MASARYKOVA UNIVERZITA , FILOZOFICKÁ FAKULTA

BDI Agenti

PB016 Úvod do umělé inteligence

Veronika Hájková, 262048

13.12.2012

Agent

Pojem (racionální) agent pochází z oblasti multiagentních systémů a označuje agenta (samostatnou entitu umístěnou do určitého prostředí, která může vykonávat nějakou činnost), který na základě posloupnosti vjemů a vnitřních znalostí volí akci maximalizující očekávanou míru výkonu. Racionalitu chápeme jako schopnost agenta plnit funkce v prostředí, v němž se agent nachází, co nejadekvátnějším způsobem. V literatuře se můžeme setkat i s pojmem Inteligentní agent.
Základní charakteristické vlastnosti agenta jsou následující:

- **Autonomnost** – agent je schopen dosáhnout svých záměrů bez jakýchkoliv vnějších zásahů (bez implicitní závislosti na jakýchkoliv jiných prvcích tohoto systému), tzn. pouze v interakci s prostředím, tato vlastnost je však předmětem řady diskuzí
- **Reaktivita** – schopnost průběžné reakce na změny prostředí tak, aby dosáhl cíle
- **Intencionalita** – schopnost uvažovat o svých dlouhodobých cílech
- **Proaktivita** – schopnost ovlivňování okolního prostředí, aby jeho stav usnadňoval dosažení závěrů

Podle výše uvedených vlastností, složitosti organizace vnitřních komponent agentů a jejich racionality můžeme uvést následující rozdělení:

1) **Reaktivní agent** – agent bezprostředně reaguje na změny prostředí (nebo změny své vůči prostředí), přestože nemá vnitřní reprezentaci znalostí o tomto prostředí. Agent neobsahuje žádné moduly pro tvorbu plánů, jeho rozhodování není výsledkem výpočtů či dedukcí na základě znalostí, ale pouze reakcí na podnět.
2) **Deliberativní agent** – agent má vlastní reprezentaci prostředí, která je realizována formou databáze tvrzení o světě, do jisté míry umožňující racionální chování. Tento agent se blíží autonomnímu chování lidí. Na rozdíl od člověka, který má schopnost sebereflexe a přizpůsobení intencionality směrem k jiným jedincům, však agent nemůže volit svobodně prostředky pro plnění cílů ani postup, jakým k cíli dojde, ani nemůže zlepšovat své chování a funkcionalitu.
3) **Sociální agent** – má znalosti o systému jako celku i ostatních agentech. Udržuje a rozšiřuje poznatky o ostatních agentech a plánech a je schopen uvažovat o jejich cílech a motivacích. Poznatky jsou většinou adresy, jména agentů, specifikace jejich schopností, což slouží ke kooperaci vzájemných aktivit. Agent je také často vybaven historií předchozích interakcí (ceny transakcí, apod.), které slouží ke zmírnění negativních důsledků omezenosti vlastních zdrojů či ke snazšímu vyhledávání pomoci ostatních v budoucnu.
4) **Hybridní agent** – kombinuje některé nebo všechny vlastnosti v jeden celek. Příklad takového agenta je InteRRaP (Integration Behaviour and Rational Planning), skládá se ze dvou částí – z plánovací jednotky a reaktivní jednotky. Za běhu agent vytvoří plán a začne jej provádět. Každý krok je kontrolován reaktivní jednotkou, která sleduje, zda v systému nedošlo ke změně. Změna je signálem k přehodnocení plánu v plánující jednotce.
5) **Racionální agent** – stojí nejvýše na pomyslné hierarchii daných agentů, je speciálním případem hybridního agenta. Obsahuje plánovací jednotku, kognitivní jednotku a bázi znalostí. Je schopen se učit na základě svých poznatků plánovat činnost tak, aby dosáhl svých cílů racionálním způsobem.

Agentní prostředí

Za prostředí považujeme vše, s čím agent přichází během své činnosti do styku. Prostředí může být:

- **Plně pozorovatelné/částečně pozorovatelné** – záleží, zda agent může prostředí svými senzory zcela sledovat (všechny jeho stavy) či jen částečně)
- **Statické/dynamické** – statické prostředí se může měnit jenom díky akcím agenta, dynamické i bez zásahu
- **Deterministické/nedeterministické** – u deterministického prostředí je predikovatelný jeho následný stav ze znalosti momentálního stavu prostředí a vykonané agentní akce, u nedeterministického nelze následný stav predikovat
- **Diskrétní/spojité** – diskrétní má konečný, spočetný počet stavů, spojité nekonečný.

