



Programovanie kognitívnych agentov

(motivácia a prehľad projektu)

Peter Novák

Computational Intelligence Group,
Clausthal University of Technology,
Nemecko

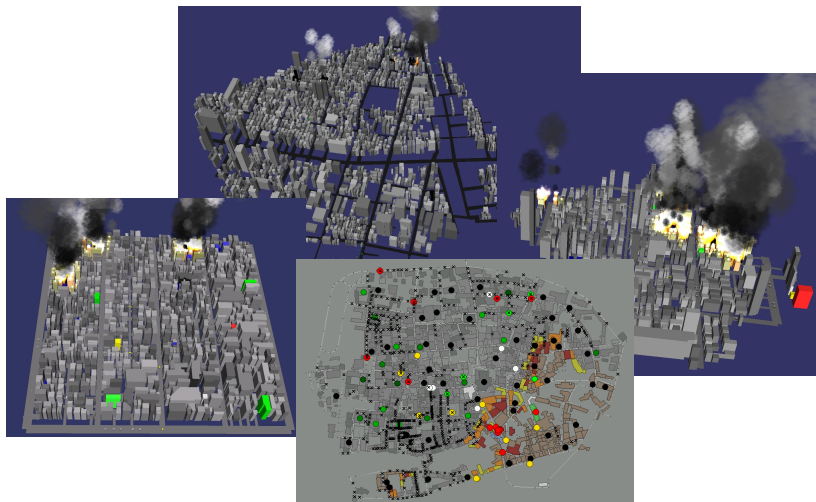
19. december, 2008
TAM 2008



Agenda

- 1 Motivácia
- 2 Programovanie agentových systémov
- 3 Experimentálne aplikácie
- 4 Záver

Príklad domény: RoboCup Rescue SimLeague



Doména (z pohľadu individuálneho agenta)

1 MAS

1 koordinácia $\rightsquigarrow$ komunikácia

$\rightsquigarrow$ reprezentácia znalostí a usudzovanie

2 heterogénny systém

$\rightsquigarrow$ rôznorodosť KRR techník

2 prostredie

1 neštruktúrované $\rightsquigarrow$ ťažkosti s plánovaním

2 dynamické $\rightsquigarrow$ dôraz na reaktívnosť

Konflikté požiadavky

plánovanie vs. reaktívnosť

Doména (z pohľadu individuálneho agenta)

1 MAS

1 koordinácia $\rightsquigarrow$ komunikácia

$\rightsquigarrow$ reprezentácia znalostí a usudzovanie

2 heterogénny systém

$\rightsquigarrow$ rôznorodosť KRR techník

2 prostredie

1 neštruktúrované $\rightsquigarrow$ ťažkosti s plánovaním

2 dynamické $\rightsquigarrow$ dôraz na reaktívnosť

Konflikté požiadavky

plánovanie vs. reaktívnosť

Kognitívny agent - pracovná definícia

(Inteligentný) agent

- autonómny
- reaktívny
- pro-aktívny

Kognitívny (inteligentný) agent

- telo - senzory/aktuátory $\rightsquigarrow$ prostredie
- znalosti - stav prostredia, postoje/ciele $\rightsquigarrow$ mentálny stav
- dynamika - konanie akcií $\rightsquigarrow$ správania $\rightsquigarrow$ nedeterminizmus
- paralelizmus - $\rightsquigarrow$ množstvo naraz bežiacich správání

Agentovo-orientovaný programovací jazyk

plánovanie vs. **reaktívnosť**
~> **hybridná architektúra**

Robustný reaktívny výpočtový model umožňujúci integráciu **heterogénnych KRR technológií** +

- *teoretické vlastnosti* - vhl'ad do vlastností systému ~> verifikácia, analýza, design
- *použitelnosť v praxi* - podpora tradičných SE techník, modularita, integrácia s externými systémami

Rôzne techniky reprezentácie znalostí a usudzovania sú vhodné pre rôzne aplikačné domény.

Súčasný stav (BDI)

Belief-Desire-Intention

architektúra agenta = (znalosti + ciele + zámery) + interakcie

- **znalosti:** model sveta
- **ciele:** popisy situácií, ktoré by agent rád dosiahol
- **zámery:** situácie a plány, ktoré agent práve sleduje

Pragmatické prístupy (JACK, Jadex):

- rozšírenie objektovo-orientovaného jazyka
- imperatívny štýl, sémantika, slabé KRR

Teoreticky orientované prístupy (AgentSpeak(L), 3APL, etc.):

- deklaratívny štýl, sémantika $\rightsquigarrow$ verifikácia
- KRR=logika, “from scratch” $\rightsquigarrow$ nepraktické, integrácia s prostredím

Súčasný stav (BDI)

Belief-Desire-Intention

architektúra agenta = (znalosti + ciele + zámary) + interakcie

- **znalosti:** model sveta
- **ciele:** popisy situácií, ktoré by agent rád dosiahol
- **zámary:** situácie a plány, ktoré agent práve sleduje

Pragmatické prístupy (JACK, Jadex):

- rozšírenie objektovo-orientovaného jazyka
- imperatívny štýl, sémantika, slabé KRR

Teoreticky orientované prístupy (AgentSpeak(L), 3APL, etc.):

- deklaratívny štýl, sémantika $\rightsquigarrow$ verifikácia
- KRR=logika, “from scratch” $\rightsquigarrow$ nepraktické, integrácia s prostredím

