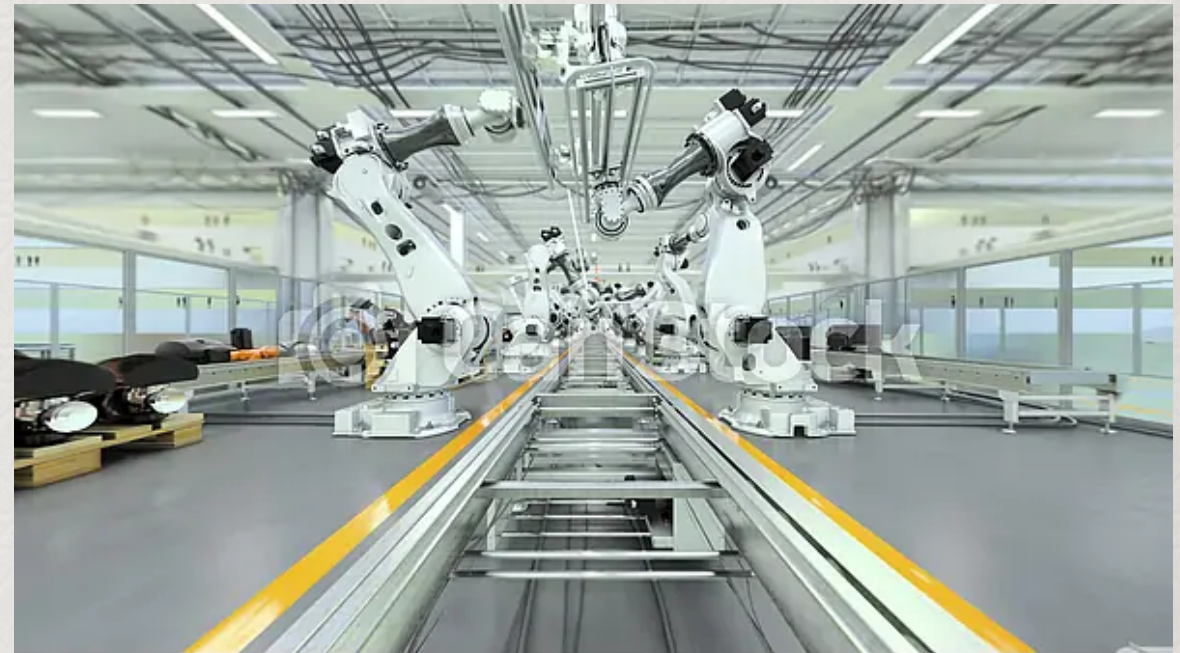




Concretamente o domínio de interação do(s) agente(s), é denominado “domain environment”. Em geral este domínio é finito, fechado, e faz parte do processo de design de um sistema automatizado especificá-lo.

Exemplo: Podemos ter como alvo um sistema inteligente para a segurança do prédio da Enga. Mecânica. Este seria o nosso "environment". O nosso domínio de aplicação seria o sistema de segurança da Poli, do qual o nosso sistema é parte.





© CanStockPhoto.com - csp71279368



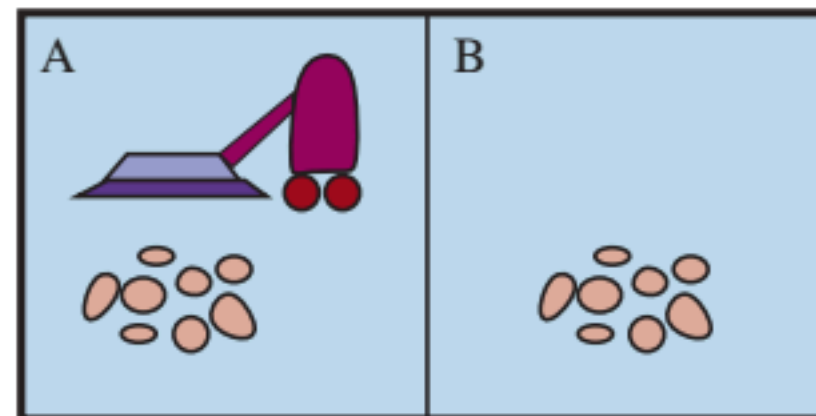


Figure 2.2 A vacuum-cleaner world with just two locations. Each location can be clean or dirty, and the agent can move left or right and can clean the square that it occupies. Different versions of the vacuum world allow for different rules about what the agent can perceive, whether its actions always succeed, and so on.

