

Искусственный интеллект

Тема 3: Интелигентни агенти

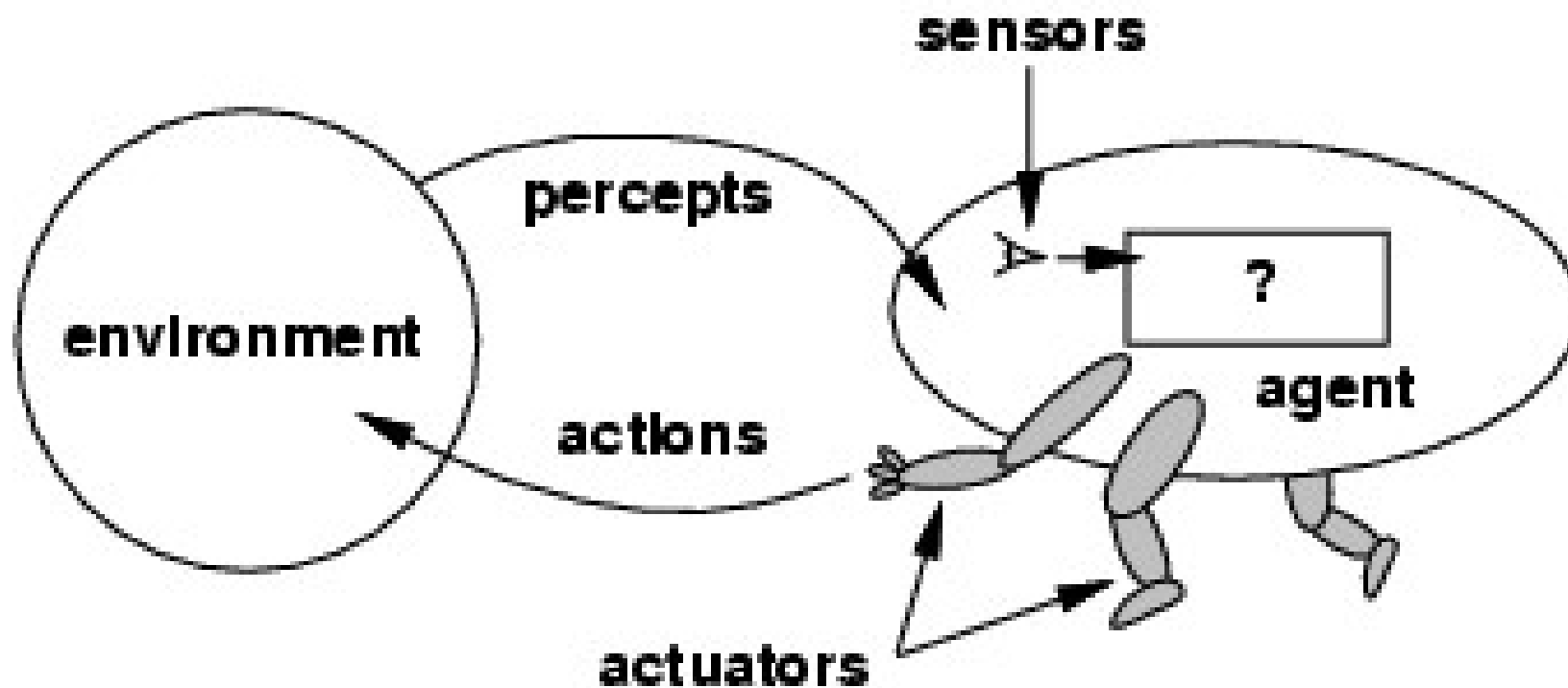


Съдържание

- Агенти и околна среда
- Рационалност
- Мярка за ефективност (Performance measure),
Околна среда (Environment), Актуатори, Сензори
- Типове околна среда
- Типове агенти

Агент

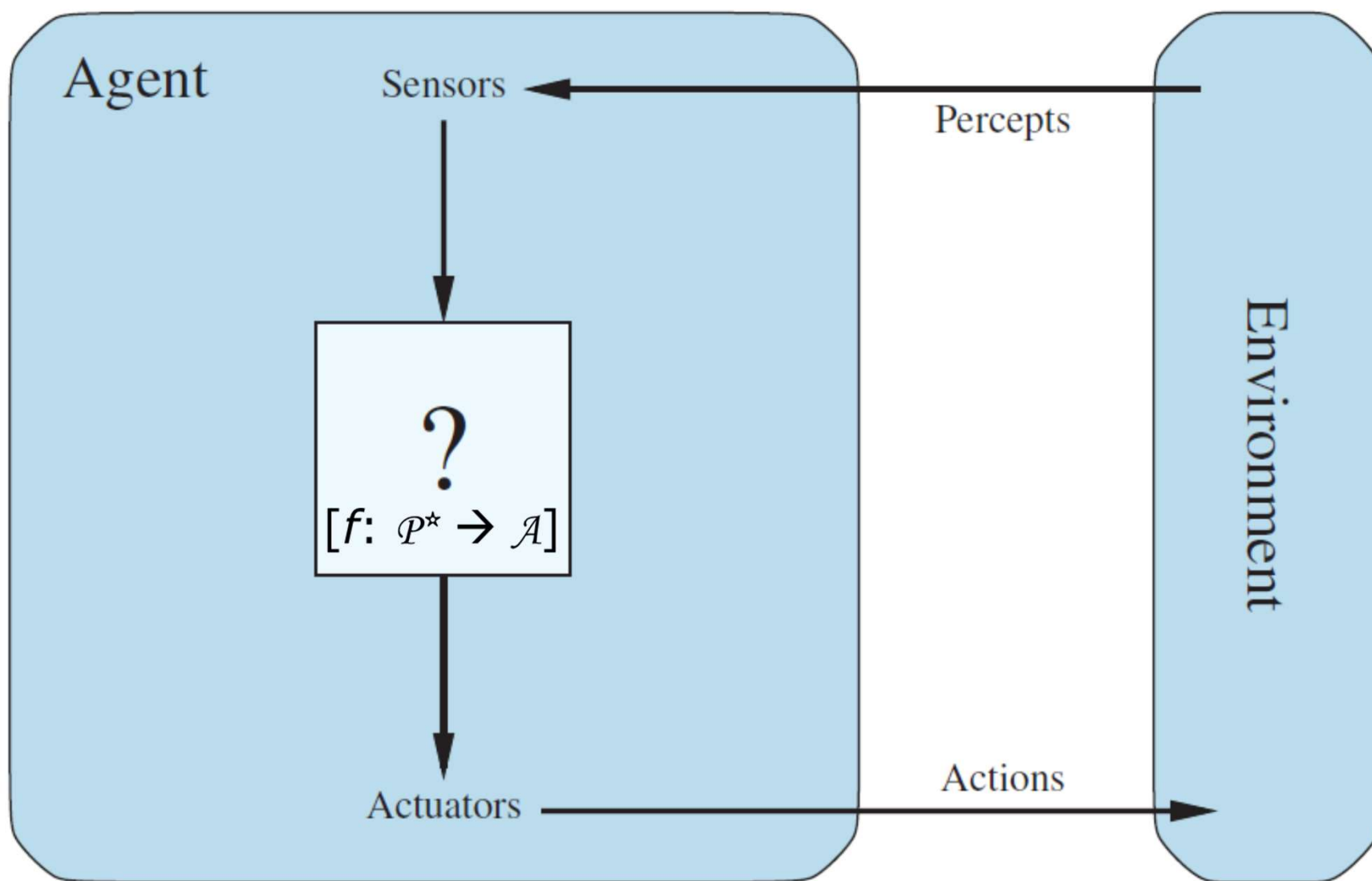
- Дефиниция за агент ?
- Човек : възприятия, актуатори
- Робот : възприятия, актуатори



