

Agentes Inteligentes

Capítulo 2



Sumário

- Definição de Agente
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

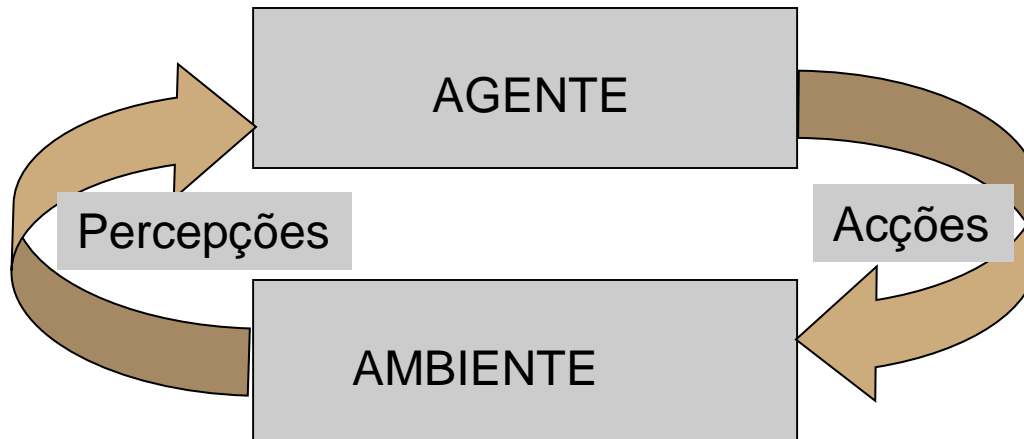
Definições de Agentes (do dicionário)

- “*Agente é Aquele que opera*”, ou “*Tudo o que age*”, ou
- “*Aquele que é encarregue dos negócios de outrem*”.

Definição de Agentes

- Um agente é tudo o que é capaz de captar/perceber o ambiente onde se encontra através de sensores e actua nesse ambiente através de actuadores

O Que é um Agente?



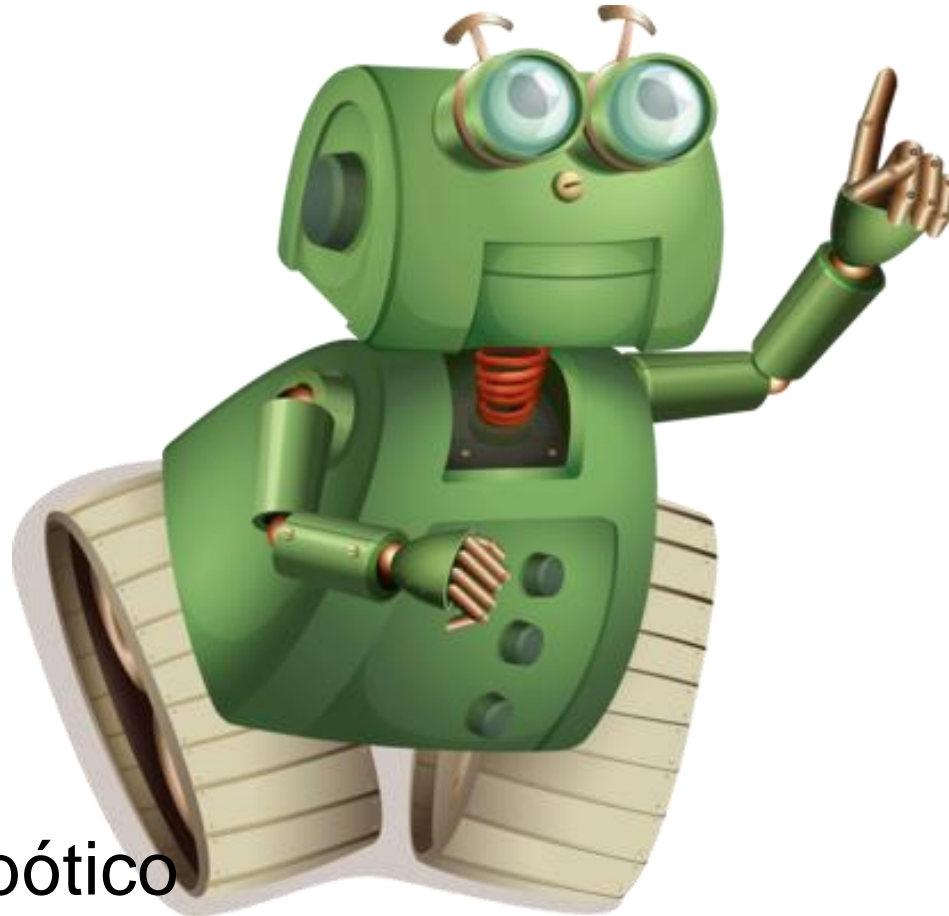
O aspeto importante dos agentes é que eles devem ser não só inteligentes mas também “autónomos”, ou seja, capazes de agir de forma independente de outros agentes e do utilizador.

Agentes....



- Agente humano
 - Sensores: olhos, orelhas e outros órgãos
 - Actuadores: mãos, pernas, boca e outras partes do corpo

Agentes....



- Agente robótico
 - Sensores: câmaras e infravermelhos
 - Actuadores: partes motoras

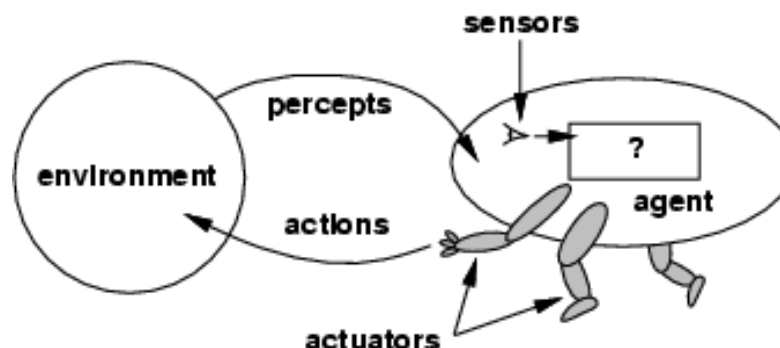
Sumário

- Definição de agentes
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

Agentes e Ambientes



Agentes e ambientes

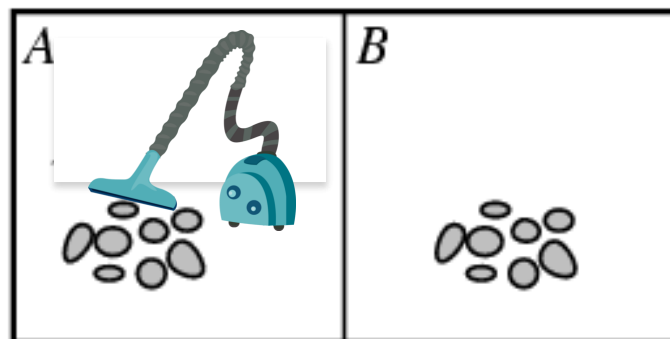


- A **função agente** mapeia uma sequência de percepções em acções:

$$[f: P^* \rightarrow \mathcal{A}]$$

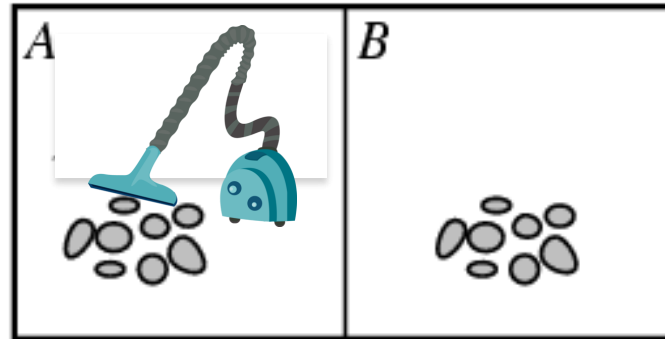
- O **programa agente** é executado numa **plataforma** para produzir f
- agente = plataforma + programa

Mundo do aspirador



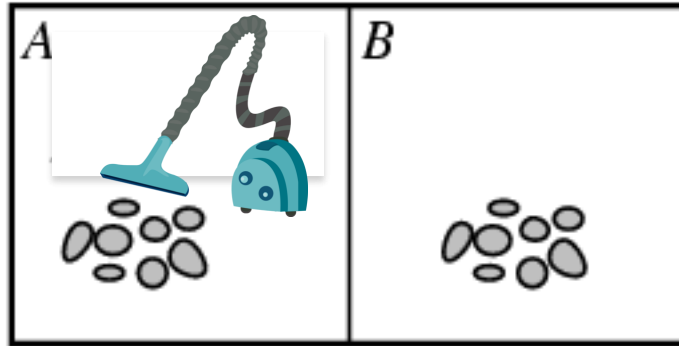
- Percepções [localização, conteúdo]
 - Localização: A, B
 - Conteúdo: Limpo, Sujo
- Acções: Esquerda, Direita, Aspirar

Agente aspirador



Sequência de Percepções	Acção
[A, Limpo]	Direita
[A, Sujo]	Aspirar
[B, Limpo]	Esquerda
[B, Sujo]	Aspirar
[A, Limpo],[A, Limpo]	Direita
[A, Limpo],[A, Sujo]	Aspirar
...	...

Agente aspirador



Função AgenteAspirador ([posição, estadoLocal])

devolve acção

Se estadoLocal = **Sujo** **então** **devolve** **Aspirar**

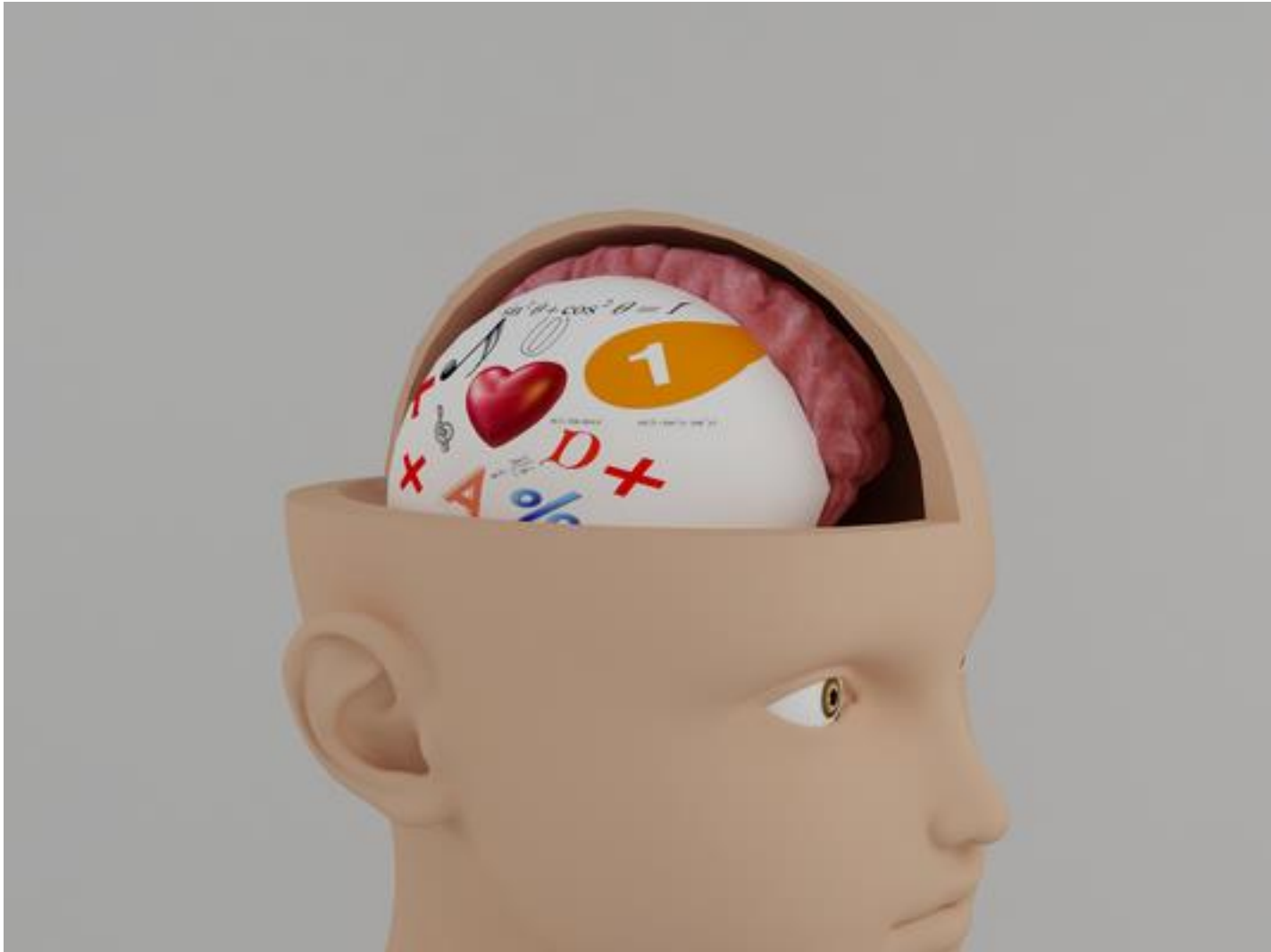
Senão se posição = **A** **então** **devolve** **Direita**

Senão se posição = **B** **então** **devolve** **Esquerda**

Sumário

- Definição de agentes
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

Racionalidade



Agentes “racionais”

- Um agente deve procurar fazer “**o que está certo**”, baseado nas suas percepções e nas acções que pode tomar
- A acção certa é aquela que dá maior **expectativa** de sucesso ao agente

Agentes “racionais”

- ***Medida de desempenho***: critério **objectivo** que mede o sucesso do comportamento do agente
- Por ex^o, medida de desempenho do agente aspirador pode ser a sujidade aspirada, tempo utilizado, electricidade consumida, ruído gerado, etc.