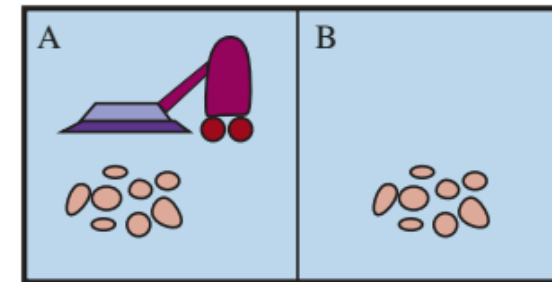


Figure 2.2 A vacuum-cleaner world with just two locations. Each location can be clean or dirty, and the agent can move left or right and can clean the square that it occupies. Different versions of the vacuum world allow for different rules about what the agent can perceive, whether its actions always succeed, and so on.

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
⋮	⋮
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck
⋮	⋮



Marvín Mínsky, introduziu o conceito de "frames" como um ente capaz de processar "percept sequences", e estes se tornaram mais tarde uma referência para a Inteligência Artificial Distribuída (ou sistemas multi-agentes inteligentes).



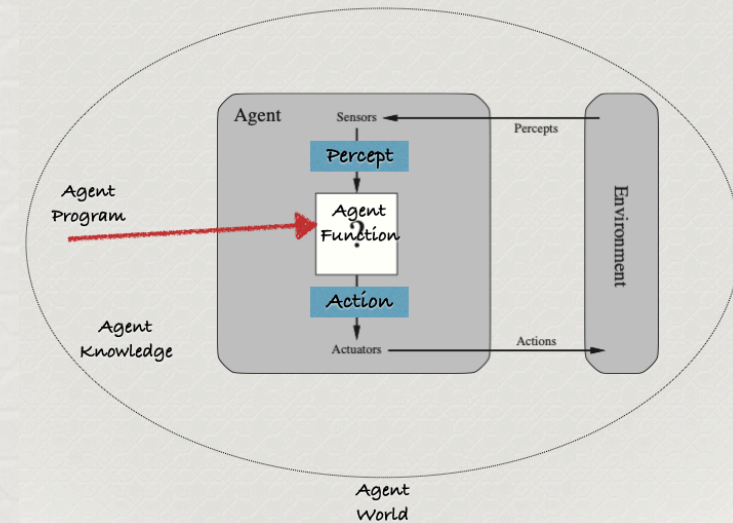
Marvin Minsky (1927-2016)

Grandes obras:

Perceptrons, Society of Mind, [The Emotion Machine](#),
Computation: Finite and Infinite Machines



Um agente é dito “racional” se,



For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Russell, Stuart; Norvig, Peter. Artificial Intelligence: A Modern Approach, eBook, Global Edition (p. 58). Pearson Education. Kindle Edition.



A percepção do domínio que envolve o agente ou "percept sequence" indica os "problemas" (tasks) que o agente deve resolver. Supõe-se portanto que o agente (inteligente) deve selecionar de forma "racional" a(s) ação(ões) que deve usar para dar uma resposta a este mesmo ambiente.

O conjunto de "problemas" (tasks) que um agente deve resolver constitui o seu "task environment".