Koordinace/Kooperace/Komunikace

Koordinace je proces v rámci multiagentního systému, jenž má za cíl vhodně přidělovat limitované zdroje a úlohy ostatním agentům tak, aby došel k výsledku optimální cestou - účelně, bez chaosu. Někdy je koordinace chápána jako komunikační proces sloužící k racionálnímu chování systému jako celku. Základním předpokladem je přítomnost agenta nebo agentů, kteří mají jisté sociální povědomí o ostatních agentech a dokáží uvažovat o důsledcích jejich předpokládaného chování.
Kooperace je proces, při kterém agenti vyjednávají a rokují o společném řešení problémů nebo konfliktů. Mohou být řízeni centrálně nebo decentralizovaně. Každý agent ve skupině má přesně definovanou roli, kterou musí plnit. Jedním z koordinačních procesů je např. vyjednávání – komunikační proces vedoucí k dosažení dohody (cíle).
Komunikace je velmi důležitý proces pro multiagentní systémy. Na komunikaci jsou závislé jednotlivé protokoly, ale také koordinace a kooperace. Vlastní komunikace je chápána jako proces, během kterého si dva nebo více agentů vyměňují informace formou elementárních komunikačních zpráv. Jazykem komunikace je tzv. ACL (Agent Communication Language). Samotný přenos zprávy probíhá standartní síťovou cestou a na přenosu zprávy se podílí následující vrstvy:

1) **Fyzická vrstva** – nejnižší, přenos realizuje jako posloupnost bitů, využívá nejnižší vrstvy modelu ISO/OSI – fyzickou, linkovou a síťovou
2) **Transportní vrstva** – specifikuje protokoly na aplikační úrovni, např. kódování, mezi tyto protokoly patří např. HTTP (Hypertext Transmition Protocol) a WAP (Web Application Protocol)
3) **Vrstva komunikační architektury** – pro komunikaci vrstvy pro vzdálené volání procedur (Remote Procedute Call) v různých podobách. XML RPC, Unix RPC, Java RMI nebo CORBA
4) **Vrstva ACL** – vrstva nesoucí danou informaci vyjádřenou komunikačním jazykem, přenášená informace obsahuje metainformace (transportní informace, použitý jazyk v obsahu zprávy atd.) a samotný obsah sdělení

Softwarový model BDI a BDI agent

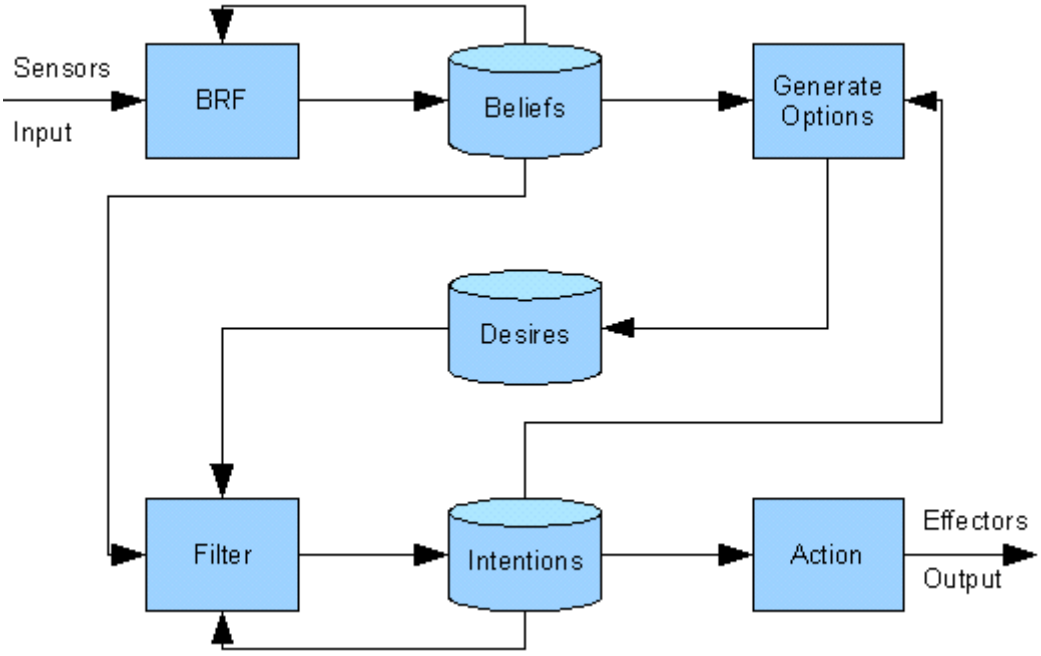
Softwarový model Belief-Desire-Intention (BDI) je přístup ke tvorbě softwarově implementovaných multiagentních systémů inspirovaný teoriemi filosofa M. E. Bratmana. BDI agent je zvláštní druh deliberativního či racionálního softwarového agenta, jehož jednání vychází z jeho představ (Beliefs), přání (Desires) a záměrů (Intentions).

