

# Vmesnik za povezavo simuliranega sistema stavbne avtomatike z več agentnim sistemom vodenja

*Domen Zupancič<sup>1,2</sup>, Matjaž Gams<sup>3</sup>, Mitja Luštrek<sup>3</sup>*

Mednarodna podiplomska  
šola Jožefa Stefana<sup>1</sup>  
Jamova 39  
SI-1000 Ljubljana

Robotina d.o.o.<sup>2</sup>  
OIC-Hrpelje 38  
SI-6240 Kozina

Inštitut Jožef Stefan<sup>3</sup>  
Jamova 39  
SI-1000 Ljubljana

Tel: +386 1 477 3230

e-mail: domen.zupancic@ijs.si

## POVZETEK

Uporaba metod umetne inteligence je ključna pri sodobnih sistemih vodenja pametnih hiš. Cilji vodenja so nasprotujoči: doseganje čim manjša porabe energije na eni strani in čim večja stopnja udobja na drugi strani. Umetna inteligenca se pojavlja na več stopnjah: upravljanje senzorskih omrežij, luščenje podatkov iz množice senzorjev, uporaba strojnega učenja za izdelavo modelov napovedovanja (npr. prisotnosti) ali za izdelavo dinamičnih modelov sistema (npr. dinamični model spreminjanja temperature v prostoru), več agentni sistemi. Predstavljeni vmesnik za povezavo simulacijskega sistema z več agentnim sistemom za vodenja omogoča vpeljavo metod umetne inteligence v pametno stavbo na modularen način.

## 1 UVOD IN SORODNO DELO

Sistemi stavbne avtomatike (BAS - Building Automation System), še posebno vodenje takšnih sistemov (BEMS - Building Energy Management Systems) je zelo interesantna tema, saj za razliko od preteklih let te tehnologije bolj in bolj prodirajo na trg tudi za privatna stanovanja. Centralni nadzorni sistemi se uporabljajo že dalj časa, posebno v poslovnih prostorih in v objektih zahtevnejših in predvsem premožnejših lastnikov.

Poleg transparentnega pregleda delovanja sistemov in interakcije z uporabnikom, sodobni sistemi s pomočjo zmogljivih računalniških sistemov omogočajo zahtevnejše pristopke k vodenju tako enostavnih, še posebej pa kompleksnih sistemov, kjer gre za vodenje množice naprav - aktuatorjev na podlagi informacij, ki jih zbira množica senzorjev.

Problemi, ki jih morajo reševati sistemi vodenja v stavbah so: energija, udobje in zanesljivost vodenja. Pregled takšnih sistemov je izvedel Dounis et al. [1] in poudarja, da je umetna inteligenca pri vodenju eden izmed ključnih pristopov k učinkovitemu vodenju, poleg tega pa omogoča vključitev uporabnika v sistem vodenja. Veliko je bilo izdelanih študij, kjer so uporabljali agentni pristop k vodenju sistemov, posebno na področju upravljanja stavb

[5], upravljanja obremenitev električnih omrežij [6], upravljanja porabe virov [7], upravljanje senzorskih omrežij [8], idr.

V prispevku bo prikazana združitev simulacijskega sistema in sistema vodenja s programskimi agenti. Simulacijski model je izdelan v programu EnergyPlus[2], sistem vodenja je izdelan v okolju JADE [3], simulacija pa teče v okolju BCVTB [4].

