



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Милан Бјелица

**МЕТОДЕ РЕАЛИЗАЦИЈЕ
КОНТЕКСТУАЛНИХ ПЛАТФОРМИ И
КОНТЕКСТУАЛНИХ КОРИСНИЧКИХ
СПРЕГА ЗА ПРИМЕНЕ У УРЕЂАЈИМА
ПОТРОШАЧКЕ ЕЛЕКТРОНИКЕ**

ДОКТОРСКА ДИСЕРТАЦИЈА

Нови Сад, 2012

Nije zadatak potrošača da zna šta želi.

Steve Jobs (1955-2011)

Mojim roditeljima, za bezrezervnu ljubav i podršku.

Bratu Marku, kao dokaz da je istrajnost najvažnija.

Nadi i Vasiliju, mojim životnim saputnicima, i onima koji će mi se tek pridružiti.

*Zahvaljujem se na saradnji, pomoći i savetima
mom mentoru, prof. dr Nikoli Tesliću,
doc. dr Ištvanu Papu,
kolegi i kumu Petru Mariću, za motivaciju i trpeljivost,
kolegama Bojanu Mrazovcu, Ivanu Kaštelanu, Milošu Pilipoviću i Nebojši Košutiću za
nesebično učešće i pomoć tokom trajanja istraživanja,
prof. dr Vladanu Zdravkoviću, za neobično korisne savete o korisničkom iskustvu,
celokupnom i-Novu timu na saradnji i podršci i
kolegama i osoblju instituta RT-RK i Fakulteta tehničkih nauka, što su zajedno doprineli
kreiranju nezaboravnog radnog okruženja.*



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:		
Идентификациони број, ИБР:		
Тип документације, ТД:	Монографска документација	
Тип записа, ТЗ:	Текстуални штампани материјал	
Врста рада, ВР:	Докторска дисертација	
Аутор, АУ:	Милан Бјелица	
Ментор, МН:	проф. др Никола Теслић	
Наслов рада, НР:	Методe реализације контекстуалних платформи и контекстуалних корисничких спрега за примене у уређајима потрошачке електронике	
Језик публикације, ЈП:	Српски / латиница	
Језик извода, ЈИ:	Српски	
Земља публиковања, ЗП:	Република Србија	
Уже географско подручје, УГП:	Војводина	
Година, ГО:	2012	
Издавач, ИЗ:	Ауторски репринт	
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	12 / 361 / 325 / 33 / 151 / 1	
Научна област, НО:	Електротехника и рачунарство	
Научна дисциплина, НД:	Рачунарска техника	
Предметна одредница/Кључне речи, ПО:	контекстуално рачунарство, потрошачка електроника, аутоматизација домаћинства, интеракција човек-рачунар	
УДК		
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, ВН:		
Извод, ИЗ:	У оквиру дисертације представљене су методе реализације контекстуалних платформи и контекстуалних корисничких спрега за уређаје потрошачке електронике. Предложени су алгоритми семантизације контекста и процене заузетости корисника, архитектура решења и резултати евалуације.	
Датум прихватања теме, ДП:		
Датум одбране, ДО:		
Чланови комисије, КО:	Председник: проф. др Владимир Ковачевић	
	Члан: проф. др Јован Ђорђевић	
	Члан: проф. др Миодраг Темеринац	Потпис ментора
	Члан: проф. др Мирослав Поповић	
	Члан, ментор: проф. др Никола Теслић	



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, ANO :		
Identification number, INO :		
Document type, DT :	Monographic publication	
Type of record, TR :	Textual printed material	
Contents code, CC :	PhD thesis	
Author, AU :	Milan Bjelica	
Mentor, MN :	PhD Nikola Teslic	
Title, TI :	Methods of implementation of context-aware platforms and context-aware user interfaces for applications in consumer electronics	
Language of text, LT :	Serbian	
Language of abstract, LA :	Serbian	
Country of publication, CP :	Republic of Serbia	
Locality of publication, LP :	Vojvodina	
Publication year, PY :	2012	
Publisher, PB :	Author's reprint	
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	12 / 361 / 325 / 33 / 151 / 1	
Scientific field, SF :	Electrical Engineering	
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW :	context-awareness, pervasive computing, consumer electronics, home automation, interruptions, human-computer interaction	
UC		
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N :		
Abstract, AB :	In this dissertation methods of implementation of context-aware platforms and context-aware user interfaces for applications in consumer electronics are presented. The work includes the proposition of algorithms for context semantization and user availability estimation, architecture of the solution as well as the results of evaluation experiments.	
Accepted by the Scientific Board on, ASB :		
Defended on, DE :		
Defended Board, DB :	President: PhD Vladimir Kovacevic	
	Member: PhD Jovan Djordjevic	
	Member: PhD Miodrag Temerinac	
	Member: PhD Miroslav Popovic	Menthor's sign
	Member, Mentor: PhD Nikola Teslic	

SADRŽAJ

POGLAVLJE 1. STANJE U OBLASTI	46
1.1 Kontekstualne platforme.....	46
1.1.1 Arhitektura kontekstualne platforme	47
1.1.1.1 Nivo mreže	49
1.1.1.2 Srednji nivo	53
1.1.1.3 Algoritmi za zaključivanje	60
1.1.1.4 Aplikativni nivo	61
1.1.1.5 Nivo korisničke sprege.....	67
1.2 Kontekstualne korisničke sprege.....	69
1.3 Oblast potrošačke elektronike	72
1.4 Kontekstualne platforme u potrošačkoj elektronici	76
1.4.1 Kontekstualne platforme namenjene inteligentnim okruženjima.....	76
1.4.2 Kontekstualne aplikacije za mobilne uređaje.....	85
1.5 Kontekstualne korisničke sprege u potrošačkoj elektronici	86
1.6 Problem asinhronih zahteva	91
1.6.1 Otkrivanje i upravljanje asinhronim zahtevima	93
1.6.2 Objavljivanje procenjenog stanja zauzetosti korisnika	95
1.6.3 Realizovana okruženja i primene u potrošačkoj elektronici	96
1.7 Doprinosi disertacije u odnosu na oblast.....	97

