UDK621.3:(53+54+621+66), ISSN0352-9045 Informacije MIDEM 35(2005)4, Ljubljana

RAZUMNI HIŠNI POMOČNIK

Uporaba teorije o razumnih agentih za izvedbo inteligentne hišne naprave

[1]Konrad Steblovnik, [2]Jurij Tasič, [3]Damjan Zazula

[1]Gorenje program Point, Velenje
[2]Fakulteta za elektrotehniko, Ljubljana
[3]Fakulteta elektrotehniko, računalništvo in informatiko, Maribor

Kjučne besede: modalna logika, večmodalna logika, usmerjeno vedênje, agent, razumni agent, BDI-model, BDI arhitektura, praktično razmišljanje, ciljna usmerjenost, ciljno usmerjeni agenti, inteligentni hišni pomočnik, povezane naprave, porazdeljeni sistemi, interoperabilnost, večagentni sistemi.

Izvleček: Vsebina članka temelji na teoriji in izvedbi razumnega agenta in njeni možni uporabi na področju inteligentnih hišnih naprav. Najprej opisujemo nekaj glavnih značilnosti agentov kot samostojnih računalniških sistemov, agentnih skupnosti in komunikacijo med agenti, ter nekatere glavne motive za uporabo te tehnologije na področju inteligentnih hišnih naprav. V nadaljevanju povzemamo teorijo o namenskem ali usmerjenem vedęnju in modalno logiko kot primerno orodje za opis miselnih stanj agentov. Iz te logike izhaja večmodalna, BDI-logika, ki predstavlja osnovni formalizem za obravnavo BDI-modela razumnega agenta. Jedro članka je posvečeno razumnim agentom, ki so zgrajeni na osnovi BDI-modela, to je modela prepričanja, želje in namena (*BDI - Belief, Desire, Intention*). Nato podajamo kratek pregled metodologij in razvojnih orodij za načrtovanje agentov in agentih sistemov. Posebej predstavljamo razvojni orodji JADE kot hrbtenični sistem agente skupnosti in Jadex kot stroj za razmišljanje, ki temelji na teoriji BDI-modela. Na koncu opisujemo še možno izvedbo razumnega hišnega pomočnika (RHP) in njegov posebni primerek razumnega pomočnika pranja (RPP). Sistem je izveden kot agentna skupnost, sestavljena iz procesnega agenta, ki skrbi za izvajanje ciljnega hišnega procesa (recimo RPP), ter njegovih agentnih pomočnikov - nadzorovanih agentov, ki opravljajo posamezna opravila v okviru ciljnega procesa - recimo pranja perila. V članku želimo pokazati, da lahko samostojne in povezane hišne naprave obravnavamo kot večagentne sisteme.

Intelligent Home Assistant

Theory of Rational Agent Application for Intelligent Home Appliance Implementation

Key words: modal logic, multimodal logic, intentional attitude, agent, rational agent, BDI model, BDI architecture, practical reasoning, goal oriented agents, intelligent home assistant, connected appliances, distributed systems, interoperability, multiagent systems.

Abstract: This paper focuses on the theory and implementation of rational agents, and their possible application in the area of intelligent home appliances. We firstly describe some essential features of agents as autonomous computer systems, agent societies and inter-agent communication, and also some ground reasons to use this technology in the area of intelligent home appliances. Then, we present a short overview of intentional notion and modal logic, used as an appropriate formalization of the agent's mental state description. A derivation of this logic is multi-modal BDI logic which basically formalizes the description of the BDI model of rational agents. The central part of our article discusses rational agents based on the BDI model - the model of *Belief, Desire,* and *Intention*. We continue with a short description of agent design methodologies and development tools. A special attention is paid to JADE, an agent development tool, which is applied as the middleware layer for agent society, and Jadex serving as a BDI reasoning engine. In the final part, our paper reveals a possible application of the Rational Home Assistant, and the Rational Washing Assistant as a special instance. The system is implemented as agent society, and uses a processing agent for performing the target home process, and his assistants - supervised agents that perform different task of the target process, such as laundry washing. In the conclusion, we argue that the autonomous and connected appliances can be treated as multi-agent systems.

1. Uvod

Agentni sistemi in tehnologije spadajo na področje, ki bo po nekaterih virih pomembno vplivalo na razvoj naslednjih računalniških generacij. Postajajo popularni za reševanje širokega področja kompleksnih problemov ali za nadzor kompleksnih sistemov, porazdeljenih sistemov, sistemov s spremenljivim okoljem, sistemov, kjer se programska oprema integrira v izvajalnem času, in podobno. V Gorenju menimo, da je to zelo obetavna in primerna tehnologija bodočnosti za načrtovanje povezanih naprav in porazdeljenih sistemov, primernih za realizacijo inteligentnega doma. V našem primeru gre za naprave bele tehnike, kot so pralni stroj, sušilec perila, hladilnik, zamrzovalnik, kuhalnik in pečica. Te naprave postajajo s pomočjo elektronskih sklopov, ki so zasnovani z novimi visokointegriranimi polprevodniškimi tehnologijami, in vgrajenimi zmogljivimi mikrokrmilniki vse bolj učinkovite. Takšna strojna oprema pomeni temelj za uvedbo programskih rešitev, ki zahtevajo visoko stopnjo zmogljivosti. V aparate bele tehnike bomo kmalu pričeli vgrajevati upravljalno-krmilne enote, ki bodo zasnovane tudi na 32-bitnih arhitekturah RISC in to z močno in aplikaciji primerno periferijo. Tako bomo razvili naprave z visokim nivojem inteligence, medsebojno pa jih bomo povezali v inteligentne sisteme povezanih hišnih aparatov. Prepričani smo, da je omenjena agentna tehnologija povsem primerna in perspektivna za takšne naloge. Najbolj nas zanima, kako z danimi računalniškimi tehnologijami doseči *inteligentnost naprav* in kako v sistemih *povezanih naprav* zagotoviti *interoperabilnost*.

Ukvarjali se bomo torej z določenim razredom računalniških sistemov, ki jih poznamo kot *agente* in jih lahko še natančneje opredelimo kot razumne agente. O računalniških agentih govorimo zato, ker lahko izvajajo *neodvisna ali avtonomna dejanja*, s katerimi dosegajo svoje cilje, za katere smo jih načrtovali. Grobo rečeno, sposobni so se odločati sami zase o tem, kaj naj napravijo v določeni situaciji. Imenujemo jih *razumni agenti*, ker se lahko *dobro (razumno) odločajo* o tem, kaj bodo storili. Izbirajo vedno najboljša možna dejanja.

Danes že lahko srečamo razumne agente na mnogih področjih in v mnogih aplikacijah. Zelo znana je Nasina misija *'Deep Space'* (DS1) /McBurney - 2005/. Agentna tehnologija je na primer čedalje bolj uporabna v elektronskem poslovanju preko interneta, pri upravljanju in nadzoru zračnega prometa, v telekomunikacijskih omrežjih, poslovnih procesih in medicinskih storitvah /Rao - 1995/, /Wooldridge - 1998/, /McBurney - 2005/. Zelo znan je (agentni) sistem za nadzor zračnega prometa OASIS na letališču v Sydneyu /Rao - 1995/.

Za preučevanje agentno usmerjenih sistemov se pojavljajo številni različni pristopi -/Bratman, 1988/, /Doyle, 1992/, /Rao in Georgeff, 1992/, /Shoham, 1993/. Zelo znana arhitektura za izvedbo razumnega agenta je BDI-arhitektura, ki vsebuje miselna vedęnja, kot so prepričanje, želja in namen (Belief, Desire, Intention). Ta vedênja predstavljajo agentove informacijsko in motivacijsko komponento ter komponento preudarjanja. Miselna vedênja določajo obnašanje sistema in lahko postanejo kritična, kadar hočemo doseči ustrezni in optimalni učinek sistema, če pri tem sistemski viri omejujejo proces preudarjanja /Bratman, 1987; Kinny in Georgeff, 1991/.

Za formalizacijo agentov se pojavljajo različni logični okviri. S pomočjo ustrezne logike lahko predstavimo lastnosti razumnih BDI-agentov in razmišljamo o njih na jasen, nedvoumen in dobro definiran način. Najbolj znana logična formalizma za obravnavo razumnih agentov sta logika KARO /Meyer - 2005/ in LORA, ki sta jo razvila Anand Rao in Georgeff v /Rao - 1991/ in /Rao - 1992/. KARO je kratica za Knowledge, Actions, Results in Opportunities (znanje, dejanja, rezultati, priložnosti), LORA pa za: Logics Of Rational Agents (logika o razumnih agentih).

Agent je računalniški sistem, ki je sposoben opraviti prilagodljiva in avtonomna dejanja v dinamičnem, nepredvidljivem in odprtem okolju. Agentne tehnologije so naravne razširitve obstoječih komponentno zasnovanih pristopov in imajo potencial, da bodo v veliki meri vplivale na življenje in delo vseh nas. Temu ustrezno je to področje eno najbolj dinamičnih in vznemirljivih v današnji računalniški znanosti /Luck - 2003/.

Računalniški agenti so nameščeni v stalno spreminjajočem se okolju /Wooldridge - 1997/. Agenti zaznavajo okolje. Imajo samo delno, mogoče napačno informacijo o okolju in so sposobni izvajati omejene napovedi o tem, kaj bo v prihodnosti veljalo. Lahko delujejo na okolje zato, da ga spremenijo, vendar imajo v najboljšem primeru samo delni nadzor nad rezultati svojih dejanj. Koncept takšnega agenta je prikazan na sliki 1.

Slika 1: Agent in njegovo okolje.

Agent mora mogoče izvajati nasprotujoče si naloge. Ima na voljo več različnih odločitev. Od njega zahtevamo, da se odloča pravočasno in da se zna pravočasno odzvati na okolje - deluje v realnem času. Poznamo štiri ključne lastnosti agentov, ki delujejo v večagentnem okolju. To so: avtonomnost, socialnost, odzivnost in usmerjeno delovanje. Razumnim agentom, ki delujejo v sodobnih okoljih, običajno pripisujemo še naslednje lastnosti: mobilnost, verodostojnost, dobronamernost, razumnost in prilagodljivost (učenje). Te lastnosti opredeljujejo razumne agente kot inteligentne sisteme. Pri načrtovanju večagentnih sistemov se običajno ukvarjamo z agentnimi sistemi na dveh nivojih. Na *mikro nivoju* se ukvarjamo z vprašanji, kako načrtujemo agente z zgornjimi lastnostmi. To je področje arhitekture razumnih agentov. Na *makro nivoju* pa obravnavamo področje skupnosti agentov. To je področje porazdeljene umetne inteligence. Ti dve področji sta med seboj tesno povezani.