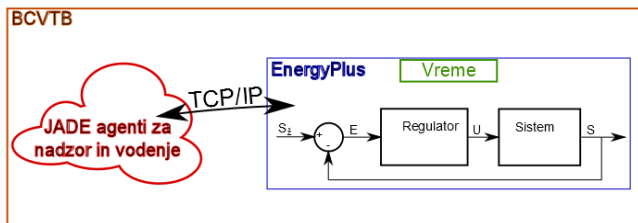
## 2 SIMULACIJSKI MODEL SISTEMA

Simulacijski model sistema vsebuje fizikalni model zgradbe, izdelan v programskem okolju EnergyPlus. EnergyPlus model zgradbe vsebuje simulacijske parametre, kot so: časovni korak, časovni okvir simulacije, katerega priredimo glede na podatke o vremenu in podatke o prisotnosti (podatki morajo biti datumsko in časovno usklajeni), podatke o lokaciji objekta, konstrukcijske podatke o materialih oken, sten, stropov in pripadajoče parametre podatke o izolaciji, geometrijske podatke ter podatke o orientaciji objekta glede na smer neba, podatke o notranji opreми posameznih prostorov - pohištvo, podatke o električnih napravah in močeh delovanja, parametre termostatov posameznih grelnih teles v povezavi s sistemom HVAC z pred nastavljenimi vrednostmi, parametri posameznih grelcev ter nastavitve za izmenjavo podatkov z zunanjimi programi, kot so BCVTB in formate poročil ob koncu simulacije. Poleg tega model vsebuje tudi parametre regulatorjev, na podlagi katerih EnergyPlus regulira stanja sistema glede na želene vrednosti in glede na vrednosti stanj v prejšnjem časovnem koraku. EnergyPlus torej izvaja regulacijo in simulacijo. Težava, s katero se soočamo, je nastavljanje zelenih vrednosti v pravem časovnem trenutku, čimer pravimo vodenje sistema.

## 3 POVEZAVA SIMULACIJSKEGA SISTEMA Z VEČ AGENTNIM SISTEMOM VODENJA

Regulacija in vodenje enostavnega sistema je shematično prikazano na sliki 1, kjer je  $S_z$  zelena vrednost stanja,  $S$  je

dejanska vrednost stanja,  $E$  je razlika med dejansko in želeno vrednostjo stanja (napaka).

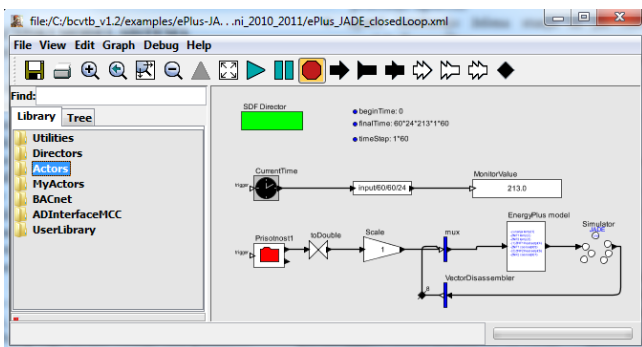


Slika 1: Povezava sistema vodenja s simulacijskim sistemom

Modri okvir predstavlja simulacijski del, rdeči oblak predstavlja agentno arhitekturo za vodenje in nadzor sistemov, vse skupaj pa se nahaja v simulacijskem okolju BCVTB, ki združuje oba sistema. Komunikacija med EnergyPlus modelom sistema in med več agentnim sistemom vodenja poteka preko TCP/IP komunikacijskih vtičnic, ki skrbijo za zanesljivo povezavo. Zanesljiva povezava je najbolj pomembna zaradi časovne sinhronizacije. Vsak časovni korak:

- EnergyPlus izračuna stanja sistema in jih posreduje agentom
- agenti izračunajo zelena stanja in jih vrnejo modelu EnergyPlus

BCVTB skrbi za komunikacijo v primernih časovnih trenutkih. Simulacijsko okolje BCVTB je izdelano za programsko okolje Ptolemy II [12], v katerem modul "Simulator" skrbi za vzpostavitev povezave zunanjega programa in za komunikacijo.



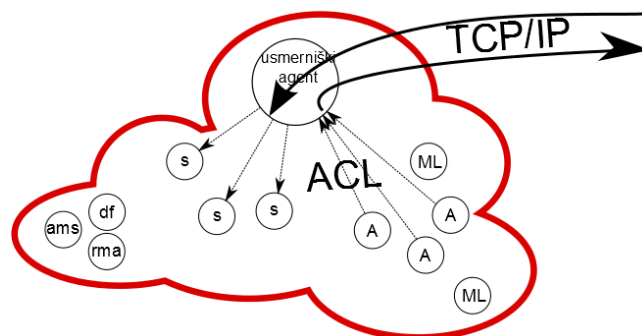
Slika 2: Ko-simulacija: EnergyPlus model sistema in vodenje z JADE programskimi agenti v okolju BCVTB

### 3.1 Usmerniški agent

Usmerniški agent je izdelan za transformacijo sporočil iz TCP/IP protokolnega sklada v ACL sporočila in je izdelan po vzoru, ki ga je predstavil Wetter [4] za ko-simulacijo z drugimi programi (EnergyPlus, Matlab, Dymola,...) ter predstavlja vlogo usmerjevalnika. Ob zagonu simulacije na shemi, ki je predstavljena na sliki 2, vsak od modulov "EnergyPlus model" in "Simulator JADE" izdela pripadajočo konfiguracijsko datoteko, v kateri zapiše podatke o gostitelju in vratih ter izdela strežniško vtičnico.