Агент

- Какво може да прави всеки агент?
 - Възприема околния свят и изпълнява действия.
- Как става това?
 - Възприема околния свят чрез сензори,
 - Интерпретира информацията въз основа на знание и информация за контекста
 - използва мярка за успешност, модели и стратегии за да взема решения,
 - решенията се прилагат чрез действия.

Агент



Агент

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

Програмата на агента се изпълнява с помощта на хардуер
за да се реализира функцията f

agent = architecture + program

Кое то всъщност се свежда до

агент = хардуер + софтуер

іRobot: автоматична прахосмукачка



iRobot: автоматична прахосмукачка

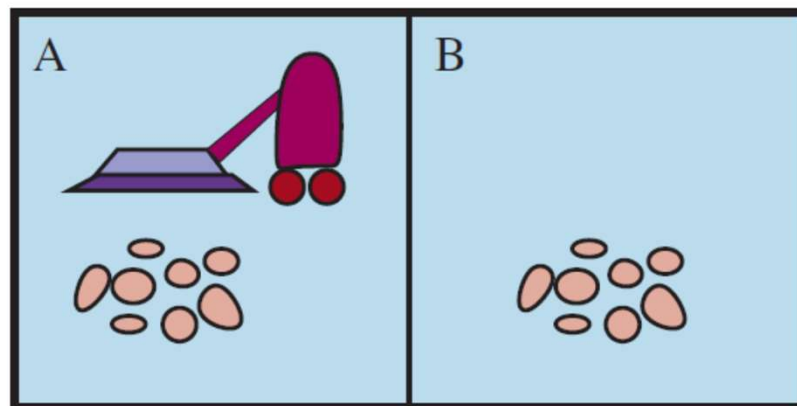


Figure 2.2 A vacuum-cleaner world with just two locations. Each location can be clean or dirty, and the agent can move left or right and can clean the square that it occupies. Different versions of the vacuum world allow for different rules about what the agent can perceive, whether its actions always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

- Възприятия: местоположение и съдържание, е.г., [А, Мръсно]
- Действия: *Ляво, Дясно, Засмукване, ПоОр*

iRobot: автоматична прахосмукачка



- **Рационално** поведение: това което максимизира постигането на заложената цел, при наличната към момента информация.
- Правилно е това което води към успех!!!
- Мярка за успешност: Обективен критерии за успешност на поведението на агента

За прахосмукачката:

- ☐ количеството изчистен прах,
- ☐ времето което е отнело,
- ☐ изразходваната електроенергия,
- ☐ количеството шум, и др.

Рационален агент

- **Рационален агент:**

За всяка възможна последователност от възприятия, рационалният агент ще избере действие което се очаква да максимизира избраната мярка за успешност, като се вземе в предвид наличната до момента информация и всяко друго налично знание по проблема, което агента има на разположение.

- **Рационален агент не е агент, който знае всичко, т.е. с безкрайно знание!**

Рационален агент

Рационален агент:

- Агентът може да предприема действия за да модифицира бъдещи възприятия за да получи полезна информация (сбор на информация, изследване)
- Агентът е автономен ...
- ...ако поведението му се определя от собствения му опит, способността му да се учи и адаптира.

Рационален агент:

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.

PEAS (Performance measure, Environment, Actuators, Sensors)

=

Мярка за ефективност, Околен свят, Изпълнителни механизми, Сензори

Пример:

Автоматичен шофьор на такси

- Performance measure? (Критерии, измерители)
- Environment? (околен свят)
- Actuators? (изпълнителни механизми)
- Sensors? (сензори)

■ Автоматичен шофьор на такси:

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

Figure 2.4 PEAS description of the task environment for an automated taxi driver.