Članek je zasnovan tako, da v drugem poglavju pokaže uporabnost modalne logike v teoriji vedênja. Tretje poglavje je posvečeno razumnim agentom in njihovi BDI-arhitekturi. V četrtem poglavju predstavimo orodja za načrtovanje agentov in njihovih skupnosti, v petem pa razvijemo idejo o inteligentnem hišnem pomočniku. Šesto poglavje sklene naš prispevek.

2. Teorija vedenja in modalna logika

Zgoraj navedene lastnosti agentov opredeljujemo kot šibko notacijo. Strožja notacija agentnosti se ukvarja z lastnostmi, ki jih pripisujemo ljudem; to so: znanje, mišljenje, nameni, želje ipd. Agentom pripisujemo mišljenjska (mentalna) stanja, ki jih sicer pripisujemo ljudem. Želimo načrtovati in zgraditi sistem naprav, porazdeljenih sistemov in naprav, povezanih v inteligentni dom. Sistem naj bo namenski ali usmerjeni, vodijo pa ga stanja mišljenja agentov. Izraz namenski ali usmerjeni sistem (lahko tudi intencijski sistem) je skoval filozof Dennett /Denett - 1987/. Razložimo ga lahko s primerom naslednjih dveh stavkov, ki opredeljujeta dejavnosti človeka: "Dragica je vzela s sabo

dežnik, ker je *menila*, da bo deževalo." in "Konrad trdo dela, ker *hoče* pripraviti doktorsko disertacijo." Ta dva stavka izražata človeško miselnost, na osnovi katere je obnašanje človeka predvidljivo, hkrati pa razlaga njegovo obnašanje z atributi vedênja, kot so mišljenje, hotenje, upanje, strah itd. Oblike vedênja, ki so opisane s takšno obliko človeške miselnosti, se imenujejo namenski ali usmerjeni pojmi ali nameni. Sistem, ki opisuje stvari, za katere lahko predvidimo obnašanje z metodo pripisovanja mišljenja, želja in razumnega obnašanja, pa imenujemo namenski ali usmerjeni sistem. Pripisovanje pojmov, kot so *mišljenje, prosta volja, nameni, zavedanje, sposobnost* ali *hotenje,* stroju je legitimno, če to izraža informacije o stroju na enak način, kot pa jih izraža o človeku /Wooldridge - 1995, McCarthy - 1978/. Takšno pripisovanje lastnosti je uporabno, kadar pomaga pri razumevanju zgradbe stroja, njegovega obnašanja v preteklosti in prihodnosti, ali pa če pomaga pri njegovih izboljšavah. Namenski ali usmerjeni pojmi so abstrakcijska orodja, ki nam ponujajo primeren in vsakdanji način za opisovanje, razlago in napovedovanje obnašanja v kompleksnih sistemih. Agent je sistem, ki ga najbolj primerno opišemo z njegovo usmerjeno držo. V ta namen opredelimo dve kategoriji vedênja: *informacijsko vedênje* (mišljenje in znanje) in *usmerjeno vedênje* (želje, namen, dolžnosti, zavezanost).

Formalno osnovo za obravnavanje usmerjenega vedęnja agentov predstavlja modalna logika in njena razširitev v večmodalno logiko. V navadni izjavni ali predikatni logiki so izjave v kateremkoli modelu ali resnične ali neresnične. V naravnem okolju pa ločimo različne vrste ali oblike resnic, kot na primer: "*biti nujno res*"; "*vedeti, da je res*"; "*verjeti, da je res*" in "*bo res v prihodnosti*". Takšne situacije opisujemo s pomočjo modalne logike, kjer imamo poleg operatorjev navadne izjavne logike še operator nujnosti ($\Box$ - škatla) in slučajnosti ($\Diamond$ - karo). Izjave v osnovni modalni logiki so definirane v naslednji Backus Naurjevi obliki (BNF):

$$\varphi \equiv \bot \mid T \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid \Box\varphi \mid \Diamond\varphi.$$

Med operatorjema $\Box$ in $\Diamond$ velja naslednja relacija: $\Box\varphi \leftrightarrow \neg \Diamond\neg\varphi$. To preberemo takole: *'Škatla φ je isto kot ne karo ne φ'.* (nujno res je enako kot ni mogoče, da ni res). Ostale izjavne oblike so dovolj jasne in jih ne bomo posebej razlagali. Primer iz realnega sveta bi lahko na osnovi zgornjega opisa izbrali takole: *'Nujno je da bo jutri deževalo.'* $\leftrightarrow$ *'Ni res, da je mogoče, da jutri ne bo deževalo.'* Modalna logika na drugačen način izraža lastnosti sveta. Razvili so jo filozofi, ki jih je zanimala razlika med *nujno resnico* in *slučajno resnico.* Ključna oseba pri snovanju modalne logike je bil C. I. Lewis /Hajdinjak - 2004/, idejo uporabljati modalno logiko za sklepanje o znanju pa je uvedel Jaako Hintikka /Hintikka - 1975/. *Nujna resnica* predstavlja nekaj, kar ne more biti drugače, *slučajna resnica* pa je nekaj, kar bi verjetno lahko bilo drugače. V modalni logiki je zelo pomembna izjava **K**: $\Box(\varphi \rightarrow \psi) \wedge \Box\varphi \rightarrow \Box\psi$. Včasih jo zapišemo tudi na ekvivalenten način v obliki $\Box(\varphi \rightarrow \psi) \rightarrow \Box\varphi \rightarrow \Box\psi$. Ta izjava se v večini knjig imenuje **K** ali Kripkejev aksiom, in sicer v čast logika S. Kripkeja, ki je uvedel tako imenovano semantiko možnih svetov. Pomembne so še naslednje izjavne oblike **T**: $\Box\varphi \rightarrow \varphi$, kar je refleksija, **D**: $\Box\varphi \rightarrow \Diamond\varphi$, čemur pravimo serija, **4**: $\Box\varphi \rightarrow \Box\Box\varphi$, kar imenujemo tranzitivnost in **5**: $\Diamond\varphi \rightarrow \Box\Diamond\varphi$, ki nakazuje evklidnost. Poznamo različne logične modele, kjer velja različna kombinacija teh izjavnih oblik. Pri obravnavi teorije o razumnih agentih sta pomembna dva modela, in sicer KD45 ali logika prepričanja in KT45 ali epistemska logika ali logika znanja. V epistemski logiki običajno uporabljamo namesto operatorja $\Box$ operator K oziroma K_i. Izjavna oblika $K_i\varphi$ pomeni, da agent i ve, da φ velja. Operator slučajnosti $\Diamond$ pa zapišemo kot $\neg K_i \neg$ ali včasih tudi M_i.

Kombinacija prvostopenjske izjavne logike in modalne logike z večjim številom modalnih operatorjev sestavlja večmodalno logiko, s pomočjo katere lahko obravnavamo razumne agente. V ta namen so bile vpeljane različne večmodalne logike. Najbolj znani sta logiki KARO, ki poleg prepričanja uporablja še znanje /van der Hoek - 2002/ in logika LORA - logika o razumnih agentih, ki je zelo lepo razložena v /Wooldridge - 1998/. V LORI lahko na primer zapišemo $\exists i \cdot$ (Bel *i Sončno(ljubljana)),* kar pomeni, da obstaja agent i, ki je prepričan, da je v Ljubljani sončno vreme. Podobno lahko zapišemo, da je *vsakdo* prepričan, da je v Ljubljani sončno: $\forall i \cdot$ (Bel *i Sončno(ljubljana)).* Poleg operatorja *Bel* za prepričanje uporabljamo v LORI še operatorja *Des* za željo in *Int* za namen. LORO sestavljata poleg prvostopenjske komponente in BDI-komponente še časovna komponenta in komponenta dejanj. Zapišemo lahko, recimo, tudi izjavo: *Int i* $\varphi \Rightarrow$ *Bel i* $\Diamond\varphi$ (če *i* namerava φ, potem je *i* prepričan, da je φ mogoč), ki ustreza značilnemu odnosu med nameni in željami.

3. Razumni agenti in BDI-arhitektura

Pripisovanje usmerjenih pojmov, kot so mišljenje, prosta volja, nameni, zavedanje, sposobnost ali hotenje, stroju je torej legitimno, če to izraža enake informacije o stroju, kot jih izraža o človeku. Te lastnosti pripišemo stroju kot razumnemu agentu, ki deluje na osnovi BDI-modela, to je modela "*prepričanj, želja in namenov*". Obnašanje človeka vsekakor najbolje modelira *tehnika razumnih agentov*, ki s pomočjo logične formalizacije prepričanj, želja in namenov tehniko usmerjanja k cilju izboljša. V /Laza - 2003/ so obravnavani preudarni agenti, ki se morajo odzivati na dogodke. Ti se dogajajo v okolju. Prevzemati morajo pobudo glede na svoje cilje, sodelovati morajo z drugimi agenti (ali ljudmi) in uporabljati predhodne izkušnje za doseganje zastavljenih ciljev. Inteligentni razumni agenti so obravnavani v /Pokhar - 2005/ kot primer modeliranja sveta s pomočjo agentov, ki premorejo- lastna miselna stanja. Agent deluje razumno. S tem doseže svoje cilje, če le ima znanje o svetu in sposobnost, da izvaja planirana dejanja. Wooldridge meni, da si lahko agente predstavljamo kot sisteme, ki so nameščeni in vključeni v neko okolje - agenti niso sistemi, ki ne bi imeli okolja /Wooldridge - 2000/. Agenti torej naj zaznavajo svoje okolje in imajo nabor možnih

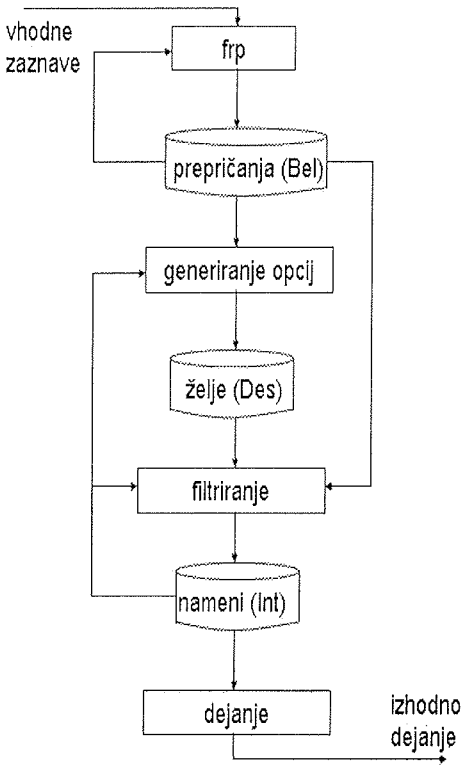
dejanj, katera lahko izvajajo tako, da vplivajo na okolje (slika 1). Razumni agent izvaja dejanja, ki na osnovi njegovega znanja o okolju, v katerem deluje, povečujejo njegove možnosti za uspeh. Dejanja razumnega agenta so odvisna od agentovih predhodnih izkušenj, agentovih informacijah o okolju, dejanj, ki jih ima agent na razpolago, in ocene koristi in možnosti za uspeh teh dejanj.