POGLAVLJE 2. ALGORITMI ZA OTKRIVANJE KONTEKSTA..... 100

2.1	Osnove algoritma za semantizaciju konteksta	100
2.1.1	Semantizacija konteksta zasnovana na polisama	101
2.1.2	Automatska semantizacija konteksta	104
2.2	Proširenja algoritma za semantizaciju konteksta.....	107
2.2.1	Jednodimenzioni P-model	108
2.2.2	Višedimenzioni P-model	112
2.2.3	Personalizacija semantizacije konteksta	115
2.3	Algoritam za procenu zauzetosti korisnika	117
2.4	Fizička realizacija algoritama	120
2.4.1	Skript jezici za realizaciju kontekstualnih polisa.....	122
2.4.1.1	Lua, XML i SMIL.....	123
2.4.1.2	Usporedna analiza primene skript jezika u okviru kontekstualne platforme	125
2.4.1.3	Kontekstualne polise u XML jeziku	132
2.4.1.4	Gramatika kontekstualne polise	135
2.4.1.5	Semantika kontekstualne polise	137
2.4.1.6	Fizička realizacija algoritama	140

POGLAVLJE 3. MODEL ARHITEKTURE PLATFORME..... 142

3.1	Analiza zahteva za kontekstualnu platformu.....	142
3.1.1	Zahtevi kontekstualne platforme po Dey.....	143
3.1.2	Heuristike za prihvatanje rešenja od strane korisnika.....	143
3.1.3	Funkcionalni zahtevi.....	146
3.1.3.1	Tehnički zahtevi	147
3.1.3.2	Semantizacija konteksta.....	148
3.1.3.3	Podrška za senzore	149
3.1.3.4	Prvobitno i naknadno podešavanje.....	150
3.1.3.5	Korisnička sprega.....	151
3.1.3.6	Podrška za aktuatore	152
3.1.3.7	Podrška za aplikacije.....	153
3.1.3.8	Povezivanje sa okruženjem.....	154
3.1.3.9	Sertifikacioni zahtevi	154
3.1.3.10	Ostali zahtevi	154
3.2	Slučajevi korišćenja.....	155

3.2.1	Strane učesnice (<i>actors</i>).....	155
3.2.2	Slučajevi korišćenja iz grupe 1 – Obavljanje redovnih aktivnosti	156
3.2.3	Slučajevi korišćenja iz grupe 2 – Uvid u stanje rada platforme	158
3.2.4	Slučajevi korišćenja iz grupe 3 – Podešavanje rada platforme	160
3.2.5	Slučajevi korišćenja iz grupe 4 – Treniranje platforme	162
3.2.6	Slučajevi korišćenja iz grupe 5 – Upotreba kontekstualnih aplikacija.....	163
3.2.7	Slučajevi korišćenja iz grupe 6 – Pristup zaključcima platforme	165
3.2.8	Slučajevi korišćenja iz grupe 7 – Prilaganje dodatnih informacija	166
3.2.9	Slučajevi korišćenja iz grupe 8 – Pristup sensorima i aktuatorima	167
3.2.10	Slučajevi korišćenja iz grupe 9 – Prikaz u grafičkoj korisničkoj sprezi.....	169
3.3	Arhitektura kontekstualne platforme.....	170
3.3.1	Pregled fizičke arhitekture	173
3.3.1.1	Centralna kontekstualna platforma.....	176
3.3.1.2	Kontekstualni moduli	177
3.3.1.3	Prezentacioni moduli.....	179
3.3.2	Arhitektura programske podrške kontekstualne platforme	180
3.3.2.1	Nivo fizičkih događaja	183
3.3.2.2	Nivo kontekstualnih parametara.....	186
3.3.2.3	Nivo semantizacije	188
3.3.2.4	Nivo agregacije	194
3.3.2.5	Fizički mrežni nivo.....	197
3.3.2.6	Simbolički nivo	202
3.3.2.7	Nivo adaptacije.....	205
3.3.2.8	Nivo akcionih modela	208
3.3.2.9	Kontekstualna programska sprega (CTX API).....	210
3.3.2.10	Aplikativni nivo	211
3.3.2.11	Nivo korisničke sprege.....	222
3.3.3	Neke važne sekvence u okviru kontekstualne platforme	224
3.3.4	Pomoćna programska podrška	230
	POGLAVLJE 4. PRIMER REALIZACIJE.....	232
4.1	User Awareness Kit	232
4.1.1	Fizička arhitektura kontekstualne platforme UAK	233
4.1.2	UAK ekosistem.....	236
4.1.3	Rukovaoci uređajima	237
	POGLAVLJE 5. EKSPERIMENTALNI REZULTATI I DISKUSIJE	240

Sadržaj

5.1	Merenje performansi rešenja	240
5.1.1	Metodologija merenja performansi rešenja	241
5.1.2	Rezultati merenja performansi rešenja	243
5.1.3	Diskusija i zaključci.....	247
5.2	Merenje odziva rešenja	248
5.2.1	Metodologija merenja odziva rešenja	249
5.2.2	Rezultati merenja odziva rešenja	253
5.2.3	Poređenje vremena odziva sa postojećim rešenjima.....	262
5.2.4	Diskusija i zaključci.....	264
5.3	Merenje tačnosti semantizacije	265
5.3.1	Metodologija merenja tačnosti semantizacije.....	265
5.3.2	Rezultati merenja tačnosti semantizacije.....	267
5.3.3	Diskusija dobijenih rezultata	272
5.3.4	Poređenje rezultata semantizacije sa postojećim rešenjima.....	273
5.4	Uporedni prikaz rešenja po potencijalu za razvoj kontekstualnih aplikacija	274
5.4.1	Metodologija evaluacije potencijala za razvoj kontekstualnih aplikacija	275
5.4.2	Rezultati evaluacije potencijala za razvoj kontekstualnih aplikacija	286
5.4.2.1	Rezultati evaluacije po kriterijumima	286
5.4.2.1	Rezultati pojedinačne evaluacije.....	293
5.4.2.2	Rezultati evaluacije prema ukupnoj oceni	299
POGLAVLJE 6. PRIMERI PRIMENE REŠENJA.....		300
6.1	Kontekstualne korisničke sprege	300
6.1.1	Sprega za privlačenje pažnje korisnika.....	300
6.1.2	Sprega za objavljivanje zauzetosti korisnika	303
6.2	Kontekstualne aplikacije	304
6.2.1	Komunikacioni klijent	305
6.3	Ostale komercijalne primene.....	308
POGLAVLJE 7. ZAKLJUČAK		310
LITERATURA		314
BIOGRAFIJA.....		344