Пример: Медицинска диагностична система

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers

Пример: Робот за сортиране/пренасяне на части

- **Мярка за успешност:**

Процентът от частите поставени в правилните кошове (сортирани правилно)

- **Околен свят:**

Конвейерната лента с части, контейнери за части

- **Актуатори:**

Ръка с много степени на свобода (стави) и грипер (механизъм за захващане)

- **Сензори**

Камера, сензори за ъглово преместване и скорост на ставите и позицията на ставите и грипера

Пример: Интерактивен учител по английски език

- **Мярка за успешност:**
Резултатите на студентите на теста
(Увеличете максимално!)
- **Околен свят:**
Група студенти
- **Актуатори:**
Дисплей (упражнения, предложения, корекции)
- **Сензори:**
Клавиатура, мишка

Типове околна среда

- **Напълно наблюдаема** (vs. частично наблюдаема):
Сензорите на агента имат достъп то цялостното състояние на средата във всеки момент от времето.
- **Детерминистична** (vs. stochastic):
Състоянието на средата е напълно определено в рамките на всяко състояние действията на агента. Ако средата е детерминистична с изключение на действията на други агенти то средата е стратегическо
- **Епизодично** (vs. sequential):
Опитът на агента е епизодичен – във всеки епизод агента възприема и след това изпълнява едно единствено действие. Изборът на действие зависи само от конкретния епизод.

Типове околна среда

- **Статична** (vs. динамична): Средата не се променя докато агентът разсъждава какво да предприеме. (Средата е **полу-динамична** ако средата не се променя с времето но се променя поведението и производителността на агента)
- **Дискретна** (vs. непрекъснатата): Ограничен брой от разграничени, ясно дефинирани усещания и действия.
- **Един агент** (vs. мултиагент): Един агент опериращ сам в средата.

Най-труден случай: частично наблюдаема, многоагентна, стохастична, последователна, динамична, непрекъснатата и непозната среда.

Типове околна среда

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
-
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent
-