Belief-Desire-Intention model lidského usuzování, který zahrnuje zvolení cíle a plánování, jak ho dosáhnout, byl vytvořen filosofem Michaelem E. Bratmanem. Tato teorie souvisí s tzv. lidovou psychologií, s „teorií teorie“ – máme nějaké subjektivní vnímání okolního světa, to se ale může lišit od skutečnosti. Důležitou roli v praktickém usuzování hrají záměry, které nejsou převoditelné na představy a přání. Záměry se od přání liší určitou mírou závazku.

Základní jednotkou softwarového BDI modelu je BDI agent, který má následující vlastnosti, je:

- **Deliberativní** – uchovává symbolickou reprezentaci svého prostředí, případně další znalosti, je řízen určitou vnitřní inteligencí
- **Řízený logikou** – jeho akce řídí logický kalkul
- **Racionální** – vykonává akce, které jsou v jeho nejlepším zájmu s ohledem na představy, které má o světě (omezeně racionální) – nemá k dispozici neomezené informace ani neomezený čas
- **Zaměřený na cíl** – k dosažení svých přání či cílů

Struktura BDI agenta



Beliefs (představy) – jsou informace daného agenta, jeho představy o stavu světa, bývají reprezentovány symbolicky. Např. v implementaci PRS jsou podobné jako fakta v PROLOGu.

Desires (přání) – čeho by chtěl agent dosáhnout, stavy, které by chtěl, aby se uskutečnily. Tato přání mohou být krátkodobé či dlouhodobé (cíle). Agent nemusí být schopen dosáhnout všech přání, dokonce se mohou vylučovat.

Intentions (záměry) – záměry konání, které mohou vést ke splnění přání a cílů. Záměry se od přání liší jistým druhem závazku. Intence mohou sdílet agenti usilující o stejný cíl.
Plánovač (Means-End Reasoner/Analyser) je důležitou součástí, který na základě představ, přání a záměrů sestavuje plán, jak cílů dosáhnout. Například u PRS architektury se můžeme setkat s již naimplementovanými plány pro jednotlivé záměry a plánovač v průběhu z této zásoby plánů pouze vybírá.
Rozhodování agentů je řízeno BDI logikou, která se může lišit podle konkrétní implementace.
1) Rozšíření temporální logiky větvícího se času (CTL – Computer Tree Logic)pro zachycení stromové hierarchie možných akcí.
2) Modální logika, stojí na konceptu možných světů pro reprezentaci představ, přání a záměrů.
Např. pravidlo:

Pokud Agent „i" věří, že je ventil 32 otevřený, tak by „i" měl mít Intenci, aby „j" také věřil, že je ventil 32 otevřený.

se dá vyjádřit v BDI logice takto:

(Bel i Otevřený(ventil32)) => (Int i (Bel j Otevřený (ventil32))

Implementace a architektury BDI

BDI je obecná teorie, která může být realizovaný různými způsoby, pomocí architektur, jazyků a nástrojů.