Medida de desempenho



- Por ex^o, medida de desempenho do agente aspirador pode ser a sujeidade aspirada, tempo utilizado, electricidade consumida, ruído gerado, etc.

Agentes racionais

- Maximizar a medida de desempenho

Por cada sequência de percepções possível, um agente racional deve seleccionar uma acção que é suposto maximizar a sua medida de desempenho, dada a informação disponibilizada pela sequência de percepções e eventualmente pelo conhecimento que o agente possui.

Agentes racionais

- Racionalidade \neq omnisciência
 - Percepções podem não disponibilizar conhecimento que é importante
- Racionalidade \neq clarividência
 - Resultado de uma acção pode não estar de acordo com o esperado
- Logo, racionalidade \neq sucesso

Agents Racionais

- Racionalidade \Rightarrow exploração, aprendizagem, autonomia

Racionalidade versus Autonomia: o agente “autónomo”

- Um agente é **autónomo** se o seu conhecimento for determinado apenas pela sua experiência (com capacidade de aprender e adaptar-se)
 - Agentes podem tomar acções para obter informações úteis (recolha de informação, exploração)

Sumário

- Definição de agentes
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

Caracterização de um agente

- PEAS
 - Performance (desempenho)
 - Environment (ambiente)
 - Actuators (actuadores)
 - Sensors (sensores)

PEAS: agente Prof^a de Inglês

- Desempenho
 - Notas dos alunos no teste
- Ambiente
 - Conjunto de alunos
- Actuadores
 - Monitor: exercícios, sugestões, correcções
- Sensores
 - Teclado: respostas



PEAS: Agente médico

- Desempenho
 - Saúde do paciente, custos
- Ambiente
 - Paciente, hospital, funcionários
- Actuadores
 - Monitor: questões, testes, diagnósticos, tratamentos
- Sensores
 - Teclado: sintomas, respostas



PEAS: Agente taxista

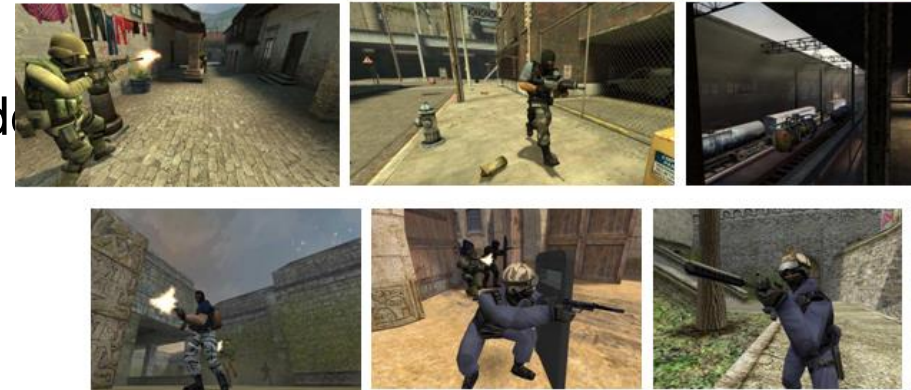
- Desempenho
 - Segurança, destino, lucros, legalidade, conforto
- Ambiente
 - Clientes, estradas, trânsito, transeuntes, tempo
- Actuadores
 - Volante, acelerador, travão, buzina, pisca
- Sensores
 - GPS, conta km, velocímetro, nível do depósito, temperatura do óleo



PEAS: Agente NPC in a First-Person Shooting Game



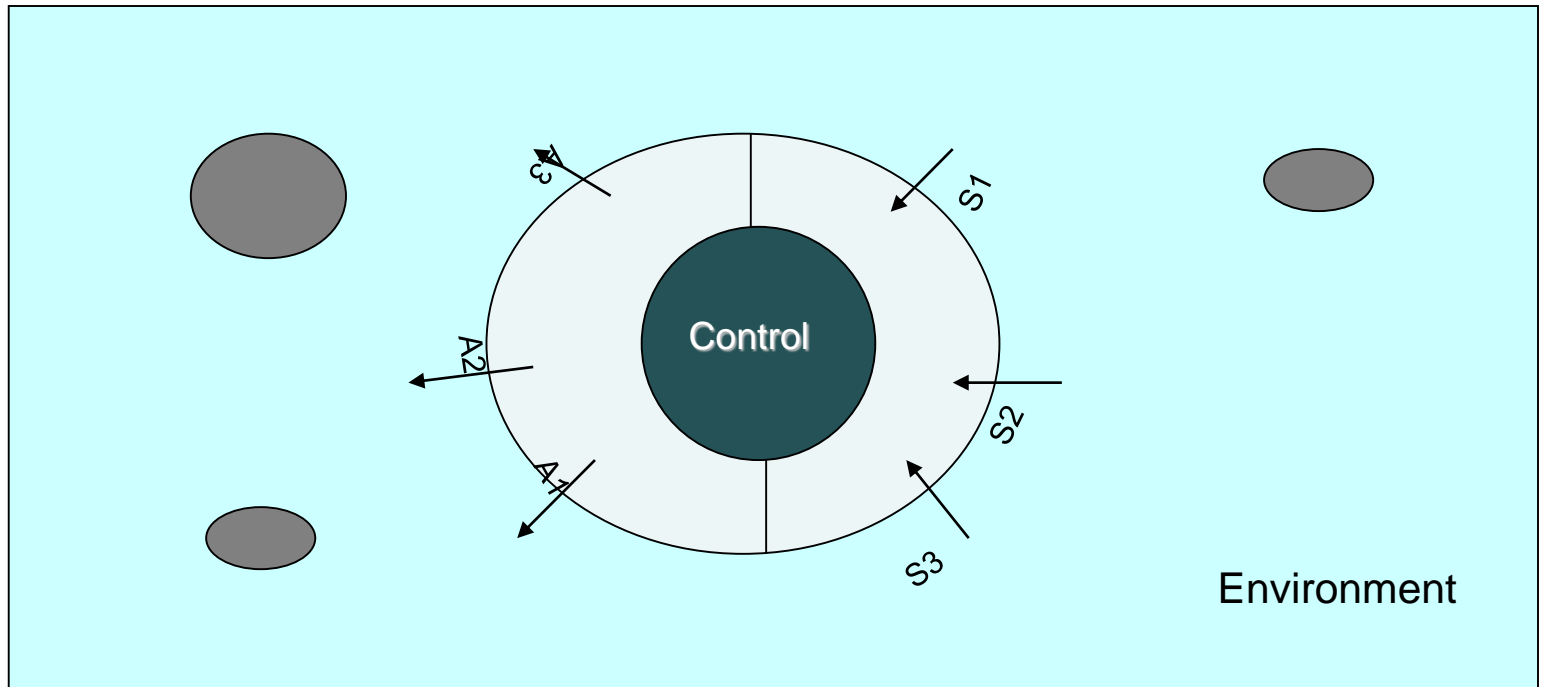
- Desempenho
 - Sobrevivência, Pontos, Credibilidade
- Ambiente
 - Ambiente do jogo
- Actuadores
 - Mover para frente/trás, lados, ir para um waypoint, rodar, saltar e abaixar, apontar, disparar, mudar de arma
 - Enviar mensagem a colega de equipa, colocar/desarmar bomba
- Sensores
 - Posição, Campo de visão de 90°, Percepção outros agentes (amigos/inimigos), Percepção armas/equipamento no chão, Radar, Percepção de colegas de equipa, Localização da bomba, Comunicação, Percepção mensagens via Radar ou Chat



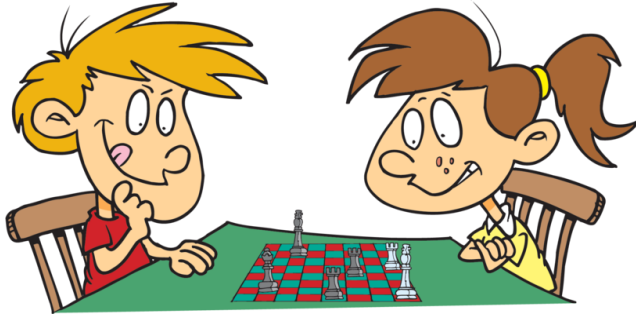
Sumário

- Definição de agentes
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

Os agentes habitam um “Ambiente”

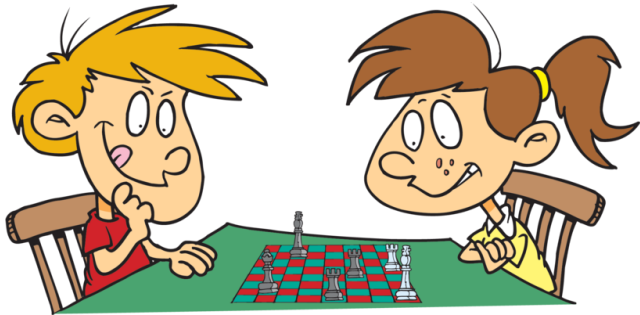


Tipos de Ambientes– *Observável* *vs. Parcialmente Observável*



- Num ambiente “observável” os sensores do agente dão acesso ao estado completo do ambiente em cada instante de tempo, pelo que não é necessário manter um estado interno sobre o mundo.
- Ou seja, o agente consegue a cada instante obter informação correcta e actualizada do mundo que o rodeia.
- A maioria dos ambientes não são totalmente observáveis tendo em conta o aparelho sensorial comum.
- Quanto mais observável é um ambiente mais fácil a criação de agentes que nele operem.

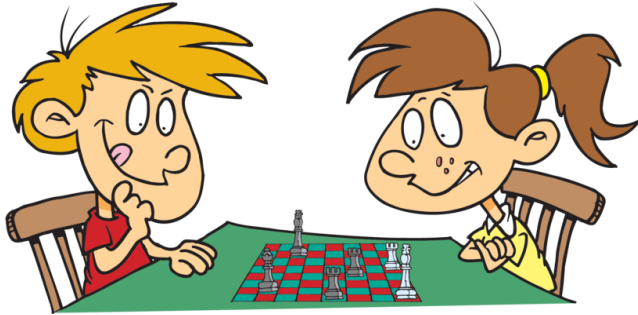
Tipos de Ambientes— *Estático* *versus Dinâmico*



- **Estático** (vs. dinâmico): o ambiente não é alterado enquanto o agente decide que acção vai tomar.
- ambiente é **semi-dinâmico** se o ambiente permanece inalterado com a passagem do tempo mas a qualidade do desempenho do agente é alterada
- Note-se que num ambiente dinâmico o mundo está em constante alteração, e portanto as acções dos agentes falham frequentemente.

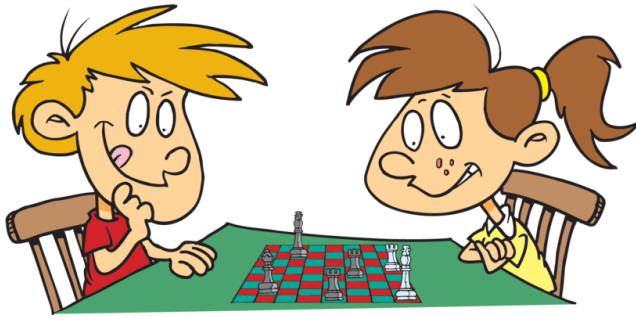
Tipos de Ambientes–

Determinístico vs. estocástico



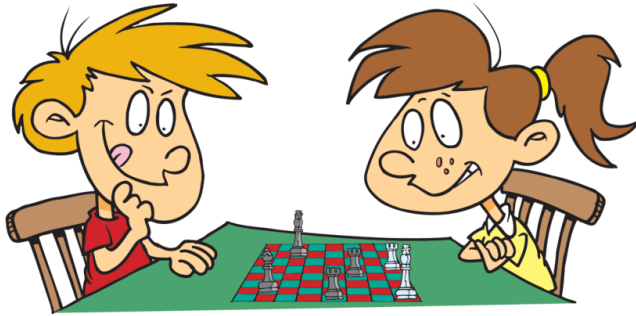
- **Determinístico** (vs. estocástico): O estado seguinte do ambiente é determinado somente em função do estado actual e da acção executada pelo agente – não há incerteza para o agente sobre o estado do mundo quando o agente executa uma acção.
- Se o ambiente é sempre determinístico excepto para as acções de outros agentes, então o ambiente é **estratégico**.
- O mundo físico é olhado como não determinístico.
- Os ambientes não determinísticos são bastante mais complexos de lidar quando da criação dos agentes.