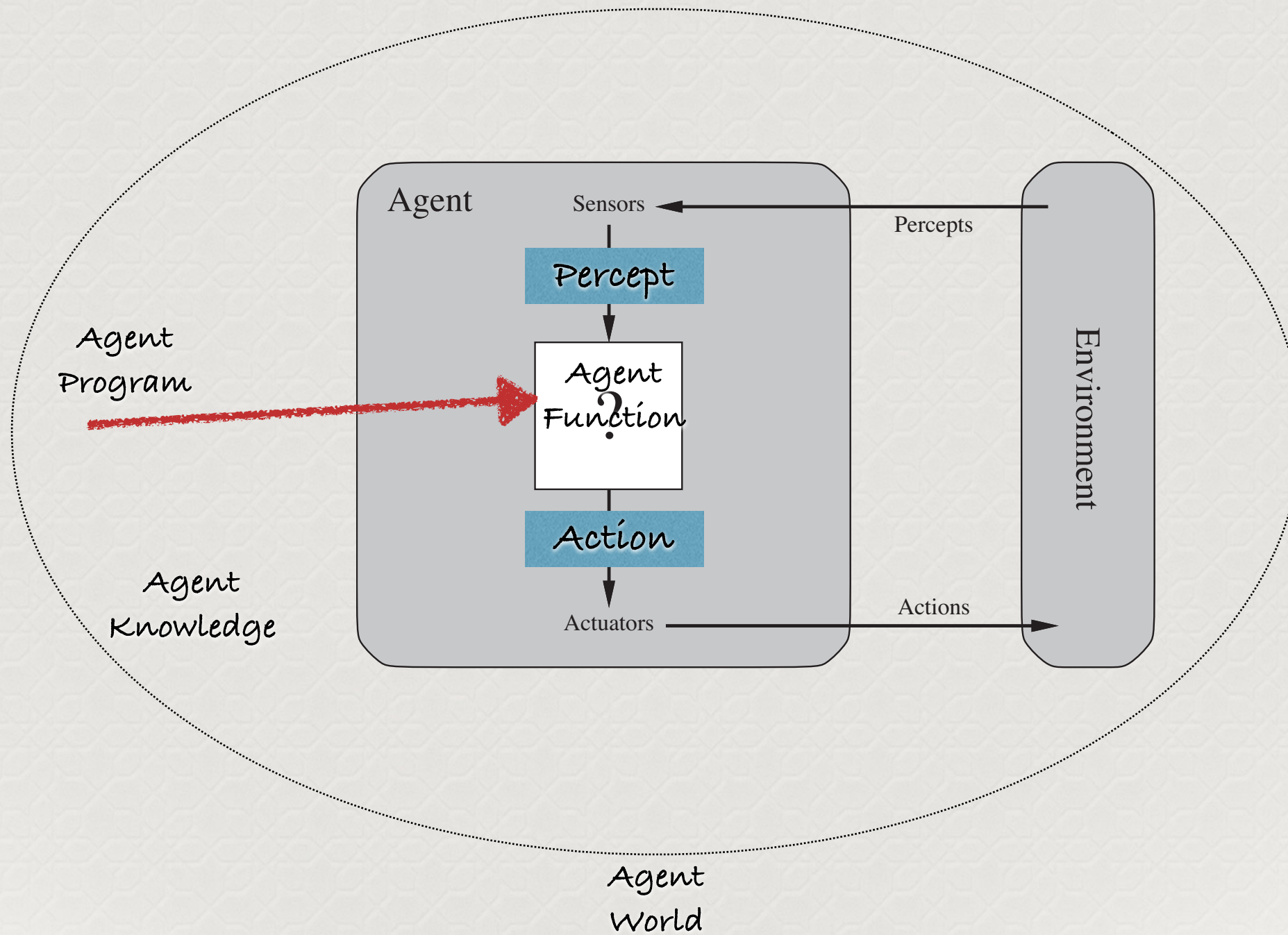


Portanto, o design (conceitual) de cada agente consiste na definição da sua performance, seu (task) environment, as ações que pode executar e os sensores que usa para detectar as "percept sequences". Em inglês isso consiste dos parâmetros: performance, environment, actions and sensors, e resulta na sigla PEAS.





Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice





Diferentes tipos de agentes:

Reflexivos (simple reflex agents)