Zatem kliče pripadajoč program - odjemalec, kateri uporabi zgoraj zapisano datoteko za izdelavo odjemalčeve vtičnice. Odjemalec se izvaja, dokler ne pride do napak ali do konca simulacije.

Usmerniški agent posreduje stanja simuliranega sistema EnergyPlus senzorskim agentom. Senzorski agenti posredujejo informacije aktuatorskim agentom. Aktuatorski agenti pa glede na algoritem vodenja vračajo želene vrednosti (eng. set-points) preko JADE usmerniškega agenta simuliranemu sistemu EnergyPlus. EnergyPlus na podlagi modela sistema izračuna nova stanja sistema in simulacijski korak se zaključi. Podrobnejši prikaz oblaka agentov je prikazan na sliki 3. Poleg treh agentov AMS, DF in RMA, ki so nujni za delovanje JADE platforme, se v oblaku nahajajo še senzorski agenti (S), aktuatorski agenti (A), agenti za strojno učenje (ML) in usmerniški agent. Vsi agenti med seboj lahko komunicirajo z uporabo Agentnega Komunikacijskega Jezika (eng. Agent Communication Language - ACL), ki je definiran v FIPA specifikacijah [13].



Slika 3: Usmerniški, senzorski, aktuatorski, ML in JADE agenti

Usmerniški agent preko ACL sporočil komunicira samo z senzorskimi in aktuatorskimi agenti, ki so opisani v naslednjih podpoglavjih. Med pripravo simulacijskega sistema je potrebno definirati povezave med stanji, ki jih oddaja EnergyPlus in senzorji. V ta namen moramo kreirati datoteko *variables\_mapping.cfg*, ki ima XML strukturo, prikazano na sliki 4, v kateri se vidi primer povezave dveh aktuatorskih agentov *jadename* s spremenljivko *bcvtbname*, definirano v EnergyPlus in dveh senzorskih agentov *jadename* s spremenljivko *bcvtbname*. Vir aktuatorskih spremenljivk - *source* je v usmerniškem agentu (*JadeGateway*), medtem ko je vir senzorskih spremenljivk v simulatorju EnergyPlus (BCVTB, *Ptolemy*).

Velikost vhodnega oziroma izhodnega vektorja modulov, prikazanih na sliki 2, je odvisna od števila senzorjev in aktuatorjev in je skladna s konfiguracijsko datoteko, opisano zgoraj. Ravno tako pa je odvisna velikost BSD sporočila med usmernikom in simulatorjem, ki se izmenjuje preko TCP/IP komunikacijskih vtičnic. Format

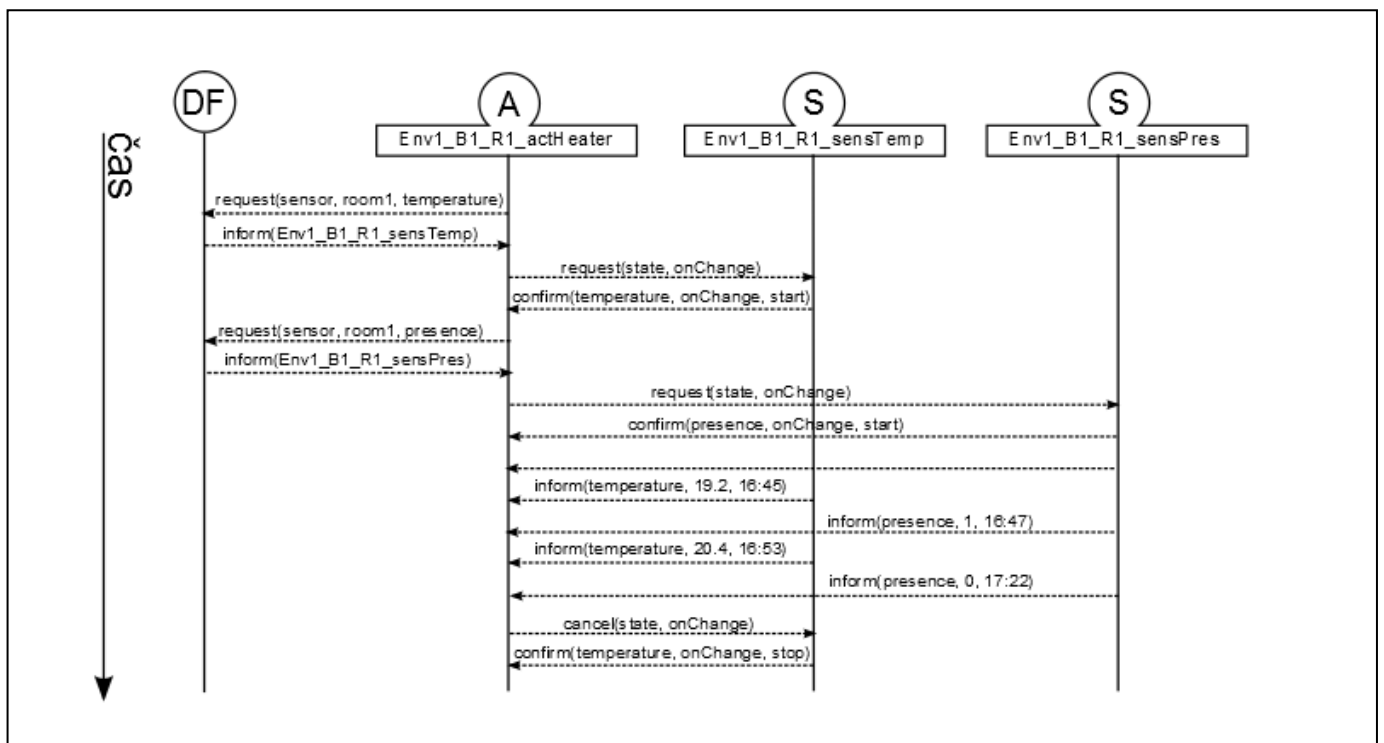
BSD sporočila katerega usmerniški agent prejme oziroma pošlje je definiran na spletni strani [13] in vsebuje naslednje vrednosti, ločene s presledkom:

