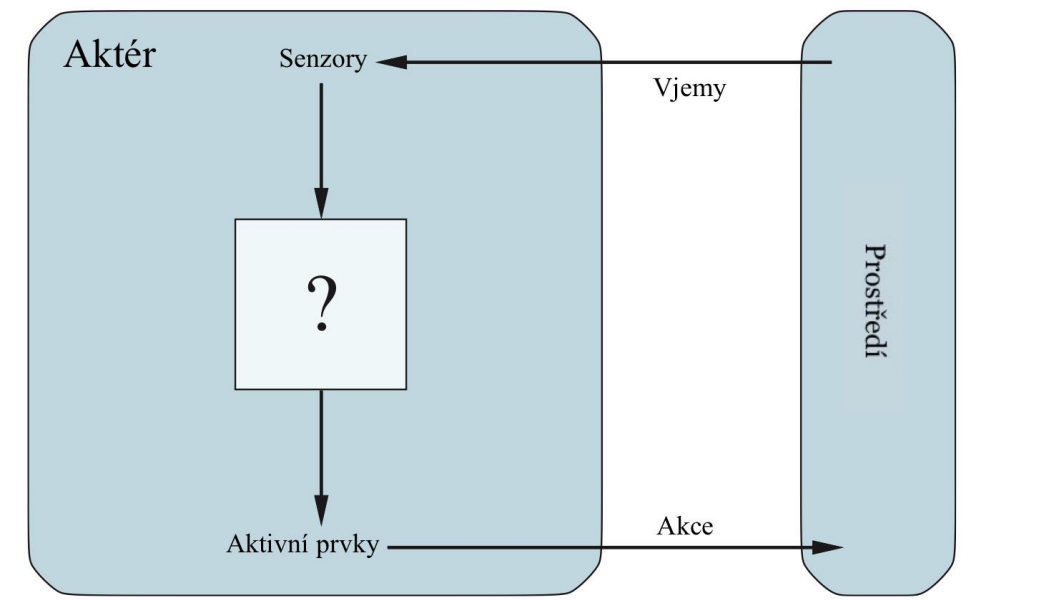
IV130 Přínosy a rizika inteligentních systémů

Inteligentní aktéři

15. března 2024

Aktéři (agenti)

- **Aktér** je cokoli, co vnímá **prostředí**, v němž působí, pomocí **vjemů** přes **senzory** a toto prostředí mění pomocí **aktivních prvků**, které vyvolávají **akce** v tomto prostředí.



Aktéři interagují s prostřednictvím prostřednictvím vjemů a akcí (činností).

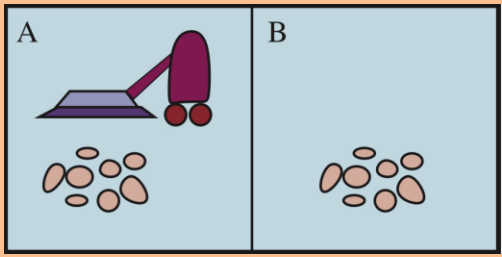
Obrázky a schémata z Russell-Norwig: AI A Modern Approach, 4th ed., 2021

Vjemy a funkce aktéra

- Vjemy jsou jediným zdrojem informací o prostředí
- Akce mohou záviset pouze na zabudované znalosti bez vazby na prostředí nebo na vjemech (případně na posloupnosti vjemů)
- Chování aktéra je popsáno **funkcí aktéra**, která jakoukoli posloupnost vjemů transformuje do akce
- Tato transformace je *implementována* pomocí **programu aktéra**

(Funkce aktéra je vnější charakterizací aktéra, program aktéra je její vnitřní implementace.)