Типове околна среда

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

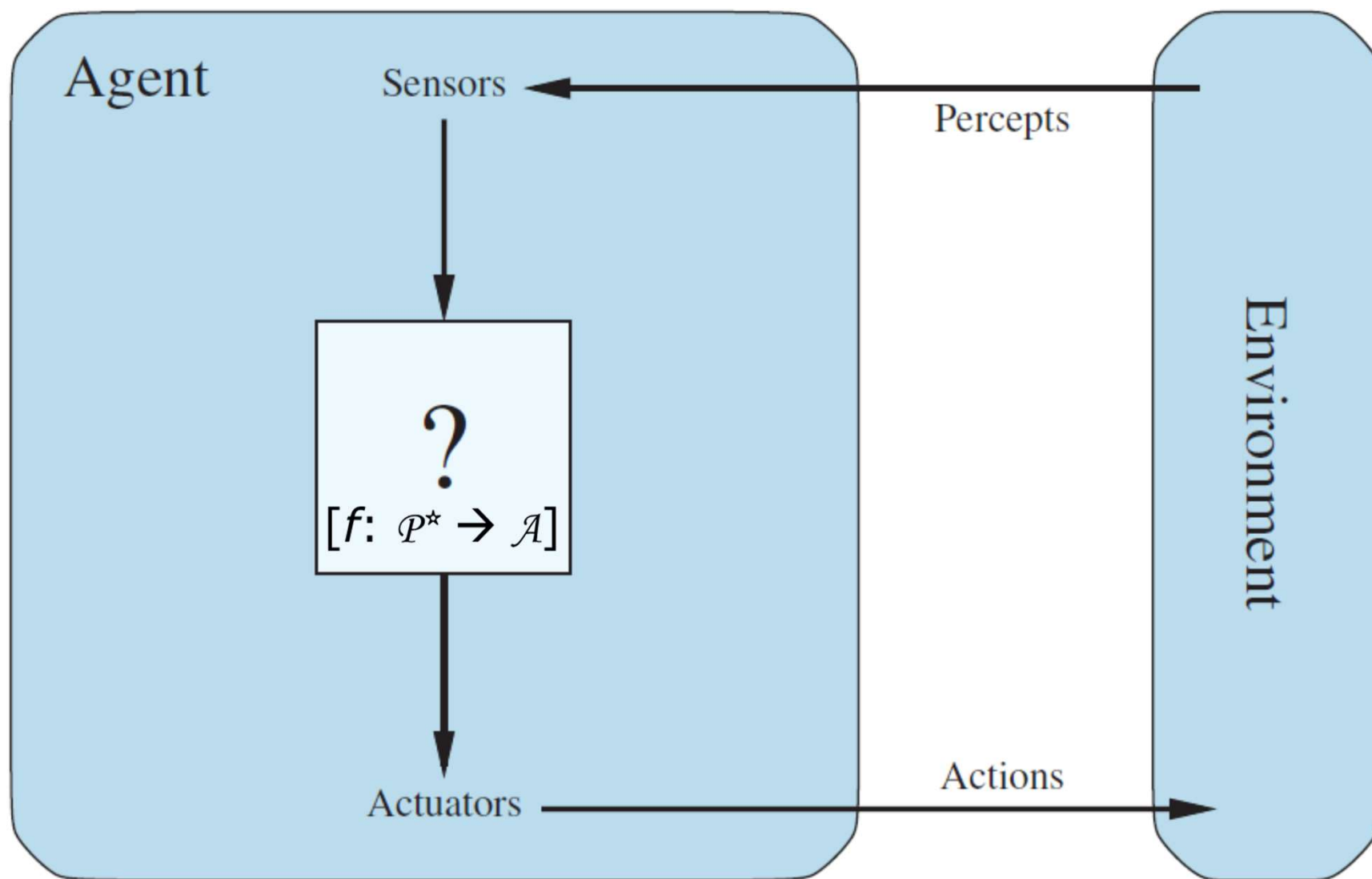


Агенти - програми

Функции на агента

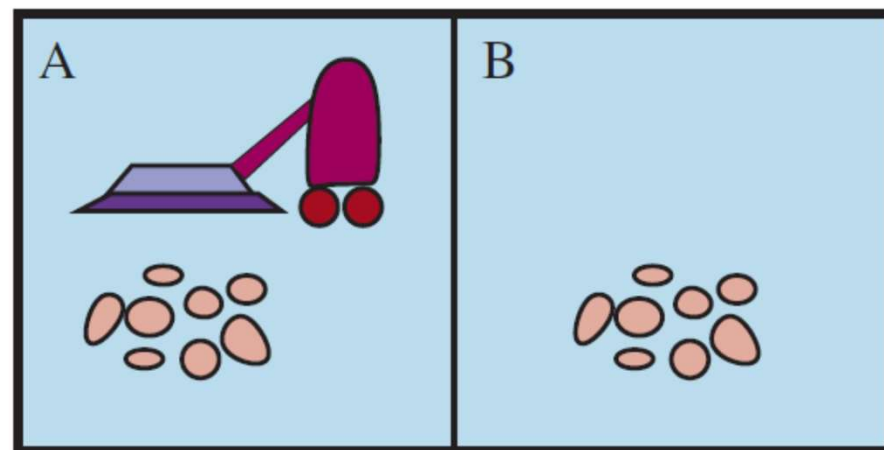
- **Агентът** е напълно определен (специфициран) от функцията си да „преобразува“ последователност от възприятия и наблюдения в действия.
- **Рационален агент**: автономна система която е има целенасочено поведение при реализирането на зададена цел.
- **Цел**: Открий начин да реализираш рационален агент който чевръсто изпълнява доставки

Структура на програмата Агент



Агентите взаимодействат с околната среда чрез сензори и изпълнителни механизми

Агент чрез таблица на съответствията Table-lookup agent (Таблични агенти)



Агент чрез таблица на съответствията

Table-lookup agent



Percept sequence	Action
$[A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Dirty}]$	\textit{Suck}
$[B, \textit{Clean}]$	\textit{Left}
$[B, \textit{Dirty}]$	\textit{Suck}
$[A, \textit{Clean}], [A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Clean}], [A, \textit{Dirty}]$	\textit{Suck}
\vdots	\vdots
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	\textit{Suck}
\vdots	\vdots

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure ???. The agent cleans the current square if it is dirty, otherwise it moves to the other square. Note that the table is of unbounded size unless there is a restriction on the length of possible percept sequences.

Агент чрез таблица на съответствията

Table-lookup agent



function TABLE-DRIVEN-AGENT(*percept*) **returns** an action

persistent: *percepts*, a sequence, initially empty

table, a table of actions, indexed by percept sequences, initially fully specified

append *percept* to the end of *percepts*

action \leftarrow LOOKUP(*percepts*, *table*)

return *action*

■ Недостатъци:

- Огромни таблици
- Изисква се дълго време за да се създаде таблицата
- Не е автономен
- Даже когато е способен на самообучение се изисква дълго време за да се научат отделните редове от таблицата

Основни типове агенти

Разглеждаме 5 основни типа агенти:

- **Обикновен рефлексен агент** (Simple reflex agents) -- Рефлексните агенти реагират директно на възприятията.
- **Агенти използващи модели** (Model-based reflex agents) -- Агентите използващи модели поддържат вътрешни състояния за да проследяват аспекти на околния свят които не са очевидни за конкретно моментно възприятие.
- **Агенти със зададена цел** (Goal-based agents). Стремят да постигнат зададената конкретна цел.
- **Агенти оптимизиращи очакваната полезност** чрез целеви функции (Utility-based agents) -- използват целеви функции с които могат да оценяват степента на полезност и да се стремят към повишаването на степента на удовлетвореност.
- **Самообучаващи се агенти**

Рефлексни агенти - Reflex agents

function REFLEX-VACUUM-AGENT(*[location, status]*) returns an action

if *status* = *Dirty* then return *Suck*

else if *location* = *A* then return *Right*

else if *location* = *B* then return *Left*



Рефлексни агенти - Reflex agents

Percept sequence	Action
$[A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Dirty}]$	<i>Suck</i>
$[B, \textit{Clean}]$	<i>Left</i>
$[B, \textit{Dirty}]$	<i>Suck</i>
$[A, \textit{Clean}], [A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
\vdots	\vdots
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Clean}]$	<i>Right</i>
$[A, \textit{Clean}], [A, \textit{Clean}], [A, \textit{Dirty}]$	<i>Suck</i>
\vdots	\vdots

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

Рефлексни агенти - Reflex agents

Базиран на система от правила, дефинирани чрез

- **условие - правило за действие** (condition–action rules),
- **ако ... тогава ... правила** (if–then rules),
- **ситуация - правило за действие** (situation–action rules),
или
- **продукционна логика на действие** (productions).