Tipos de Ambientes– *Episódico vs. Sequencial*



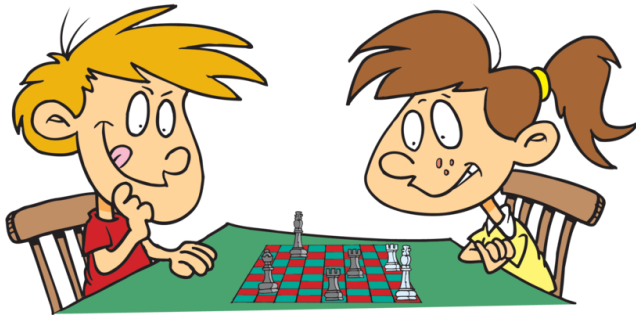
- **Episódico** (vs. sequencial): A experiência do agente está dividida em episódios atômicos (em que cada episódio consiste em percepção+acção do agente) e a escolha de cada acção em cada episódio depende apenas do próprio episódio.

Tipos de Ambientes– *Discreto versus Contínuo*



- **Discreto** (vs. contínuo): O agente tem um número limitado de percepções e acções distintas que estão claramente definidas.

Tipos de Ambientes– *Single versus Multi-Agente*



- **Agente único** (vs. multi-agente): Só existe um agente no ambiente.

Tipos de Ambientes

	Xadrez com relógio	Análise de Imagem	Condutor de táxi
Completamente observável?	Sim	Sim	Não
Determinístico?	Estratégico	Sim	Não

Tipos de Ambientes

	Xadrez com relógio	Análise de Imagem	Condutor de táxi
Episódico?	Não	Sim	Não
Estático?	Semi	Semi	Não

Tipos de Ambientes

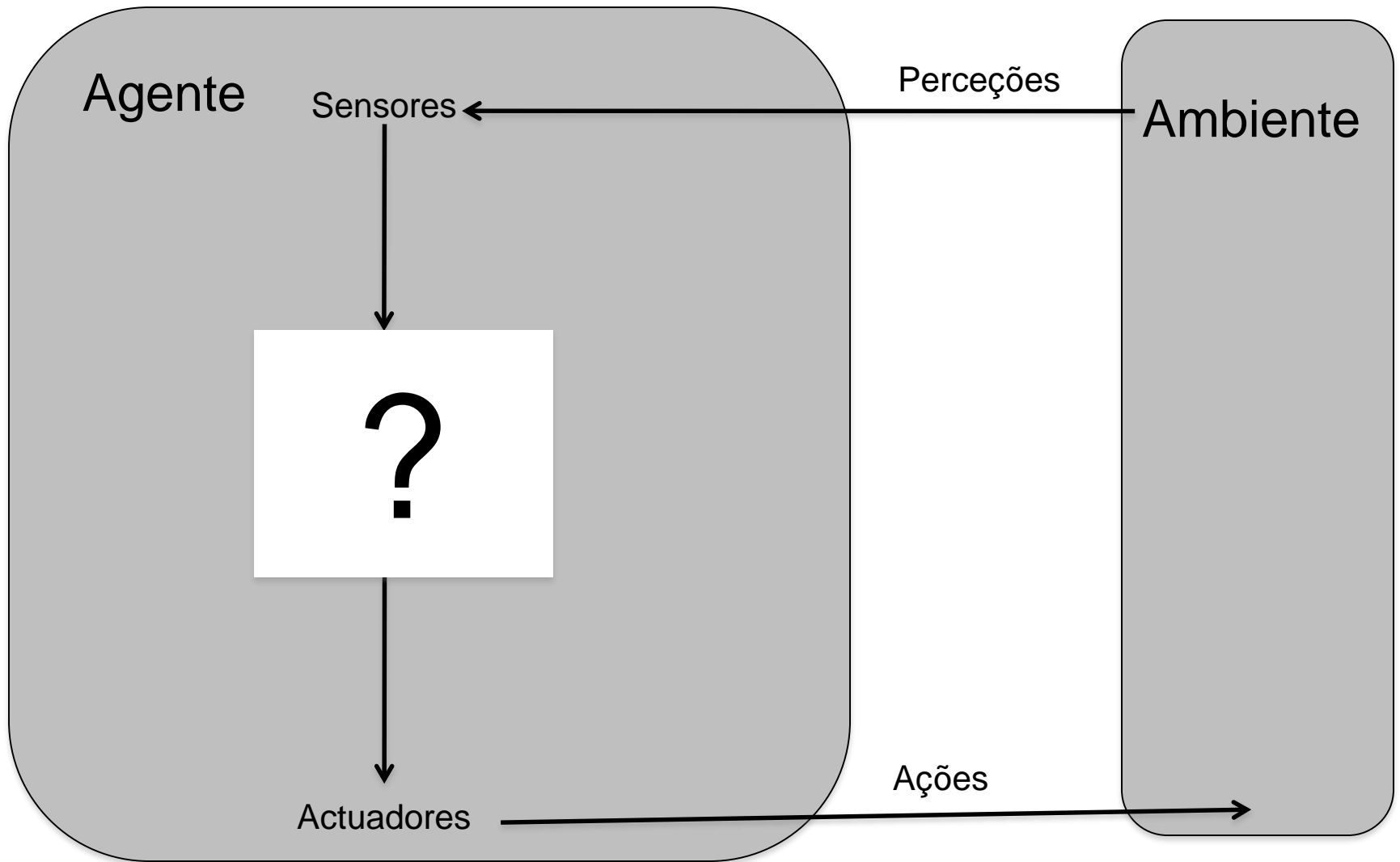
	Xadrez com relógio	Análise de Imagem	Condutor de táxi
Discreto?	Sim	Não	Não
Agente único?	Não	Sim	Não

Tipos de Ambientes

- O tipo de ambiente determina o tipo de agente
- O mundo real é:
 - Parcialmente observável (vs. completamente observável)
 - Estocástico (vs. determinístico)
 - Sequencial (vs. episódico)
 - Dinâmico (vs. estático)
 - Contínuo (vs. discreto)
 - Multi-agente (vs. agente único)

Sumário

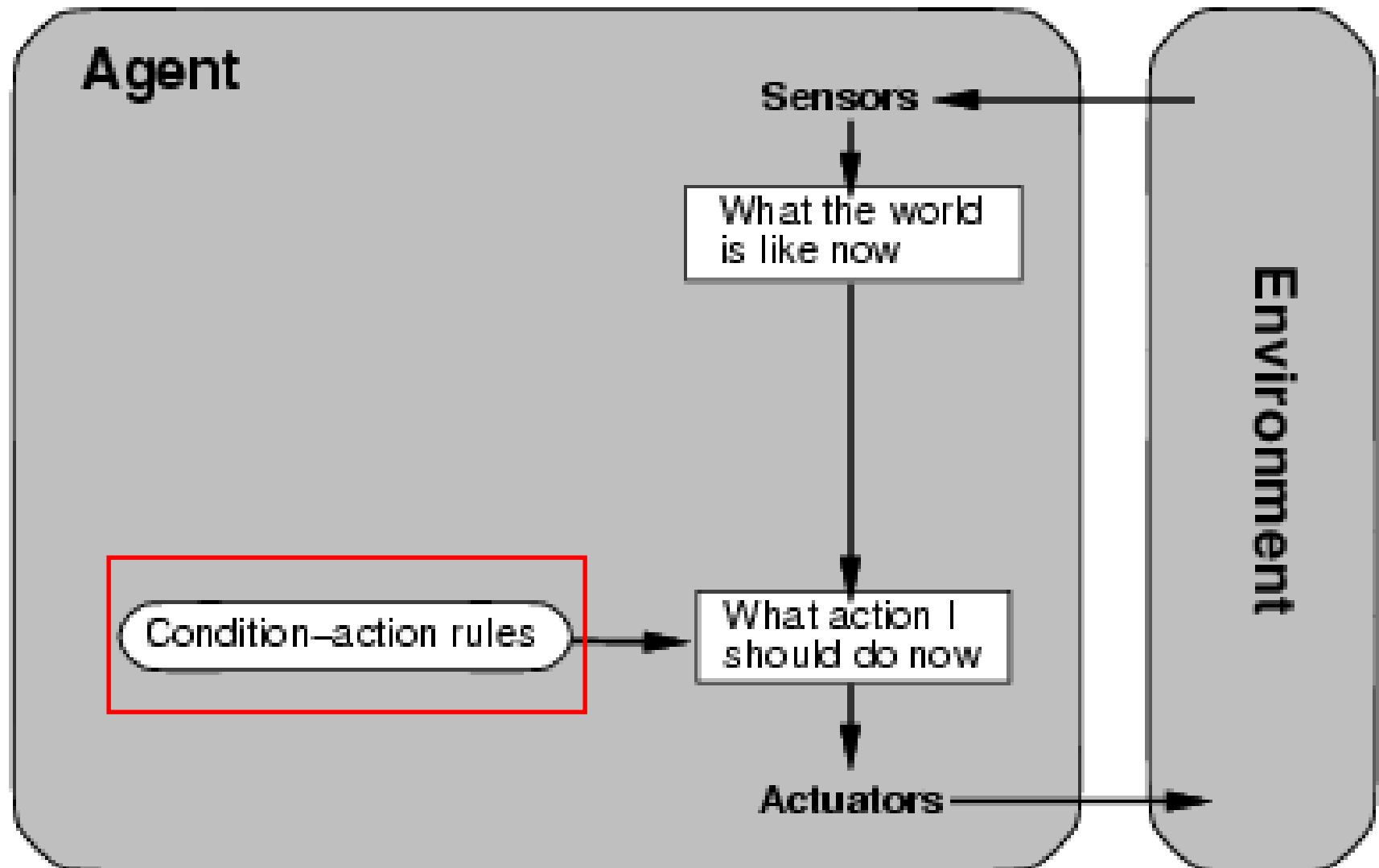
- Definição de agentes
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes



Tipos de Agentes

- Agentes reflexos simples
- Agentes reflexos baseados em modelos
- Agentes baseados em objetivos
- Agentes baseados em utilidade
- Agentes com aprendizagem

Agentes de reflexos simples



Agentes de reflexos simples

Função AgenteReflexoSimples (*percepção*)
devolve *acção*

Estático: *regras* (conj^o de regras condição-acção)

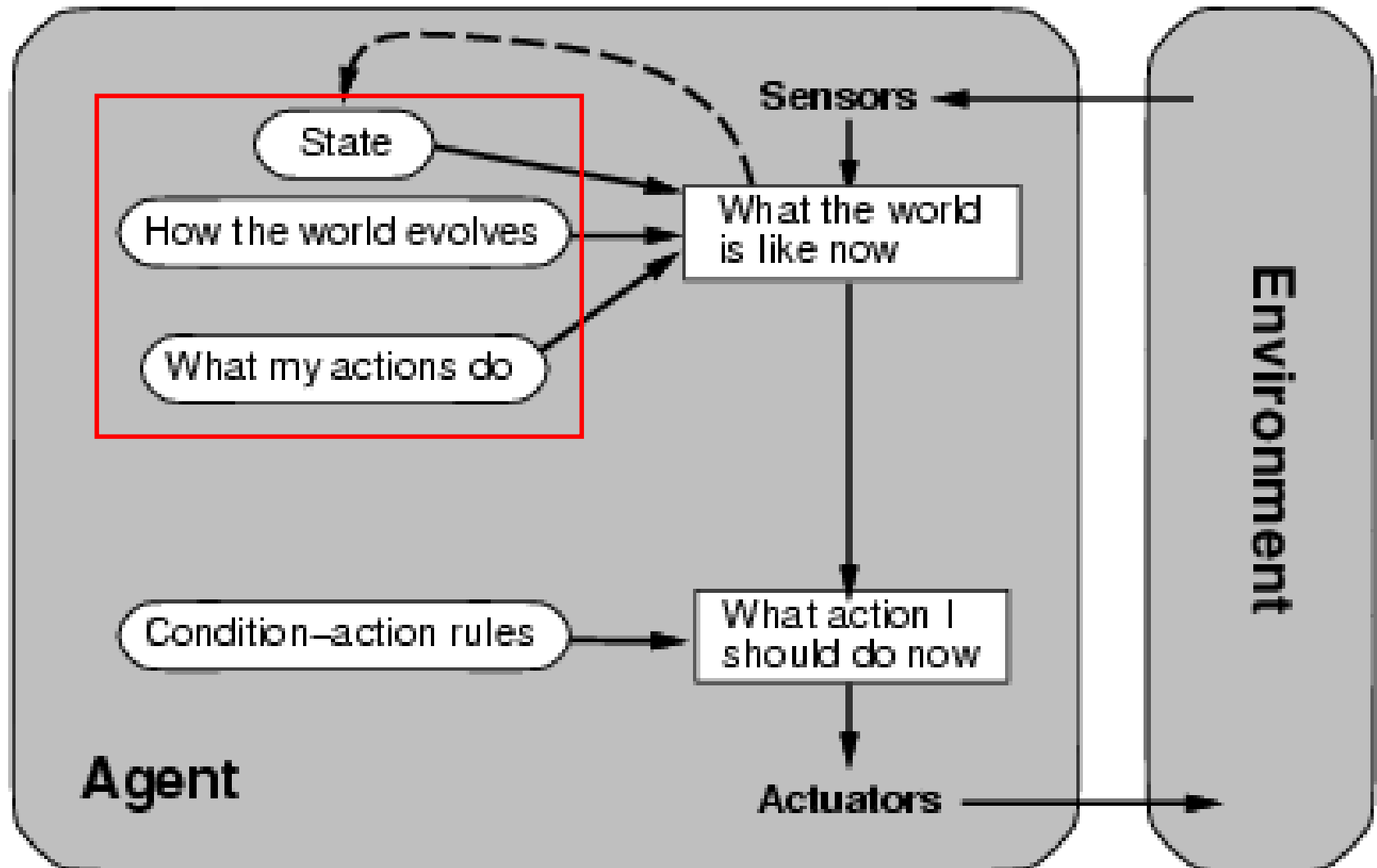
estado \leftarrow InterpretarInput(*percepção*)