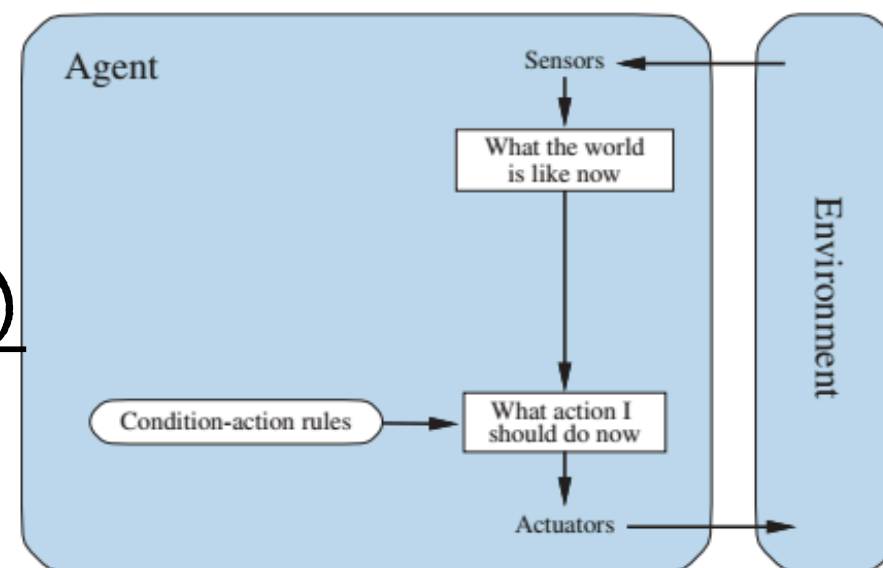
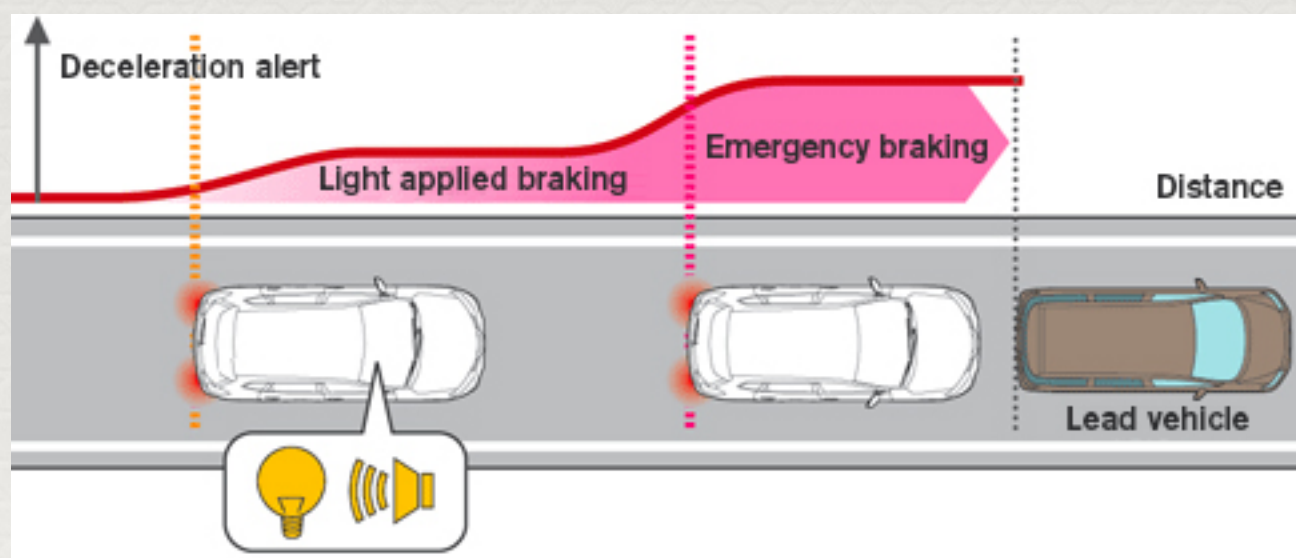


Figure 2.9 Schematic diagram of a simple reflex agent. We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

if car-in-front-is-braking then initiate-braking





Diferentes tipos de agentes:

Agentes baseados em modelos
(Model-based agents)

Nesse caso o agente guarda estados relacionados com sequências perceptivas, e, mesmo que não possa "sensoriar" todo o domínio pode ainda decidir o que fazer.

