

ARHITEKTURA SISTEMA OpUS

Aleš Tavčar^{1,2}, Jure Šorn¹, Tea Tušar¹, Tomaž Šef¹, Matjaž Gams^{1,2}

Odsek za inteligentne sisteme, Institut »Jožef Stefan«¹

Jamova cesta 39, 1000 Ljubljana, Slovenija

Mednarodna podiplomska šola Jožefa Stefana²

Jamova cesta 39, 1000 Ljubljana, Slovenija

e-mail: {ales.tavcar, tea.tusar, tomaz.sef, matjaz.gams}@ijs.si

POVZETEK

Obstoječi sistemi hišne avtomatizacije oz. pametnih stavb ne omogočajo naprednih funkcionalnosti, ki bi jih od takih sistemov pričakovali. Trenutne rešitve omogočajo zgolj spremljanje stanja sistemov in okolja v hiši ter krmiljenje hišnih naprav preko mobilnih naprav in spleta. Definiranje urnikov delovanja je prepuščeno samim uporabnikom, kar se običajno odraža v večji porabi in neučinkovitem delovanju. V pričujočem prispevku so predstavljeni načini za nadgradnjo sistemov hišne avtomatizacije z inteligentnimi metodami učenja navad uporabnikov in metodami optimizacije delovanja. Tak sistem je zmožen spremljanja obnašanja uporabnikov, se učiti njihovih navad in prilagajati delovanje glede na spreminjajoče se življenjske navade in potrebe.

1 UVOD

Trenutne komercialne rešitve hišne avtomatizacije ne ponujajo naprednih funkcionalnosti, ki bi jih pričakovali v takih sistemih. Namesto pametnega predvidevanja navad in potreb uporabnika obstoječi sistemi ponujajo zgolj krmiljenje hišnih naprav ter nadzor stanja v zgradbi. Slednji poteka preko različnih pametnih naprav in v redkih primerih preko spletnih vmesnikov. Nastavljanje režima delovanja in scenarijev je prepuščeno samim uporabnikom, iz česar običajno sledi, da so urniki nastavljeni površno in zato neučinkovito. Poleg tega se navade uporabnikov neprestano spreminjajo. V določenem obdobju lahko npr. uporabniki začnejo prihajati domov kasneje; ob nespremenjenem urniku to pomeni, da se začne ogrevanje hiše prezgodaj, kar se odraža v večji porabi energentov in posledično v višjih obratovalnih stroških. Poleg tega se vedno bolj uveljavlja uporaba fotovoltaike kot komplementaren vir energije. Predpostavljamo, da lahko z inteligentnim kombiniranjem različnih virov energije zmanjšamo stroške obratovanja in hkrati ohranimo zadovoljivo stopnjo udobja uporabnikov.

Za reševanje zgornjih nalog predlagamo sistem, ki je sposoben sprotnega spremljanja dogajanja v hiši, učenja navad uporabnikov, prilagajanja delovanja glede na spremembe v navadah uporabnikov in optimiziranja delovanja celotnega sistema hišne avtomatizacije.

Sistem OpUS, ki je bil razvit v okviru projekta e-storitve za gospodarstvo, je samostojna in robustna rešitev, ki se lahko vgradi v široko paleto obstoječih sistemov hišne avtomatizacije. Združuje več inovativnih komponent, kjer vsaka skrbi za določen vidik inteligentnega upravljanja pametne stavbe.

V nadaljevanju prispevka najprej opravimo kratek pregled sorodnega dela. Nadaljujemo s predstavitvijo arhitekture sistema OpUS in na kratko opišemo naloge posameznih komponent sistema.

2 SORODNI SISTEMI

Obširen pregled sodobnih sistemov vodenja v pametnih zgradbah je opisal Dounis et.al [1]. Za razliko od klasičnih načinov vodenja, je za optimalno, prediktivno ali adaptivno vodenje potrebno imeti model zgradbe. Temu pristopu sledimo v tem prispevku.

Vodenje sistemov z uporabo podatkov in znanj o uporabnikih in okolju predstavlja nove smernice raziskav in razvoja tako imenovanega vseprisotnega in prodornega računalništva (ang. ubiquitous computing, pervasive computing), saj sodobne naprave, senzorji in aktuatorjih, ki se vse bolj množično pojavljajo v zgradbah (senzorji prisotnosti, senzorji gibanja, senzorji odprtosti oken, senzorji na mobilnih telefonih, osebne vremenske postaje itd.) omogočajo beleženje najrazličnejših informacij in kopičenje znanj tako o obnašanju posameznega uporabnika, kot o obnašanju sistema. Uporaba takšnih znanj se izkorišča v sistemih, ki spodbujajo uporabnike k zmanjšanju porabe energije s spodbujanjem k na primer nižanju zelenih temperatur ogrevanja ali pa k izbiri primernih prostorov v službi za potrebe sestankov (manj ljudi - manjši prostor - manj energije za ogrevanje) [3]. Znanje o uporabnikih se izkorišča za gradnjo modelov uporabnikovega obnašanja in uporabo le-teh pri vodenju in adaptaciji sistemov ogrevanja, razsvetljave, prezračevanja in ogrevanja sanitarne vode [4,5]. Prihranki energije se gibljejo med 5-30%.