PRILOG: SENZORSKE TEHNOLOGIJE.....346

Sadržaj

SPISAK SLIKA

Slika 1. Prikaz slojevite arhitekture kontekstualne platforme	48
Slika 2. Tipična arhitektura nivoa mreže kontekstualne platforme	53
Slika 3. Arhitektura srednjeg nivoa zasnovanog na agentima.....	55
Slika 4. Arhitektura reflektivnog srednjeg nivoa	56
Slika 5. Arhitektura srednjeg nivoa zasnovanog na metapodacima	57
Slika 6. Arhitektura srednjeg nivoa zasnovanog na prostoru uređenih n-torki	58
Slika 7. Arhitektura adaptivnog srednjeg nivoa	59
Slika 8. Tipična arhitektura kontekstualne platforme zasnovane na OSGI okruženju	60
Slika 9. Aplikacija za prodavnice: ekran na osnovu lokacije ukazuje na najbliže proizvode i omogućava pregled i poređenje (Black, 2009, str. 2).....	66
Slika 10. Adaptivna korisnička sprega za odabir muzičkog sadržaja, istaknutog u zavisnosti od sličnosti albuma i afiniteta korisnika (Strober, 2010, str. 388)....	70
Slika 11. Opipljive korisničke sprege: Video editor (desno) (Zigelbaum, 2007, str. 2) i <i>Tern</i> , slagalica za programiranje (levo) (Horn, 2007, str. 2).....	72
Slika 12. Izgled Control4 <i>Composer</i> aplikacije.....	79
Slika 13. Odeljak <i>Programming</i> u <i>Composer</i> aplikaciji.....	80
Slika 14. Izgled Control4 <i>Navigator</i> aplikacije	81
Slika 15. Arhitektura Control4 sistema preslikana na zahteve kontekstualnih platformi	82

Spisak slika

Slika 16. Izgled ekrana <i>Vacation</i> aplikacije (gornji red) i <i>Wakeup</i> aplikacije (donji red) na uređaju sa ekranom osjetljivim na dodir u okviru <i>Control4 Navigator</i> aplikacije.....	83
Slika 17. <i>Kinect</i> uređaj sa označenim sensorima (slajd sa Microsoft E3 konferencije).....	87
Slika 18. <i>Playstation 3</i> igračka konzola i <i>Playstation Move</i> uređaj	88
Slika 19. Korisničke sprege i aplikacije za augmentiranu realnost: a) TwittARound, b) Layar, c) NearestTube, d) TAT Augmented ID, e) Wikitude AR Travel Guide i f) CarFinder.....	90
Slika 20. UML dijagram aktivnosti algoritma za automatsko podešavanje verovatnoća stanja okruženja.....	105
Slika 21. Ilustracija ažuriranja verovatnoće stanja okruženja nakon tri uzastopne deklaracije (iterativni postupak)	107
Slika 22. Grafik P-modela za stanje okruženja „korisnik posmatra TV program“ u zavisnosti od kontekstualnog parametra sistemskog vremena t	110
Slika 23. Ukupan broj korisnika sa priključenim <i>Skype</i> klijentom u toku dana (Borderless, 2008).....	111
Slika 24. Grafik P-modela za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ u zavisnosti od kontekstualnog parametra sistemskog vremena t , sa verovatnoćama za svaki sat korišćenja. 111	
Slika 25. Grafik P-modela za stanje okruženja „korisnik koristi komunikacionu aplikaciju na TV prijemu“ (siva površina na slici), generisan na osnovu međuzavisnosti P-modela za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ (isprekidana linija) i P-modela za stanje okruženja „korisnik posmatra TV program“ (puna linija).	112
Slika 26. Grafici polinoma za verovatnosni otklon verovatnoće stanja okruženja „korisnik posmatra TV program“ u zavisnosti od godina starosti korisnika, za korisnike muškog pola (isprekidana linija) i za korisnike ženskog pola (puna linija).....	114
Slika 27. Grafik višedimenzionog P-modela za stanje okruženja „korisnik posmatra TV program“ u zavisnosti od kontekstulnih parametara sistemskog vremena (t)	

i godina starosti korisnika (x), za korisnike muškog pola (levo) i za korisnike ženskog pola (desno)	114
Slika 28. UML dijagram aktivnosti algoritma za direktni metod saopštavanja cene asinhronog zahteva	120
Slika 29. Primerni skript pisan u jeziku Lua	124
Slika 30. Primerni skript pisan u jeziku XML.....	125
Slika 31. Primerni skript pisan u jeziku SMIL	125
Slika 32. Arhitektura sistema korišćenog za procenu upotrebljivosti skript jezika	126
Slika 33. Moduli programske podrške korišćenog sistema sa izmenama potrebnim za procenu upotrebljivosti skript jezika.....	126
Slika 34. Brzina odziva na događaj (u ciklusima procesora)	129
Slika 35. Brzina pokretanja skripta (u ciklusima procesora).....	129
Slika 36. Kompleksnost sintakse za skriptove pisane u jezicima Lua, XML I SMIL (prva tri reda – Halstead metrike).....	130
Slika 37. Primer jedne jednostavne kontekstualne polise	133
Slika 38. Gramatika kontekstualne polise u EBNF notaciji	135
Slika 39. Sintaksni dijagram kontekstualne polise (1)	136
Slika 40. Sintaksni dijagram kontekstualne polise (2)	137
Slika 41. Grupe slučajeva korišćenja kontekstualne platforme	155
Slika 42. Slučajevi korišćenja u grupi SK1 – Obavljanje redovnih aktivnosti.....	156
Slika 43. Slučajevi korišćenja u grupi SK2 – Uvid u stanje rada platforme	158
Slika 44. Slučajevi korišćenja u grupi SK3 – Podešavanje rada platforme.....	160
Slika 45. Slučajevi korišćenja u grupi SK4 – Treniranje platforme.....	163
Slika 46. Slučajevi korišćenja u grupi SK5 – Upotreba kontekstualnih aplikacija	164
Slika 47. Slučajevi korišćenja u grupi SK6 – Pristup zaključcima platforme.....	165
Slika 48. Slučajevi korišćenja u grupi SK7 – Prilaganje dodatnih informacija	166
Slika 49. Slučajevi korišćenja u grupi SK8 – Pristup senzorima i aktuatorima	168
Slika 50. Slučajevi korišćenja u grupi SK9 – Prikaz u grafičkoj korisničkoj sprezi	169
Slika 51. Pogled na kontekstualnu platformu sa najvišeg nivoa hijerarhije.....	171