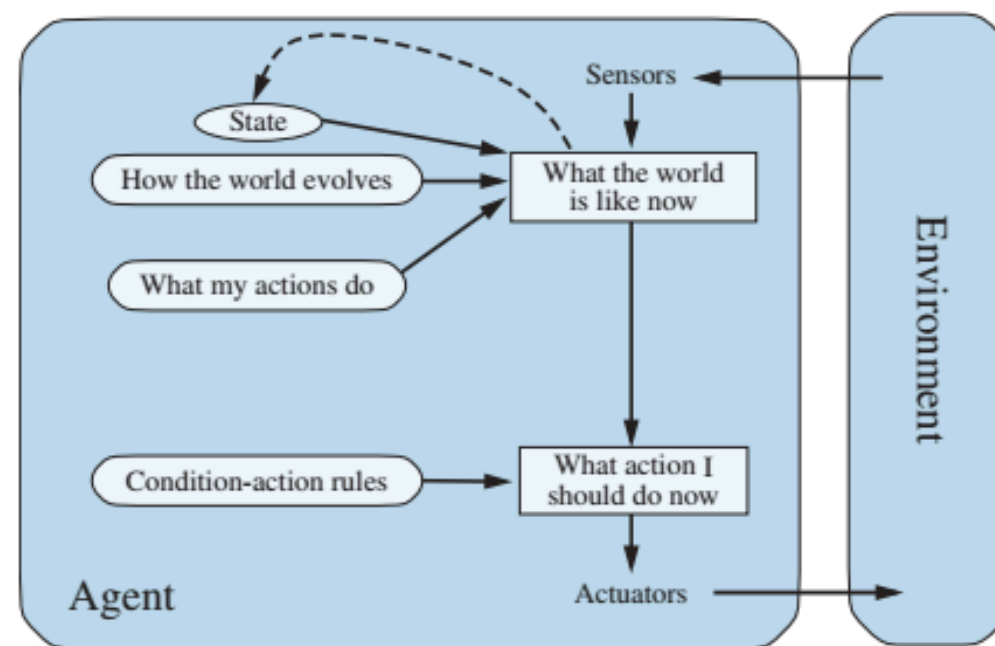


Figure 2.11 A model-based reflex agent.



Internet Superhighway





Diferentes tipos de agentes:

Agentes baseados em objetivos (Goal-based agents)

A prioridade do "agent program" pode estar ligada a atingir um determinado objetivo. Portanto a escolha do que fazer é direcionada pela aproximação a este objetivo.

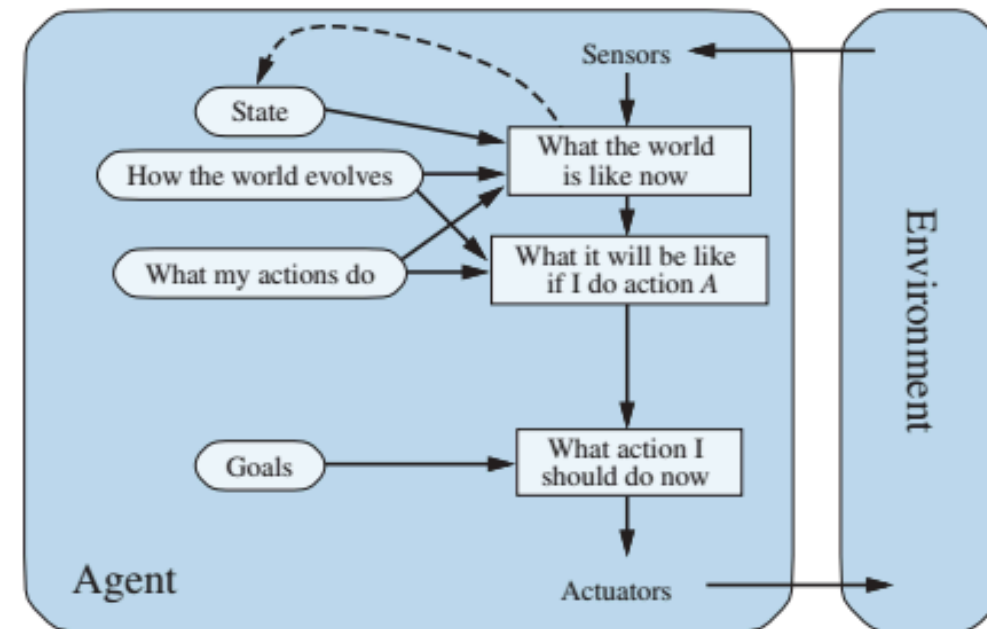


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.