Veliko projektov na temo izvedbe testnih pametnih hiš in stanovanj je bilo že dokončanih. Leta 1990 so izdelali Neural Network House [6], kjer so uporabljali nevronske mreže za inteligentno vodenje sistemov. Sledila sta IHome[7] in MavHome[8], temelječa na inteligentnem večagentnem pristopu nadzora in vodenja sistemov z uporabo

tehniki za modeliranje in napovedovanje uporabnikovega obnašanja in akcij. Gator Tech Smart House [9] je splošno uporaben študijski projekt za raziskavo tehnik vseprisotnega računalništva (ang. pervasive computing). Eden zadnjih projektov - ThinkHome [10] uporablja širok nabor podatkov o okolju, vremenu in uporabniku za namene študije vodenja pametnih domov.

Vsi sistemi se povečini osredotočajo zgolj na določene vidike upravljanja stavbe. ThinkHome, na primer, poskuša predvidevati temperaturo, ki bo za uporabnika najudobnejša. Poleg tega poskuša predvidevati, kdaj bo nek uporabnik prisoten. Sistem OpUS je obširnejši, saj poskuša modelirati večje število parametrov pametne hiše obenem pa v algoritme vodenja vključuje uporabniške navade in optimizacijske algoritme.

3 ARHITEKTURA SISTEMA

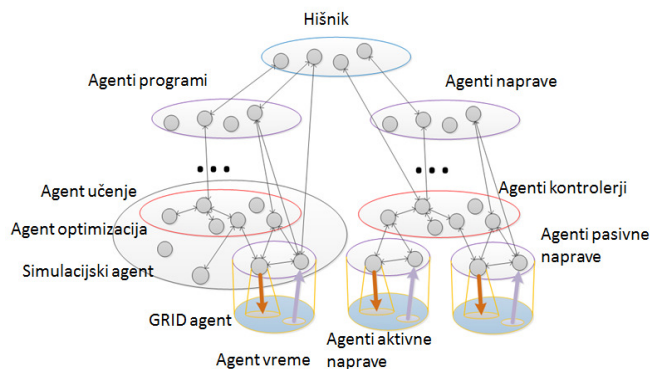
Arhitektura sistema OpUS je definirana s hierarhično urejenim več-agentnim sistemom [11]. Agenti so avtonomne entitete, ki so sposobne zaznavanja in interakcije z okolico skladno z njihovimi preferencami, lastnostmi in aktivnimi cilji. Sposobni so samostojnega razmišljanja in sodelovanja za doseg skupnih ciljev. Vsak agent v OpUS agentni arhitekturi je določen z agentno ovojnico, ki natančno definira posameznega agenta. Ovojnica določa vhodne podatke, ki jih agent zahteva, akcije, ki jih lahko izvaja in izhodne podatke, ki jih lahko posreduje.

3.1 Hierarhično urejena agentna arhitektura

Agentna arhitektura ni samo skupek agentov, ki določa hierarhične odnose med njimi, ampak skrbi za beleženje stanja vsakega agenta, določa način komunikacije med njimi in omogoča podporo za sprotno simuliranje dinamike v prototipnem okolju.