Behavioural State Machines/Jazzyk

Behavioural State Machines/Jazzyk

Štíhly programovací systém podporujúci jasné rozdelenie medzi *reprezentáciou znalostí* a *správaniami* agenta.

usudzovanie vs. výpočtový model

BSM agent: $\mathcal{A} = (\mathcal{M}_1, \dots, \mathcal{M}_n, \mathcal{P})$

KR modul $\mathcal{M} = (\mathcal{S}, \mathcal{L}, \mathcal{Q}, \mathcal{U})$

- $\mathcal{S}$ - množina mentálnych stavov
- $\mathcal{L}$ - jazyk reprezentácie znalostí
- $\mathcal{Q}$ - query operátory $\models: \mathcal{S} \times \mathcal{L} \rightarrow \{\top, \perp\}$,
- $\mathcal{U}$ - update operátory $\odot: \mathcal{S} \times \mathcal{L} \rightarrow \mathcal{S}$.

Behavioural State Machines/Jazzyk (pokr.)

mental state transformer:

$$\models_i \varphi \longrightarrow \odot_j \psi$$

when query_i **module**_i [{ φ }] **then** update_j **module**_j [{ ψ }]

$\tau_1 | \tau_2$ nedeterministický výber, $\tau_1 \circ \tau_2$ zret'azenie,
{...} nestovaná syntax, makrá

prechodový systém nad stavmi $\sigma = \langle \sigma_1, \dots, \sigma_n \rangle$ indukovaný
updatmi $\oplus \psi$ z programu $\mathcal{P}$

Sémantika BSM/Jazzyk programu (operačný pohľad)

Sekvencia $\sigma_1 \xrightarrow{\odot_1} \sigma_2 \xrightarrow{\odot_2} \dots \xrightarrow{\odot_{i-1}} \sigma_i \xrightarrow{\odot_i} \dots$ je **beh** BSM.

Agent (BSM) je charakterizovaný množinou všetkých **behov**.

Behavioural State Machines/Jazzyk (pokr.)

mental state transformer:

$$\models_i \varphi \longrightarrow \odot_j \psi$$

when query_i **module**_i [{ φ }] **then** update_j **module**_j [{ ψ }]

$\tau_1 | \tau_2$ nedeterministický výber, $\tau_1 \circ \tau_2$ zreťazenie,
{...} nestovaná syntax, makrá

**prechodový systém nad stavmi $\sigma = \langle \sigma_1, \dots, \sigma_n \rangle$ indukovaný
updatmi $\oplus \psi$ z programu $\mathcal{P}$**

Sémantika BSM/Jazzyk programu (operačný pohľad)

Sekvencia $\sigma_1 \xrightarrow{\odot_1} \sigma_2 \xrightarrow{\odot_2} \dots \xrightarrow{\odot_{i-1}} \sigma_i \xrightarrow{\odot_i} \dots$ je **beh** BSM.

Agent (BSM) je charakterizovaný množinou všetkých **behov**.

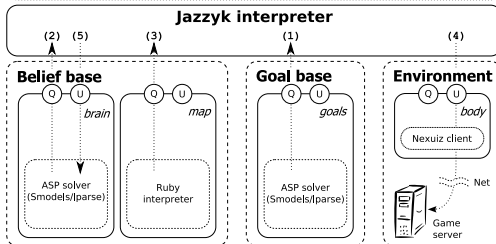
Jazzbot: Bot v simulovanom 3D prostredí.

- testbed pre použitie **heterogénnych KRR** v agentovej doméne
- individuálny agent
- testbed pre **NMR/ASP/EKB aplikácie** v dynamickom prostredí
- **zaujímavé, bohaté, dynamické** prostredie

Agent program:

```

when believes goals(Obj) [{find(Obj)}] and (1)
    believes brain(Obj) [{see(Obj)}] and (2)
    query map(Object, Dist) [{Dist=get_distance_of(Obj)}] (3)
then {
    act body(Dist) [{move forward Dist}] , (4)
    update brain(Obj) [{keeps(Obj)}] (5)
}
    
```



URBI-Bot: **simulovaný e-Puck robot**

⇒ smerom k programovaniu kognitívnych robotov

URBI

Jednoduchý event-based programovací jazyk.

- ⇒ nezávislý na OS, ľahko integrovateľný, modulárny - komponenty pre robotické HW moduly

URBI-Bot (e-Puck robot)

- podobná štruktúra ako Jazzbot, i.e. využíva nemonotónne usudzovanie (ASP)
- kód pre simulátor je priamo prenositeľný na reálneho robota
- dôraz na dynamiku prostredia



Zhrnutie

- BSM = **nová paradigma pre vývoj APL**
 - generický jazyk pre **reaktívne systémy** +
 - podpora **designových vzorov** pre programovanie BDI agentov
 - využitie **heterogénnych KRR** technológií
- **proof-of-concept štúdie** simulovaných robotov: *Jazzbot*, *URBI-Bot*
- **praktické použitie NMR/ASP/EKB** v agentovej doméne: robotika, hry (non-player-characters)

Prebiehajúca a budúca práca

- 1 DCTL*: logika pre BSM/Jazzyk
 - ↪ hybrid modálnej temporálnej logiky CTL* a Harellovej Dynamickej Logiky
 - verifikácia
 - knižnica dizajnových vzorov pre BSM/Jazzyk
- 2 pravdepodobnostné BSM
- 3 experimenty smerom k MAS: Multi-Agent Programming Contest
- 4 **aplikácie v zábavnej robotike**: Rovio, Nao



Ďakujem za pozornosť...

<http://jazzyk.sourceforge.net/>