Spisak slika

Slika 52. Osnovni elementi fizičke arhitekture kontekstualne platforme i njihova ožičenja	174
Slika 53. Blok dijagram fizičke arhitekture centralne kontekstualne platforme ..	177
Slika 54. Blok dijagram fizičke arhitekture kontekstualnog modula tipa A.....	178
Slika 55. Blok dijagram fizičke arhitekture kontekstualnog modula tipa B.....	179
Slika 56. Blokovska i nivojska podela programske podrške na najvišem nivou hijerarhije	181
Slika 57. Nivojska organizacija bloka za semantizaciju.....	181
Slika 58. Nivojska organizacija bloka za kontekstualizaciju.....	182
Slika 59. Blok dijagram programske podrške nivoa fizičkih događaja	183
Slika 60. UML dijagram klasa za realizaciju nivoa fizičkih događaja	184
Slika 61. Blok dijagram programske podrške nivoa kontekstualnih parametara .	186
Slika 62. UML dijagram klasa za realizaciju nivoa kontekstualnih parametara ..	187
Slika 63. Blok dijagram programske podrške nivoa semantizacije	189
Slika 64. UML dijagram klasa modula za upravljanje kontekstualnim parametrima, modula za upravljanje P-modelima i modula za automatsku semantizaciju ..	190
Slika 65. UML dijagram klasa modula za evaluaciju polisa i konfiguracionog modula	192
Slika 66. UML dijagram klasa modula za objavu verovatnoće stanja i modula za treniranje	193
Slika 67. Blok dijagram programske podrške nivoa agregacije	194
Slika 68. UML dijagram klasa agregacionog modula za procenu zauzetosti korisnika i modula za upravljanje agregacijom	195
Slika 69. UML dijagram klasa modula za upravljanje okruženjima	196
Slika 70. Blok dijagram programske podrške fizičkog mrežnog nivoa.....	198
Slika 71. UML dijagram klasa koje realizuju senzorski i aktuatorski SDK i senzorsku i aktuatorsku spregu	199
Slika 72. UML dijagram klasa modula za upravljanje uređajima i modula za upravljanje događajima.....	201
Slika 73. Blok dijagram programske podrške simboličkog nivoa.....	202
Slika 74. UML dijagram klasa modula za evidenciju događaja, modula za evidenciju uređaja i modula za evidenciju okruženja.....	203

Slika 75. UML dijagram klasa modula za upravljanje entitetima	204
Slika 76. Blok dijagram programske podrške nivoa adaptacije	205
Slika 77. UML dijagram klasa modula za adaptaciju.....	206
Slika 78. Izvod iz Lua skripta kojim se realizuje jedan akcioni model	209
Slika 79. Blok dijagram programske podrške nivoa akcionih modela.....	210
Slika 80. UML dijagram klasa kontekstualne programske sprege	211
Slika 81. Blok dijagram programske podrške aplikativnog nivoa	212
Slika 82. UML dijagram klasa modula za integraciju sa sistemskom spregom....	213
Slika 83. Ilustracija sintakse primitiva za uspostavljanje i raskidanje veze, kao i primitive za dobavljanje spiska senzora	214
Slika 84. Primer jednog HTTP zahteva i odgovora modula za integraciju sa drugim sistemima.....	215
Slika 85. UML dijagram sekvence pri uspostavljanju i obradi HTTP API zahteva i isporuci odgovora	216
Slika 86. UML dijagram klasa modula za upravljanje ekranima	217
Slika 87. UML dijagram klasa modula za interpretaciju i modula za upravljanje aplikacijama.....	218
Slika 88. Funkcija <i>LampFade</i> iz kontekstualne aplikacije <i>Luminosity</i>	219
Slika 89. Funkcija <i>LampUpdate</i> iz kontekstualne aplikacije <i>Luminosity</i>	220
Slika 90. Funkcija <i>StateMachine</i> iz kontekstualne aplikacije <i>Luminosity</i>	221
Slika 91. Ulazna tačka kontekstualne aplikacije <i>Luminosity</i>	222
Slika 92. Ilustracija ažuriranja korisničke sprege na prezentacionoj jedinici sa centralne kontekstualne platforme.....	223
Slika 93. UML dijagram sekvence za ilustraciju mehanizma pokretanja kontekstualne aplikacije iz sistemske korisničke sprege.....	225
Slika 94. UML dijagram sekvence za izračunavanje stepena zauzetosti korisnika	226
Slika 95. UML dijagram sekvence za izračunavanje nove verovatnoće stanja.....	227
Slika 96. UML dijagram sekvence za ažuriranje režima rada grafičke aplikacije na prezentacionoj jedinici u zavisnosti od verovatnoće stanja okruženja.....	228
Slika 97. UML dijagram sekvence za deklaraciju novog stanja od strane posmatrača korišćenjem Android aplikacije na mobilnom telefonu	229