Обикновен рефлексен агент – Simple reflex agents

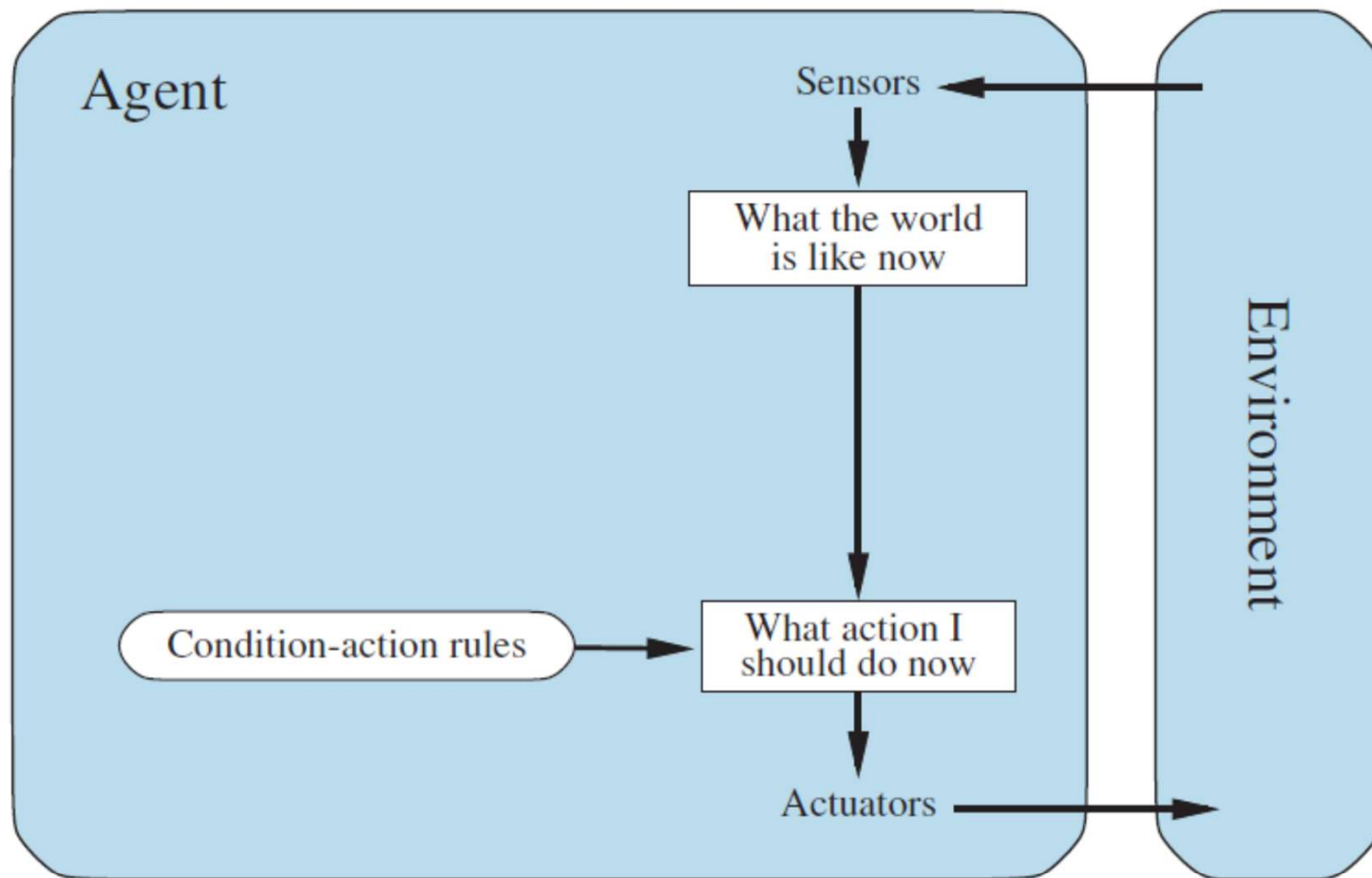


Figure 2.9 Schematic diagram of a simple reflex agent. We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

Обикновен рефлексен агент – Simple reflex agents

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
persistent: *rules*, a set of condition–action rules

state \leftarrow INTERPRET-INPUT(*percept*)

rule \leftarrow RULE-MATCH(*state*, *rules*)

action \leftarrow *rule*.ACTION

return *action*

- INTERPRET-INPUT функцията – създава абстрактно описание на текущото състояние въз основа на възприетата информация
- RULE-MATCH функцията – връща първото правило от набора което съвпада с зададеното състояние.

Обикновен рефлексен агент – Simple reflex agents

Предимства:

- Лесни за реализация
- Изискват малко ресурси

Недостатъци:

- Риск от „зацикляне“, попадане в безкраен цикъл
- Ограничена възможност за работа при промяна на условията/средата
- Ограничена интелигентност

Агенти използващи модели

Model-based reflex agents

Основна идея: *да се следи останалата част от света, която не може да се наблюдава в момента, т.е.*

Агентът трябва да поддържа представа за „модел на света“, т.е. вътрешно състояние което зависи от историята на възприятията, с цел да се моделират и използват някои аспекти, които не могат да бъдат наблюдаване в момента.

За актуализация на вътрешното състояние се изискват два вида „знание“:

- информация „Как се променя света?“ и
- „Как действията на агента оказват влияние на света?“.

В частично наблюдаем свят обикновено търсим някъкво „приблизително“ текущо състояние.

Необходимо е да се справяме с несигурност и непълнота дължаща се на частичната наблюдаемост на света.

Агенти използващи модели

Model-based reflex agents

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action

persistent: *state*, the agent's current conception of the world state

model, a description of how the next state depends on current state and action

rules, a set of condition–action rules

action, the most recent action, initially none

state \leftarrow UPDATE-STATE(*state*, *action*, *percept*, *model*)