3. 1. BDI-model

Georgeff pravi, da postaja BDI-model verjetno najbolj znan in preučevan model razumnega agenta s praktičnim razmišljanjem /Georgeff - 1999/. Obstaja veliko število razlogov za ta uspeh, vendar je mogoče najbolj pomembno, da ta model kombinira ugledni filozofski model praktičnega razmišljanja, ki ga je razvil Bratman /Bratman - 1987/, številne izvedbe BDI-modela kot so originalna Bratmanova arhitektura IRMA (Intelligent Resource-bounded Machine Architecture) /Bratman - 1988/ in najbolj znana izvedba BDI-modela v sistemu PRS (PRS - Procedural Reasoning Machine) /Georgeff - 1987/) ter več uspešnih aplikacij, vključno s slavno diagnozo napak vesoljskega taksija in sistem zračnega upravljanja OASIS /Rao - 1995/, in končno elegantno abstraktno logično semantiko - teorijo, ki jo zelo strogo formalizira družina BDI-logik. Na kratko bomo opisali generični BDI-model in enostavno izvedbo BDI-agenta - njegovo programsko komponento.

3. 2. Generični BDI-model

Generični BDI-model je narisan na sliki 2.



Slika 2: Generični BDI-model.

V nadaljevanju predstavlja zapis $\gamma(T)$ moč množice tipa *T*, *x* pa predstavlja kartezijski produkt. Določimo lahko glavne komponente agentove nadzorne zanke v generičnem modelu s slike 2. Proces, s katerim agent posodablja prepričanje, je formalno modeliran s *funkcijo revizije prepričanja - frp* : $\gamma(Bel) \times zaznave \rightarrow \gamma(Bel)$. Trenutno prepričanje in trenutne zaznave določajo novo množico prepričanj. Funkcijo *preudarjanja* agenta razdelimo na dve ločeni funkcionalni komponenti: *generiranje opcij* - pri tem agent generira množico možnih alternativ, in *filtriranje* - pri tem agent izbira med kompetitivnimi alternativami. Formalno -funkcijo za generiranja *opcij* zapišemo kot: *opcije:* $\gamma(Bel) \times \gamma(Int) \rightarrow \gamma(Des)$. Da bi agent izbiral konkurenčne opcije, uporablja funkcijo *filtriranja,* ki jo zapišemo takole: *filter :* $\gamma(Bel) \times \gamma(Des) \times \gamma(Int) \rightarrow \gamma(Int)$. Dejansko je to proces preudarjanja, ki ga izvaja agent. Agentovo razmišljanje o sredstvih za doseganje ciljev podaja planska funkcija: *plan:* $\gamma(Bel) \times \gamma(Int)$ *-> Plan.* Pri izvedbi agenta moramo zagotoviti pravočasno posodabljanje prepričanja, ponovnega tehtanja opcij in ponovnega planiranja.

3. 3. Izvedba razumnega agenta

Na osnovi razlage iz podpoglavja 3.2 lahko opredelimo tako imenovano *programsko komponento* razumnega agenta BDI, oziroma način, kako lahko agenta zgradimo. Na sliki 3 vidimo osnovno nadzorno zanko agenta. Agent neprenehoma izvaja postopek, v katerem opazuje svet, se odloča, kateri bo naslednji namen, ki ga bo poskušal udejanjiti. Določi neke vrste plan za doseganje postavljenega cilja in potem ta plan izvaja.

1. **while true do**
2. agent opazuje svet;
3. agent posodablja notranji model sveta;
4. agent preudarja o tem, katere namene naj udejanji v naslednjem koraku;
5. agent razmišlja o sredstvih za doseganje ciljev in zasnuje načrt za udejanjanje namena;
6. agent izvaja načrt;
7. **end while;**

Slika 3: Splošna nadzorna zanka agenta.

Osnovno izvajalno zanko s slike 3 lahko razgradimo na detajlnejše agentove posege, s čimer pridemo do mnogo realnejšega algoritma. Prikazuje ga slika 4.

1. B := B_0; /* B_0 so začetna prepričanja */
2. I := I_0; /* I_0 so začetni nameni */
3. **while true do**
4. pridobi naslednjo zaznavo ρ;
5. B := frp(B, ρ);
6. D := opcija (B,I);
7. I := filter(B, D, I);
8. π := plan(B,I);
9. **while not** (prazen(π) **or** uspešen(I, B) **or** nemogoč(I, B)) **do**

```
10.        α := glava (π);
11.        izvaja(α);
12.        π := konec(π);
13.        pridobi naslednjo zaznavo ρ;
14.        B := frp(B, ρ);
15.        if pretehta(I,B) then
16.              D := opcije(B,I);
17.              I := filter(B, D, I);
18.        end-if
19.        if not smiseln(π, I, B) then
20.              π := plan(B, I)
21.        end-if
22.    end-while
23. end-while
```

Pomen posameznih oznak:

- *B* je spremenljivka, ki vsebuje agentovo trenutno prepričanje.
- *D* je spremenljivka, ki vsebuje agentovo trenutno željo.
- *I* je spremenljivka, ki vsebuje agentov trenutni namen.
- *ρ, ρl, ...* predstavljajo zaznave.
- *Plan* je množica, ki jo sestavljajo predpogoji in izhodni pogoji.
- Telo plana je dejanski recept za izvedbo plana.
- Če je π plan, potem lahko z njim opravimo naslednje operacije:
- *pred(π), izhod(π), telo(π),*
- *prazen(π), izvaja(π),*
- *glava(π), konec(π),*
- *smiseln(π, I, B).*
- *frp* – funkcija revizije prepričanja.

Slika 4: Nadzorna zgradba previdnega in smotrno zavezanega agenta.

V zunanji izvajalni zanki agent zaznava okolje in posodablja prepričanje, generira želje oziroma cilje in izbira konkurenčne opcije s pomočjo funkcije *filter*. Iz prepričanja generira plan za doseganje namenov. Pogoji za izvajanje notranje zanke so trije: izvaja se, dokler ni plan prazen, uspešen ali nemogoč. Ti pogoji določajo previdnega in smotrno zavezanega agenta. V notranji zanki agent ponovno posodablja prepričanje, pretehta opcije in ponovno planira. Te posege mora opravljati v ustreznih, pravočasnih časovnih intervalih. Na tem modelu in algoritmu temelji razvojno orodje jadex za agentne sisteme, ki ga opisujemo v naslednjem poglavju.

4. Orodja za načrtovanje agentov in agentih skupnosti

4. 1. Metodologije za agentno orientirano načrtovanje programske opreme

S pomočjo agentno orientiranega pristopa lahko načrtujemo fleksibilne sisteme z zamotanim in s popolnim vedênjem. Načrtovanje večagentih sistemov vključuje zamotane postopke in več nivojev načrtovanja. Danes so na voljo agentno orientirana razvojna orodja, ki omogočajo, da si to tehnologijo prisvaja tudi industrija. Rezultati so obetavni in to usmerja tudi nas pri študiju agentne tehnologije kot alternative poti za načrtovanje inteligentnih naprav – razumnih hišnih pomočnikov. Združenje AOSE (Agent Oriented Software Engineering) – agentno orientirano načrtovanje programske opreme – združuje dejavnosti s področja metodologij načrtovanja programske opreme, razvojnih orodij in programskih jezikov za področje agentno orientiranih sistemov. Posvetovanja AAMAS (Autonomous Agents and Multiagent Systems) obravnavajo zelo široka področje agentih sistemov.

Običajno predstavljajo predlagane metodologije razširitev obstoječih objektno orientiranih metodologij, ki vključujejo koncepte agentnega načrtovanja /Massonet - 2002/. Vse te metodologije imajo naslednje skupne pristope in postopke za načrtovanje večagentnega okolja, kakor jih navajamo v nadaljevanju.

Organizacija: Organizacija agente skupnosti predstavlja skupnost agentov, ki sodelujejo za skupne namene. To je virtualna zgradba, ker nima vgrajenega posebnega nadzornega sistema, ki bi tej zgradbi ustrezal. Storitve zagotavlja in skupaj dosega skupnost agentov, ki organizacijo sestavljajo. Njena zgradba se izraža skozi moč povezav med sestavnimi deli in pa vedęnjskimi in koordinacijskimi mehanizmi sodelovanja med njimi.

Vloga: Vloga določa zunanje značilnosti agenta v določenem kontekstu njegove obravnave. Agent lahko igra večje število vlog, pa tudi večje število agentov lahko igra isto vlogo.

Cilji: Cilje lahko enačimo z željami agenta. Za vsakega agenta določimo njegove cilje in delne cilje.

Vhodne zaznave in izhodna dejanja: Vhodne zaznave in izhodna dejanja predstavljajo agentov stik s svetom oziroma okoljem. Stanje sveta zaznavajo vhodne zaznave, z njimi agent posodablja svoje znanje in prepričanje. Izhodna dejanja temeljijo na nivoju agentovega znanja in prepričanja; z njimi agent deluje na okolje. Zaznave in dejanja se izvajajo kot plani agenta.

Sodelovanje z drugimi agenti: Agenti v večagentnem sistemu sodelujejo med sabo tako, da si izmenjujejo sporočila. Uporabljamo dogovorjene protokole. V našem primeru uporabimo platformo JADE /Bellifemine - 1999/, ki izkorišča komunikacijski standard FIPA.

V literaturi so obravnavane predvsem naslednje metodologije za načrtovanje agentov in agentih sistemov, ki združujejo zgornje pristope: Prometheus, MaSE, Tropos, AUML, GAIA /Shehory - 2005/.