- verzija okolja Ptolemy,
- zastavica, ki govori o tipu sporočila (normalna operacija, konec simulacije, napaka),
- število vrednosti tipa *double*,
- število vrednosti tipa *integer* (vedno 0),

- število vrednosti tipa *boolean* (vedno 0),
- simulacijski čas v sekundah,
- vrednosti tipa *double*
- konec, določen z znakom "\n".

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BCVTB-variables>
<!-- The next elements receive the variables from JADE gateway - setpoint values -->
  <variable source="JadeGateway">
    <JadeGateway bcvtbname="TSetHeat1" jadenname="Env1_B1_Room1_ActHeater"/>
  </variable>
  <variable source="JadeGateway">
    <JadeGateway bcvtbname="TSetCoo1" jadenname="Env1_B1_Room1_ActChiller"/>
  </variable>
  .....
<!-- The next elements send the sensor states from BCVTB (EnergyPlus) to JadeGateway -->
  <variable source="Ptolemy">
    <JadeGateway bcvtbname="ENVIRONMENT" jadenname="Env1_OutdoorTemperature"/>
  </variable>
  <variable source="Ptolemy">
    <JadeGateway bcvtbname="ZSF1" jadenname="Env1_B1_Room1_SensTemp"/>
  </variable>
  .....
</BCVTB-variables>
```

Slika 4: Izsek iz konfiguracijske datoteke *variables\_mapping.cfg*, ki povezuje spremenljivke stanja s senzorskimi agenti in zelene vrednosti z aktuatorskimi agenti



Slika 5: Časovni potek iskanja primernih sensorjev s pomočjo DF agenta, sklepanje pogodb med aktuatorji in senzori, obveščanje sensorjev o stanjih ter prekinitve pogodb

### 3.2 Senzorski agent

Senzorski agent (v nadaljevanju senzor) ima vlogo računalniškega predstavnika fizične naprave - senzorja. Ima dve nalogi: Prejemanje informacij o stanjih sistema (temperatura, vlaga, prisotnost, itd.) od simulatorja preko usmerniškega agenta in posredovanje teh informacij agentom, kateri te informacije zahtevajo ali s katerimi ima sklenjeno pogodbo o obveščanju. Pogodbe sklepajo z aktuatorskimi agenti, ki zahtevajo informacije v vsakem simulacijskem trenutku.

### 3.3 Aktuatorski agent

Aktuator vsebuje več možnih stanj, vendar ima v vsakem stanju nalogo, da v vsakem časovnem trenutku izračuna želeno vrednost. Mogoča so različna stanja, ki predstavljajo algoritme vodenja, kot so: urnik, vključeno, izključeno, glede na trenutno senzorsko stanje enega senzorja in s pravili, glede na klasifikacijski model in trenutna senzorska stanja več senzorjev, itd. Na začetku stanja, v katerem se agent pojavi, s primernimi senzorji, katere pridobi preko DF agenta, sklene pogodbo o obveščanju. Vsak časovni korak v simulaciji tako pridobi informacije o senzorskih vrednostih in glede na algoritem stanja priredi aktuatorske vrednosti, ter jih preko usmerniškega agenta pošlje simulatorju. Slika 5 prikazuje časovno zaporedje sporočil, med aktuatorjem, DF agentom in dvema senzorjema. Aktuator potrebuje za delovanje podatke o prisotnosti in temperaturi. Najprej pri DF agentu dobi imena senzorskih agentov, ki nudijo te informacije. Potem z vsakim od teh senzorjev sklene pogodbo o obveščanju. Ko ne potrebuje več informacij, prekine pogodbo.

### 3.4 ML agent

ML agent je namenjen strojnemu učenju. Kadar aktuator za svoje delovanje potrebuje klasifikacijski model, sporoči ML agentu vir informacij, na osnovi katerih naj bo klasifikacijski model zgrajen. ML poišče informacije, zgradi model in aktuatorju vrne lokacijo datoteke. Aktuator potem ta model uporabi v algoritmu za izračunavanje zelenih vrednosti.