Spisak slika

Slika 98. Namenski računar kojim je realizovana centralna kontekstualna platforma	235
Slika 99. UAK ekosistem	236
Slika 100. Linux skript korišćen za merenje performansi sistema	243
Slika 101. Rezultati merenja vremena pokretanja sistema: na X osi je prikazan redni broj merenja, dok je na Y osi prikazano vreme u sekundama (plava boja – vreme pokretanja celokupnog uređaja i programske podrške; crvena boja – vreme pokretanja aplikacije kontekstualne platforme).....	244
Slika 102. Poređenje vremena pokretanja sistema predložene platforme i platformi trenutno dostupnih na tržištu: na X osi je prikazan redni broj merenja, dok je na Y osi prikazano vreme u sekundama	244
Slika 103. Rezultati merenja količine zauzete memorije: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan utrošak memorije (u Kb)	245
Slika 104. Rezultati merenja broja pokrenutih niti. na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan broj pokrenutih niti	245
Slika 105. Rezultati merenja broja otvorenih datoteka: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan broj otvorenih datoteka	246
Slika 106. Rezultati merenja zauzetosti procesora: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazano zauzeće procesora (u %)	246
Slika 107. Struktura vremena potrebnog za pokretanje modernog digitalnog TV prijemnika (Jo, 2009).....	247
Slika 108. Izgled ispitnog Okruženja A: simulacija dnevne sobe sa postavljenim senzorima (optička kamera, mikrofonski niz, 3D kamera) i prezentacionom jedinicom priključenom na TV prijemnik	251
Slika 109. Izgled ispitnog Okruženja B: Scena za <i>Luminosity</i> aplikaciju sa postavljenom 3D kamerom kao jedinim senzorom	252
Slika 110. Vreme odziva rešenja za ispitni Slučaj A: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama).....	253
Slika 111. Jedan primer frejma za početak merenja odziva za ispitni Slučaj A...	254
Slika 112. Jedan primer frejma za završetak merenja odziva za ispitni Slučaj A	255
Slika 113. Očekivana raspodela odziva rešenja u ispitnom slučaju A	256

Slika 114. Vreme odziva rešenja za ispitni Slučaj B: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama)	257
Slika 115. Jedan primer frejma za početak merenja odziva za ispitni Slučaj B....	257
Slika 116. Jedan primer frejma za završetak merenja odziva za ispitni Slučaj B.	258
Slika 117. Očekivana raspodela odziva rešenja u ispitnom slučaju B.....	258
Slika 118. Vreme odziva rešenja za ispitni Slučaj C sa jednim učesnikom na sceni: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama), za slučaj kretanja unutar scene (plavo) i za slučaj ulaska na scenu (crveno).....	259
Slika 119. Vreme odziva rešenja za ispitni Slučaj C sa dva učesnika na sceni: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama), za slučaj kretanja unutar scene (plavo) i za slučaj ulaska na scenu (crveno).....	260
Slika 120. Šest uzastopnih frejmova iz ispitnog Slučaja C: Na slici se vidi zavisnost osvetljenosti lampe od ulaska korisnika u zonu lampe u svakom frejmu	261
Slika 121. Poređenje vremena odziva na naredbu za uključivanje svetla predloženog rešenja (crveno) i sistema <i>Control4</i> (plavo): na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama).....	263
Slika 122. Simulirano okruženje dnevne sobe.....	267
Slika 123. Dobijeni nivoi zauzetosti u toku poslepodneva eksperimentalnog dana u simuliranom okruženju dnevne sobe (10 – minimalna zauzetost, 0 – maksimalna zauzetost). Crvena linija označava prosečan nivo zauzetosti dobijen od strane posmatrača. Isprekidana crna linija označava prosečan nivo zauzetosti dobijen samoprocenom učesnika, dok plava linija označava vrednosti izračunate od strane platforme.....	268
Slika 124. Nivoi zauzetosti izračunati od strane platforme u toku poslepodneva eksperimentalnog dana u simuliranom okruženju dnevne sobe (10 – minimalna zauzetost, 0 – maksimalna zauzetost). Zelena linija označava rezultat zasnovan na Postavci 1. Žuta linija označava rezultat zasnovan na Postavci 2. Siva linija označava rezultat zasnovan na Postavci 3.	268

Spisak slika

Slika 125. Zavisnost vrednosti nivoa tolerancije ε (x osa) i uspešnosti prepoznavanja nivoa zauzetosti korisnika (y osa) za svako od stanja okruženja	271
Slika 126. Grafički prikaz stepena ispunjenosti prvog kriterijuma (<i>make vs buy</i>): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo	287
Slika 127. Grafički prikaz stepena ispunjenosti drugog kriterijuma (<i>make vs buy</i>): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo	288
Slika 128. Grafički prikaz stepena ispunjenosti trećeg kriterijuma (<i>make vs buy</i>): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo	289
Slika 129. Grafički prikaz stepena ispunjenosti četvrtog kriterijuma (<i>make vs buy</i>): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo	290
Slika 130. Grafički prikaz stepena ispunjenosti petog kriterijuma (<i>make vs buy</i>): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo	292
Slika 131. Grafički prikaz stepena ispunjenosti šestog kriterijuma (cena): zelenom bojom prikazan je nivo efikasnosti cene – što je nivo veći, cena je niža	293
Slika 132. Grafički prikaz pojedinačne evaluacije za okruženje Control4.....	293
Slika 133. Grafički prikaz pojedinačne evaluacije za okruženje Adhoco	294
Slika 134. Grafički prikaz pojedinačne evaluacije za okruženje AMX.....	294
Slika 135. Grafički prikaz pojedinačne evaluacije za okruženje Crestron	295
Slika 136. Grafički prikaz pojedinačne evaluacije za okruženje Oblo.....	295
Slika 137. Grafički prikaz pojedinačne evaluacije za okruženje Philips Retail Solutions	296
Slika 138. Grafički prikaz pojedinačne evaluacije za okruženje Subtle.....	296
Slika 139. Grafički prikaz pojedinačne evaluacije za okruženje SOCAM.....	297
Slika 140. Grafički prikaz pojedinačne evaluacije za okruženje CORTEX.....	297
Slika 141. Grafički prikaz pojedinačne evaluacije za okruženje Gaia	298
Slika 142. Grafički prikaz pojedinačne evaluacije za predloženo rešenje.....	298