4. 2. Programska orodja za programiranje agentih sistemov

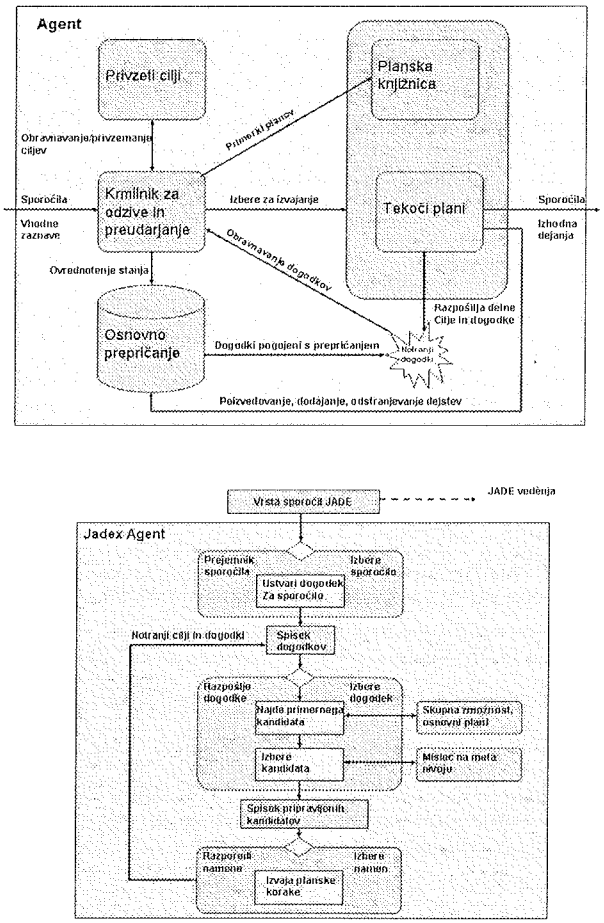
Obstaja vrsta orodij za razvoj agentno orientiranih sistemov, od razvojnih okolij do programskih jezikov /Dastani - 2005/.

Naštejmo jih nekaj, medtem ko nekoliko natančneje opisujemo JADE in jadex, ki ju kasneje uporabimo za izvedbo agentne skupnosti "večagenti pomočnik pranja": programski jeziki 3APL, AgentSpeak, JASON in programska okolja JACK, JADE, jadex.

4. 3. JADE in jadex

Jadex označujemo tudi kot BDI-stroj za razmišljanje /Dastani - 2005/. BDI-model je zelo primeren za opisovanje agentovih mišljenjskih stanj /Braubach - 2004/. Želje (cilji) agenta predstavljajo njegovo motivacijsko držo in so glavni vir agentovih dejanj. Predstavitev ciljev in rokovanje z njimi igra torej osrednjo vlogo pri ciljno orientiranih analizah in tehnikah modeliranja, kot je jadex. Jadex je vgrajen (ali nadgrajen) na platformo JADE. Zasnovan je torej na BDI-modelu in integrira agentne teorije, ki so objektno orientirane in opisane z jezikom XML. Eksplicitna predstavitev ciljev dovoljuje razmišljanje in upravljanje s cilji. Platforma JADE /Bellifemine - 1999/ se osredotoči na vpeljavo referenčnega modela FIPA, ki predstavlja zahtevano komunikacijsko infrastrukturo in platformo za storitve, kot je na primer upravljanje z agenti, ter množico razvojnih in testnih orodij. Namenoma pušča odprte številne stvari, ki se tičejo notranjega koncepta agenta, in s tem nudi enostavni, opravilno zasnovani model, v okviru katerega lahko razvijalec zasnuje kakršnokoli obnašanje agenta. Zaradi tega je zelo primeren za izvedbo stroja za razmišljanje. Medtem ko je JADE agentna platforma, pripravljena za stik z zunanjostjo, ki vključuje komunikacijo in agentovo upravljanje, lahko stroj za razmišljanje obvladuje notranjost agenta. Obnašanje razumnega agenta, zasnovanega na BDI-modelu, določajo torej prepričanja, plani in cilji. Te tri komponente so združene v skupnost, ta pa opredeljuje skupne sposobnosti sistema, ki jih lahko potem modularno izkoristimo. Abstraktna jadexova arhitektura ter njegov abstraktni model sta narisana na sliki 5. Jadex temelji na generičnem BDI-modelu razumnega agenta, ki smo ga opisali v poglavju 3. Navzven predstavlja jadex črno škatlo, ki sprejema in oddaja sporočila. Na ta način zaznava okolje in vanj posega s svojimi dejanji. Vsakemu sporočilu sledi notranji dogodek, ki ga prestreže in obravnava krmilnik dogodkov. Ta skrbi za odzivno in preudarno vrednotenje dogodkov. Nove dogodke lahko prožijo tudi cilji in posebni pogoji notranjega prepričanja agentov. Krmilnik za odzive in preudarjanje izbira na osnovi ovrednotenih dogodkov plane, ki se izvajajo. Ti lahko dosegajo in posodabljajo osnovno prepričanje, pošiljajo sporočila drugim agentom, kreirajo nove cilje ali delne cilje in prožijo notranje dogodke. Mehanizem odzivanja in preudarjanja je v splošnem enak za vse agente. Vedênje posameznega agenta torej določajo samo njegova prepričanja, cilji in plani, ki si jih v nadaljevanju poglejmo malo podrobneje.

Prepričanja: Prepričanje odseva osvojena znanja in je shranjeno, da je dosegljivo vsem planom. Agenti lahko poizvedujejo po prepričanjih. Jadex ne vsiljuje logične predstavitve prepričanja. Namesto tega lahko uporabimo običajne javine objekte, ki opisujejo temeljna prepričanja. Objekti torej shranjujejo dejstva ali množice dejstev, ta pa so opisana z dogovorjenimi podatkovnimi strukturami. Podatke, ki ustrezajo prepričanjem, lahko posodabljamo pod vplivom novih dejstev, tako da jih spreminjamo, dodamo ali odstranimo. Po njih lahko poizvedujemo s pomočjo objektnega jezika za oblikovanje poizvedb (OQL - Object Query Language). Prepričanja lahko uporabimo kot vhodne podatke za misleči stroj tako, da določimo stanja prepričanj kot predpogoje za plane ali kot pogoje za dosego ciljev. Stroj nadzoruje spremembe prepričanj in samodejno prireja cilje in plane.



Slika 5: Jadexova abstraktna arhitektura in njegov izvajalni model.

Cilji: Jadex zasleduje splošno idejo, ki pravi, da so cilji dejanske in trenutne želje agenta. Kateremukoli cilju, ki ga ima, agent neposredno priredi primerna dejanja. Cilju sledi, dokler je agent aktiven ali pa se deaktivira oziroma ga zastavljeni cilj ne zanima več. Za razliko od večine drugih sistemov jadex ne predpostavlja, da morajo biti cilji med sabo skladni. Da lahko razlikujemo med ravnokar privzetimi cilji (to je želenimi) in pa zasledovanimi cilji, uvajamo življenjsko dobo cilja, ki jo sestavljajo stanja cilja: *opcija, aktiven* in *začasno odložen.* Kadar postane cilj privzet, postane opcija, ki se doda agentovi zgradbi, želja. Mehanizem preudarjanja je odgovoren za prehode stanj na poti do vseh privzetih ciljev.

Plani: Miselni stroj upravlja vse dogodke, kot so sprejem sporočil ali aktiviranje ciljev, tako da izvaja primerne plane. Jadex uporablja plansko knjižnico, ki vsebuje agentove plane. Vsak plan ima glavo, ki določa okoliščine, pod katerimi se plan izbere, in telo plana, ki določa dejanja za izvajanje. Najpomembnejši del glave plana so cilji in/ali dogodki, ki upravljajo plan. V planih je določeno, katera agentova izhodna dejanja je treba izvajati in katere vhodne vrednosti je treba zaznavati.

5. Razumni hišni pomočnik kot večagentni sistem

Razumni agent zna izbirati najboljša možna dejanja. Porazdeljene in povezane inteligentne naprave lahko pri tem sodelujejo in skupaj izbirajo dejanja, ki najbolje vodijo do skupne koristi. Pokazati želimo, da je interoperabilnost ne samo koristna, ampak že kar nujna pri vodenju takega porazdeljenega sistema. Pomembni pogoj za interoperabilnost naprav je združljivost, ki je dosežena s standardiziranimi pristopi pri načrtovanju. V tem poglavju želimo torej predstaviti možno izvedbo razumnega agenta s pomočjo razvojnega orodja JADE, ki zagotavlja interoperabilnost porazdeljenih naprav kot agentnih sistemov. Jadex, ki je stroj za razmišljanje, pa prispeva nivo, ki je nameščen nad hrbtenično infrastrukturo JADE in ponuja načrtovanje agentov, zasnovanih na BDI-modelu. Omogoča načrtovanje ciljno orientiranih agentov, ki z mehanizmi preudarjanja izbirajo najboljša možna dejanja in se torej odločajo razumno. S takšno platformo lahko načrtujemo inteligentne sisteme. Orodje dodatno omogoča še, da programiramo v javi. S tem imamo v rokah pomembne razvojne načrtovalske mehanizme za razširitve naših agentov. S pomočjo takšne platforme bomo predstavili enostavni primer razumnega hišnega pomočnika pranja, zgrajenega v obliki agentne skupnosti. Programiranje agentnih sistemov ni preprosto opravilo, saj moramo načrtovati številne usklajene sisteme (agente), njihovo miselno zgradbo - prepričanja in cilje ter njihove medsebojne odnose (objektni poizvedovalni jezik), njihova dejanja - plane (java), sodelovanje in komunikacijo z drugimi agenti (FIPA, JADE). Želimo torej prikazati možno izvedbo razumnega agenta s pomočjo BDI-arhitekture in večagenti sistem kot agentno skupnost v realnem okolju hišne naprave. V našem primeru je to pralni stroj, ki ga upravlja razumni pomočnik pranja (RPP), vgrajen v okolje pralnega stroja. Na osnovi dogovora -dialoga z uporabnikom poskrbi za pranje perila v skladu z uporabnikovimi željami in v skladu z navodili za pranje izbranega in vloženega perila. RPP zna iz vgrajene planske knjižnice izbrati najboljši možni plan za pranje vloženega perila, pri čemer upošteva uporabnikove želje tako, da doseže zastavljene cilje na najboljši možni način.