regra \leftarrow EmparelhaRegra(*estado*, *regras*)

acção \leftarrow RegraAcção[*regra*]

devolve *acção*

Agentes reflexos baseado em modelos



Agentes reflexos baseado em modelos

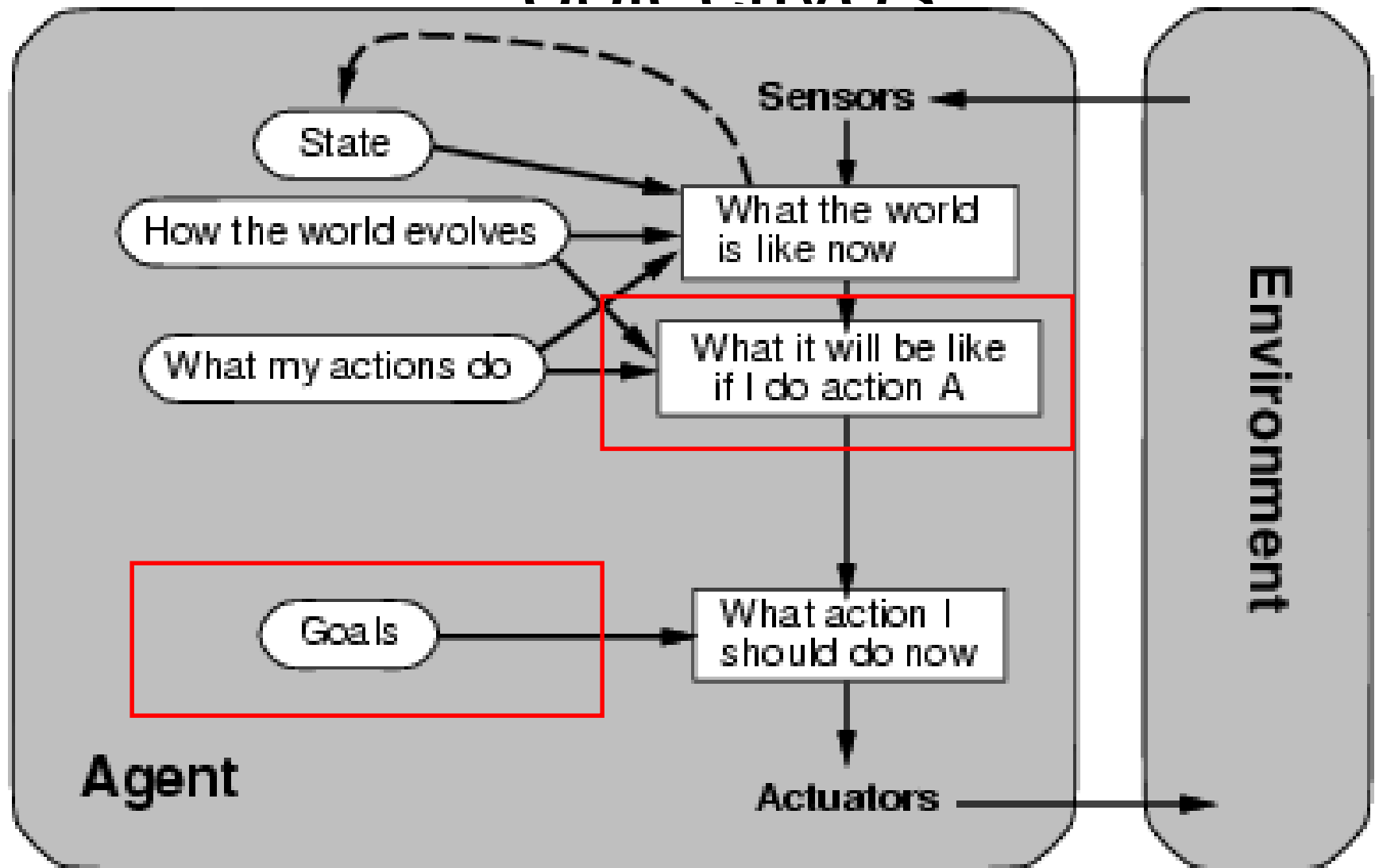
Função AgenteReflexosBaseadoEmModelos (*percepção*)
devolve *acção*

Estático: *estado* (descrição do estado do mundo)
regras (conj^o de regras condição-acção)
acção (a acção mais recente)

estado ← **ActualizaEstado**(*estado*, *acção*, *percepção*)
regra ← EmparelhaRegra(*estado*, *regras*)
acção ← RegraAcção[*regra*]
devolve *acção*

Também chamados agentes de reflexos simples com estado interno.

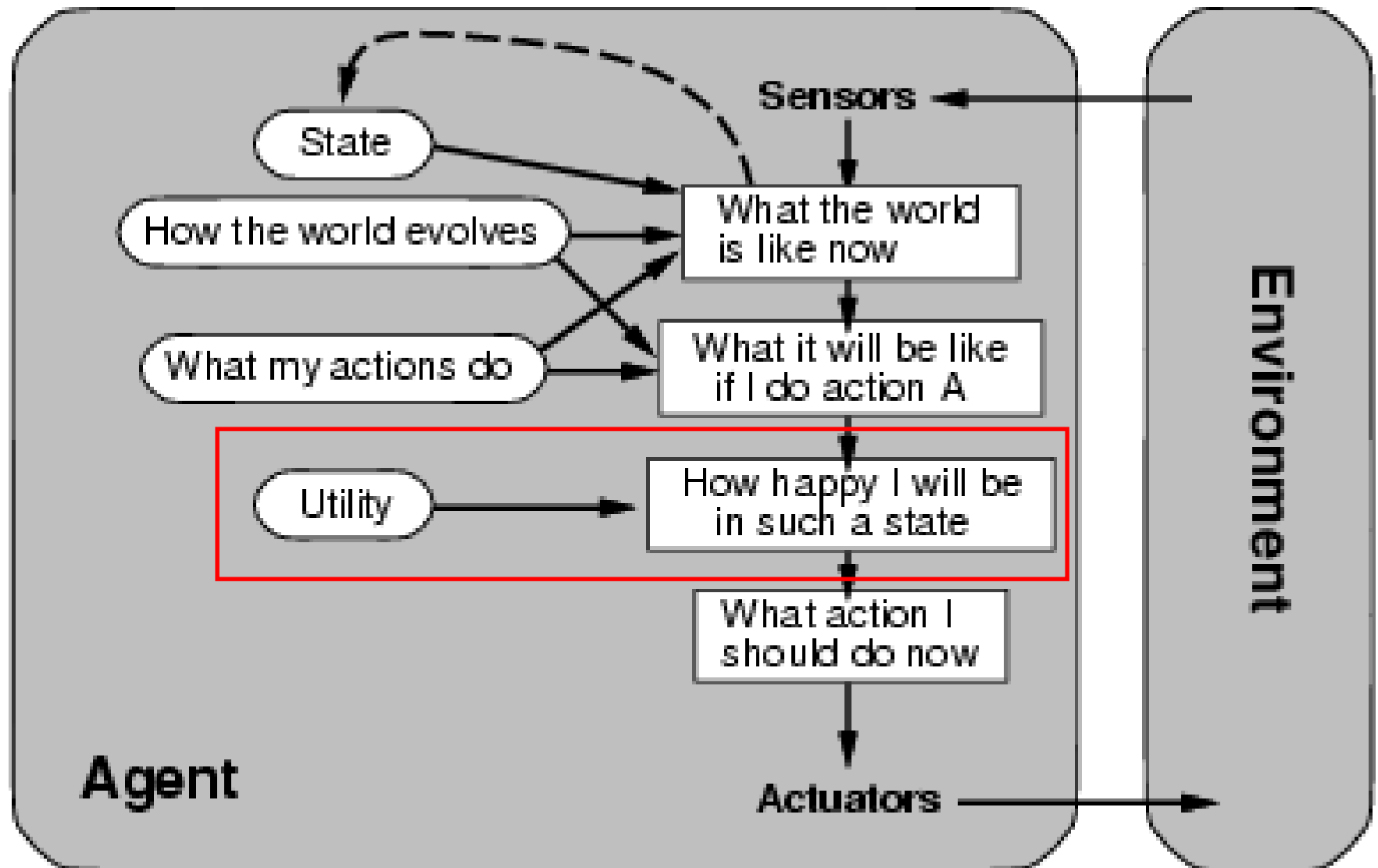
Agentes baseados em objetivos



Agente baseado em objectivos

- Agente tem um (ou mais) objectivo(s)
- Por ex^o, considere-se um agente taxista que cujo objectivo é chegar a um destino
 - Chegando a um cruzamento, o agente decide avançar, virar à direita ou virar à esquerda em função do objectivo
- Acrescenta a um agente de reflexos simples considerações sobre o futuro, a fim de alcançar os objectivos

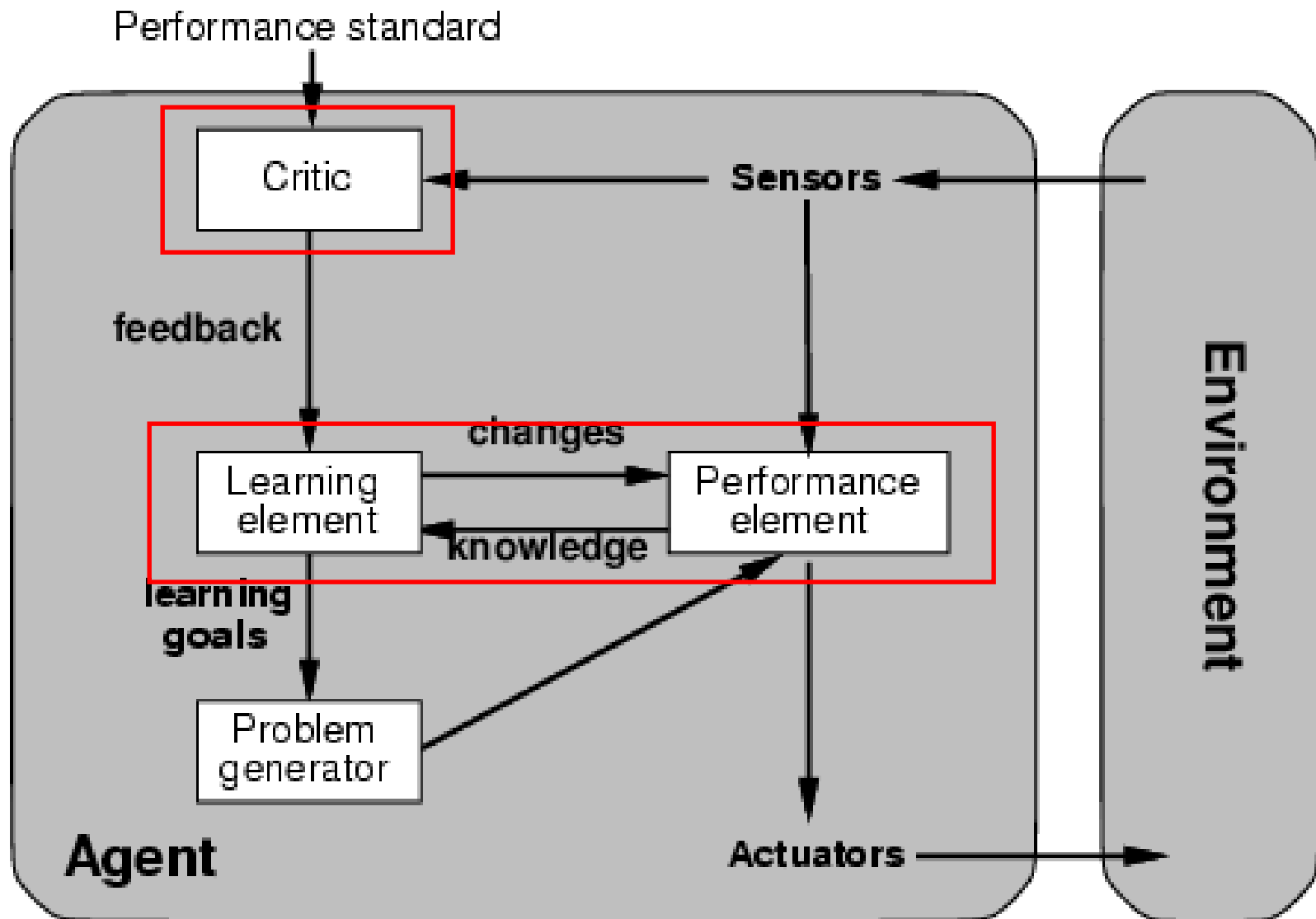
Agentes baseados em utilidade



Agente baseado em utilidade

- Agente tem uma função de utilidade
- A função de utilidade permite estabelecer preferências entre sequências de estados que permitem atingir os mesmos objectivos
- Por ex^o, considere-se um agente taxista que pretende chegar a um destino
 - A função de utilidade permite distinguir as diferentes formas de chegar ao destino, em função do tempo, da despesa, da segurança, ...