Algoritmus podle Wooldridge

Tento algoritmus navrhl Michael Wooldridge. Předpoklad, ze kterého tento přístup vychází, je podobně jako u deliberativního agenta takový, že plán je posloupnost akcí vedoucí k dosažení nějakého zvoleného záměru. Záměr je směrován podle agentových přání a aktuálního stavu prostředí. Základní algoritmus má jednoduchý tvar, nelze ho však chápat jako univerzální návrh pro realizaci agenta, spíše jako ukázku možného přístupu k realizaci agenta jako entity řízené nekonečným cyklem.

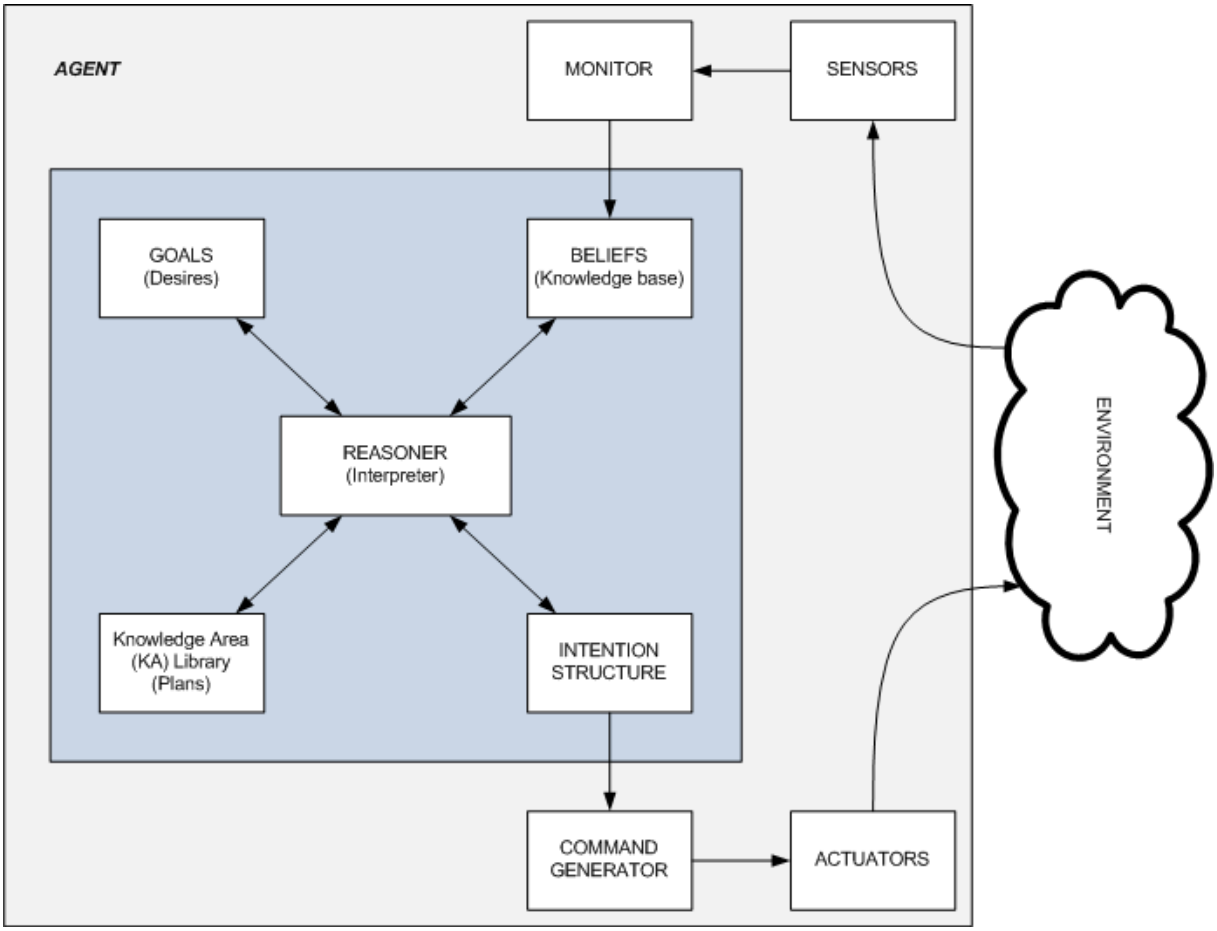
```
opakuj  {
            přijmi vjem z okolí
            uprav vnitřní model prostředí
            vyber záměr
            sestav plán pro dosažení záměru
            spusť plán
        }
```