Slika 143. Rezultati evaluacije rešenja prema ukupnoj oceni	299
Slika 144. Ilustracija prelaza stanja u algoritmu rada aplikacije <i>Luminosity</i>	301
Slika 145. Prikaz rada aplikacije <i>Luminosity</i> u laboratorijskim uslovima	302
Slika 146. Ilustracija prelaza stanja u algoritmu rada aplikacije <i>LightBeat</i>	303
Slika 147. Izgled kontekstualne korisničke sprege za objavljivanje zauzetosti korisnika	304
Slika 148. Izgled komunikacione aplikacije i situacija u režimu <i>Aktivan</i>	306
Slika 149. Izgled komunikacione aplikacije i situacija u režimu <i>Pasivan</i>	306
Slika 150. Izgled komunikacione aplikacije i situacija u režimu <i>Isključen</i>	307
Slika 151. Izgled komunikacione aplikacije i situacija u režimu <i>Isključen</i> , sa korisnikom koji na zahtev izaziva prikaz spiska propuštenih obaveštenja	307

Spisak slika

SPISAK TABELA

Tabela 1. Tipovi fizičkih senzora u upotrebi	51
Tabela 2. Istraživanja u kojima se realizuju algoritmi sa mogućnošću primene u logičkim sensorima	52
Tabela 3. Pregled istraživanja algoritama zaključivanja sa primenom u kontekstualnim platformama	62
Tabela 4. Sistemi za automatizaciju domaćinstva kao osnova kontekstualnih platformi	78
Tabela 5. Koeficijenti polinoma koji predstavlja P-model za stanje okruženja “korisnik posmatra TV program”	109
Tabela 6. Koeficijenti polinoma koji predstavlja P-model za stanje okruženja “korisnik ima potrebu za korišćenjem komunikacione aplikacije”	111
Tabela 7. Koeficijenti polinoma za verovatnosni otklon verovatnoće stanja okruženja “korisnik posmatra TV program” za korisnike muškog pola (γ) i ženskog pola (δ)	113
Tabela 8. Primer tabele za određivanje cene asinhronog zahteva u zavisnosti od stanja okruženja	119
Tabela 9. Primenljivost skript jezika u okviru kontekstualne platforme	131
Tabela 10. Opis semantike kontekstualne polise	139
Tabela 11. Minimalni zahtevani parametri centralnog kontrolera	176
Tabela 12. Minimalni zahtevani parametri mikrokontrolera	178
Tabela 13. Kontekstualizaciona proširenja Lua jezika	208

Spisak tabela

Tabela 14. Fizičke karakteristike realizovane centralne kontekstualne platforme	234
Tabela 15. Fizičke karakteristike realizovanog kontekstualnog modula tipa A .	235
Tabela 16. Realizovane senzorske biblioteke za UAK kontekstualnu platformu	237
Tabela 17. Realizovane aktuatorske biblioteke za UAK kontekstualnu platformu	237
Tabela 18. Realizovane senzorske biblioteke za UAK kontekstualnu platformu	238
Tabela 19. Primeri tipičnih situacija koje je moguće detektovati korišćenjem rukovalaca sensorima, virtuelnih senzora i kontekstualnih polisa	239
Tabela 20. Statistički pokazatelji merenja za ispitni Slučaj A	255
Tabela 21. Statistički pokazatelji merenja za ispitni Slučaj B	256
Tabela 22. Statistički pokazatelji merenja za ispitni Slučaj C	260
Tabela 23. Statistički pokazatelji poređenja vremena odziva	263
Tabela 24. Srednja apsolutna greška i uspešnost prepoznavanja stepena zauzetosti	268
Tabela 25. Spisak stanja okruženja u posmatranom periodu identifikovanih od strane posmatrača.....	269
Tabela 26. Matrica konfuzije procene nivoa zauzetosti u zavisnosti od stanja okruženja datog samoprocenom i izveštajem posmatrača.....	270
Tabela 27. Zavisnost tolerancije ϵ i uspešnosti prepoznavanja stanja zauzetosti, od zahtevanog broja diskretnih stanja zauzetosti.....	271
Tabela 28. Ocene evaluacije zahteva kontekstualne platforme po Dey	287
Tabela 29. Ocene evaluacije opštih zahteva kontekstualne platforme.....	288
Tabela 30. Ocene evaluacije integrabilnosti i primenljivosti za razvoj kontekstualnih aplikacija potrošačke elektronike	289
Tabela 31. Ocene evaluacije podrške za senzore	290
Tabela 32. Ocene evaluacije podrške za aktuatore	291
Tabela 33. Ocene evaluacije cene rešenja.....	292

SKRAĆENICE

AI – Artificial Intelligence
API – Application Programming Interface
AS – Availability Score
CCD – Charge Coupled Device
CES – Consumer Electronics Show
CGI – Common Gateway Interface
CPL – Call Processing Language
CRUD – Create, Read, Update and Delete
CSS – Cascading Style Sheets
DALI – Digital Addressable Lighting Interface
DC – Direct Current
DDR – Double Data Rate
DK – Development Kit
DMX – Digital MultipleX
EAN – European (International) Article Number
EBNF – Extended Backus – Naur Form
EC – Energy Controller
ECI – Estimated Cost of Interruption
EPG – Electronic Program Guide
FSM – Finite State Machine
GND - Ground
GPRS – General Packet Radio Service
GPS – Global Positioning System
GSM – Global System for Mobile communications
GTK – Gimp ToolKit
GWT – Google Web Toolkit
HC – Home Controller
HCI – Human Computer Interaction
HDMI – High Definition Multimedia Interface
HTML – HyperText Markup Language
HTTP – HyperText Transfer Protocol

Skraćenice

HVAC – Heating, Ventillation and Air Conditioning
IEEE – Institute of Electrical and Electronics Engineers
IP – Internet Protocol
IR – Infra Red
JPEG – Joint Photographic Experts Group
JSON – JavaScript Object Notation
LED – Light Emitting Diode
MAE – Mean Absolute Error
MIT – Massachusetts Institute of Technology
NIR – Near Infra Red
OS – Operating System
OSGI – Open Services Gateway Initiative
OTS – Off The Shelf
PID – Process IDentifier
PIR – Passive Infra Red
PHP – Hypertext Preprocessor
PMU – Power Management Unit
PoE – Power over Ethernet
POSIX – Portable Operating System Interface
PWM – Pulse Width Modulation
QoC – Quality of Context
QR code – Quick Response code
RF4CE – Radio Frequency for Consumer Electronics
RFID – Radio Frequency Identifier
RGB – Red Green Blue
ROLL – Routing Over Low power and Lossy networks
RSSI – Received Signal Strength Indication
RTC – Real Time Clock
SDK – Software Development Kit
SEA2M – Speech Enhancement Algorithms for Array of Microphones
SIM – Subscriber Identity Module
SMIL – Synchronized Multimedia Integration Language
SMS – Short Message Service
STB – Set-Top Box
TCP – Transmission Control Protocol
TOF – Time of flight
TVE – Tangible Video Editor
UAK – User Awareness Kit
UI – User Interface
UML – Unified Modelling Language
UPnP – Universal Plug and Play
USB – Universal Serial Bus
UMTS – Universal Mobile Telecommunications System
UHF – Ultra-High Frequency
UV – Ultra violet
WLAN – Wireless Local Area Network
XML – eXtensible Markup Language