Příklad: Automatický vysavač na dvou lokacích podlahy



- Vjemy dávají polohu, A nebo B
- a informaci, zda je v lokaci smetí
- Akce spočívají ve vysání smetí
- a dvojici pohybů: vlevo a vpravo.
- Příklady vjemů: (A,čisto), (B,čisto), (A, smetí), atd.
- Tři možné akce: Doprava, Doleva, Sání

- Funkce aktéra může být reprezentována jednoduchou tabulkou (závisející na poloze a smetí, nebo jen poloze)

Racionální aktéři

- *Racionální aktér* dělá „správné“ věci
- „Správnost“ je v AI vymezena tzv. *konsekvencionalismem*: aktérovo chování je posuzováno na základě důsledků (konsekvencí) jeho chování
- Vyjádření charakteru důsledků je zahrnuto v nějaké chrakterizaci působení vzhledem k *preferencím*, kvantifikovat to lze *nákladovou funkcí,* jejíž minimalizace/maximalizace popisuje chování
- Preference je obecně vhodné volit tak, aby odpovídaly přímo žádoucímu výsledku (např. čistá podlaha namísto maximalizace množství vysátého smetí)

Racionalita

Racionální chování aktéra závisí na

- preferencích / nákladové funkci, což určuje kritéria úspěchu,
- jeho výchozí znalosti prostředí,
- akcích, které může aktér provádět,
- a posloupnosti vjemů až do daného okamžiku.

Racionální aktér volí pro každou možnou posloupnost vjemů akci, u níž se očekává maximalizace nákladové funkce (preferencí), a činí tak na základě dosavadní posloupnosti vjemů a možných vestavěných znalostech zahrnutých v programu aktéra.

Vševědoucnost

- Racionalita se liší od *vševědoucnosti* ve smyslu znalosti *skutečného* výsledku akcí aktéra (nezávisle na výsledku popsatelném vjemy).
- Racionalita se liší od dokonalosti: racionalita maximalizuje *očekávaný* výsledek, zatímco dokonalost *skutečný* výsledek.
- Racionalita závisí na sběru informací z prostředí, což umožňuje chování, které ovlivňuje budoucí vjemy (směrem k dosažení záměrů)

Učení (se)

- Racionální aktér nejen sbírá informace o prostředí, může se i *učit* z toho, jaké vjemy mu přicházejí
- Zkušenosti dovolují modifikovat vestavěnou počáteční konfiguraci (která může některé znalosti o prostředí obsahovat), a tím vylepšovat své chování
- Extrémní případ: prostředí je kompletně známo na začátku a výsledky akcí jsou plně predikovatelné – vjemy nebo učení jsou pak zbytečné (existuje to u některých jednoduchých organismů)

Autonomie aktéra

- Apriorně zabudovaná znalost v aktérovi je nepřímo úměrná jeho *autonomii*
- Autonomní aktér dokáže napravovat chybnou nebo neúplnou počáteční znalost o prostředí
- Autonomie odpovídá např. vytváření reflexů na základě zkušenosti namísto reflexů vrozených a neměnných
- Autonomní aktér jako aktér *nezávislý* na výchozích znalostech
- Učení umožňuje autonomnímu aktérovi uspět ze stejného výchozího stavu (se stejnou výchozí konstrukcí) v různě se měnícím prostředí

Prostředí

- Prostředí odpovídají druhu řešeného úkolu (prostředí úkolu)
- Racionální aktéři jsou „řešením“ konkrétní podoby prostředí, v němž se má úkol plnit
- Prostředí bude charakterizováno pomocí čtveřice parametrů NPAS

➢ N – nákladová funkce (ukazatel plnění záměru)

➢ P – prostředí (parametry ovlivňující chování i výsledek)

➢ A – vymezení aktivních prvků aktéra ovlivňujících prostředí

➢ S – senzory vymezující možné vjemy, kterými aktér získává informace o prostředí

Příklad chrakterizace NPAS prostředí pro automatické taxi

Typ aktéra	Nákladová funkce	Prostředí	Aktivní prvky	Senzory
Řidič taxi	Bezpečná, podle pravidel, rychlá, pohodlná jízda, maximalizace zisku, minimalizace dopadu na ostatní na silnici	Cesty, další provoz, policie, chodci, zákazníci, počasí	Řízení, plyn, brzdy, signalizace směru, klakson, displej, řeč	Kamery, radar, tachoměr, GPS, senzory v motoru, mikrofony, dotyková obrazovka

NPAS popis prostředí řešení úkolů pro automatizovaného řidiče taxi.