Architektura IRMA (Bratman, Israel, Pollack 1988)

Architektura IRMA (Intelligent Resource-Bounded Machine Architecture) je jednou z prvních architektur agentů navrženou pro práci v reálném (tj. v otevřeném a nedeterministickém) prostředí. S tím souvisí omezený čas, který má agent při rozhodování, ale také omezené znalosti prostředí. Je to také první architektura, ve které se objevuje pojem BDI pro agentovy představy o věrohodnosti informací z okolního prostředí a o výsledcích vlastních akcí.
Vlastní architektura IRMA obsahuje čtyři klíčové symbolické složky: *knihovnu parciálních plánů* vedoucích k dosažení cílového stavu (záměru agenta), a explicitní reprezentaci přestav, přání a

záměrů. Navíc obsahuje logickou jednotku pro usuzování o okolním prostředí(usuzovač), analyzátor prostředků a cílů (means-end analyser), jež určuje který plán může být použit k dosažení záměru a analyzátor možností (opportunity analyser), který monitoruje okolí. Dále IRMA obsahuje filtrační jednotku (filtering process), která je zodpovědná za výběr podmnožiny akcí konzistentních s aktuálními záměry agenta. Poslední jednotkou je deliberativní jednotka (deliberation process), která sestavuje nové plány respektující nové možnosti. Pokud uvažujeme analyzátor možností jako část, která zastupuje agentovy představy, jednotku prostředků a cílů jako nástroj pro plnění agentových přání a konečně filtrační jednotku jako jednotku pro agentovy záměry, je zřejmá souvislost mezi architekturou IRMA a BDI přístupem. Vyskytují se tu však dva problémy. Prvním je schopnost agenta reagovat na změnu prostředí změnou plánu, nebo dokonce záměru. Druhým problémem je nutnost sledování konečných či dlouhodobých cílů a při hledání parciálních plánů ověřovat, jestli je jejich případné provedení v souladu s těmito dlouhodobými cíli.

PRS architektura



Architektura PRS (Procedural Reasoning System)zahrnuje prvky BDI přístupu k budování agentních systémů. Jako pokus o formální specifikaci této architektury byl navržen výpočetní systém zvaný dMars (distributed Multi-Agent Reasoning System). PRS umožňuje usuzovat o prostředcích a cílech podobně jako plánovače, ale její výhodou je taková reaktivita, která je nezbytná pro přežití ve vysoce dynamických a nejistých světech. Pracuje navíc s knihovnou procesů (aplikovatelné plány). V knihovně se nacházejí parciální plány – procedurální znalosti, jedná se o posloupnosti akcí, které transformují systém z počátečního stavu do stavu cílového. Každý takový parciální plán má svou podmínku spuštění(invocation condition), tělo (posloupnost akcí) a také definovaný konečný efekt na

systém po jeho úspěšném či neúspěšném vykonání. Celý proces rozhodování je řízen interpretem. V jednotce představ jsou uchovávány znalosti o prostředí a při proměně tohoto prostředí během činnosti agenta je jednotka aktualizována. Podle agentova stavu jsou vybírána přání a pro daná přání je sestavován po částech plán, k dosažení cíle je vytvořena řada parciálních plánů. Když je plán sestaven, je zahrnut do záměrů agenta a z knihovny procesů jsou pak interpretem vybírány jednotlivé procesy k realizaci. Cyklus interpretace PRS agenta:[1]

1. V každém okamžiku je možné přehodnotit představy a cíle agenta
2. Tato změna může spustit, nebo iniciovat jako aplikovatelné některý akt z knihovny aktů
3. Některé z těchto aplikovatelných aktů může být vybrán do grafu záměrů
4. Z grafu záměrů je vybrán akt ke spuštění
5. Je vykonáván jeden krok z aktu vybraného pro spuštění
6. Každý krok představuje vykonání elementární akce v prostředí
7. Akce může způsobit vznik nového podcíle, nebo nové znalosti o prostředí
8. Nebo způsobí změnu samotného grafu záměrů

PRS byl úspěšně implementován v řadě reálných aplikací (roboti, programoví agenti), byl použit pro detekování chyb raketoplánů a v sondách pro vyhledávání ponorek. Je také výchozím systémem pro jiné systémy.