Agentes com aprendizagem



Agente com aprendizagem

- Correspondem à ideia de máquina inteligente caracterizada por Turing (1950)
 - Agente actua num mundo inicialmente desconhecido
- Elemento de aprendizagem
 - Torna o agente mais eficiente ao longo do tempo
 - Usa feedback da crítica que avalia actuação do agente de acordo com o desempenho espectável
- Elemento de desempenho
 - Responsável por seleccionar as acções do agente
 - Corresponde ao essencial de um agente
- Elemento de geração de problemas
 - Sugere acções que podem trazer informação útil

Sumário

- Definição de Agente
- Agentes e ambientes
- Racionalidade
- PEAS: caracterização de um agente
- Tipos de ambientes
- Tipos de agentes

Exemplos de Agentes

Capítulo 2- Continuação



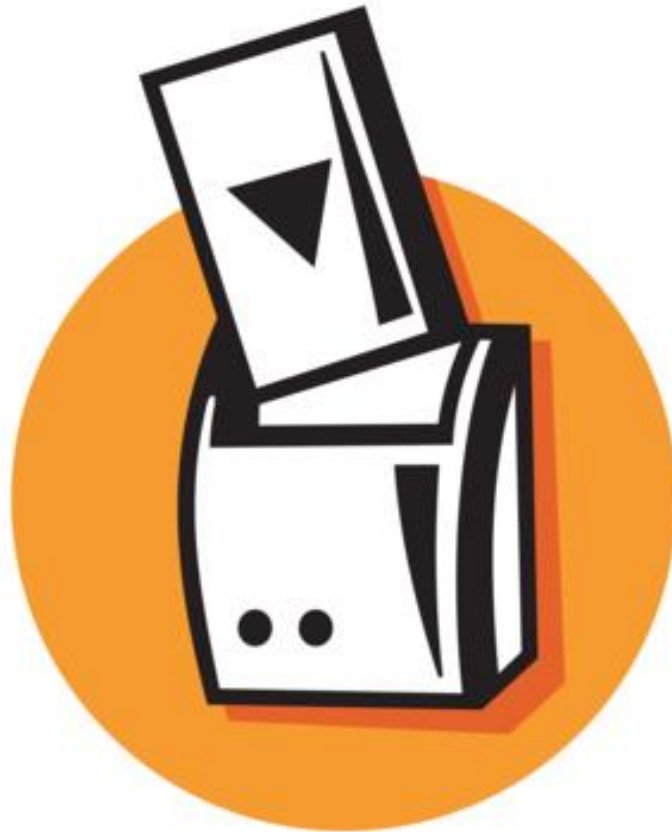
Sumário

- Exemplos de agentes (ex^{os} de exames)
 - Reflexos simples
 - Reflexos simples baseados em modelos (também chamados com estado interno)

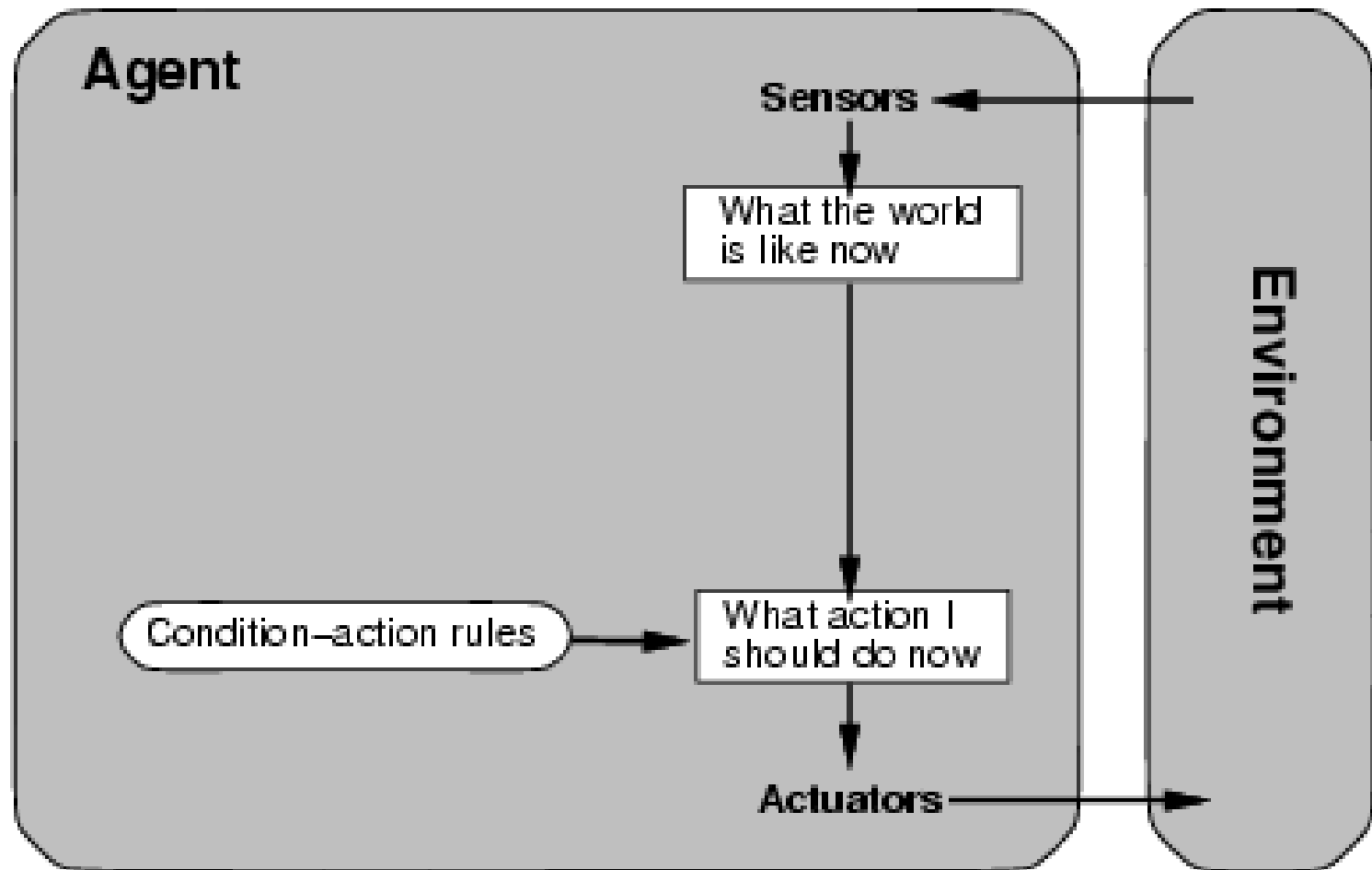
Exemplos de Agentes

- Enunciado
 - Caracterização do agente
- Exercício
 - Tipo de agente
 - Reflexos simples
 - Reflexos simples com estado interno
 - Definição da percepção
 - Definição da função agente

Agente Venda Bilhetes Metro



Agente Venda Bilhetes Metro: Agente Reflex Simples



Agente Venda Bilhetes Metro: simples

- Considere um agente que faz a **venda de bilhetes de metro**.
- Em cada percepção, o agente recebe o custo do bilhete pretendido e a quantia introduzida.
- A acção devolvida poderá ser **“QUANTIA_CERTA”** ou o **valor** correspondente ao troco ou à quantia introduzida (no caso desta ser insuficiente).