Další typy NPAS prostředí

Typ aktéra	Nákladová funkce	Prostředí	Aktivní prvky	Senzory
Systémy lékařské diagnostiky	Zdravý pacient, snížené nákklady	Pacient, nemocnice, zdravotníci	Zobrazení otázek, testy, diagnózy, léčení	Vstup symptomů a nálezů přes dotykovou obrazovku nebo hlas
Systém analýzy satelitních snímků	Správná kategorizace objektů, terénu	Satelit na orbitu, spojení se Zemí, počasí	Zobrazení scény s kategorizací	Hi-res digitální kamera
Robot podávající součástky	Procento součástek ve správných koších	Dopravní pás se součástkami; koše	Rameno s klouby a úchopy	Kamera, senzory hmatu a polohy ramen
Řízení rafinérie	Čistota, výtěžnost, bezpečnost	Rafinérie, suroviny, operátoři	Ventily, pumpy, ohřívání, vířidla, displeje	Teplota, tlak, tok, chemické senzory
Interaktivní učitel jazyka	Studentův výsledek testu	Skupina studentů, testovací agentura	Zobrazení cvičení, zpětná vazba, řeč	Vstup přes klávesnici, hlas

Příklady typů aktérů a jejich NPAS popisy.

Vlastnosti prostředí pro různé úkoly

- *Plně pozorovatelné* vs. *částečně pozorovatelné* – v závislosti na tom, zda senzory dávají informace o úplném stavu prostředí (i zcela bez senzorů – jako *nepozorovatelný* stav prostředí)
- *Jeden* nebo *více* aktérů – hodnota nákladové funkce může záviset na chování jiného aktéra (aktérů), včetně chování kompetitivního
- *Deterministické* vs. *nedeterministické* – v závislosti na tom, zda akce plně určují stav prostředí po akci (v částečně pozorovatelných prostředích je determinismus nejvýše zdánlivý); *stochastické* pro prostředí popsané pravděpodobnostmi

Vlastnosti prostředí pro různé úkoly

- *Epizodické* vs. *sekvenční* – atomické epizody se skládají z odělených dvojic vjemu a jedná akce po něm, v sekvenčních může jedno rozhodutí ovlivnit řadu nebo i všechny další
- *Statické* vs. *dynamické* – dynamické prostředí se během aktérova rozmýšlení může měnit (*polodynamické* odpovídá změně aktérovy nákladové funkce během rozmýšlení)
- *Diskrétní* vs. *spojité* – odpovídá diskrétnosti veličin času, vjemů i akcí vs. spojitosti některé z nich
- *Známé* vs. *neznámé* mohou být vlastnosti prostředí (zákony jeho chování)

Příklady prostředí úkolů

Inteligentní aktéři

Prostředí úkolu	Pozorovatelnost	Aktéři	Determinističnost	Epizodičnost	Statičnost	Diskrétnost
Křížovka	Plná	Jeden	Deterministický	Sekvenční	Statický	Diskrétní
Šachy s hodinami	Plná	Více	Deterministický	Sekvenční	Polo	Diskrétní
Poker	Částečná	Více	Stochastický	Sekvenční	Statický	Diskrétní
Backgammon	Plná	Více	Stochastický	Sekvenční	Statický	Discrétní
Automatické taxi	Částečná	Více	Stochastický	Sekvenční	Dynamický	Spojitý
Lékařské vyšetření	Částečná	Jeden	Stochastický	Sekvenční	Dynamický	Spojitý
Analýza obrazů	Plná	Jeden	Deterministický	Epizodický	Polo	Spojitý
Robot třídící součástky	Částečná	Jeden	Stochastický	Epizodický	Dynamický	Spojitý
Řízení rafinérie	Částečná	Jeden	Stochastický	Sekvenční	Dynamický	Spojitý
Výuka jazyka	Částečná	Více	Stochastický	Sekvenční	Dynamický	Diskrétní

Příklady prostředí úkolů a jejich charakteristiky.

Aktéři definovaní pomocí tabulky popisující funkci aktéra

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 ← LOOKUP(*percepts*, *table*)
  return *action*

Program TABLE-DRIVEN-AGENT pro aktéra řízeného tabulkou: pro každý nový vjem (percept) je program vyvolán a vrací akci z tabulky, kterou si celou uchovává ve své paměti.