5. 1. Opredelitev lastnosti razumnega pomočnika pranja kot večagentnega sistema

Odločili smo se torej, da izdelamo RPP kot posebni primerek razumnega hišnega pomočnika. Vendar pa je RPP samo eden izmed agentov, ki sestavljajo večagentnega pomočnika pranja (VAPP). RPP je nadzorni agent z najvišjo stopnjo *samostojnosti* in dejansko določa obnašanje celotnega sistema. Njegovi pomočniki, ki se obnašajo kot *nadzorovani agenti*, opravljajo določena pomožna opravila (opredeljujemo jih v naslednjem podpoglavju) in imajo različne stopnje samostojnosti. Imenujemo jih lahko tudi agenti delavci, saj so zadolženi za čisto rutinska opravila. RPP zna planirati sam zase in za nadzorovane agente. Nadzorovani agenti imajo določene plane, ki jih sami zase ne znajo posodobiti. Okolje VAPP naseljujejo naslednji agenti: *RPP, Agent_Termostat, Agent_Za_Vodo, Agent_Za_Meritev_Čiste_vode, Agent_Za_Prašek_ Mehčalo, Agent_Za_Vrata_Stroja, Agent_Za_Motor, Agent_Za_Vlaganje_Perila, Agent_GUI.* Njihove lastnosti opredeljujemo v naslednjem podpoglavju. Lastnosti celotnega okolja VAPP in posameznih agentov v agentni skupnosti lahko opišemo izhajajoč iz lastnosti agentov v večagentnem okolju, kakor smo jih navedli v poglavju 1. Okolje VAPP in njegov nadzorni agent RPP se morata obnašati *dobronamerno* in *verodostojno*, kar pomeni, da RPP uporabniku ne bo dovolil izbrati parametrov pranja izven priporočenega obsega in bo v primeru, da eden izmed članov skupnosti odpove, še vedno optimalno opravil svoje delo, če bo to mogoče. Sicer pa bo zaključil pranje na najboljši možni način. Agent se mora znajti tudi v nepredvidenih situacijah in poiskati razumne rešitve. Glede na postavljene želje se bo obnašal *razumno*, kar pomeni, da bo izbral najbolj obetavna dejanja z namenom, da doseže postavljene cilje. Vedno bo znal poiskati razumne rešitve. Glede na uporabnikove želje bo RPP posodabljal svoje prepričanje in se s tem *učil*. RPP bo znal na osnovi dogovora z uporabnikom *samostojno* opraviti svoje naloge in doseči končno stanje, ki pomeni kvalitetno in po željah uporabnika oprano perilo. RPP se mora do drugih agentov v svojem okolju znati obnašati *socialno*, kar dosežemo s komunikacijo po predpisih agentnega komunikacijskega protokola FIPA. Lahko tudi rečemo, da RPP deluje *usmerjeno*, saj zasleduje samo en cilj, to je kvalitetno oprano perilo. *Agent_GUI* in *Agent_Za_Vlaganje_Perila* sta delno samostojna, delno pa tudi nadzorovana agenta te skupnosti. Usmerja ju sicer RPP, vendar pa se dogovarjata z uporabnikom oziroma spremljata proces vlaganja perila samostojno in se pri tem tudi učita.

VAPP, ki ga upravlja RPP, ima vse lastnosti inteligentnega agentnega sistema. Takšen agentni sistem lahko realiziramo z enotno ali na s porazdeljeno strojno opremo.

5. 2. Organizacijska zgradba večagentnega pomočnika pranja

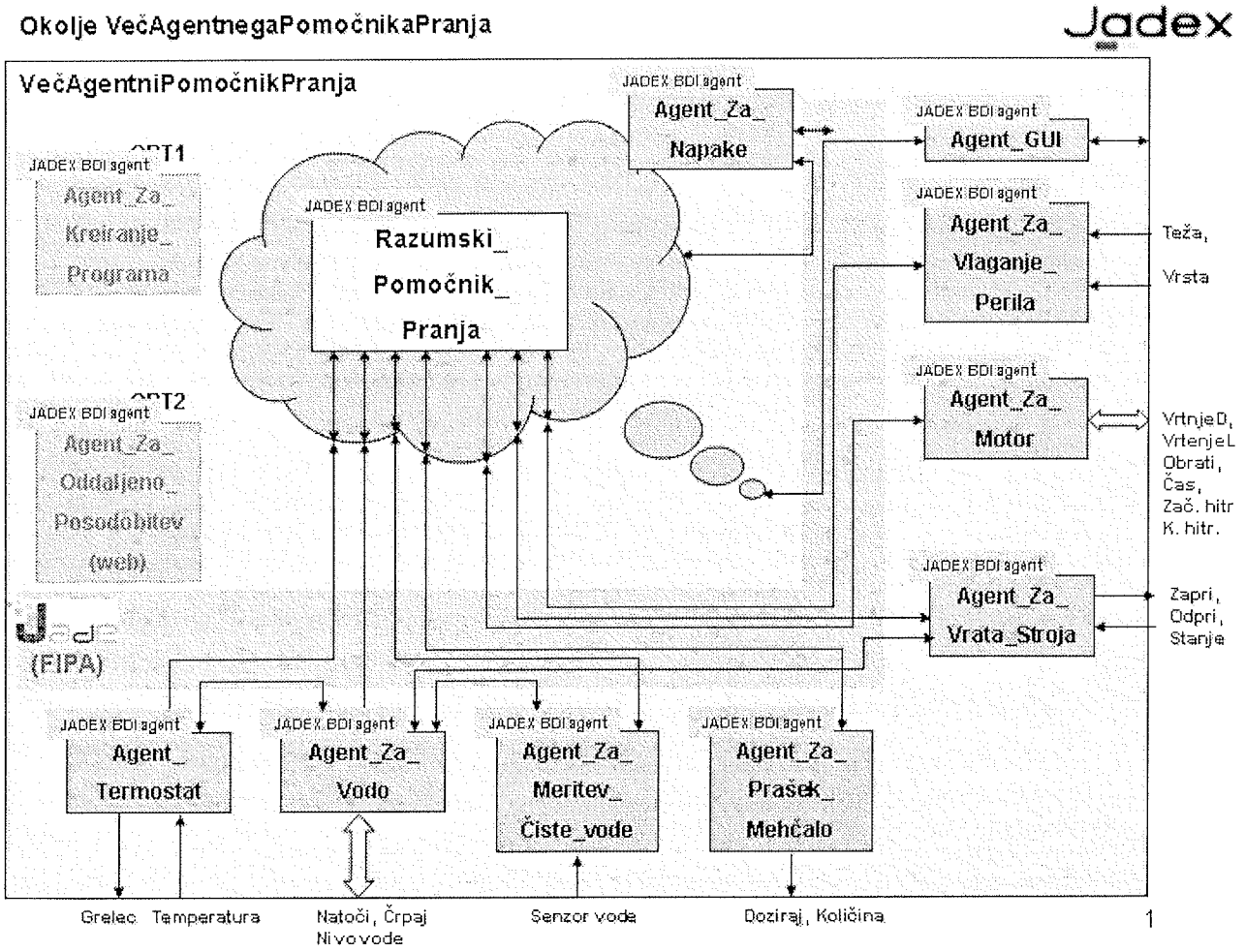
Zgradba oziroma organizacija VAPP je prikazana na sliki 6. Njegovo jedro predstavlja RPP, ki smiselno usmerja delovanje našega hišnega pomočnika, in pa večje število agentov, ki v procesu pranja samostojno izvajajo posamezna opravila pod nadzorom RPP. Gre za delna opravila pri pranju: *Agent_Termostat* je agentni podsistem za uravnavanje temperature vode v bobnu, *Agent_Za_Vodo* skrbi za pris-

otnost določene količine vode v bobnu, *Agent_Za_Meritev_Čiste_vode* meri nivo motnosti vode med pranjem, *Agent_Za_Prašek_Mehčalo* dozira potrebno količino praška ali mehčala, *Agent_Za_Vrata_Stroja* zaklene vrata bobna, ko se poteka pranje, *Agent_Za_Motor* vrti boben, *Agent_Za_Vlaganje_Perila* spremlja vlaganje perila, *Agent_GUI* pa omogoča pogovor med RPP in uporabnikom. Vsi ti nadzorovani agenti komunicirajo z RPP s pomočjo platforme JADE, dogovarjati pa se znajo tudi med sabo. RPP torej udejanja svoj plan in vpliva na okolje s posredovanjem nadzorovanih agentov. Ti so povezani s fizičnim svetom in z drugimi agenti, kar pomeni, da zaznavajo fizični svet pralnega stroja in izvajajo dejanja. Agent termostat na primer zaznava temperaturo vode in s pomočjo grelca vzdržuje zahtevano temperaturo. S prepričanjem nadzorovanih agentov lahko delno upravlja RPP, saj se sami običajno ne učijo, ker opravljajo "rutinska dela", ki so imajo v naprej določen plan. Poleg RPP imata samo *Agent_Za_Vlaganje_Perila* in *Agent_GUI* lastno prepričanje, ki si ga lahko sama posodabljata in se tako učita.