Agente Venda Bilhetes Metro

- Percepções

Estrutura com: custo e quantia

Em LISP:

```
(destruct percepcao  
          custo quantia)
```

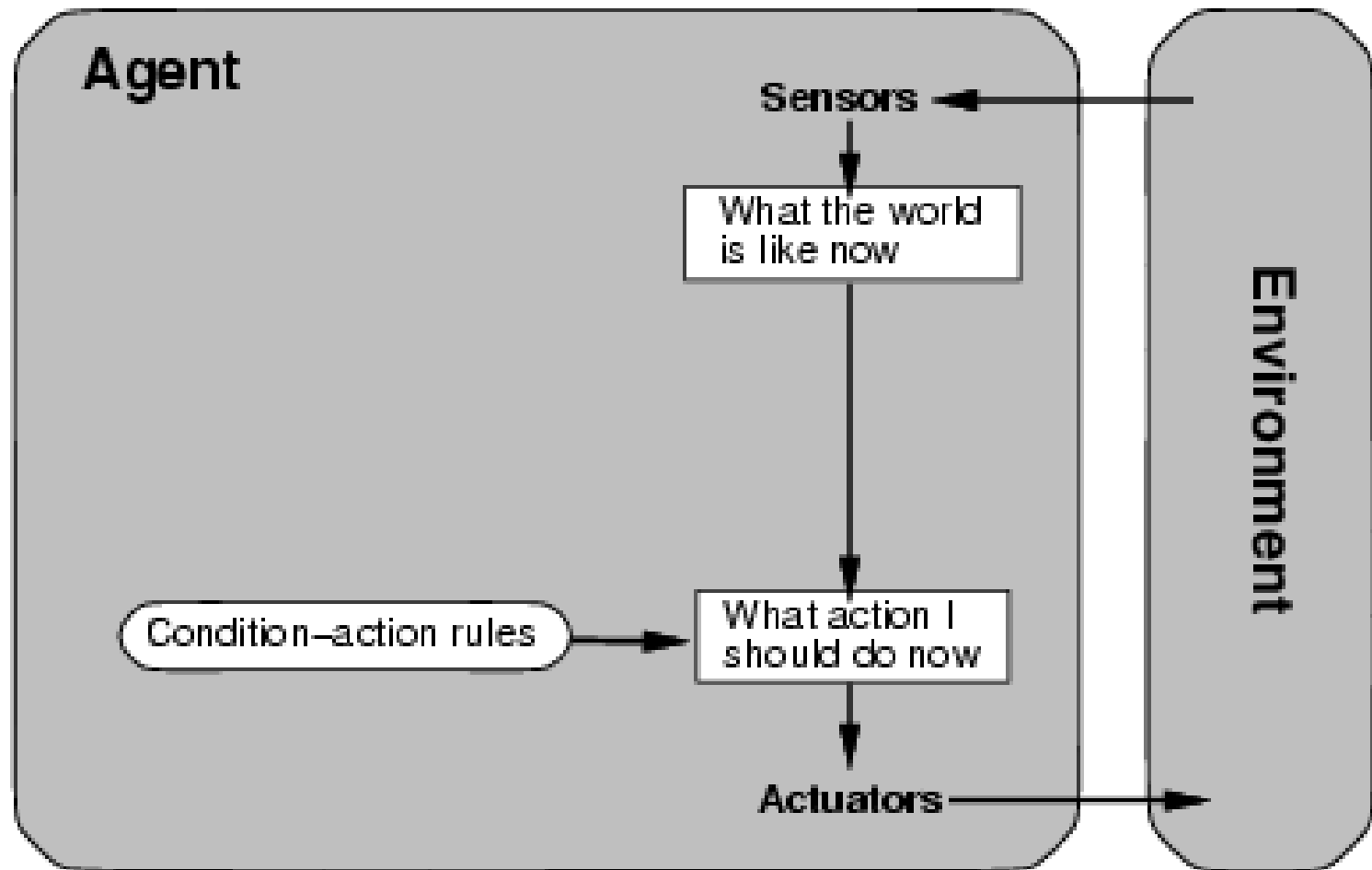
- Acções

- QUANTIA_CERTA

- Valor troco

- Valor quantia

Agente Venda Bilhetes Metro: Agente Reflex Simples



Agente Venda Bilhetes Metro

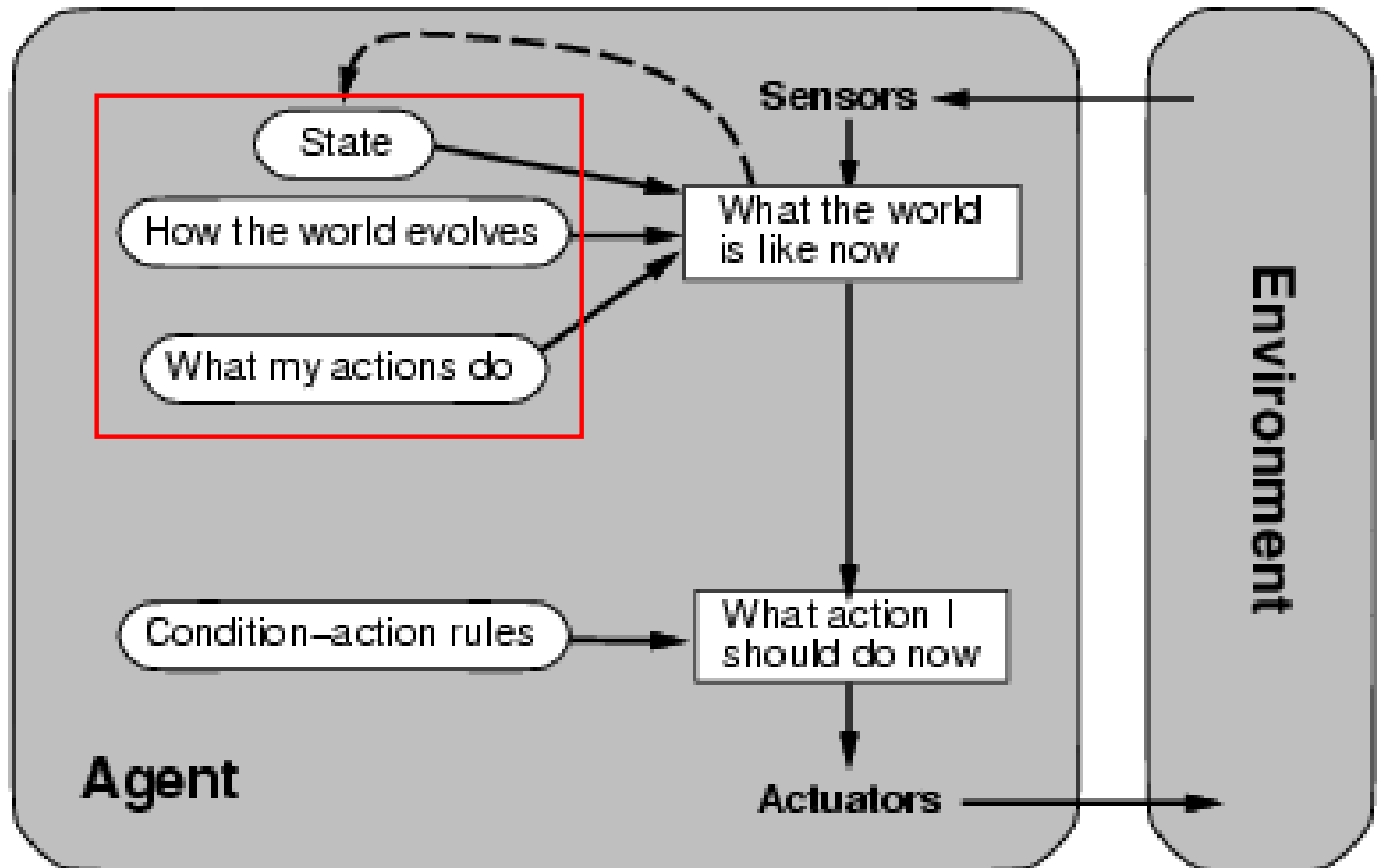
```
(defun agente (p)
  (let ((custo (percepcao-custo p))
        (quantia (percepcao-quantia p)))
    (cond ((= custo quantia) `QUANTIA_CERTA)
          ((< custo quantia) (- quantia custo))
          (t quantia))))
```

Agente Venda Bilhetes Metro

```
> (setf p1 (make-percepcao :custo 0.70 :quantia 1))
#S(PERCEPCAO :CUSTO 0.7 :QUANTIA 1)
> (setf p2 (make-percepcao :custo 0.70 :quantia 0.5))
#S(PERCEPCAO :CUSTO 0.7 :QUANTIA 0.5)
> (setf p3 (make-percepcao :custo 0.70 :quantia 0.7))
#S(PERCEPCAO :CUSTO 0.7 :QUANTIA 0.7)

> (agente p1)
0.3
> (agente p2)
0.5
> (agente p3)
QUANTIA_CERTA
```


Agentes reflexos baseado em modelos



Agentes com estado interno

- Como criar um estado interno em LISP?
 - Usando **closures** (ambientes léxicos)

```
(defun cria-agente (<inicialização>)
  (let ((<var-estado> <valor>)...)
    #'(lambda (<percepção>)
        <actualiza vars-estado>
        <devolve acção>)))
```

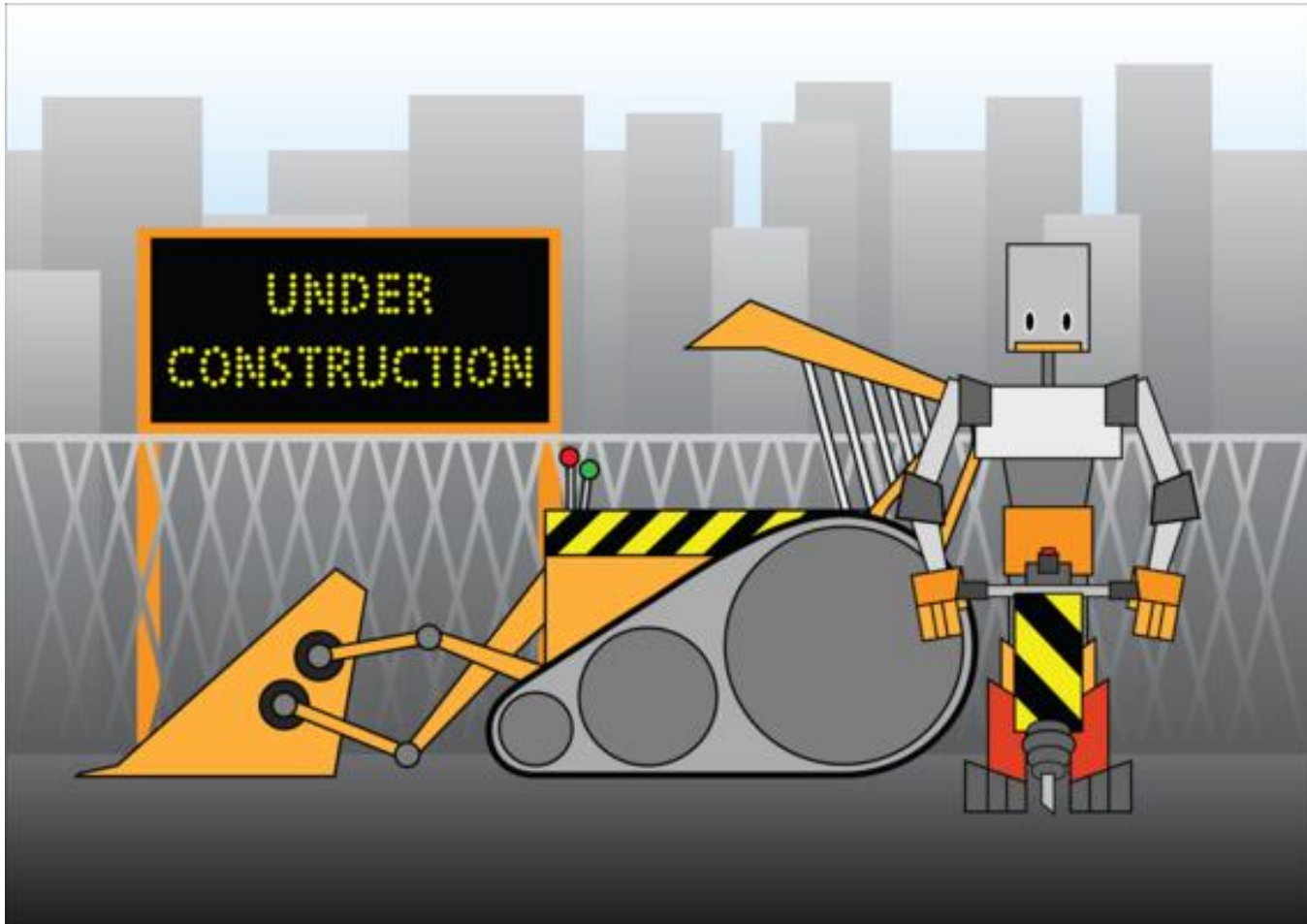
Agentes com estado interno

- Exemplos de **closures** (ambientes léxicos)

```
(defun cria-contador ()  
  (let ((contador 0))  
    #'(lambda () (incf contador))))
```

```
(defun cria-acumulador (acum)  
  #'(lambda (inc) (incf acum inc)))
```

Agente Robô em Fábrica de Brinquedos



Agente Robô em Fábrica de Brinquedos

- Considere um agente **robot numa fábrica de brinquedos.**
- Os sensores do robot detectam quando é que um brinquedo passou pela passadeira.
- O robot recebe informação dos seus sensores e quando tiverem passado 3 brinquedos necessita de **EMPACOTAR** o brinquedo.

Agente Robô que Empacota Brinquedos

- Percepção

```
(destruct percepcao  
  (passa-brinquedo nil))
```

- Acções

- EMPACOTAR
- NAO_FAZ_NADA

- Agente de reflexos simples com estado interno (nº de brinquedos por empacotar)

Agente Robô que Empacota Brinquedos

```
(defun cria-agente ()  
  (let ((n-brinquedos 0))  
    #'(lambda (p)  
      (when (percepcao-passa-brinquedo p)  
        (incf n-brinquedos))  
      (cond ((= n-brinquedos 3)  
              (setf n-brinquedos 0) 'EMPACOTAR)  
            (t 'NAO_FAZ_NADA))))))
```

Agente Robô que Empacota Brinquedos

```
(defun cria-agente ()
```

```
  (let ((n-brinquedos 0))
```

← Estado

```
    #'(lambda (p)
```

```
      (when (percepcao-passa-brinquedo p)
```

```
        (incf n-brinquedos))
```

```
      (cond ((= n-brinquedos 3)
```

```
        (setf n-brinquedos 0) 'EMPACOTAR
```

```
        (t 'NAO_FAZ_NADA))))))
```

Acções

Agente Robô que Empacota Brinquedos

```
> (setf p1 (make-percepcao))
#S(PERCEPCAO :PASSA_BRINQUEDO NIL)
> (setf p2 (make-percepcao :passa_brinquedo t))
#S(PERCEPCAO :PASSA_BRINQUEDO T)

> (setf ag-emp (cria-agente))
#(FUNCTION :LAMBDA (p) ...)
> (funcall ag-emp p1)
NAO_FAZ_NADA
> (funcall ag-emp p2)
NAO_FAZ_NADA
> (funcall ag-emp p2)
NAO_FAZ_NADA
> (funcall ag-emp p2)
EMPACOTAR
```

Agente num Parque de Estacionamento



Agente num Parque de Estacionamento

- Considere um agente que faz a gestão de um **parque de estacionamento**.
- O sensor detecta quando entra um carro e quando sai um carro.
- A acção devolvida poderá ser **LEVANTAR_ENTRADA**, **LEVANTAR_SAIDA**, **LEVANTAR_ENTRADA_SAIDA** (entra um carro e sai outro ao mesmo tempo), **ESPERAR** (não há lugar) ou **NAO_FAZ_NADA**.

Agente num Parque de Estacionamento

- Percepção

```
(defstruct percepcao  
  entra-carro sai-carro)
```

- Acções

- LEVANTAR_ENTRADA
- LEVANTAR_SAIDA
- LEVANTAR_ENTRADA_SAIDA
- ESPERAR
- NAO_FAZ_NADA

- Agente de reflexos simples com estado interno
(nº de lugares livres)

Agente num Parque de Estacionamento

```
(defun cria-agente
```

(n)

*Estado Interno:
nº de carros*

Agente num Parque de Estacionamento

```
(defun cria-agente (n)
  #'(lambda (p)
    (let ((entra (percepcao-entra-carro p))
          (sai (percepcao-sai-carro p)))
```

*Estado Interno:
nº de carros*

*Estado Interno:
nº de carros*

Agente num Parque de Estacionamento

```
(defun cria-agente (n)  
  #'(lambda (p)  
    (let ((entra (percepcao-entra-carro p))  
          (sai (percepcao-sai-carro p)))  
      (cond ((and entra sai)  
              `LEVANTAR_ENTRADA_SAIDA)
```

Agente num Parque de Estacionamento

```
(defun cria-agente (n)
  #'(lambda (p)
    (let ((entra (percepcao-entra-carro p))
          (sai (percepcao-sai-carro p)))
      (cond ((and entra sai)
              'LEVANTAR_ENTRADA_SAIDA)
            (sai (incf n) 'LEVANTAR_SAIDA))
```


Agente num Parque de Estacionamento

```
(defun cria-agente (n)
  #'(lambda (p)
    (let ((entra (percepcao-entra-carro p))
          (sai (percepcao-sai-carro p)))
      (cond ((and entra sai)
              `LEVANTAR_ENTRADA_SAIDA)
            (sai (incf n) `LEVANTAR_SAIDA)
            (entra (if (zerop n)
                        `ESPERAR
                        (progn
                          (decf n)
                          `LEVANTAR_ENTRADA) ) )
```

Agente num Parque de Estacionamento

```
(defun cria-agente (n)
  #'(lambda (p)
    (let ((entra (percepcao-entra-carro p))
          (sai (percepcao-sai-carro p)))
      (cond ((and entra sai)
              `LEVANTAR_ENTRADA_SAIDA)
            (sai (incf n) `LEVANTAR_SAIDA)
            (entra (if (zerop n)
                        `ESPERAR
                        (progn
                          (decf n)
                          `LEVANTAR_ENTRADA) ) )
              (t NAO_FAZ_NADA) ) ) ) )
```

Exemplos mais completos...