Softwares e sistemas de roteamento inteligente.



Diferentes tipos de agentes:

Agentes baseados em serviço
(Utility-based agents)

Nesse caso os agentes têm restrições sobre as ações que executam, relacionadas com a prestação de serviços.

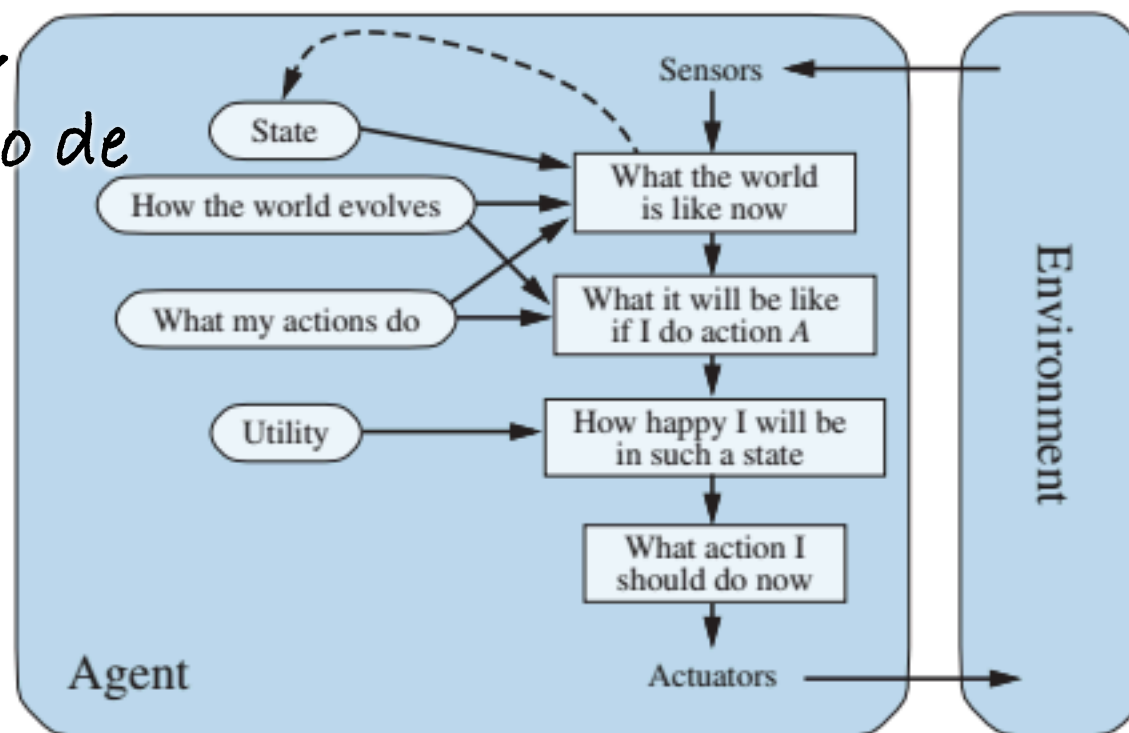


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.



Telemedicina



Diferentes tipos de agentes:

Agentes que aprendem
(Learning agents)

Nesse caso os agentes devem melhorar a sua performance e eventualmente o próprio "agent program", tomando como base os tasks já resolvidos com sucesso.