Struktura aktérů

- Aktér = architektura aktéra + program
- Základní výzvou pro AI je nacházet způsob, jak co nejmenším programem realizovat racionální chování směřující k plnění záměrů
- Čtyři základní typy aktérů popisující většinu inteligentních systémů:

➢ Jednoduší reflexivní aktéři

➢ Reflexivní aktéři pracující s modelem

➢ Aktéři realizující plnění cílů

➢ Aktéři založení na hodnocení užitku

Jednoduchý reflexní aktér

- Akce jsou voleny na základě aktuálního vjemu (korektnost jejich chování závisí na plné pozorovatelnosti prostředí)

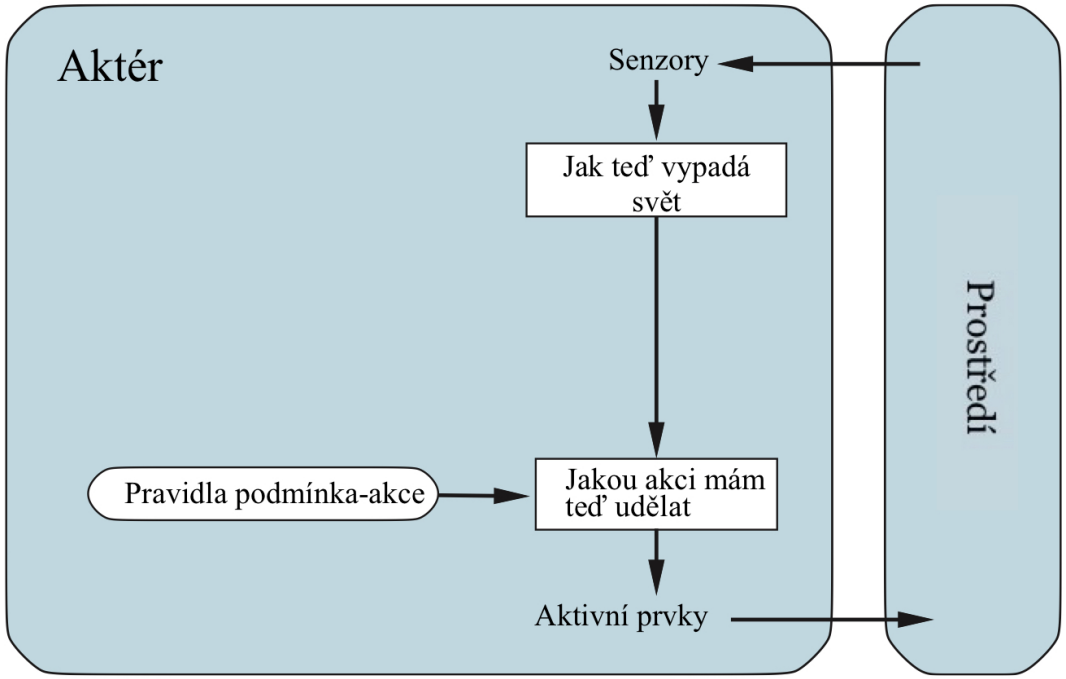


Schéma jednoduchého reflexního aktéra.
Obdélníky jsou vnitřní stavy aktérova rozhodování,
ovály představují zabudované informace používané procesem.

Jednoduchý reflexní aktér

- Diagram popisuje strukturu kódu v aktérovi

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

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Jednoduchý reflexní aktér. Pracuje na základě pravidla, jehož podmínka odpovídá aktuálnímu stavu, který je dán vjemem (*percept*).