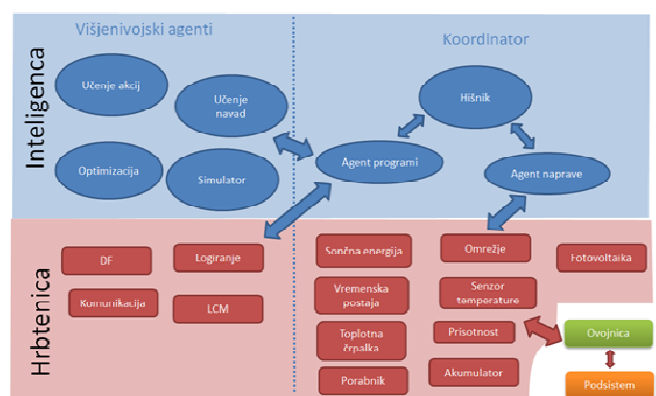
Sistem OpUS sestavlja šest različnih tipov agentov. Agent **pasivna naprava** je naprava, ki zgolj beleži in posreduje določene podatke. Primer takega agenta je senzor temperature zraka. Agent **aktivna naprava** je naprednejši in omogoča upravljanje z napravo. Primer takega agenta je toplotna črpalka, ki se ga lahko vklaplja, izklaplja in nastavlja na določeno temperaturo. Naslednji tip agentov je **program**, slednji združuje agente, ki ponujajo različne servise znotraj arhitekture. V to kategorijo spadajo agent dnevnik, ki beleži spremembe v sistemu, agent upravljanja sistema, agenti za učenje, agenti za optimizacijo, simulacijo ipd. Četrty tip agentov **hišnik** so upravljalovski agenti, ki omogočajo nadzor in upravljanje določenih sklopov pametne stavbe. Odvisno od razdelitve in same hierarhične strukture, lahko ti agenti upravljajo sistem ogrevanja, določeno sobo ali pa celotno zgradbo. Agent hišnik, ki je v agentni hierarhiji postavljen najvišje, skrbi tako za koordinacijo med ostalimi višjenivojskimi agenti, kot tudi za komunikacijo z zunanjim svetom. Zgradba je namreč lahko vključena v pametno mesto, ki lahko od nje zahteva aktivno pogajanje o cenah energentov ipd.

Zadnja dva tipa agentov sta še podporni in komunikacijski agenti, ki so lastni arhitekturi in skrbijo za prenašanje sporočil med agenti, beleženje posameznih agentov, posredovanje podatkov, ontologij ipd. Simbolična shema arhitekture je prikazana na sliki 2. Sama arhitektura je zasnovana tako, da je omogočeno enostavno dodajanje in odstranjevanje agentov v sistem.



Slika 1: Hierarhična več-agentna arhitektura.

Sistem je logično razdeljen na dva dela, kot je prikazano na sliki 3. Zgornji nivo vsebuje tako imenovano inteligenco, torej agente, ki skrbijo za generiranje novih, za uporabnika ustrežnejših urnikov. Spodnji nivo (t. i., hrbtnica) vsebuje agente vodenja, ki skrbijo za izvajanje urnikov in sprotno sinhronizacijo z realnim okoljem. Pomembna predpostavka v sistemu je ta, da lahko sistem ob morebitnem izpadu zgornjega dela še vedno nemoteno deluje in skrbi za upravljanje z napravami v zgradbi.



Slika 2: Povezovanje agentov znotraj arhitekture.

3.2 Učenje navad uporabnika

V sistem OpUS je vključen modul za učenje navad uporabnika. Slednji poskuša zgraditi model obnašanja uporabnika na podlagi opazovanja prisotnosti v hiši in v posameznih sobah. Zgrajeni model lahko za določeno obdobje napove verjetnost, da je uporabnik prisoten ali odsoten. Napoved prisotnosti uporabnika omogoči celo vrsto dodatnih storitev, ki so v sodobnem sistemu hišne avtomatizacije nujne. Natančna napoved časa odhoda uporabnika omogoča varčevanje z energijo, saj lahko sistem

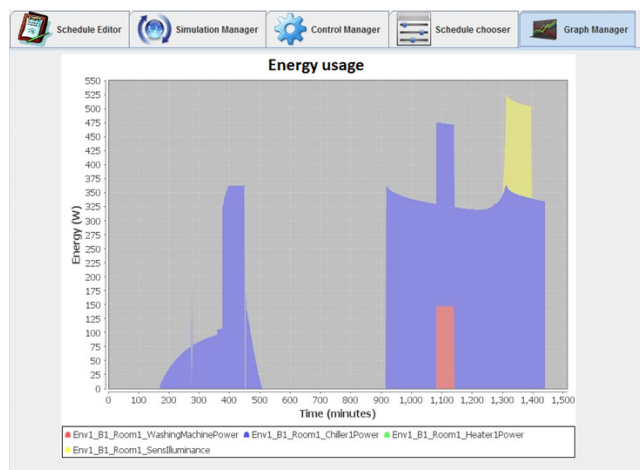
predčasno izklopi ogrevanje. Podobno lahko natančna napoved časa prihoda omogoči povečanje občutka udobja pri uporabnikih.