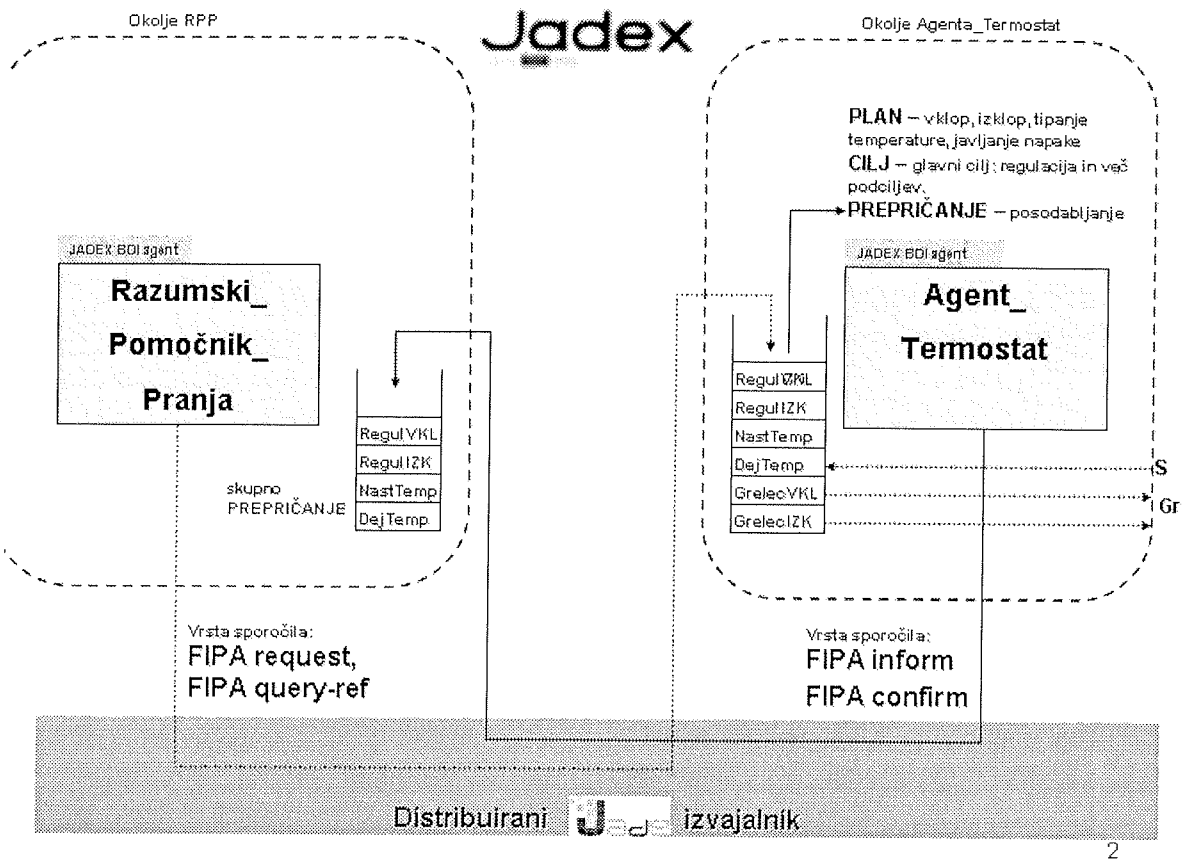
Na sliki 7 je narisana povezava med RPP ter agentom *Agent_Termostat*. Agenta RPP in *Agent_Termostat* sta med sabo povezana preko porazdeljenega izvajalnika JADE. RPP lahko pošlje v sistem za ogrevanje vode dve vrsti FIPA-sporočil. Eno je tipa "FIPA request", s katerim RPP posreduje ukaze, na primer 'vklopiinsegrej 34'. Ta ukaz vklopi proces segrevanja vode in regulacijo na zahtevani temperaturi. Drugi ukaz je tipa "FIPA query-ref"; ta pošlje poizvedbo o stanju naslovljenega agenta. *Agent_Termostat* lahko odgovori na dva načina: z informacijskim sporočilom tipa "FIPA inform" ali s potrditvijo odziva na ukaz, tj. s sporočilom tipa "FIPA confirm".

5. 2. Načrtovanje agenta *Agent_Termostat*

Agent_Termostat predstavlja sistem za ogrevanje vode v VAPP in je poseben primerek nadzorovanega agenta. To ni enostavni odzivni agent, ki bi skrbel samo za vklop in izklop gretja vode, ampak tvori podsistem za ogrevanje vode v napravi - pralnem stroju. Opis postopkov načrtovanja ostalih agentov, med njimi tudi RPP, zaradi obsežnosti tukaj izpuščamo. V skladu z metodologijo načrtovanja agentne programske opreme Prometheus določimo najprej vlogo agenta *Agent_Termostat*, ki jo igra v okviru agentne skupnosti hišnega pomočnika, ki ga nadzira RPP. Ta vloga je zapisana v tabeli 1.



Slika 6: Organizacija večagentnega pomočnika pranja.



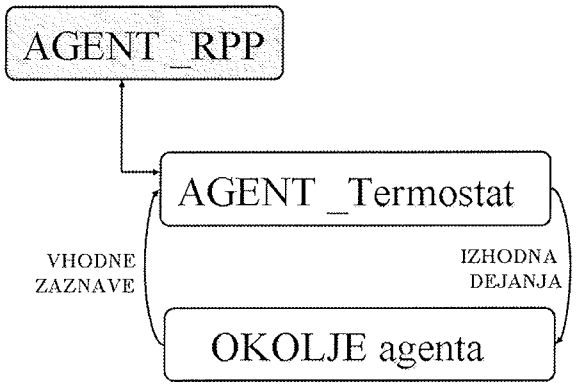
Slika 7: Zasnova agenta za segrevanje vode – Agent_Termostat.

Tabela 1: Vloga agenta Agent_Termostat.

***Agent_Termostat* zaznava okolico:**	
- Zaznava temperaturo vode	Temperaturni senzor meri temperature. Strojna oprema pretvori temperaturo v obliko, ki jo *Agent_Termostat* lahko izmeri.
- Poizve za nivo vode	Dejansko *Agent_Za_Vodo* posreduje informacij o o nivoju vode preko sistema JADE.
- Zaznava moč grelca	Zaznava delovanje grelca – moč na grelcu.
- Sprejema ukaze od RPP	Vklop, izklop, nastavljena temperatura.
***Agent_Termostat* vpliva na okolico:**	
- Vklopi ali izklopi grelec	Grelec vklopi, če je temperatura nižja od nastavljene spodnje meje. Izklopi ga, ko je temperature višja od nastavljene zgornje meje.
- Javi napako	Javi napako grelca, nivoja vode ali tipala za temperaturo tako, da obvesti agenta *Agent_Za_Napake,* ta pa vzpostavi stik z zunanjim, posebnim agentom *Agent_Servis*, ki je zadolžen za izpeljavo vzdrževanja dodeljenih mu naprav.
- Javi stanje v RPP	Javlja svoje stanje agentu v RPP.
***Agent_Termostat* izvaja naslednje plane:**	
Vzdržuje stanje "pripravljen".	
Sprejema ukaze od RPP in mu posreduje stanje sistema za ogrevanje.	
Pri *Agentu_Za_Vodo* preveri nivo vode.	
Vklopi grelec in segreje vodo na zahtevano temperaturo.	
Izvaja regulacijo temperature.	
Javi napako agentu *Agent_Za_Napake* in preide v stanje pripravljen.	
Izklopi regulacijo in preide v stanje pripravljen.	

Agent_Termostat je glede na stopnjo njegove avtonomije torej nadzorovani agent. Nadzoruje ga RPP. Na osnovi splošne zgradbe za razumnega agenta iz poglavja 3, slika 3, lahko narišemo njegovo zunanjo simbolično zgradbo na sliki 8.

Sistemu za ogrevanje vode oziroma agentu *Agent_Termostat* določimo cilje in njegovo notranjo zgradbo. To zgradbo kaže slika 9. Povezanost in pomen postavljenih ciljev sta s sliko dovolj pojasnjena, zato ju ne opisujemo posebej.



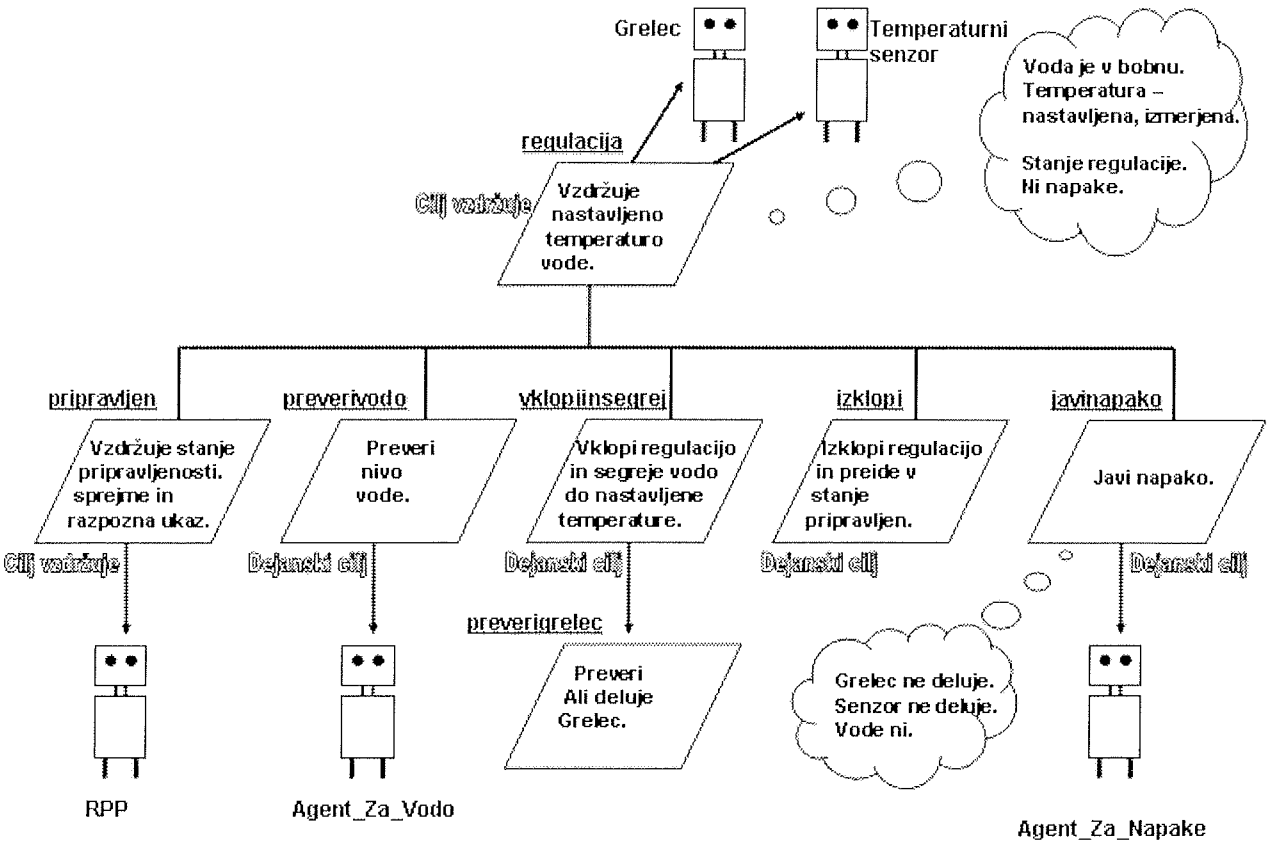
Slika 8: Simbolična zgradba agenta Agent_Termostat.