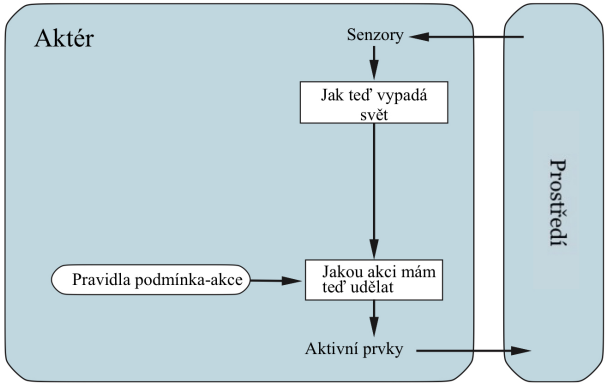
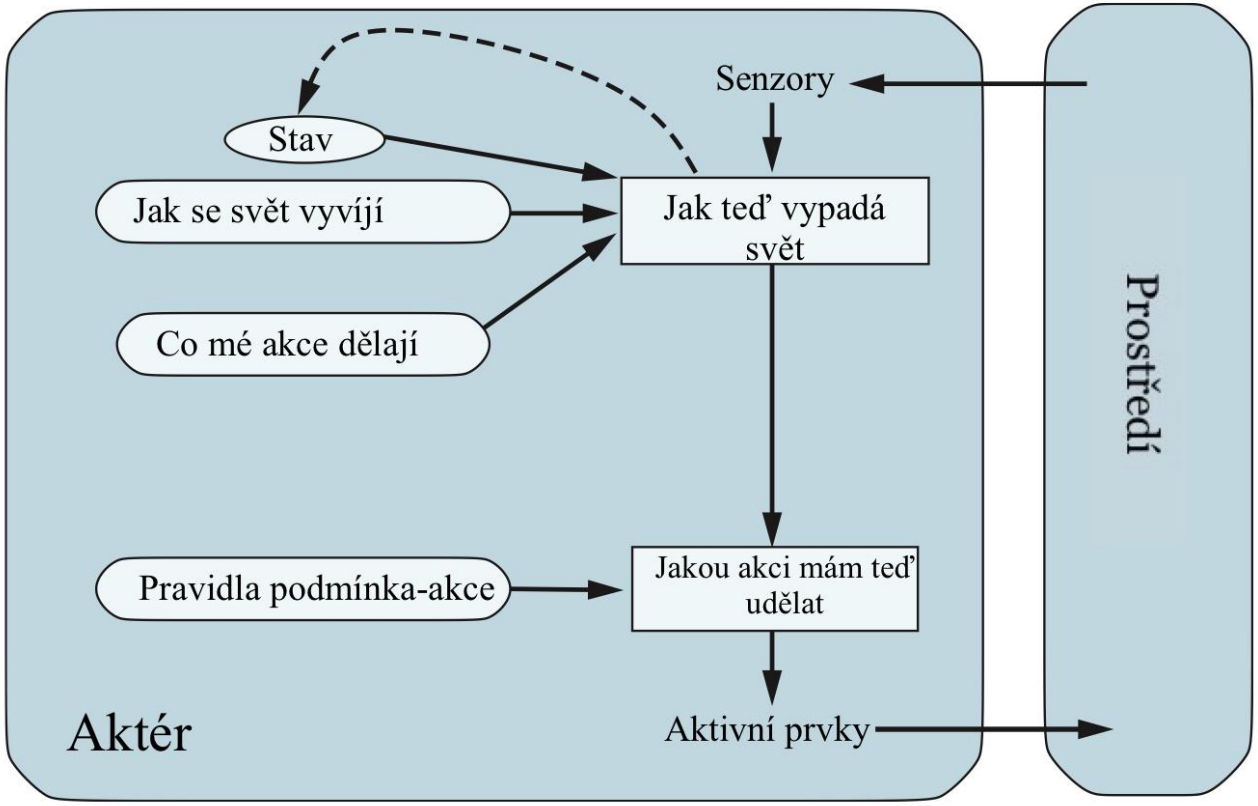


Schéma jednoduchého reflexního aktéra.
Obdélníky jsou vnitřní stavy aktérova rozhodování,
ovály představují zabudované informace používané procesem.

Reflexní aktér založený na modelu

- Částečná pozorovatelnost se dá řešit tím, že si držíme model části světa, který nevidíme
- Řeší to interní stav aktéra



Reflexní aktér založený na modelu.