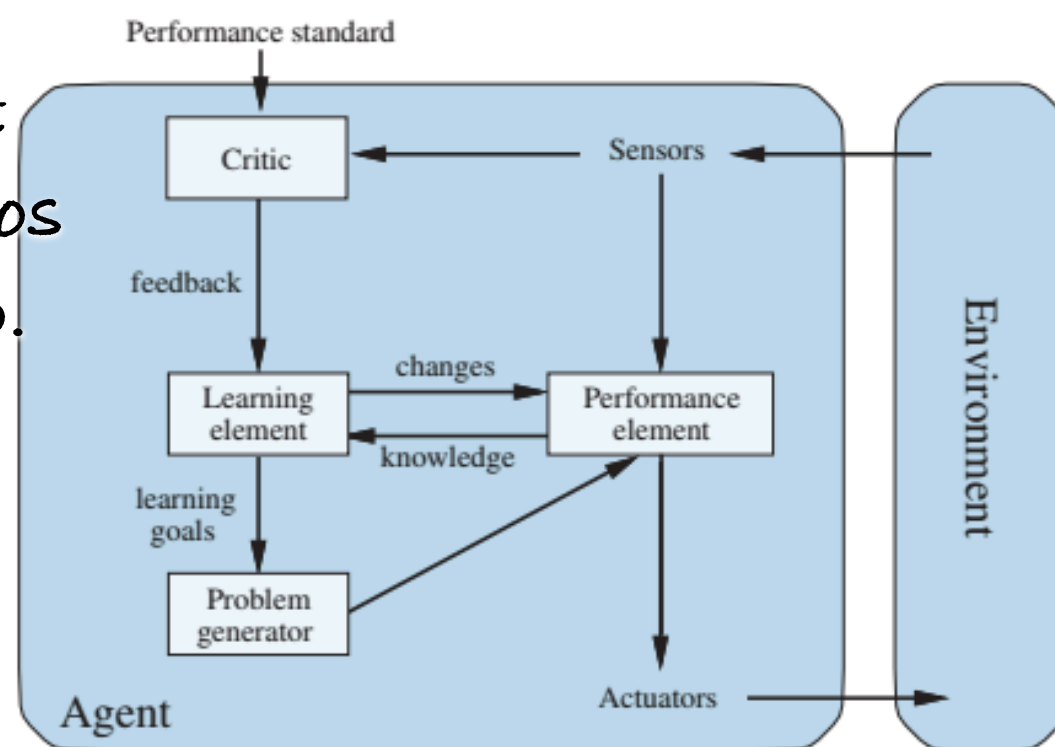


Figure 2.15 A general learning agent. The "performance element" box represents what we have previously considered to be the whole agent program. Now, the "learning element" box gets to modify that program to improve its performance.



Acessem o site da disciplina e vejam o exercício de programação: i) vá ao site mostrado abaixo (link na página da disciplina no edisciplinas) Vamos exercitar os fatos e regras, a versão mais simples da representação de conhecimento.

d-l@b Virtual Lab Página inicial Programação Newsletter

PMR 3510 - INTELIGÊNCIA ARTIFICIAL

Virtual Lab
Neste laboratório virtual vamos exercitar os conceitos básicos da programação lógica em PROLOG. Esta é uma, ou talvez a maneira mais direta de prototipar programas lógicos, que, para se tornarem aplicativos de software precisariam ou serem inseridos em uma outra linguagem, como o Python, ou C, ou serem integralmente migrados para uma linguagem híbrida como o PyLog. Siga a ordem dos módulos e faça os exercícios nos notebooks respectivos.

- [Módulo 1: Fatos e regras](#)
- [Módulo 2: Procedimentos](#)
- [Módulo 3: Estruturas \(listas e grafos\)](#)
- [Módulo 4: Busca \(A, A*\)](#)
- [Módulo 4: Planning](#)

Swish Prolog



ii) também no edisciplinas tem um link para o tutorial Prolog como mostrado abaixo; na versão online free do tutorial e no capítulo 1 escolha o knowledge base 1.

www.learnprolognow.org/lpnpage.php

Learn Prolog Now!

by Patrick Blackburn, Johan Bos, and Kristina Striegnitz

- LPNI Home
- Free Online Version
- Paperback English
- Paperback Français
- Teaching Prolog
- Prolog Implementations
- Prolog Manuals
- Prolog Links
- Thanks!
- Contact us

[next] [prev] [prev-tail] [tail] [up]

Chapter 1 Facts, Rules, and Queries

This chapter has two main goals:

1. To give some simple examples of Prolog programs. This will introduce us to the three basic constructs in Prolog: facts, rules, and queries. It will also introduce us to a number of other themes, like the role of logic in Prolog, and the idea of performing unification with the aid of variables.
2. To begin the systematic study of Prolog by defining terms, atoms, variables and other syntactic concepts.