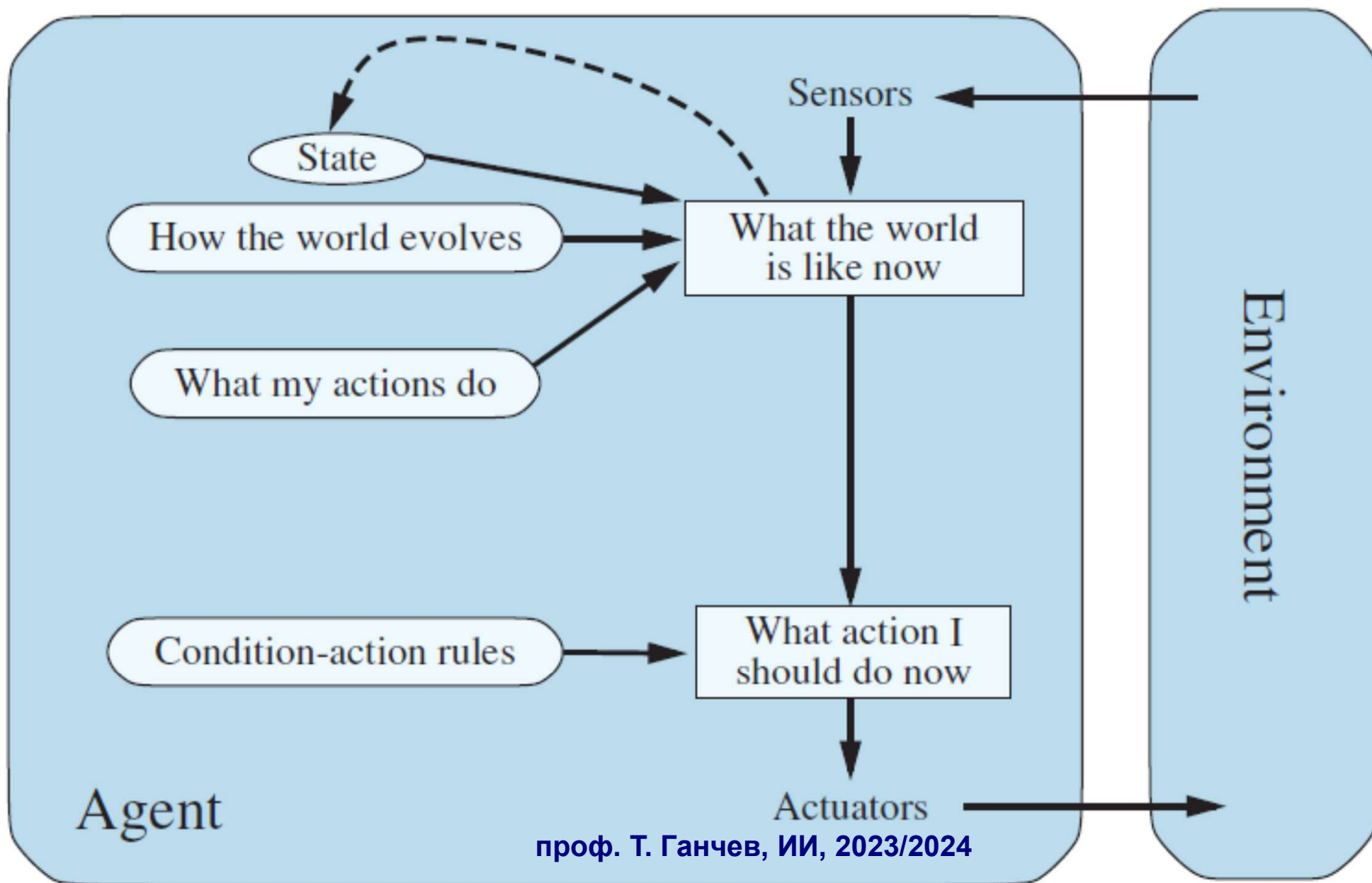
rule \leftarrow RULE-MATCH(*state*, *rules*)

