

BDI AGENTI

Veronika Hájková
262048

Softwarový model BDI

- ▶ přístup ke tvorbě softwarově implementovaných multiagentních systémů
- ▶ inspirovaný teoriemi filosofa M. E. Bratmana
- ▶ teorie souvisí s tzv. **lidovou psychologií**, s „teorií teorie“
- ▶ záměry se od přání liší určitou mírou závazku
- ▶ základní jednotkou BDI agent

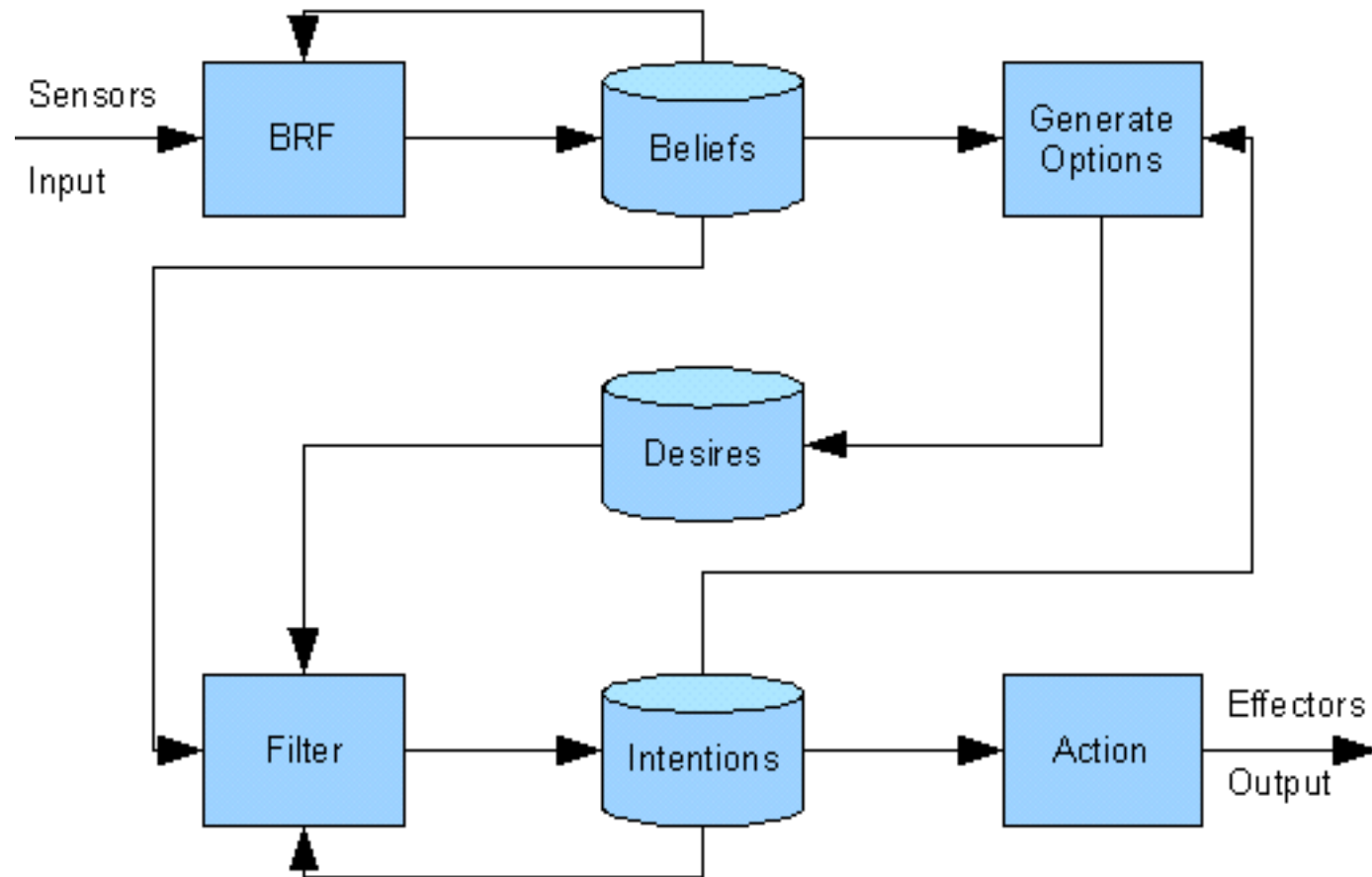


BDI agent

- ▶ zvláštní druh deliberativního či racionálního softwarového agenta, jehož jednání vychází z:
 - ❖ **představ (Beliefs)** – jeho představy o stavu světa
 - ❖ **přání (Desires)** – krátkodobé či dlouhodobé cíle
 - ❖ **záměrů (Intentions)** – záměry konání, které mohou vést ke splnění přání a cílů
 - ❖ **Plánovač (Means-End Reasoner/Analyser)** – na základě představ, přání a záměrů sestavuje plán, jak cílů dosáhnout
 - ❖ **BDI logika** – temporální logika, modální logika, řízeno rozhodování agentů
- ▶ **vlastnosti:**
 - ❖ Deliberativní
 - ❖ Řízený logikou
 - ❖ Racionální (omezeně)
 - ❖ Zaměřený na cíl



Struktura BDI agenta



Implementace a architektury modelu BDI

- Algoritmus podle Wooldridge
- PRS architektura
- AgentSpeak(L)
- JACK Intelligent Agents
- OASIS: Air Traffic Management Systém



Algoritmus podle Wooldridge

- navrhl Michael Wooldridge
- není univerzální návrh pro realizaci agenta, spíše ukázka možného přístupu k realizaci agenta jako entity řízené nekonečným cyklem

```
opakuj {  
    přijmi vjem z okolí  
    uprav vnitřní model prostředí  
    vyber záměr  
    sestav plán pro dosažení záměru  
    spusť plán  
}
```