Výpočetní systém pro PRS – dMARS

Prvky systému dMARS můžeme rozdělit do tří skupin:

1) Formule popisující agentovy představy (znalosti)
2) Formule popisující agentovy cíle (externí-> směrem k prostředí, interní-> směrem k vnitřním stavům agenta)
3) Plány (procedurální znalosti) – posloupnost akcí, transformují systém z nějakého stavu do jiného

Struktura plánu je složitější a je tvořena položkami Vyvolání (Invocation), Kontext (Context), Udržení cíle (Maintance), Tělo (Body), Úspěch/Neúspěch (Success/Fail). Systém dMARS vybere na základě aktuálního stavu prostředí nejprve všechny plány, které je možno vykonat – při změně prostředí či při nějaké vnitřní událostí. Z těchto plánů pak vybere ten, který nejlépe vyhovuje záměru a pro tento plán vytvoří instanci. Po vytvoření je instance umístěna na vrchol zásobníku plánů, označena jako aktivní a začne se vykonávat. Pokud je akce jedním z listů plánu a po jejím vykonání se plán dostane do koncového stavu, plán uspěje. Neuspěje naopak, pokud není v koncovém stavu a zároveň není možné v aktuálním stavu prostředí pokračovat žádnou z možných větví plánu. V obou případech (úspěch či neúspěch) je vykonána interní akce, která buď odebere aktuální plán z vrcholu zásobníku a pokračuje vykonáváním dříve přerušeného plánu nebo vytvoří plán nový a jeho instanci umístí do vrcholu zásobníku.

[1] ZBOŘIL, František ml. BDI systémy. 2006 [cit. 2011-01-21]. Dostupné online

Praktické aplikace

OASIS: Air Traffic Management System

Systém řízení letového provozu OASIS (Optimal Aircraft Sequencing using Intelligent Scheduling), který byl vyvinut pro letiště v Sydney a úspěšně otestován v dubnu roku 1995 tamtéž. OASIS přesně vypočítá odhadované časy přistání a určuje pořadí letadel k přistání, přičemž počítá s nejmenším celkovým zpožděním a navrhuje řídícímu letového provozu příslušná kontrolní opatření k dosažení tohoto pořadí. Také monitoruje a porovnává skutečný postup letadel vůči odhadovanému pořadí a upozorňuje řídícího letového provozu na významné rozdíly a vhodná opatření k nápravě situace. OASIS je navržen tak, aby reagoval na náhlé změny v prostředí (např. meteorologické podmínky nebo uspořádání dráhy) a změny uživatelských cílů (mimořádné provozní požadavky letadla – stav nouze apod.)

OASIS kombinuje umělou inteligenci, softwarové agenty a běžné softwarové techniky k tomu, aby dosáhl výrazně vyšší všestrannosti než jakýkoliv jiný systém určený k plynulosti v letovém provozu.

Architektura systému OASIS se skládá z jednoho agenta pro každé letadlo a množství globálních agentů, včetně řadiče, agenta pro modelování povětrnostních podmínek, koordinátora a hlídače trajektorie. V každém okamžiku systém pojme běh až sedmdesáti či osmdesáti agentů současně, provádí řazení a dává pokyny letové kontrole v reálném čase. Agenti letadel jsou zodpovědní za létání s letadly a globální agenti za všeobecné řízení a koordinaci letadel.[2]

Hlavní vlastnosti řešení jsou:

- Schopnost tvořit plány reagující na specifické situace, citlivé na kontext, zaměřené na svůj účel
- Rovnováha mezi chováním reaktivním a chováním zaměřeným na cíl
- Vysoká flexibilita díky vysokoúrovňovému reprezentačnímu a programovacímu jazyku

[2] 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

SEZNAM LITERATURY:

1) VOLNÁ, Eva. *Vybrané partie umělé inteligence: Multiagentové systémy, Úvod do mementiky.* [online]. Ostrava, 2005 [cit. 2012-12-12]. Dostupné z: http://www1.osu.cz/~volna/Vybrane_partie_UI_1_dil_skripta.pdf. Skripta. Ostravská univerzita.
2) 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
3) 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.
4) 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.