action \leftarrow *rule*.ACTION

return *action*

Рефлексни агенти използващи модели

Model-based reflex agents

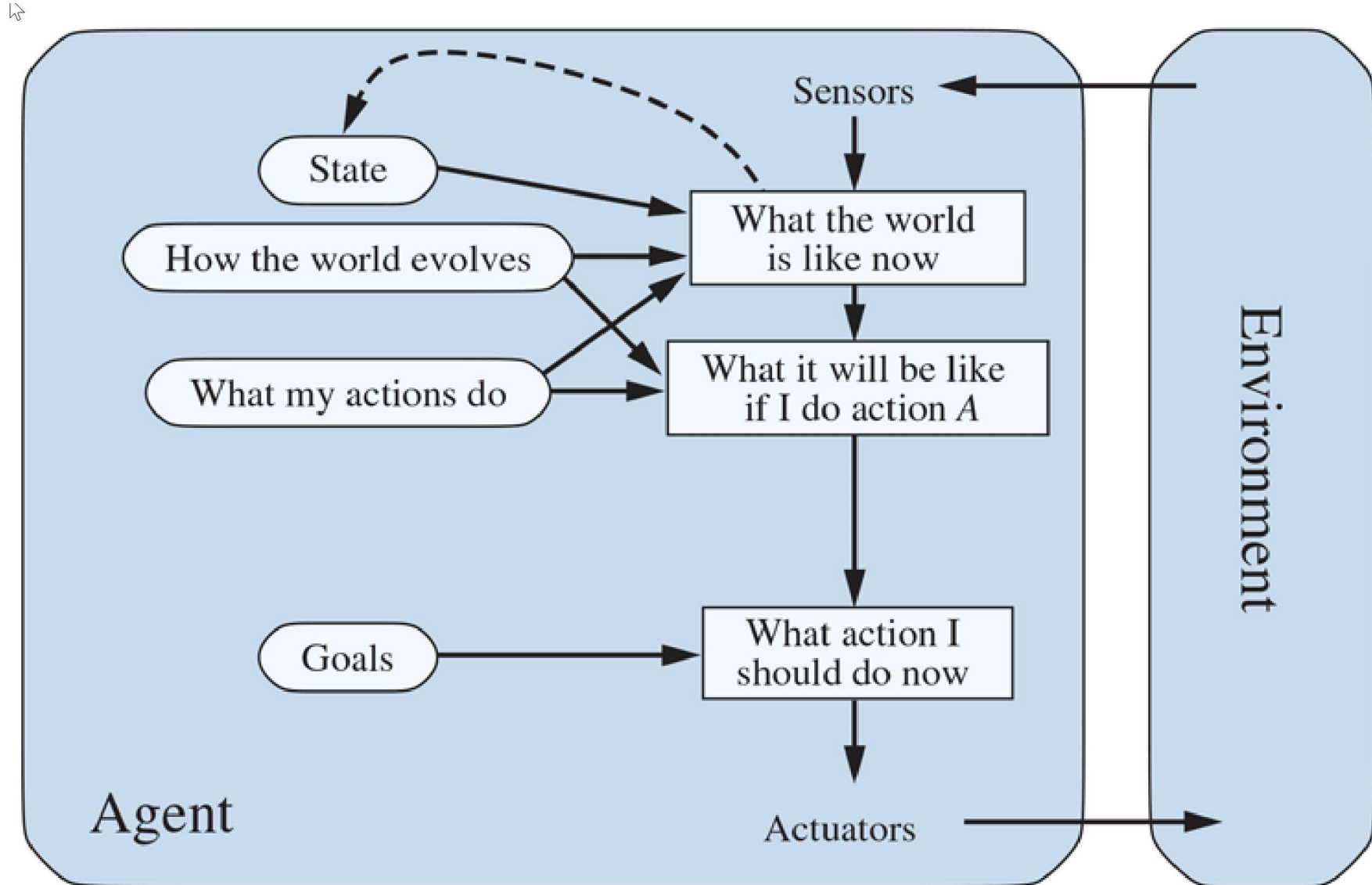


Model-based reflex agents

Предимства

Недостатъци

Model-based, Goal-based agents

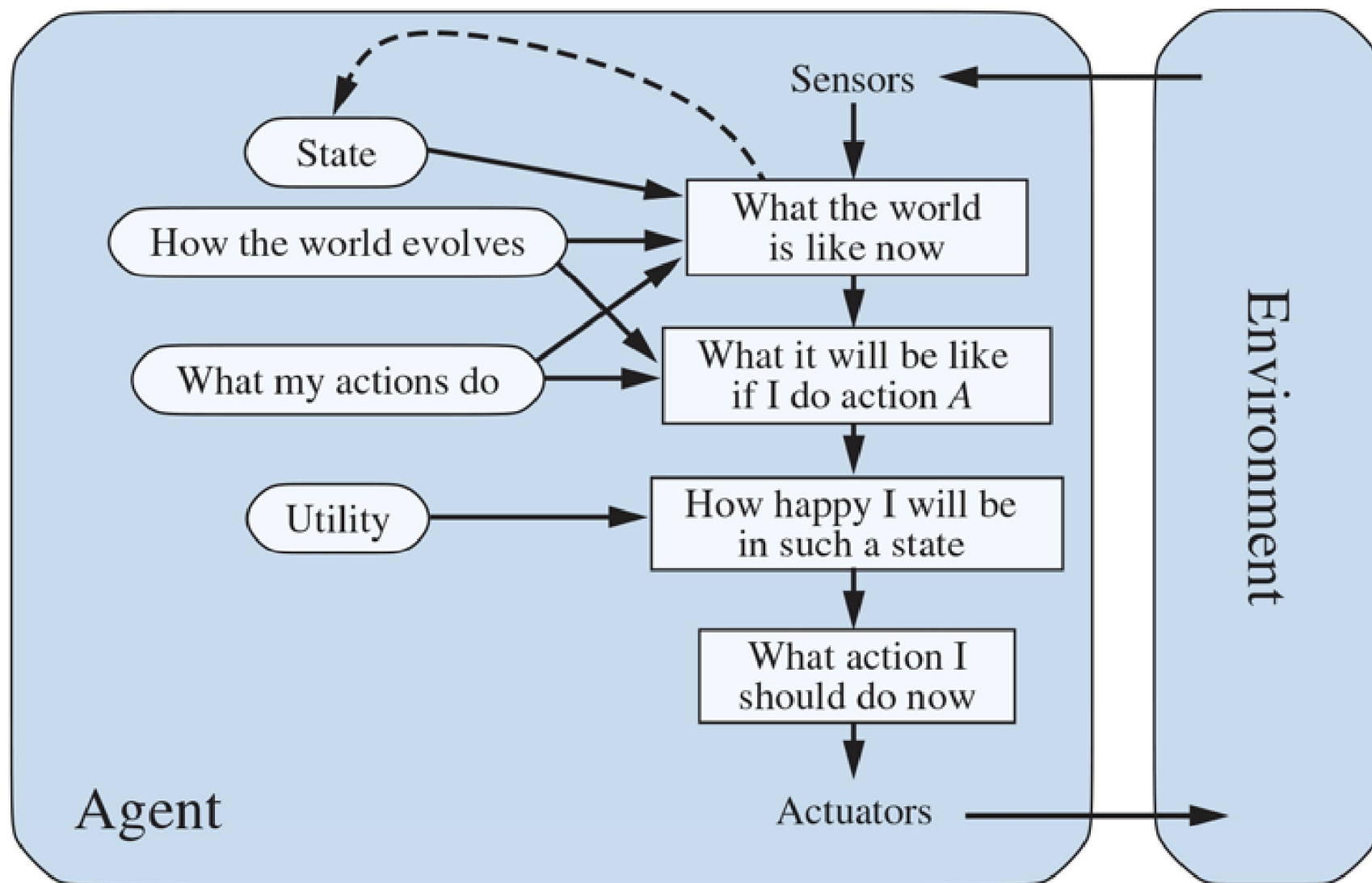


A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Model-based, Goal-based agents