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

Entretenimento

Transportes

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

Entretenimento

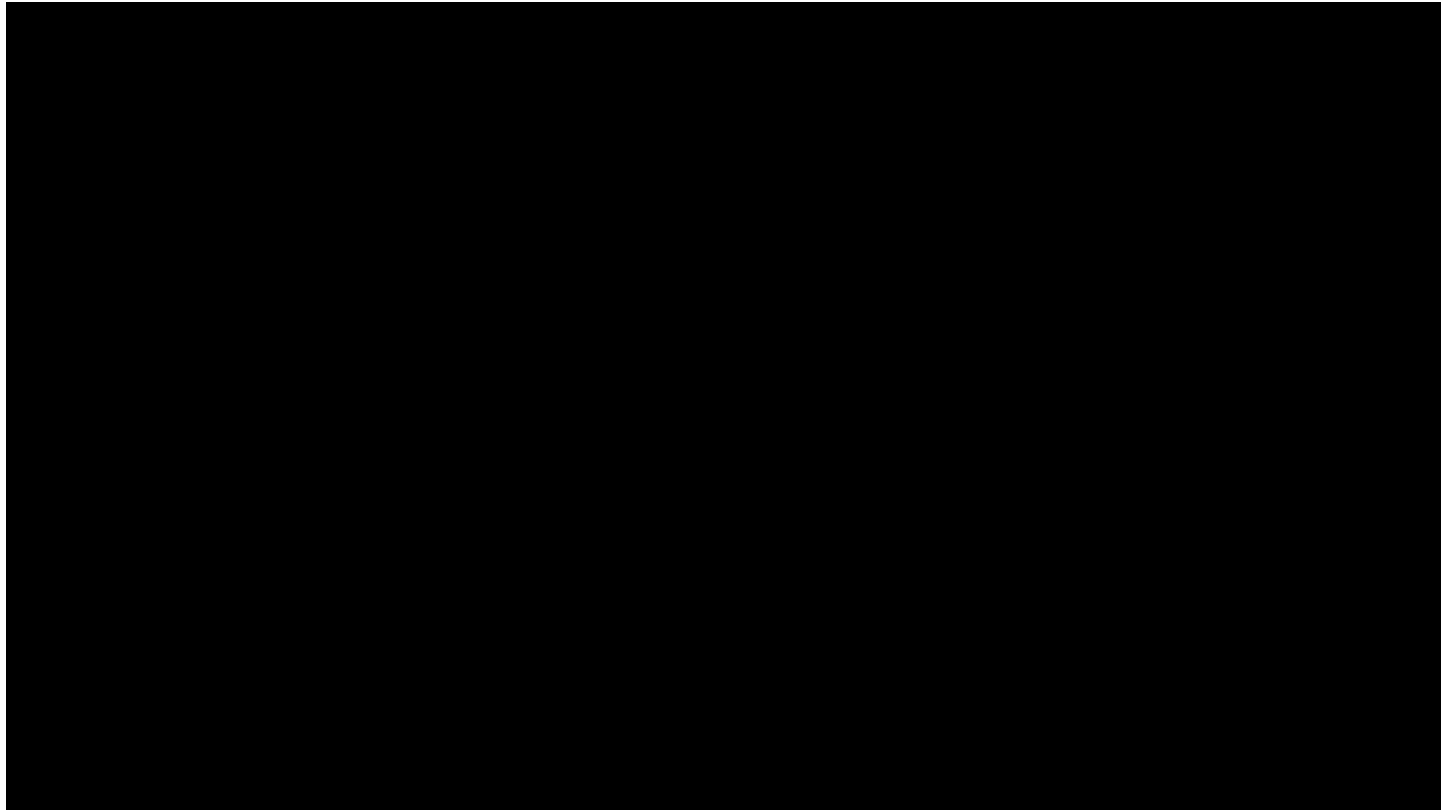
Transportes

Google car



Affective Intelligent Driving Agent

[http://www.youtube.com/watch?v=huQLyjws
kQo](http://www.youtube.com/watch?v=huQLyjws
kQo)



EducaçãoTerapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

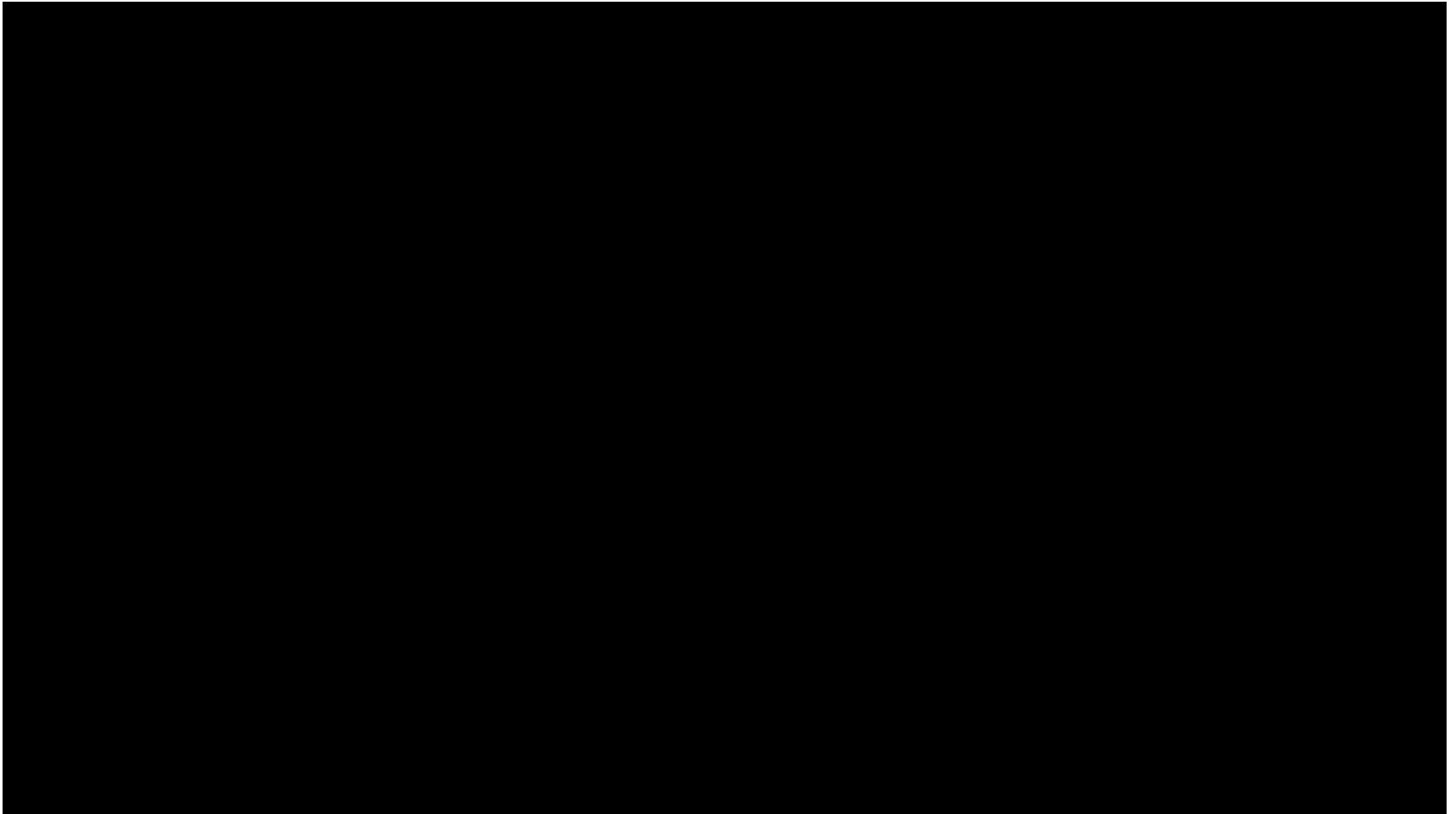
Fábricas

Casa

Entretenimento

Transportes

Agentes as Nurses



NurseBot



Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

Entretenimento

Transportes

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para Definição de Políticas

Fábricas

Casa

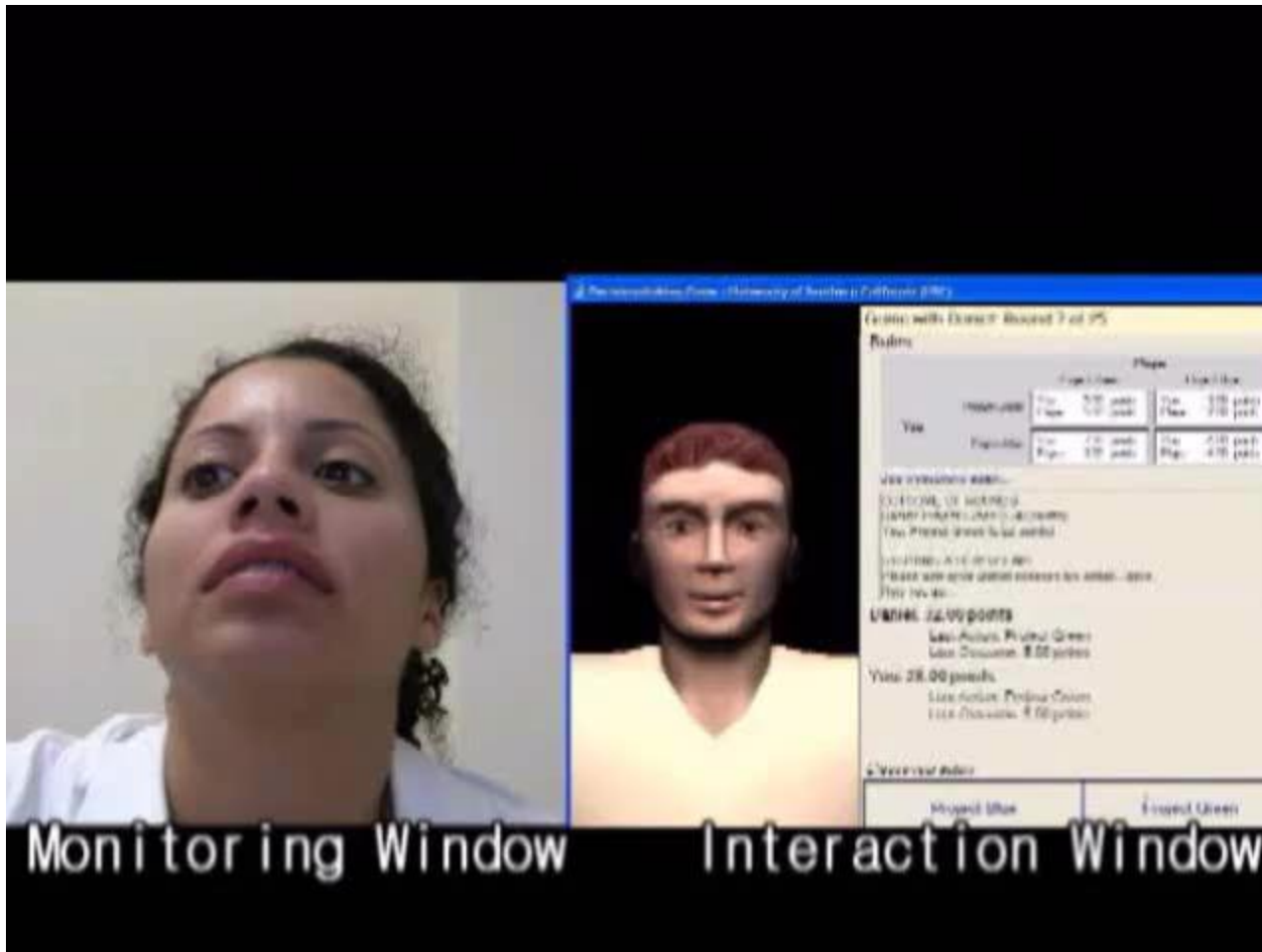
Entretenimento

Transportes

Agents for Social Simulation: Changing policies

The Virtual Bar

Agents and Collaboration with Humans



Agents for studying social processes

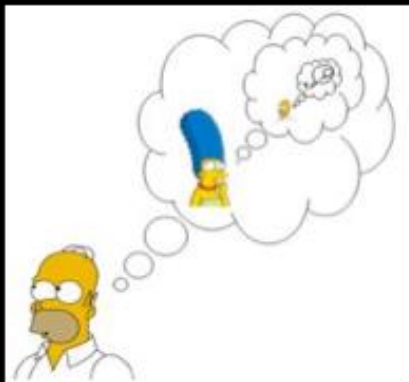
Game Theories

Classic Game Theory

Few players...



Rational decisions



Static concepts

Evolutionary Game Theory

Large populations...