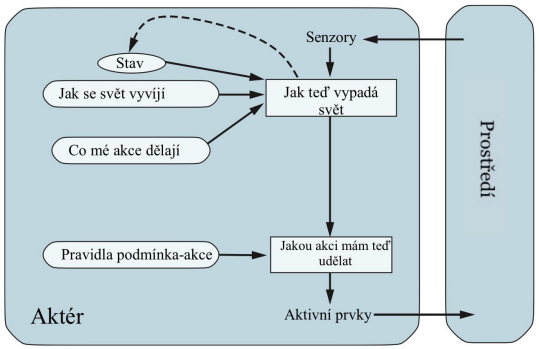
Reflexní aktér založený na modelu

- Diagram opět definuje strukturu kódu v aktérovi

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              transition_model, a description of how the next state depends on
                    the current state and action
              sensor_model, a description of how the current world state is reflected
                    in the agent's percepts
              rules, a set of condition–action rules
              action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, transition_model, sensor_model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

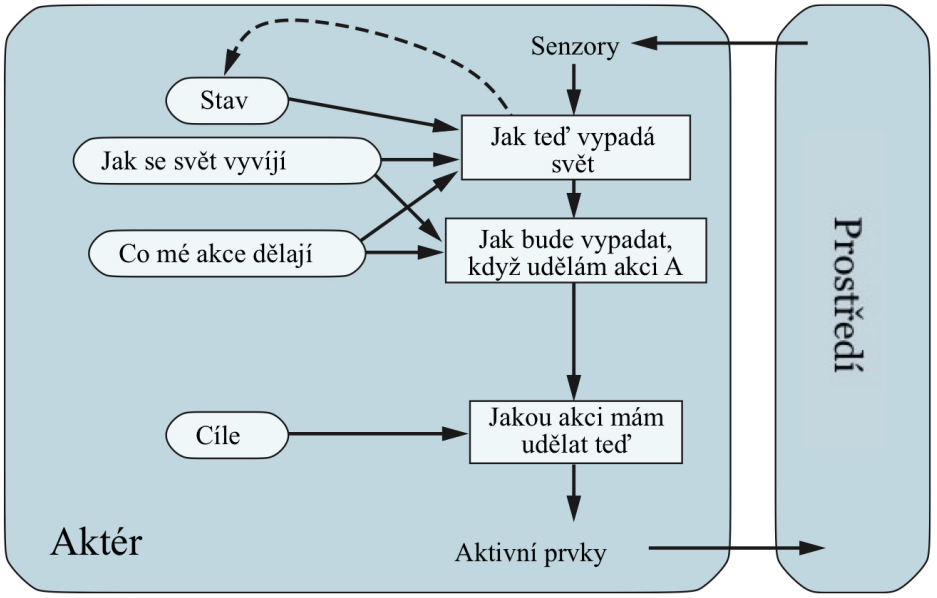
Reflexní aktér založený na modelu. Na základě vestavěného vnitřního modelu sleduje aktuální stav světa. Pak volí akci stejně jako to dělá reflexní aktér.



Reflexní aktér založený na modelu.

Aktér založený na modelu a plnění cíle

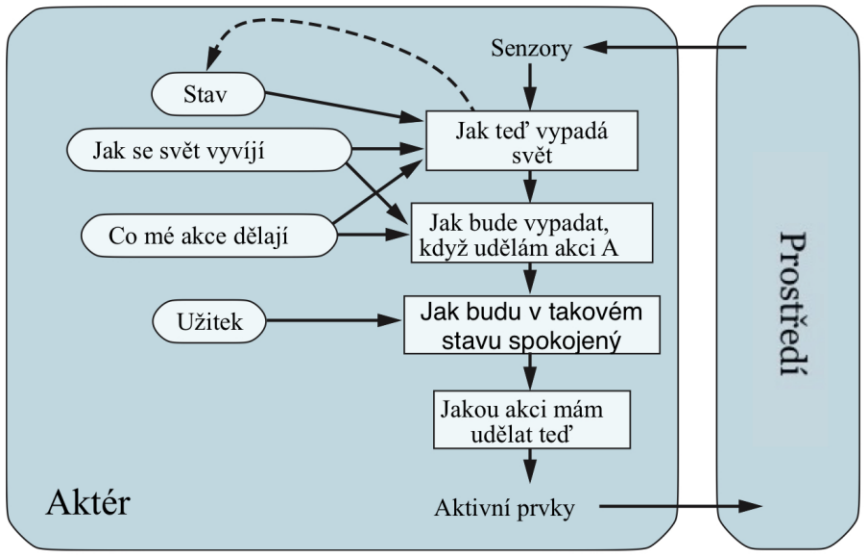
- Zahrnutí informaci o cíli, tj. popisu žádoucích situací, může výrazně pomoci s používáním modelu
- Zahrnutí cíle kombinuje určení výsledku akce s hodnocením žádoucnosti jejího výsledku



Aktér založený na modelu zaměřený na dosažení cíle. Sleduje stav světa a množinu cílů, kterých se snaží dosáhnout, vybírá si akci, která (nakonec) povede k dosažení jeho cílů.