REZIME

U okviru doktorske disertacije date su metode i algoritmi za realizaciju kontekstualnih platformi i kontekstualnih korisničkih sprega za primene u uređajima potrošačke elektronike. Osnovna motivacija za rad na ovakvoj temi leži u činjenici da se prilikom dizajna modernih uređaja potrošačke elektronike poklanja nedovoljna pažnja aspektima kao što su upotrebljivost (*usability*), stanje fizičkog okruženja uređaja, rad na osnovu dostupnosti pažnje i zauzetosti korisnika, interakcija sa drugim uređajima, kao i prilagođavanje osobinama korisnika, njegovim afinitetima i stilu upotrebe. Današnji tipičan uređaj potrošačke elektronike je samocentričan, nedovoljno svestan okruženja, direktno ili indirektno zahteva čestu interakciju sa korisnikom i njegovu pažnju, doprinoseći povećanju stresa, smanjenju produktivnosti i odvajanju korisnika od fizičkog okruženja u korist virtuelnog okruženja nametnutog novim tehnološkim mogućnostima predvođenim globalnom mrežom - Internetom.

Disertacija najpre daje pregled stanja u oblasti kontekstualnih platformi, sa posebnim osvrtom na oblast potrošačke elektronike i kontekstualne aplikacije za uređaje potrošačke elektronike. U okviru pregleda izdvojeni su elementi zajednički za mnoga istraživanja, u cilju predstavljanja tipične arhitekture kontekstualne platforme i trenutnih rešenja za pojedinačne blokove te arhitekture. Posebna pažnja posvećena je pregledu kontekstualnih korisničkih sprega, koje pripadaju nekim od najnovijih primena kontekstualnih platformi. Takođe, dat je pregled oblasti upravljanja asinhronim zahtevima, čime su upravljanje tim zahtevima u zavisnosti od trenutnog stanja zauzetosti korisnika, kao i procena zauzetosti korisnika, dati kao dodatni zahtevi

Rezime

funktionalnostima kontekstualne platforme, za njenu uspešnu primenu u oblasti potrošačke elektronike.

Algoritmi za otkrivanje konteksta, odnosno za davanje značenja (semantizaciju), predstavljaju značajan deo disertacije. Predložene su dve varijante osnovnog algoritma, koje se oslanjaju na sistem pravila u kombinaciji sa stabilima odlučivanja; prva varijanta koristi unapred pripremljene polise sa definisanim pravilima semantizacije, koje se definišu na osnovu pažljivo dizajniranih eksperimenata, dok se druga varijanta oslanja na samoprocenu korisnika i procenu posmatrača, koji prilažu informacije o stanju okruženja u postupku treniranja platforme. Obe verzije su dalje proširene statističkim modelima (P-modeli), kojima se modeluju korisničke aktivnosti specifične za uređaje potrošačke elektronike i međuzavisnosti više takvih aktivnosti u cilju tačnije procene stanja u okruženju. Takođe su date varijante u kojima algoritmi koriste personalizovane podatke, odnosno ponašaju se u skladu sa identitetom korisnika ili grupe korisnika, kao i varijante u kojima je omogućeno treniranje algoritma za detekciju stepena zauzetosti korisnika za određena stanja okruženja. U okviru evaluacije mogućih načina implementacije algoritama, detaljno su istražene mogućnosti primene skript jezika, na osnovu kojih je izabran jezik XML za realizaciju kontekstualnih polisa, i jezik Lua za realizaciju kontekstualnih aplikacija. Dati su rezultati evaluacije različitih skript jezika u smislu brzine odziva, kompleksnosti sintakse i prilagođenosti zahtevima upotrebe.

Model arhitekture kontekstualne platforme predložen je na osnovu istraživanja postojećih modela, primene novih algoritama za semantizaciju, osobina uređaja potrošačke elektronike i iskustava u realizaciji kontrolera za inteligentna okruženja. Model je izložen sistematično, počevši od heurističkih zahteva koji se odnose na prihvatanje uređaja potrošačke elektronike od strane korisnika, konkretnog skupa zahteva kontekstualne platforme i detaljne specifikacije slučajeva korišćenja koje treba podržati modelom. Dalje, dat je pregled potrebne fizičke arhitekture kontekstualne platforme, kao i detaljan opis blokovske i nivojske arhitekture programske podrške, sa prikazom UML dijagrama klasa i sekvenci gde god je to bilo potrebno. Predloženi model dat je u nivou detalja dovoljnom za komercijalnu realizaciju u celini ili u delovima.

Na osnovu predloženog modela realizovana je konkretna kontekstualna platforma UAK (*User Awareness Kit*). Ova platforma služi kao osnova za ispitivanje funkcionisanja predloženog modela u inteligentnim okruženjima, poput domaćinstva. U okviru evaluacionih eksperimenata ispitana je performansa rešenja, u smislu korišćenja fizičkih resursa uređaja potrošačke elektronike, što je osnovni pokazatelj mogućnosti upotrebe. Eksperimentalno je pokazano da realizovano rešenje radi stabilno, i pokazuje uporedivu ili veću brzinu izvršavanja u odnosu na srodne sisteme. Dalje, ispitana je brzina odziva rešenja na različite korisničke akcije u inteligentnom okruženju, gde je pokazana zadovoljavajuća brzina odziva za većinu ispitivanih slučajeva. Konačno, dati su rezultati merenja tačnosti semantizacije za okruženje dnevne sobe, gde je pokazano da postignuta tačnost merenja nivoa zauzetosti korisnika u prostoriji nadmašuje dosadašnja rešenja.