Izhajajoč iz vloge agenta in njegovih ciljev lahko določimo miselno zgradbo agenta, ki jo opredeljujejo relacije med njegovimi prepričanji, željami in nameni. Ta notranja zgradba, ki skrbi za ogrevanje vode v pralnem stroju, je prirejena razvojnemu orodju za načrtovanje agentov in agentih skupnosti Jadex. Narisana je na sliki 10. Cilji se v jadexu interpretirajo kot želje, plani pa kot nameni agenta. *Agent_Termostat* ima določena naslednja stanja: *Pripravljen, Regulacija, ČakaNaVodo in Napaka.* Agent izvaja plane, ki dosegajo delne cilje ali posodabljajo prepričanje, ki torej vplivajo na njegovo notranjo stanje mišljenja ali na BDI- zgradbo. Plani lahko vključujejo neposredno izvedbo določenih dejanj, lahko pa nanje vplivajo le posredno, tako da postavljajo delne cilje ali posodabljajo prepričanja. Na ta način agent prehaja v nove možne svetove tako da si lahko njegovo delovanje predstavljamo kot obnašanje dinamičnega sistema, katerega stanja se menjajo, kakor določa diagram v obliki drevesa. Posamezno drevo prehajanja stanj pomeni le enega izmed možnih svetov, v katerem se znajde agent. Prestopi med svetovi in s tem tudi iz enega v drugo drevo stanj so odvisni od spreminjanja prepričanj ter prilagajanja želja in poti za njihovo uresničenje.

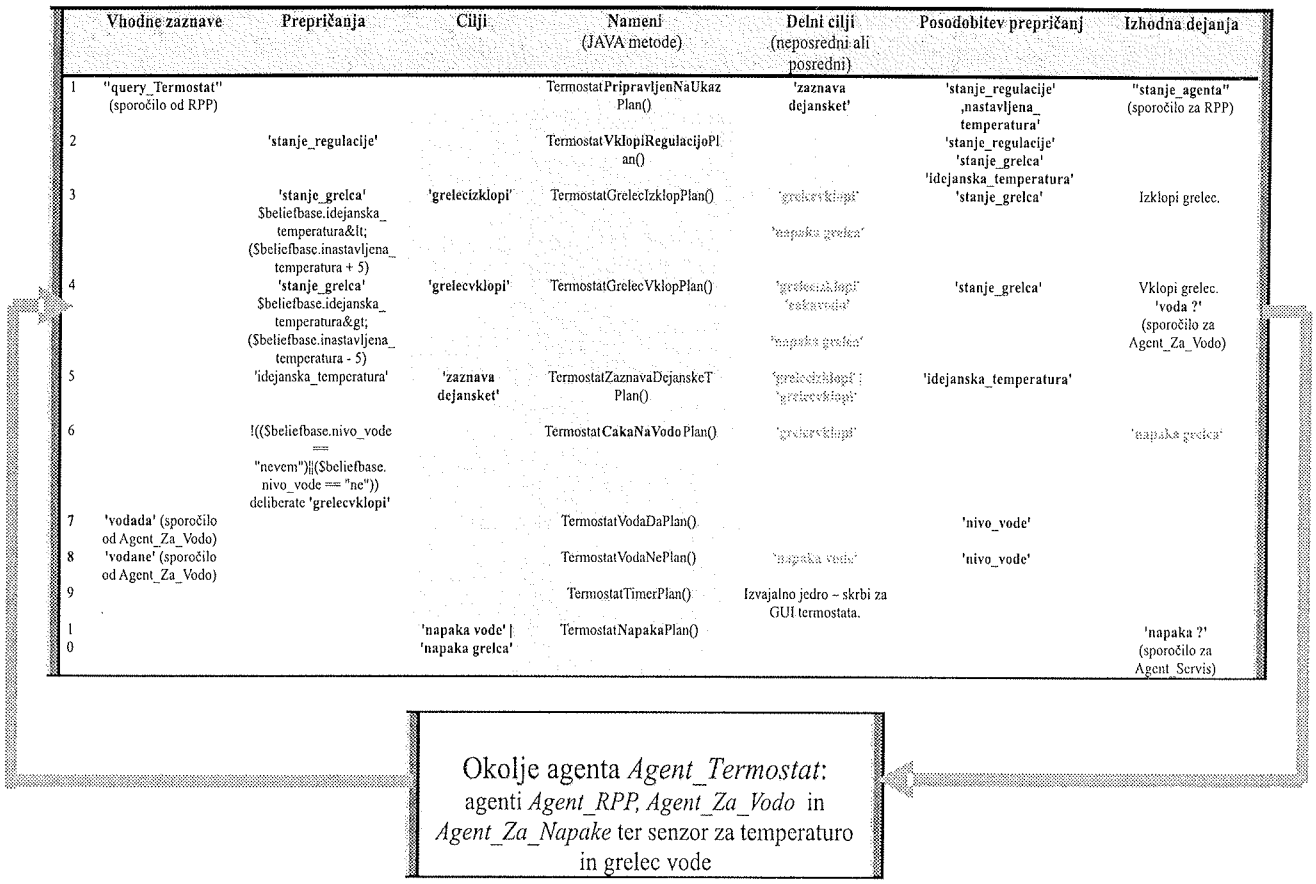
Opišimo nekaj primerov s slike 10:

Zunanja zaznava v prvi vrstici je v obliki sporočila, prejetega od agenta *Agent_RPP*. To sporočilo proži nov plan »Termostat**VklopiRegulacijo**Plan()«, pri čemer je njegova izvedba odvisna od prepričanja '**stanje_regulacije**'. To lahko obravnavamo kot dejanje, ki agenta postavi iz stanja *Pripravljen* v stanje *Regulacija*.

V tretji in četrti vrstici sta pogoja ($T_d < T_n - 5$) oziroma ($T_d > T_n + 5$), ki generirata cilja '**grelecvklopi**' oziroma '**grelecizklopi**' in s tem povzročita dejanje, ki vklopi ali izklopi grelec. Tako namreč poteka regulacija temperature vode



Slika 9: Struktura ciljev agenta Agent_Termostat *in njihova medsebojna povezanost.*



	Vhodne zaznave	Prepričanja	Cilji	Nameni (JAVA metode)	Delni cilji (neposredni ali posredni)	Posodobitev prepričanj	Izhodna dejanja
1	"query_Termostat" (sporočilo od RPP)			TermostatPripravljenNaUkazPlan()	'zaznava dejansket'	'stanje_regulacije' ,nastavljena_ temperatura'	"stanje_agenta" (sporočilo za RPP)
2		'stanje_regulacije'		TermostatVklopiRegulacijoPlan()		'stanje_regulacije' 'stanje_grelca' 'idejanska_temperatura'	
3		'stanje_grelca' $beliefbase.idejanska_ temperatura< ($beliefbase.inastavljena_ temperatura + 5)	'grelecizklopi'	TermostatGrelecIzklopPlan()	'grelecvklopi' 'napaka grelca'	'stanje_grelca'	Izklopi grelec.
4		'stanje_grelca' $beliefbase.idejanska_ temperatura> ($beliefbase.inastavljena_ temperatura - 5)	'grelecvklopi'	TermostatGrelecVklopPlan()	'grelecizklopi' 'cakanavodo' 'napaka grelca'	'stanje_grelca'	Vklopi grelec. 'voda ?' (sporočilo za Agent_Za_Vodo)
5		'idejanska_temperatura'	'zaznava dejansket'	TermostatZaznavaDejanskeTPlan()	'grelecizklopi' 'grelecvklopi'	'idejanska_temperatura'	
6		!(($beliefbase.nivo_vode == "nevem")\|\|($beliefbase. nivo_vode == "ne")) deliberate 'grelecvklopi'		TermostatCakaNaVodoPlan()	'grelecvklopi'		'napaka grelca'
7	'vodada' (sporočilo od Agent_Za_Vodo)			TermostatVodaDaPlan()		'nivo_vode'	
8	'vodane' (sporočilo od Agent_Za_Vodo)			TermostatVodaNePlan()	'napaka vode'	'nivo_vode'	
9				TermostatTimerPlan()	Izvajalno jedro – skrbi za GUI termostata.		
10			'napaka vode' | 'napaka grelca'	TermostatNapakaPlan()			'napaka ?' (sporočilo za Agent_Servis)

Slika 10: Notranja miselna zgradba agenta Agent_Termostat *v okolju VAPP.*

(T_d). Doseči mora nastavljeno temperaturo (T_n) in skrbeti, da se ne spreminja več, kot določa histereza 5°C. Zaporedje ciljev '**grelecvklopi**' in '**grelecizklopi**' se ponavlja, dokler RPP ne pošlje sporočila '**izklopi**'. Agent vzdržuje stanje *Regulacija.*

Zelo zanimiva je tudi situacija v šesti vrstici, kjer prepričanje '**nivo_vode**' = "ne vem" ali "ne" vodi v stanje *CakaNaVodo.* Dokler v bobnu ni vode, agent preudarja, ali naj vklopi grelec ali ne. Dokler ni vode, grelca seveda ne sme vklopiti.

RPP smo programirali v Borlandovem razvojnem okolju Jbuilder, v katerega smo inštalirali JADE in razvojno okolje za agentne sisteme jadex. Na ta način smo dobili zelo učinkovit sistem za načrtovanje in razvoj porazdeljenih večagentnih sistemov z agenti, ki so zgrajeni na osnovi BDI-modela. S preprostim primerom modeliranja inteligentne hišne naprave v obliki agentnega sistema smo želeli pokazati, kako je možno uporabiti perspektivno večagentno tehnologijo tudi pri snovanju porazdeljenih inteligentnih sistemov in povezanih hišnih naprav.

6. Zaključek

Raziskujemo agente in večagentne sisteme kot novo tehnologijo, s pomočjo katere bomo izdelali inteligentne hišne pomočnike. To so lahko agenti, kot je opisani razumni pomočnik pranja, ali pa tudi kuhanja ali hlajenja, ter njihova večagentna skupnost, v kateri sobivajo.