## 4 ZAKLJUČEK

Sistem smo preizkusili z simulacijskim modelom stanovanjske stavbe, pridobljenim z okoljem BCVTB, ki smo ga modificirali za naše potrebe. V simulacijo sistema smo vključili podatke o prisotnosti uporabnika stanovanja [9,10] in vremenske podatke za Ljubljana-Brnik za obdobje 2010-2011. Sistem deluje zanesljivo in omogoča modularno združevanje simulacijskih modelov z več agentnimi sistemi vodenja.

## ZAHVALA

"Operacijo delno financira Evropska unija, in sicer iz Evropskega socialnega sklada. Operacija se izvaja v okviru Operativnega razvoja človeških virov za obdobje 2007 -

2013, 1. razvojne prioritete: Spodbujanje podjetništva in prilagodljivosti, prednostne usmeritve 1.1.: Strokovnjaki in raziskovalci za konkurenčnost podjetij."

## LITERATURA

- [1] A.I. Dounis, C. Caraiscos, Advanced control systems engineering for energy and comfort management in a building environment—A review, *Renewable and Sustainable Energy Reviews*, Volume 13, Issues 6–7, August–September 2009
- [2] D. B. Crawley, C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann, Energyplus: Energy simulation program, *ASHRAE Journal*. 42, 49–56, (2000).
- [3] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, JADE: A software framework for developing multi-agent applications. Lessons learned, *Information and Software Technology*. 50(1-2), 10–21 (Jan., 2008).
- [4] Wetter, Michael. "Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation* 4.3 (2011): 185-203.
- [5] Escrivá-Escrivá, Guillermo, Isidoro Segura-Heras, and Manuel Alcázar-Ortega. "Application of an energy management and control system to assess the potential of different control strategies in HVAC systems." *Energy and Buildings* 42.11 (2010): 2258-2267.
- [6] McArthur, Stephen DJ, et al. "Multi-agent systems for power engineering applications—Part I: concepts, approaches, and technical challenges." *Power Systems, IEEE Transactions on* 22.4 (2007): 1743-1752.
- [7] Cao, Junwei, et al. "ARMS: An agent-based resource management system for grid computing." *Scientific Programming* 10.2 (2002): 135-148.
- [8] Lesser, Victor, Charles L. Ortiz, and Milind Tambe, eds. Distributed sensor networks: A multiagent perspective. Vol. 9. Springer, 2003.
- [9] D. Cook. Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*, 2011.
- [10] D. Cook and M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, 2009
- [11] C. Sorin. Towards High-Quality Multilingual Text-to-Speech. *Proc. CRIM/FORWISS Workshop on Progress and Prospects of Speech Research and Technology*. München. 1994.
- [12] Christopher Brooks, Edward A. Lee, Xiaojun Liu, Steve Neuendorffer, Yang Zhao, and Haiyang Zheng. Ptolemy II: heterogeneous concurrent modeling and design in Java. *Technical Report No. UCB/EECS-2007-7*, University of California at Berkeley, Berkeley, CA, January 2007
- [13] BCVTB: <http://simulationresearch.lbl.gov/bcvtb/releases/latest/doc/manual/ch07s11.html>. *Spletna stran*. Zadnji dostop: 5.9.2013
- [14] Poslad, Stefan, Phil Buckle, and Rob Hadingham. "The FIPA-OS agent platform: Open source for open standards." *Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*. Vol. 355. 2000.