Gradnja modela poteka tako, da sistem dlje časa beleži senzorske podatke o prisotnosti in sproti gradi verjetnostni model navad uporabnika. Prednost sprotne gradnje je v tem, da omogoča prilagajanje na spremembe v navadah uporabnika. Sistem nato dnevno uporabi zgrajeni model za napoved časa odhoda in prihoda uporabnika. S tem optimizacijskemu agentu zmanjša preiskovalni prostor, saj določi intervale, kjer z večjo verjetnostjo pride do sprememb v režimu delovanja.

3.3 Model in simulacija zgradbe

Sposobnost natančnega simuliranja realnega okolja omogoča predhodno verifikacijo strategij vodenja in izbiro tiste, ki vodi do najustreznejšega stanja sistema. V sistemu OpUS je simulacija ključna za poganjanje optimizacijskih algoritmov in sprotne izvajanja dinamike sistema.

Razviti simulacijski agent uporablja orodje EnergyPlus[13], ki omogoča simuliranje poljubnih modelov stavb in različnih naprav. Ključna naloga je zgraditi model stavbe, ki kar se da natančno odraža realno prototipno okolje. Orodje omogoča definiranje naprav s poljubnimi specifikacijami, omogoča integracijo fotovoltaike in akumulatorjev. Za čim bolj natančno izvajanje simulacije uporablja informacije o geografski lokaciji modela, realne informacije o vremenu in sončnem obsevanju. Simulacijski agent omogoča sprotne spreminjanje parametrov delovanja in izpis stanja naprav. Na sliki 3 je prikazana zaslonska maska pregleda porabe posameznih naprav znotraj simulacije.



Slika 3: Prikaz posamezne porabe tekom simulacije.

3.4 Optimizacija delovanja

Sistem OpUS uporablja optimizacijo za izračun novih urnikov, ki so za uporabnika ustrežnejši in omogočajo manjše stroške in večje udobje. Z uporabo simulatorja stavbe, ki za podani urnik vrača stroške in udobje,

večkriterijski optimizacijski algoritem poišče množico urnikov, ki predstavljajo kompromisne rešitve glede na obravnavana nasprotujoča si kriterija. Dobljene rešitve predstavljajo približek za t.i. Pareto fronto [12], na osnovi katere nato uporabnik preko uporabniškega vmesnika izbere nov, zanj najustreznejši urnik.

Algoritem gradi množico rešitev iz začetnega urnika, s preiskovanjem celotnega prostora urnikov in uporabo simulatorja za ovrednotenje generirane rešitve iz vidika porabe in udobja. Tekom izvajanja se generirane rešitve izboljšujejo in najboljše rešitve z vidika obeh kriterijev tvorijo Pareto fronto nedominiranih urnikov. Uporabnik lahko preko uporabniškega vmesnika izbere enega od teh urnikov glede na preference o načinu vodenja stavbe. Za izvajanje izbranega urnika poskrbi agentna arhitektura, ki sočasno upravlja simulirano in realno prototipno okolje.

3.5 Komunikacijski vmesnik

Eden pomembnih modulov v arhitekturi je komunikacijski vmesnik, ki skrbi za sinhronizacijo agentne arhitekture z realnim kontejnerskim prototipom. OpUS preko HTTP zahtevkov pridobiva informacije o stanju naprav priključenih na kontrolerje.

Primer zahtevka za branje temperature v prostoru:

```
http://www.exp.com/scgi/?c11025.ts01_temperature_0
```

Pri čemer je »c11025« CyBro krmilnik z naslovom 11025, ts01_temperature_0 pa je ime spremenljivke v krmilniku.

Strežnik vrne odgovor v formatu XML:

```
<data>
<var>
<name>c11025.ts01_temperature_0</name>
<value>268</value>
<description>
Measured internal temperature, multiplied
by 10 (e.g. 247 means 24.7°C).
</description>
</var>
</data>
```

Dobljene vrednosti se uporabljajo za sinhronizacijo agentne arhitekture z dogajanjem v realnem okolju, tako da se vrednosti v XML odgovoru prepisejo v ustreznega agenta-napravo.

Vodenje sistema poteka na podoben način preko http zahtevkov. Na podlagi vrednosti v izbranem urniku pošlje sistem zahtevek za spremembo delovanja naprave. Prednost takega pristopa je decentraliziranost, saj ni potrebno, da sta sistem OpUS in krmilnik locirana na istem strežniku. S tem tudi ločimo inteligentni del sistema od logičnega dela, ki skrbi zgolj za izvajanje ukazov na krmilniku.



Slika 4: Upravljanje z agenti aktuatorji.