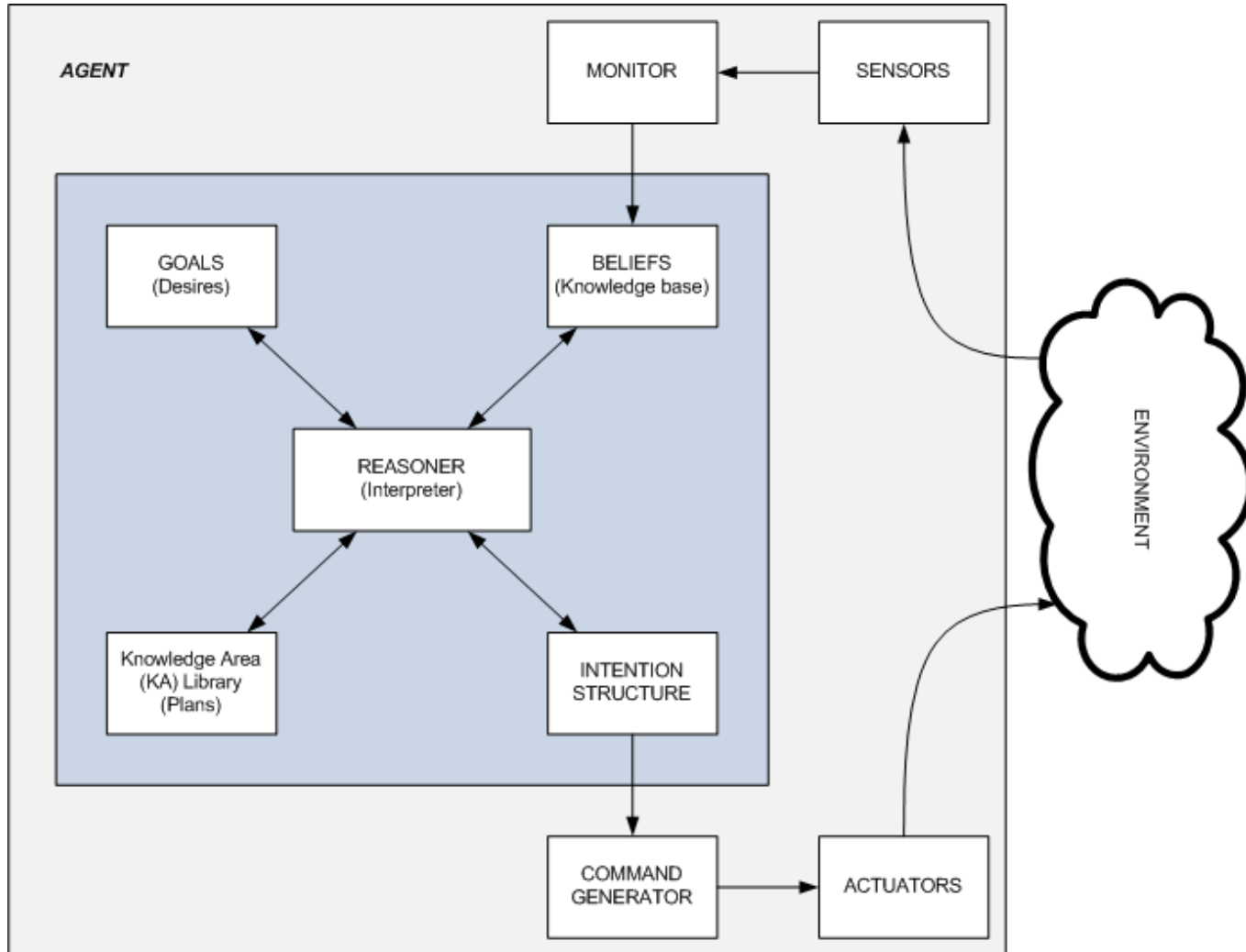


PRS architektura

- Procedural Reasoning Systém
- pokus o formální specifikaci – výpočetní systém dMars (distributed Multi-Agent Reasoning System)
- umožňuje usuzovat o prostředcích a cílech podobně jako plánovače
- reaktivita, která je nezbytná pro přežití ve vysoce dynamických a nejistých světech
- knihovna procesů aplikovatelné parciální plány:
 - ❖ podmínka spuštění (invocation condition)
 - ❖ tělo – posloupnost akcí
 - ❖ definovaný konečný efekt na systém
- proces rozhodování je řízen interpretem



PRS architektura



PRS architektura

- ▶ úspěšně implementována v řadě reálných aplikací (roboti, programoví agenti)
- ▶ detekování chyb raketoplánů a v sondách pro vyhledávání ponorek
- ▶ výchozím systémem pro jiné systémy



AgentSpeak(L)

- ▶ agentově orientovaný programovací jazyk
- ▶ Založen na logickém programování a BDI architektuře
- ▶ 1996, Anand Rao
- ▶ abstraktní programovací jazyk agenta jako pomůcka pochopení vztahu mezi implementací architektury BDI a formalizací myšlenek nad architekturou BDI – modální logikou
- ▶ jedním z nejpopulárnějších – platforma Jason



AgentSpeak(L)

```
+concert(A,V) : likes(A)
  ← !book_tickets(A,V).

+!book_tickets(A,V) : ¬busy(phone)
  ← call(V);
  ...;
  !choose_seats(A,V).
```