Najprej smo na kratko povzeli lastnosti večagentih sistemov in poudarili, da sta na področju porazdeljenih računalniških sistemov najpomembnejša medsebojna komunikacija in skupno ter porazdeljeno znanje agentov. Nadaljevali smo z obravnavo teorije namenskega ali usmerjenega vedenja in modalne logike ter njenega pomena za načrtovanje (razumnih) agentov in agentih sistemov. Njena logična razširitev je večmodalna logika, ki nam omogoča, da razmišljamo o razumnem agentu, ki temelji na BDI-modelu, to je modelu prepričanj, želja in namenov, ki so predstavljeni s tremi modalnostmi (Bel, Des in Int). Pomen logike za obravnavo agentov, na primer LORE, ki je sicer v članku nismo podrobneje predstavili, je predvsem v tem, da lahko z njeno pomočjo sistematično razmišljamo o lastnostih razumnih agentov. Opisali pa smo generični BDI-model in lastnosti ter programsko izvedbo razumnega agenta. Oboje služi kot temelj v večini agentih razvojnih orodij, ki jih v članku omenjamo. Področje agentnih sistemov je podprto s številnimi metodologijami načrtovanja, razvojnimi sistemi, orodji in programskimi jeziki. Med njimi smo nekoliko natančneje pregledali JADE in razvojni sistem Jadex, ki smo ju potem tudi uporabili za izvedbo večagentnega pomočnika pranja. Ideje, ki so nas pri tem vodile, smo osvetlili v zadnjem poglavju.

S primerom, ki ga obravnavamo v zadnjem poglavju, smo pokazali, kako lahko načrtujemo inteligentne hišne naprave kot večagentne sisteme. S tem smo postavili osnovo za nadaljnje raziskovalno delo, ko bomo na osnovi takšne arhitekture raziskovali miselno zgradbo Razumnega Hišnega Pomočnika in kot njegove posebne primerke - Razumnega Pomočnika Pranja, Razumnega Pomočnika Kuhanja in Razumnega Pomočnika Hlajenja. Obravnavali jih bomo kot posamezne razumne agente in kot pomočnike, ki delujejo kot samostojni ali nadzorovani agenti in opravljajo neko določeno hišno opravilo, se lahko učijo, vodijo dialog med človekom in strojem v čimbolj naravnem jeziku, ki si ga izpopolnjujejo, ter znajo sodelovati z drugimi hišnimi pomočniki v agentni skupnosti, ki jo bomo zgradili in preizkusili ter ovrednotili.

Viri in literatura

/Bellifemine - 1999/ F. Bellifemine, G. Rimassa, A. Poggi. JADE - A FIPA-compliant agent framework. In 4th International Conference on the Practical Applications of Agents and Multi-Agent System (PAAM-99, pages 97-108, London, UK, December 1999.

/Bratman - 1987/ M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press: Cambridge, MA, 1987

/Bratman - 1988/ M. E. Bratman, D. Israel in M. Pollack. *Plans and Resource-bounded Practical Reasoning.* Computational Intelligence. Zv. 4, str. 349 - 355. 1988.

/Braubach - 2004/ L. Braubach, A. Pokahr, D. Moldt, W. Lamersdorf. Goal Representation for BDI Agent Systems. Distributed Systems and Information System Computer Science Department, University of Hamburg. Dosegljivo na: http://vsis-www.informatik.uni-hamburg.de/publications/view.php/208

/Cohen - 1990/P.R. Cohen and H.J Levesque. Intention is choice with rcommitment. *Artificial Intelligence, 42: 213-261,1990*

/Dastani - 2005/M. Dastani, R. H. Bordini, B. van Riemsdijk. Programming Languages for Multi-Agent System. EASSS 2005, http://www.agentlink.org/happenings/easss/2005/

/Denett - 1987/ D. C. Denett. *The Intentional Stance*, The MIT Press: Cambridge, MA, 1987.

/Doyle, 1992/ J. Doyle. Rationality and its roles in reasoning. Computational Intelligence, 376 - 409, 1992.

/Fagin - 1985/ R. Fagin and J. Y. Halpern. *Belief, awareness, and limited reasoning*. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85), pages 480-490, Los Angeles, CA, 1985.

/Fikes - 1971/ R.E.Fikes and N.Nilsson. *STRIPS: A new approach to the application of theorem proving to problem solving. Artificial intelligence*, 5(2): 189-208, 1971.

/Georgeff - 1987/ M. P. Georgeff, A. L. Lansky. *Reactive reasoning and Planning.* In Proceedings of the Sixth National Conference on Artificial Intelligent (AAAI-87), strani677-682, Seattle, WA, 1987.

/Georgeff - 1999/ M. Georgeff, B. Pell, M.Pollack, M. Tambe in M. Wooldridge. *The Belief-Desire-Intention Model of Agency.* Editors Intelligent Agents V. Springer-Verlag Lecture Notes in AI Volume 1365, marec 1999.

/Hajdinjak - 2004/ Melita Hajdinjak. *Vodenje dialoga med človekom in računalnikom v naravnem jeziku.* Magistrsko delo. Univerza v Ljubljani, Fakulteta za elektrotehniko, 2004.

/Halpern - 1990/ Joseph Y. Halpern. *Knowladge and Common Knowladge in a Distributed Environment*. Journal of the Assotiation for Machinery, Vol.37, No. 3. July 1990, pp. 549-587.

/Hintikka - 1975/ J.Hintikka. *The Intentions of Intentionality and Other New Models for Modalities*. D. Reidel Publishing Company, Dordrecht. 1975.

/Huth - 2004/ Michael Huth and Mark Ryan. *Logic in Computer Science; Modelling and Reasoning about Systems*. Cambridge University Press. 2004

/Laza - 2003/ R. Laza, A. Gómez, R. Pavón, J. M. Corchado. *A Case-Based Reasoning Approach to the Implementation of BDI Agents.* Departamento de Informática University of Vigo. Dosegljivo na: http://home.earthlink.net/~dwaha/research/meetings/eccbr02-micbrw/papers/rlaza%20p27-30.doc

Kinny - 1991/ D. Kinny, M. Georgeff. Commitment and effectiveness of situated agents. In Proeceedings of the Twelfth International Joint Conference on Artificial Intelligentce,pages 82 - 88, Sydney, Australia, 1991.

/Levesque - 1984/ H. J. Levesque. *A logic of implicit and explicit belief*. In Proceedings of the Fourth International Joint Conference on Artificial Intelligence (AAAI-84), pages 198-202, Austin, TX, 1984.

/Luck - 2003/ Michael Luck, Peter McBurney and Chris Preist. *Agent Technology: Enabling Next Generation Computing*. AgentLink II. 2003.

/Massonet - 2002/ P. Massonet, Y. Deville, C.Nčve. From AOSE Methodology to Agent Implementation. Dosegljivo na: http://portal.acm.org/citation.cfm?id=544747

/McBurney - 2005/ P. McBurney, M. Luck. Introduction to Agents. EASSS 2005, http://www.agentlink.org/happenings/easss/2005/

/McCarthy - 1978/ J. McCarthy. *Ascribing mental qualities to machines*. Technical report, Stanford University AI Lab., Stanford, CA94305, 1978.

/Meyer - 2005/ J.-J. Ch. Meyer. Logics for Agents: The BDI & KARO Frameworks. EASSS 2005, http://www.agentlink.org/happenings/easss/2005/

/Ming Mao - 2005/ Ming Mao. Agent, BDI and Teamwork. Dosegljivo na: http://usl.sis.pitt.edu/ulab

/Pokhar - 2005/ Aleksander Pokahr, Lars Braubach. *Jadex Users Manual.* University of Hamburg, Nemčija. http://vsis-www.informatik.uni-hambug.de

/Rao - 1991/ Anand S. Rao, Michael P. Georgeff. Modeling rational agents within a BDI architecture. In R. Fikes and E. Sandewall, editors, Proceedings of Knowledge Representation and Reasoning (KR&R-91), pages 473-484. Morgan Kaufmann Publisher: San Mateo, CA, April 1991.

/Rao - 1992/ Anand S. Rao, Michael P. Georgeff. An abstract architecture for rational agents. In C. Rich, W Swarout, and B. Nebel, editors, Proceedings of Knowledge Representation and Reasoning (KR&R-92), pages 473-484, 1992.

/Rao - 1995/ Anand S. Rao, Michael P. Georgeff, *BDI Agents: From Theory to Practice.* April 1995. Technical Note 56. Australian Artificial Intelligence Institute. Dosegljivo na: http://www.agent.ai/doc/upload/200302/rao95.pdf

/Referenca - 1/ http://www.aamas2005.nl/

/Referenca - 2/ http://www.agentgroup.unimore.it/aose05/

/Referenca - 3/ http://www.fipa.org/

/Shehory - 2005/ O. Shehory, L. Padgham, A. Sturm, L. Sterling. Metodologies for Agent-Oriented Software Enginering. EASSS 2005, http://www.agentlink.org/happenings/easss/2005/

/Smith - 1977/ A. Smith. *The Contract net: A formalism for the control of distributed problem solving.* In Proceedings of the Fift International Conference on Artificial Intelligence (AAI.92), San Diego, CA, 1992.

Shoham - 1993/ Y. Shoham. Agent-oriented programming. Artificial Intelligence, 51 - 92, 1993.

/van der Hoek - 2002/ W. van der Hoek, M. Wooldridge. Towards a Logic of Rational Agency. Dept of Computer Science, University of Liverpool, Liverpool L69 7YF, UK. September 2002

/Wooldridge - 1992/ M. J. Wooldridge. *The Logical Modelling of Computatonal – Multiagent Systems,* A thesis submitted to the University of Manchaster for the degree of Doctor of Philosophy in the Faculty of Technology,1992.

/Wooldridge - 1995/ M. J. Wooldridge, Nicholas R. Jennings. *Intelligent Agents: Theory and Practice.* Submitted to Knowladge Engineering Review, 1995.

/Wooldridge - 1997/ M. J. Wooldridge. *Agent based computing.* Dosegljivo na: http://www.csc.liv.ac.uk/~mjw/pubs/icon98.pdf.

/Wooldridge - 1998/ M. J. Wooldridge. *Reasoning about Rational Agents.* The MIT Press Cambridge, Massachusetts London, England. 1998.

/Wooldridge - 2000/ Michael Wooldridge. *Reasoning about Rational Agents.* The MIT Press Cambridge, Massachusetts London, 2000.

/Wooldridge - 2002/ Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, Ltd. 2002

mag. Konrad Steblovnik
Gorenje program Point
Partizanska 12, 3320 Velenje
konrad.steblovnik.point @gorenje.si

Prof. dr. Jurij Tasič
Fakulteta za elektrotehniko
Tržaška 25, 1000 Ljubljana

Dr. Damjan Zazula
Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova 17, 2000 Maribor

Prispelo (Arrived): 23. 11. 2005; Sprejeto (Accepted): 08. 12. 2005