Aktér s modelem a maximalizací užitku

- Užitková funkce dovoluje zabudovat do aktéra vnější hodnocení činnosti (za předpokladu, že jsou ve shodě)
- Opět se jedná o maximalizaci očekávaného užitku
- Používá se i verze, kdy model využíván není



Aktér založený na modelu a pracující na maximalizaci užitku. Obsahuje model světa spolu s užitkovou funkcí kvantifikující preference mezi jednotlivými stavy. Vybírá si akci vedoucí k nejlepšímu očekávanému užitku, a tento nejlepší užitek se počítá způměrováním všech možných výsledných stavů vážených jejich pravděpodobností.

Učící (se) aktér

- S myšlenkou učících se strojů přišel již Alan Turing v roce 1950
- Učící (se) aktér má čtyři složky:

➢ *Učící prvek* zodpovědný za vylepšování

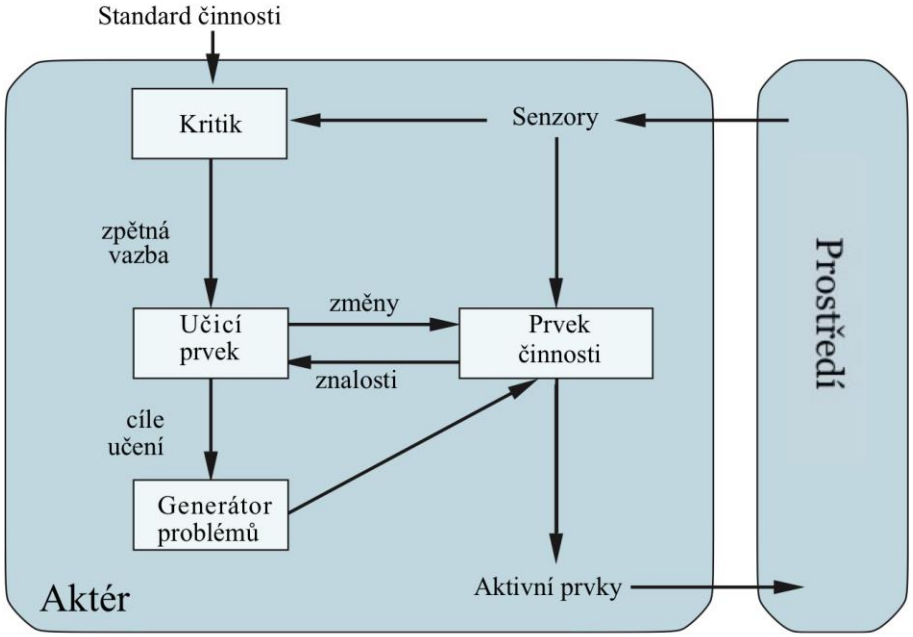
➢ *Prvek činnosti*, který vybírá externí akce

➢ *Kritik* poskytuje zpětnou vazbu učícímu prvku tak, že hodnotí činnost aktéra a navrhuje směr modifikace (komponenta oddělená od modifikací prvku činnosti)

➢ *Generátor problémů* sloužící ke zkoumání nových akcí směřujících k novým zkušenostem

Učící (se) aktér

- Učení může zahrnovat i aktérovo zkoumání „Co mé akce dělají“ nebo „Jak se svět vyvíjí“
- Externí standard činnosti může zahrnout i vnější zpětnou vazbu k činnosti
- Může do toho vstupovat i zpětná vazba z lidského chování



Obecný učící se aktér. Obdélník "prvek činnosti" odpovídá tomu, co byl dřív celý program aktéra. Obdélník "Učicí prvek" tento program modifikuje, aby zlepšil jeho činnost.