Natural selection

Strategies that do well reproduce faster



Dynamical concepts

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

Entretenimento

Transportes

Educação Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

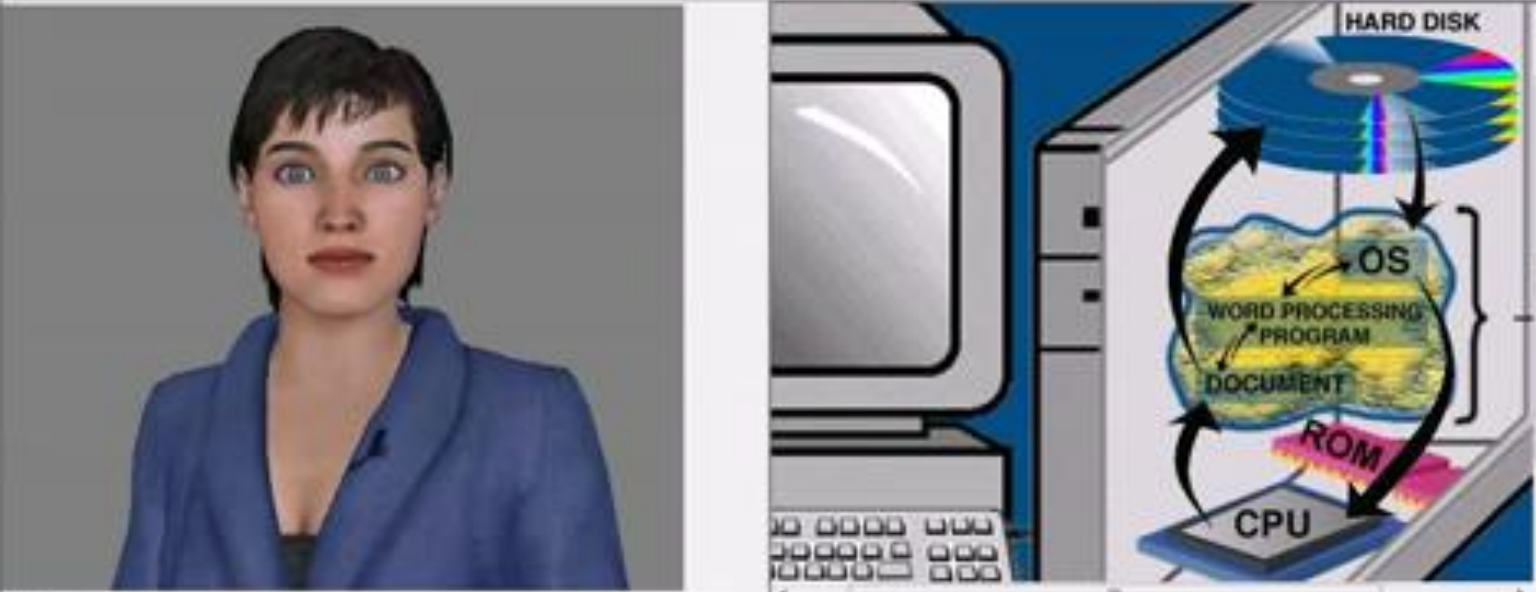
Entretenimento

Transportes

AutoTutor

File Edit Session Plugins Help

How does the operating system interact with the word processing program when you create a document?



Log of previous responses:

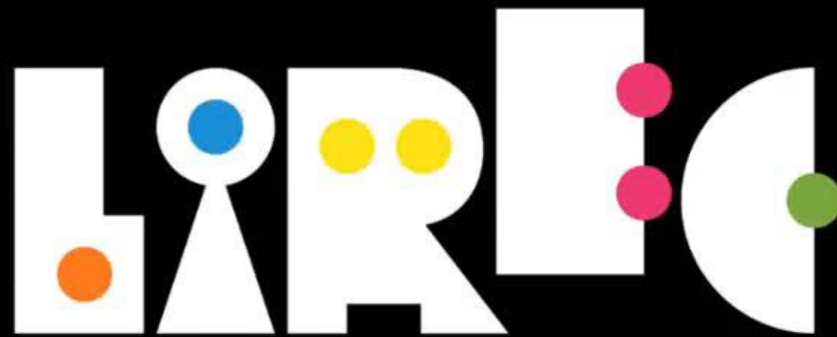
you have to work with a statistics program named business stat. The instructions to install the program say that you need a 486 50 computer with 32 megabytes of RAM. You have a 486 50 computer, but only 8 megabytes of ram. You decide that you want to upgrade your computer so you can run business stat. Consider this problem. How will your upgraded computer increase your computer's performance for running other programs?

Tutor: The operating system is first booted into ram. Then it provides services to the application programs that the user wants to run. For example, suppose that you want to use a word processing program. You want to create a document. Here's the question. How does the operating system interact with the word processing program when you create a document?

Enter your response here:

Submit

Affective Tutors: AutoTutor



LIVING WITH ROBOTS
AND INTERACTIVE COMPANIONS

Um Robô tutor de
Xadrez

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

Fábricas

Casa

Entretenimento

Transportes

Educação

Terapia & Saúde

Companheiros/Social

Simulação Social para
Definição de Políticas

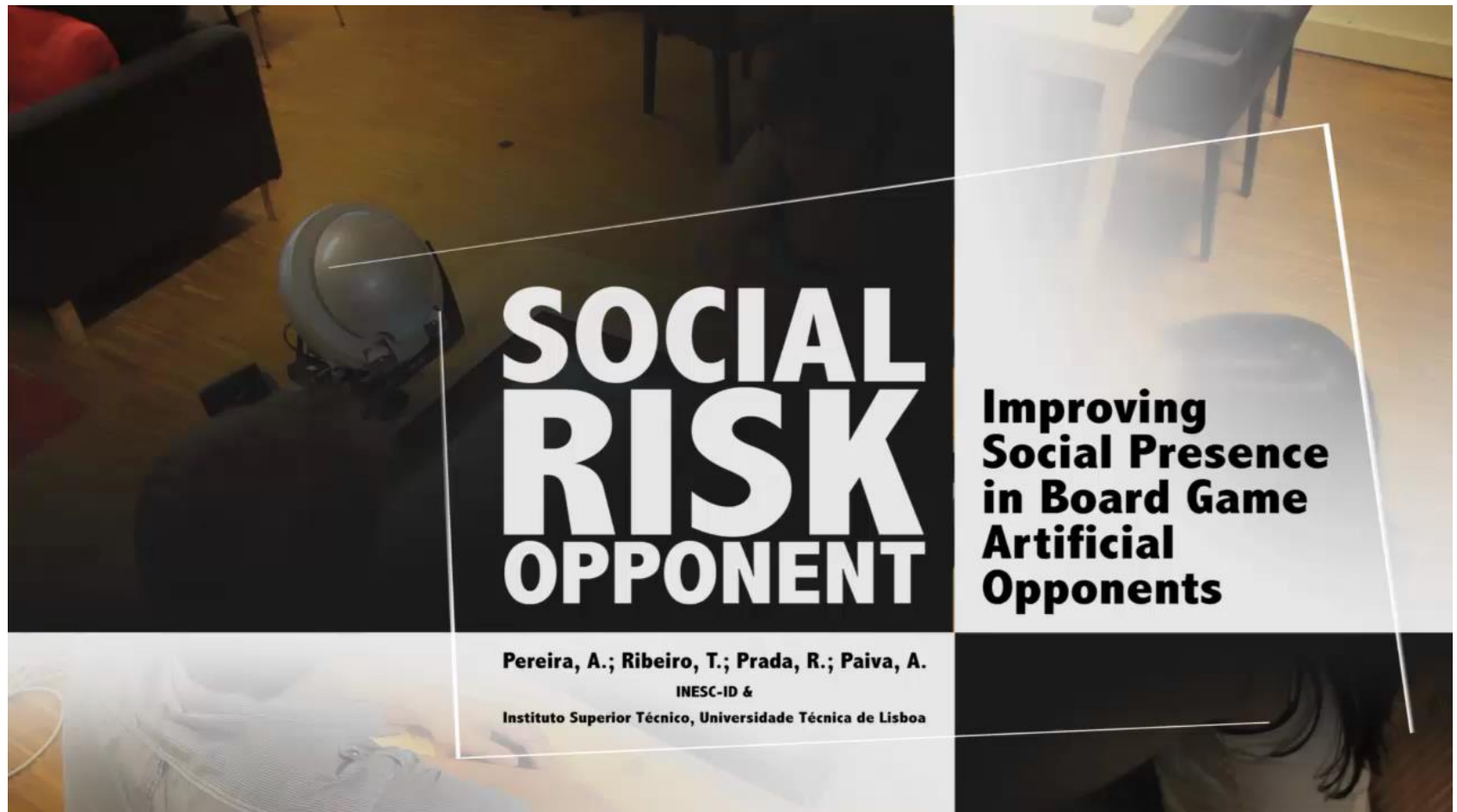
Fábricas

Casa

Entretenimento

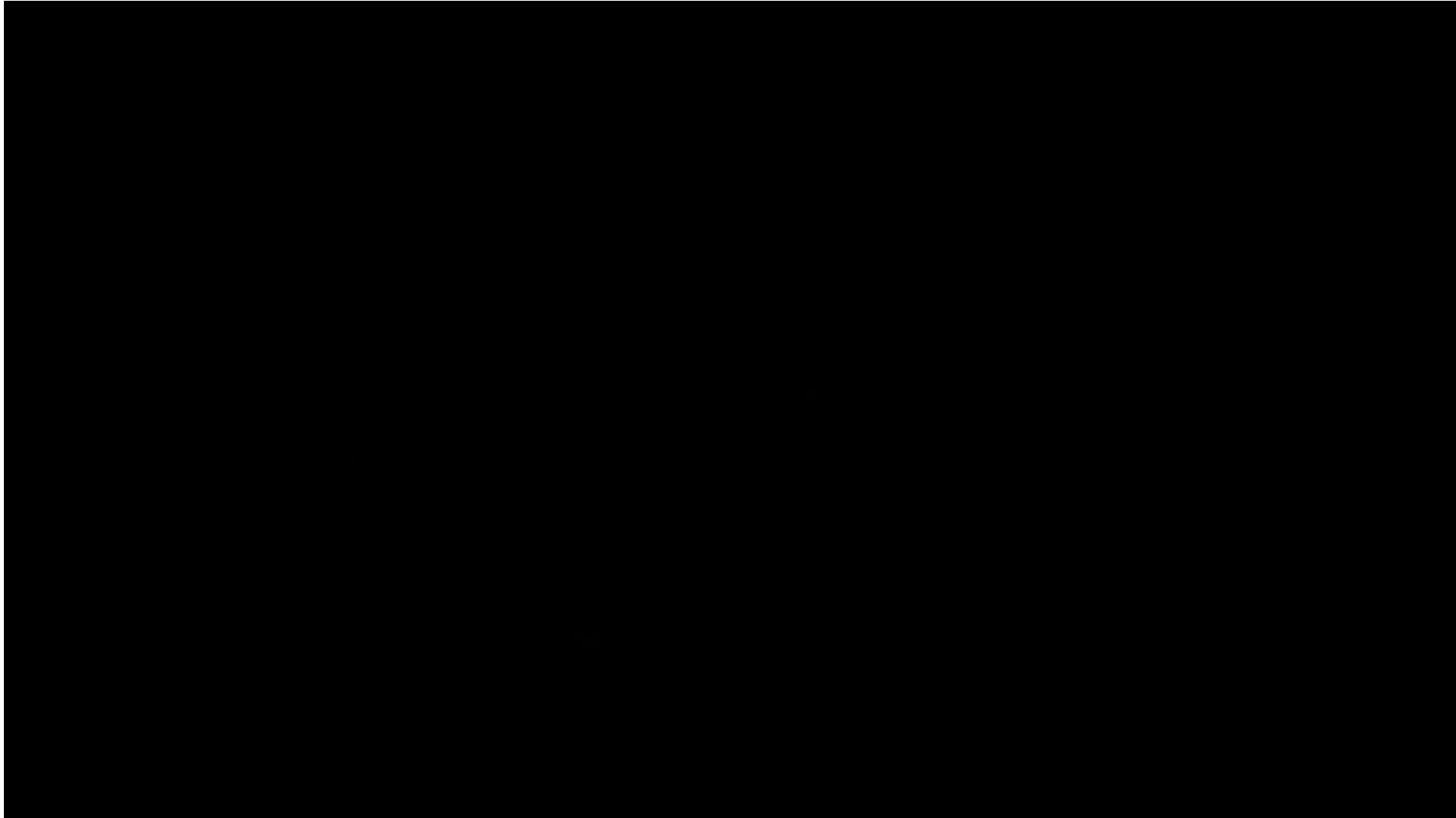
Transportes

Agents as opponents



Agents and Robots in Art

http://www.youtube.com/watch?v=_sUeGC-8dyk



Perguntas?

Outro exemplo: Agente Termóstato



Agente Termóstato

- Considere um agente **termóstato**.
- Por cada percepção, o agente recebe o valor da temperatura ambiente pretendida.
- A acção devolvida poderá ser **AQUECER, ARREFECER** ou **MANTER** em função da temperatura ambiente actual, que será imediatamente actualizada.

Agente Termóstato

- Percepção

`(defstruct percepcao
 temperatura)`

- Acções

- AQUECER
- ARREFECER
- MANTER

- Agente de reflexos simples com estado interno (temperatura ambiente actual)

Agente Termóstato

```
(defun cria-agente (temp)
  #'(lambda (p)
    (let ((nova-temp (percepcao-temperatura p)))
      (cond ((= nova-temp temp) `MANTER)
            ((< nova-temp temp)
             (setf temp nova-temp) `ARREFECER)
            (t (setf temp nova-temp) `AQUECER))))))
```