Na sliki 4 je prikazan vmesnik za upravljanje z napravami v sistemu OpUS in hkrati, preko sinhronizacijskega mehanizma, v realnem prototipnem okolju. Na zasloni maski je prikazana trenutna poraba naprav ter celotnega sistema, prikazani so podatki iz senzorjev, stanje baterije in trenutna vremenska napoved. Posamezne naprave je mogoče priklaplјati in izklaplјati ter nastavljati parametre delovanja.

4 ZAKLJUČEK

V pričujočem prispevku smo opisali celotno arhitekturo sistema OpUS in opisali glavne module sistema. Uporaba več-agentne paradigme olajša razvoj kompleksnih sistemov, kjer je pomembna sprotna komunikacija in učinkovito sodelovanje med komponentami.

Vseh pet glavnih agentov sistema je načrtovanih tako, da lahko uporabljajo storitve drugih agentov in hkrati ponudijo na razpolago svoje funkcionalnosti. Tipičen primer je simulacijski agent, ki uporablja storitve vmesnika ter podatke v agentni arhitekturi in ponuja storitve simulacije optimizacijskemu agentu ter omogoča sprotno izvajanje s strani agente arhitekture.

Takšen dinamičen in inteligenten sistem lahko znatno zniža porabo in s tem stroške inteligentne hiše. S tem pa vzpodbuja ekološko ozaveščenost uporabnikov in zmanjšuje negativne vplive na okolje.

Obstoječi tržni sistemi hišne avtomatizacije predstavljenih rešitev ne uporabljajo. Deloma zaradi višjih stroškov razvoja, kar se direktno preslika v ceno takega sistema, deloma pa zaradi trenutno nezanesljivega in nerobustnega delovanja. Sistem OpUS se uspešno spopada z navedenimi težavami in z uporabniškega stališča omogoča prijazno uporabo in veliko mero avtonomnosti.

Reference

[1] Dounis, Anastasios I., and Christos Caraiscos. "Advanced control systems engineering for energy and comfort management in a building environment—A

review." *Renewable and Sustainable Energy Reviews* 13.6 (2009): 1246-1261.

- [2] R. Baños, F. Manzano-Agugliaro, F.G. Montoya, C. Gil, A. Alcayde, J. Gómez, Optimization methods applied to renewable and sustainable energy: A review, *Renewable and Sustainable Energy Reviews*, Volume 15, Issue 4, May 2011, Pages 1753-1766, ISSN 1364-0321, <http://dx.doi.org/10.1016/j.rser.2010.12.008>
- [3] Laura Klein, Jun-young Kwak, Geoffrey Kavulya, Farrokh Jazizadeh, Burcin Becerik-Gerber, Pradeep Varakantham, Milind Tambe, Coordinating occupant behavior for building energy and comfort management using multi-agent systems, *Automation in Construction*, Volume 22, March 2012, Pages 525-536, ISSN 0926-5805, <http://dx.doi.org/10.1016/j.autcon.2011.11.012>
- [4] Lu, Jiakang, et al. "The smart thermostat: using occupancy sensors to save energy in homes." *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010.
- [5] Agarwal, Yuvraj, et al. "Occupancy-driven energy management for smart building automation." *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM, 2010.
- [6] M. C. Mozer, The neural network house: An environment that adapts to its inhabitants, *Proc AAAI Spring Symp Intelligent Environments*. pp. 110–114, (1998)
- [7] V. Lesser, M. Atighetchi, B. Benyo, B. Horling, V. L. M. Atighetchi, A. Raja, R. Vincent, P. Xuan, S. X. Zhang, T. Wagner, P. Xuan, and S. X. Zhang. The intelligent home testbed. In *Proceedings of the Autonomy Control Software Workshop*, (1999).
- [8] D. Cook, M. Youngblood, I. Heierman, E.O., K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: an agent-based smart home. In *Pervasive Computing and Communications, 2003. (PerCom 2003)*. *Proceedings of the First IEEE International Conference on*, pp. 521 – 524 (march, 2003). doi: 10.1109/PERCOM.2003.1192783.
- [9] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, The gator tech smart house: a programmable pervasive space, *Computer*. 38(3), 50 – 60 (march, 2005). ISSN 0018-9162. doi: 10.1109/MC.2005.107.
- [10] C. Reinisch, M. J. Kofler, F. Iglesias, and W. Kastner, Thinkhome energy efficiency in future smart homes, *EURASIP J. Embedded Syst.* 2011, 1:1–1:18 (Jan., 2011). ISSN 1687-3955. doi: 10.1155/2011/104617
- [11] Multi-Agent system:
https://en.wikipedia.org/wiki/Multi-agent_system
- [12] Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Nw York, 2001.
- [13] EnergyPlus.
http://apps1.eere.energy.gov/buildings/energyplus/energyplus_about.cfm