1.1 [Some Simple Examples](#)
[Knowledge Base 1](#)
[Knowledge Base 2](#)
[Knowledge Base 3](#)
[Knowledge Base 4](#)
[Knowledge Base 5](#)

1.2 [Prolog Syntax](#)
[Atoms](#)
[Numbers](#)
[Variables](#)
[Complex terms](#)

1.3 [Exercises](#)
1.4 [Practical Session](#)

[next] [prev] [prev-tail] [front] [up]



Artificial Intelligence

A Modern Approach

Stuart J. Russell and Peter Norvig

Contributing writers:
John F. Canny, Jitendra M. Malik, Douglas



Prentice Hall, Englewood Cliffs, New Jersey

- △ US Edition
- △ Global Edition
- Acknowledgements
- Code
- Courses
- Editions
- Errata
- Exercises
- Figures
- Instructors Page
- Pseudocode
- Reviews



<http://aima.cs.berkeley.edu>

Artificial Intelligence: A Modern Approach, 4th US ed.

by [Stuart Russell](#) and [Peter Norvig](#)

The [authoritative](#), [most-used](#) AI textbook, adopted by over [1500](#) schools.

[Table of Contents](#) for the US Edition (or see the [Global Edition](#))

[Preface \(pdf\)](#); [Contents with subsections](#)

I Artificial Intelligence

- 1 Introduction ... 1
- 2 Intelligent Agents ... 36

II Problem-solving

- 3 Solving Problems by Searching ... 63
- 4 Search in Complex Environments ... 110
- 5 Adversarial Search and Games ... 146
- 6 Constraint Satisfaction Problems ... 180

III Knowledge, reasoning, and planning

- 7 Logical Agents ... 208
- 8 First-Order Logic ... 251
- 9 Inference in First-Order Logic ... 280
- 10 Knowledge Representation ... 314
- 11 Automated Planning ... 344

IV Uncertain knowledge and reasoning

- 12 Quantifying Uncertainty ... 385
- 13 Probabilistic Reasoning ... 412
- 14 Probabilistic Reasoning over Time ... 461
- 15 Probabilistic Programming ... 500
- 16 Making Simple Decisions ... 528
- 17 Making Complex Decisions ... 562
- 18 Multiagent Decision Making ... 599

V Machine Learning

- 19 Learning from Examples ... 651
- 20 Learning Probabilistic Models ... 721
- 21 Deep Learning ... 750
- 22 Reinforcement Learning ... 789

VI Communicating, perceiving, and acting

- 23 Natural Language Processing ... 823
- 24 Deep Learning for Natural Language Processing ... 856
- 25 Computer Vision ... 881
- 26 Robotics ... 925

VII Conclusions

- 27 Philosophy, Ethics, and Safety of AI ... 981
- 28 The Future of AI ... 1012
- Appendix A: Mathematical Background ... 1023
- Appendix B: Notes on Languages and Algorithms ... 1030
- Bibliography ... 1033 ([pdf](#) and [bib data](#))
- Index ... 1069 ([pdf](#))

[Exercises \(website\)](#)

[Figures \(pdf\)](#)

[Code \(website\)](#); [Pseudocode \(pdf\)](#)

Covers: [US](#), [Global](#)

△ AI: A Modern Approach

Modified: Jun 09, 2021



Na próxima aula trataremos do uso da IA para resolver problemas, buscando um modelo geral que possa servir de base a agente(s) inteligente(s).




Formação dos grupos para os exercícios? .




Telemedicina: um novo desafio para a IA

Nesta aula vamos começar a tratar da resolução de problemas de forma inteligente e o uso de IA em sistemas de automação. Um conceito essencial para isso é o conceito de agentes e especialmente de agentes inteligentes. O processo de automação é dado pelo conhecimento dos dados e processos, dos agentes (inteligentes) e do chamado ciclo de automação, que consiste do sensoriamento, processamento e acionamento. Cada agente deve ser capaz de trabalhar nestes limites, tendo como destaque a captura do seu task environment, isto é, do conjunto de problemas que deve resolver. Seguindo o livro-texto, vamos falar da classificação dos agentes, sempre de "olho nas aplicações" e em especial, em aplicações ligadas a automação. O uso em telemedicina - que ilustra esta seção - é um dos desafios que se apresentam para a inteligência de máquina e IA.

 [Materia da SICStus Prolog](#)

 [Aula2](#)

[slides da Aula2](#)

 [Formação dos grupos](#)





Até a próxima aula!