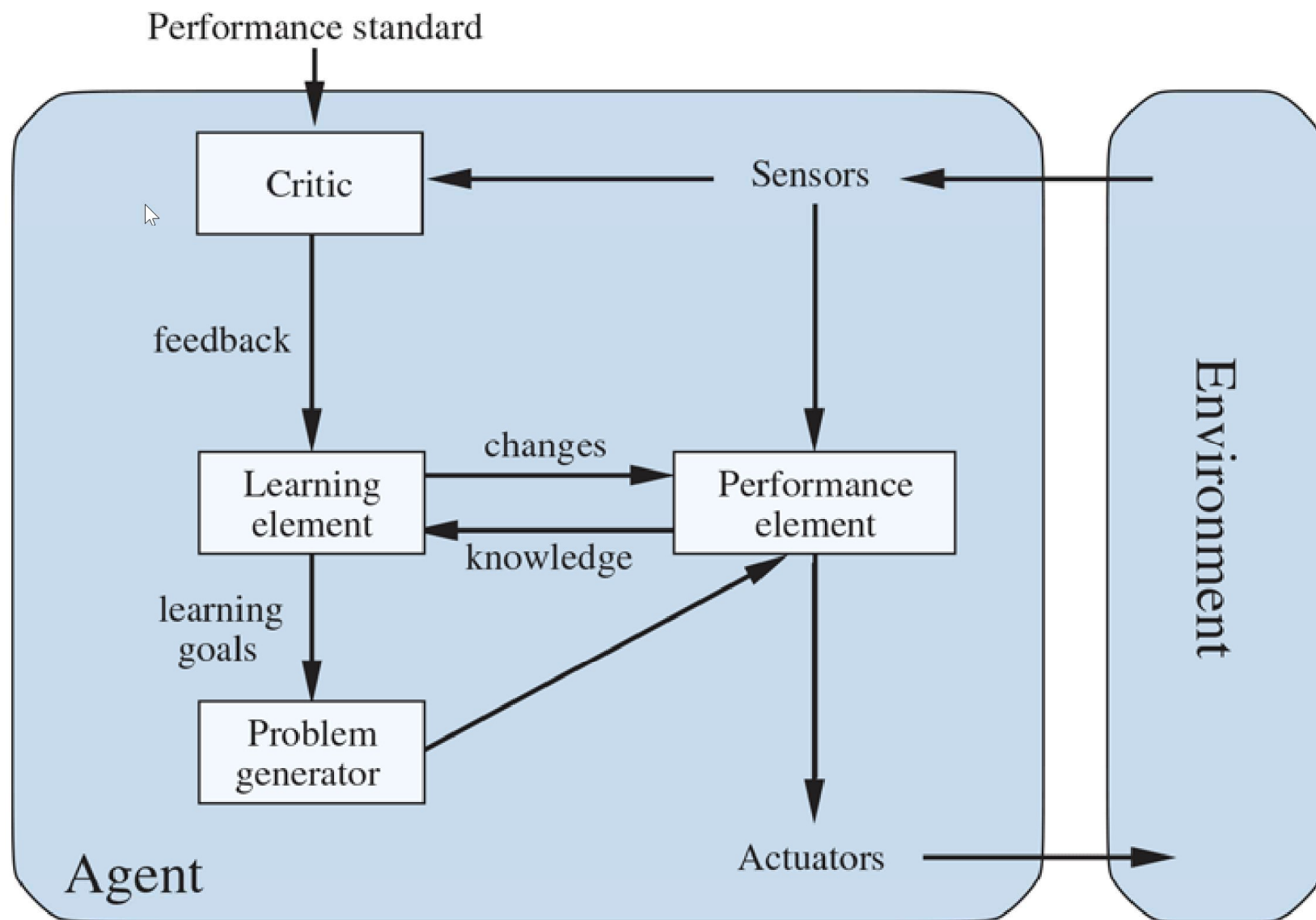
- Основни концепции при агентите реализиращи зададена цел са:
 - Ясна дефиниция на крайната цел – например дестинацията до която трябва да се достигне.
 - Търсене на последователността от състояния през които трябва да се премине
 - Планиране (обмисляне на възможностите)

Utility-based agents



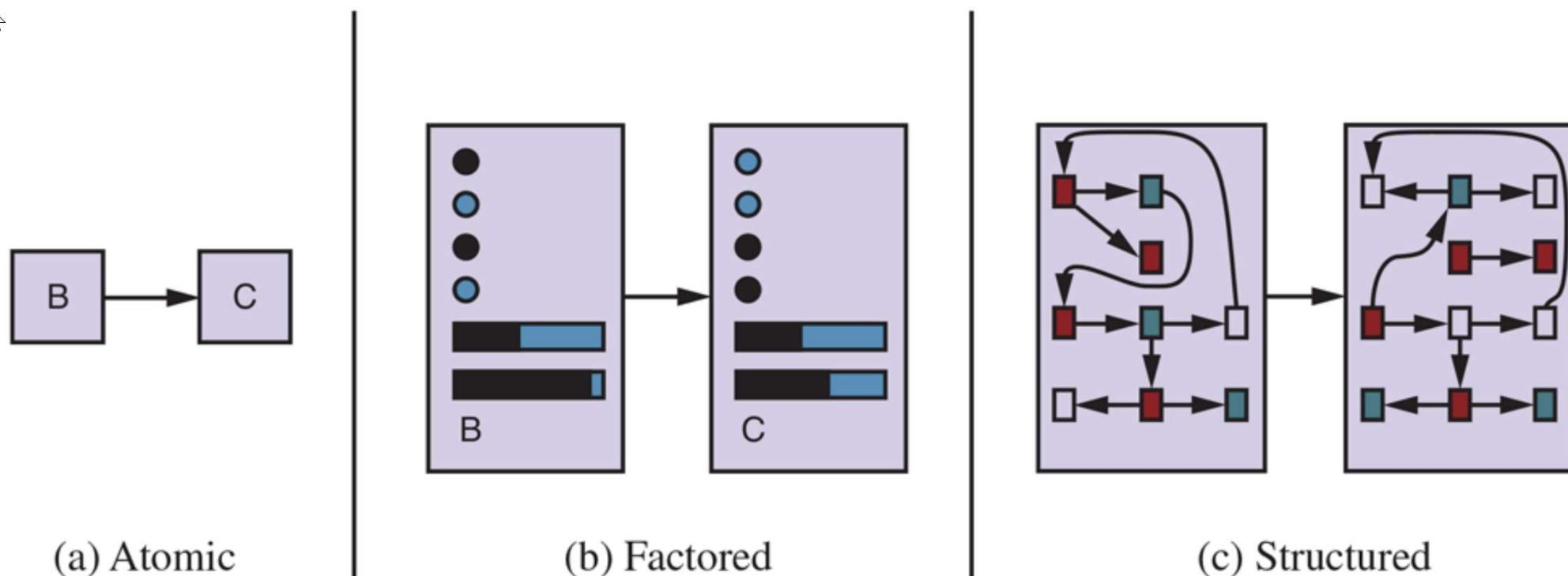
A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states,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

Learning agents



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.

Реализация на вътрешните състояния



Three ways to represent states and the transitions between them. (a) Atomic representation: a state (such as B or C) is a black box with no internal structure; (b) Factored representation: a state consists of a vector of attribute values; values can be Boolean, real-valued, or one of a fixed set of symbols. (c) Structured representation: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.