Figure 2.1: Examples of AgentSpeak(L) Plans

<u>agent</u>	→ (<u>init_bels</u> <u>init_goals</u>) * <u>plans</u>
<u>init_bels</u>	→ <u>beliefs</u> <u>rules</u>
<u>beliefs</u>	→ (<u>literal</u> ".") *
<u>rules</u>	→ (<u>literal</u> ":-" <u>log_expr</u> ".") *
<u>init_goals</u>	→ ("!" <u>literal</u> ".") *
<u>plans</u>	→ (<u>plan</u>) *
<u>plan</u>	→ ["@" <u>atomic_formula</u>] <u>triggering_event</u> [":" <u>context</u>] ["<-" <u>body</u>] "
<u>triggering_event</u>	→ ("+" "-") ["!" "?"] <u>literal</u>
<u>literal</u>	→ ["~"] <u>atomic_formula</u>
<u>context</u>	→ <u>log_expr</u> "true"
<u>log_expr</u>	→ <u>simple_log_expr</u> "not" <u>log_expr</u> <u>log_expr</u> "&" <u>log_expr</u> <u>log_expr</u> " " <u>log_expr</u> "(" <u>log_expr</u> ")"
<u>simple_log_expr</u>	→ (<u>literal</u> <u>rel_expr</u> <VAR>)
<u>body</u>	→ <u>body_formula</u> (";" <u>body_forumula</u>) * "true"

JACK Intelligent Agents

- ▶ rámeček (framework) v Javě pro vývoj multi-agentních systémů
- ▶ vyvinut v roce 1997 členy australského institutu umělé inteligence (Australian Artificial Intelligence Institute)
- ▶ pro komerční použití
- ▶ poskytuje vlastní plánovací jazyk založený na Javě a grafické plánovací nástroje



OASIS: Air Traffic Management System

- systém řízení letového provozu
- Optimal Aircraft Sequencing using Intelligent Scheduling
- pro letiště v Sydney roku 1995
- přesně vypočítá odhadované časy přistání
- určuje pořadí letadel k přistání
- monitoruje a porovnává skutečný postup letadel vůči odhadovanému pořadí



OASIS: Air Traffic

Management System

- reaguje na náhlé změny v prostředí (např. meteorologické podmínky nebo uspořádání dráhy)
- na změny uživatelských cílů (mimořádné provozní požadavky letadla – stav nouze)
- kombinuje umělou inteligenci, softwarové agenty a běžné softwarové techniky
- **Architektura systému OASIS:**
 - ❖ jeden agent pro každé letadlo
 - ❖ množství globálních agentů – řadič, agent pro modelování povětrnostních podmínek, koordinátor, hlídač trajektorie



Seznam literatury:

- ▶ BORDINI, Rafael H. a Jomi F. HÜBNER. *Jason: A Java-based interpreter for an extended version of AgentSpeak*. [online]. [cit. 2012-12-20]. Dostupné z: <http://jason.sourceforge.net/Jason.pdf>
 - ▶ VOLNÁ, Eva. *Vybrané partie umělé inteligence: Multiagentové systémy, Úvod do mementiky*. [online]. Ostrava, 2005 [cit. 2012-12-12]. Dostupné z: http://www.l.osu.cz/~volna/Vybrane_partie_UI_I_dil_skripta.pdf. Skripta. Ostravská univerzita.
 - ▶ Wikipedie: Otevřená encyklopedie. *Softwarový model Belief-Desire-Intention* [online]. 2012 [cit. 2012-12-13]. Dostupné z: http://cs.wikipedia.org/wiki/Softwarov%C3%BD_model_Belief-Desire-Intention
 - ▶ ZBOŘIL, František ml. *IV. BDI Systémy* [online]. Brno, 2005,2006 [cit. 2012-12-12]. Dostupné z: http://www.fit.vutbr.cz/~zborilf/study/AGS/AGS04_BDI.pdf. Prezentace. Vysoké učení technické.
 - ▶ ZBOŘIL, František ml. *Plánování a komunikace v multiagentních systémech* [online]. Brno, 2004 [cit. 2012-12-12]. Dostupné z: <http://www.fit.vutbr.cz/~zborilf/PhD/thesis.pdf>. Disertační práce. Vysoké učení technické. Vedoucí práce doc. Dr. Ing. Petr Hanáček.
-



Děkuji za pozornost