Na osnovu realizovane kontekstualne platforme kreirano je nekoliko kontekstualnih korisničkih sprega i kontekstualnih aplikacija. Kontekstualne korisničke sprege koriste svetlosne efekte u boji, da bi privukle pažnju korisnika ka određenom elementu inteligentnog okruženja, odnosno da bi objavile drugim korisnicima nivo zauzetosti korisnika u inteligentnom okruženju. Takođe, realizovana je komunikaciona aplikacija za TV prijemnik, u okviru koje su režimi rada postavljeni automatski, na osnovu stanja okruženja u prostoriji. Komercijalna rešenja nastala na osnovu predloženog modela pokazuju uspešnu primenu na osnovu prvih izveštaja korisnika i drugih pokazatelja prikupljenih tokom pilot rada sistema u realnom okruženju.

U okviru disertacije je pokazana mogućnost primene predloženih metoda za realizaciju kontekstualnih platformi i kontekstualnih korisničkih sprega za sisteme i uređaje potrošačke elektronike. S obzirom na sveobuhvatan prilaz izlaganju arhitekture kontekstualnih platformi, predloženi model može biti korišćen kao osnova za buduće realizacije novih aplikacija i uređaja potrošačke elektronike.

SUMMARY

In this thesis methods and algorithms for the implementation of context-aware platforms and context-aware user interfaces for consumer electronic devices are given. The motivation for working on this topic came from the fact that current design of modern consumer electronic devices fails to address vital issues such as usability, device physical context, user attention and availability, interaction with other devices in the environment and tailoring the operation to meet the characteristics of individual users, their affinities and usage patterns. Consumer electronic devices of today are overwhelmingly self-centric, mostly unaware of the environment. They tend to require frequent user interaction and to seek user attention, increasing user stress and decreasing his overall productivity. Moreover, this situation further detaches the user from his physical environment, reinforcing his virtual environment constructed with the help of modern technological advancements led by the global network – the Internet.

The thesis first gives an overview of the current state of the art in the development of context-aware platforms. Specific interest in this review is within the area of consumer electronics, with the emphasis on the context-aware applications for consumer electronic devices. The review seeks to derive building blocks that are common for many previous researches, in order to present a typical architecture of a context-aware platform including solutions for specific blocks in that architecture. Further, the thesis analyzes the available context-aware user interfaces, which are among the newest applications of context-aware computing. Review is extended to

Summary

include the problems of mediating interruptions, with the goal to set the detection of human availability and susceptibility to undesired interruptions among the newest requirements for context-aware platforms in consumer electronic devices.

Algorithms for the detection of context and giving the meaning to the parameters of context are a significant part of the thesis. A basic algorithm is proposed, based on the set of rules combined with decision trees, including two variants of the algorithm. The first variant is based on the predefined context-aware policies which define rules for context deduction. These policies are fabricated based on the specifically designed experiments. The second variant uses the self-assessment of participating users and the reports given by the observers, issuing information about the state of the environment during the training period. Both variants are further extended to include statistical probability models (P-models), which are used to accurately express those user activities specific for consumer electronic devices, especially when interleaved with other activities. Additionally, extensions are given to account for the difference among specific users or user groups, and to provide means to calculate the availability score of monitored users. The evaluation of different implementation methods of these algorithms is also given, including the application of script languages for the definition of context-aware policies and applications. Based on the detailed evaluation of the different script languages including the response time, syntax complexity and their applicability in context-aware computing, XML language for the definition of context-aware policies, and Lua language for programming context-aware applications have been selected.

Proposed model of a context-aware platform is based on a research on existing models, new algorithms for context deduction, specifics of consumer electronic devices and experiences in the implementation of different controllers for smart spaces. Model is presented using a systematic approach, starting with the enumeration of all needed heuristics which are used to provide the acceptance of new consumer electronic device or application by users. Then, a specific set of functional requirements and use cases are given. Further, the needed hardware architecture is presented, as well as the detailed description of all needed layers and blocks of software. The descriptions are followed by UML class and sequence diagrams

wherever suitable. The level of detail of the description of the proposed model is sufficient to facilitate a commercial implementation as a whole or in parts.

Based on the proposed model a specific context-aware platform UAK (*User Awareness Kit*) is implemented. This platform further serves as a testing platform, to scrutinize the functionalities and performance of the proposed model within smart spaces common to the users of consumer electronic devices, such as residential households. Within the scope of evaluation experiments first the performance of the solution was assessed, in order to ensure the stable use of physical resources, what is the first gate in providing feasibility for the use in consumer electronics. Experiments showed the stability of operation after a long run, whereas the speed of execution was comparable or exceeded that of the similar existing solutions. Further, the response time was assessed for the different user actions within a smart space, and the sufficient response times were calculated for most of the assessed use cases. Finally, the results on the accuracy of context deduction were given, showing that the accuracy of user availability estimation is higher than in the previous research.

The proposed platform enabled the creation of several context-aware user interfaces and context-aware applications. Implemented context-aware user interfaces use color lighting to attract user's attention towards an element within a smart space. Also, lighting effects are used to announce user availability scores and therefore help mediate undesired social interruptions. A context-aware communication client for TV receiver is implemented, which uses context information to set up the most appropriate operation mode. Finally, commercial solutions which emerged based on the proposed model indicate the successful application based on the first customer testimonies and other indicators collected during the course of pilot setup in the real environment.

The thesis shows the possibility of applying the proposed methods towards the successful implementation of context-aware platforms and context-aware user interfaces for systems and devices in the field of consumer electronics. The thesis gives a broad view on the architecture of context-aware platforms, which can be used as a basis for future implementations of new consumer applications and devices.

UVOD

Razvoj funkcionalnosti uređaja potrošačke elektronike (*consumer electronics*) već dugo vremena zavisi uglavnom od razvoja fizičke arhitekture (*hardware*) uređaja, odnosno tehnoloških mogućnosti za njihovu izradu. Prelaskom sa analogne na digitalnu elektroniku, nove mogućnosti uslovljene su povećanjem brzine procesora, povećanjem kapaciteta memorijskih modula za privremeno i trajno smeštanje podataka, smanjivanjem potrošnje energije uređaja (odnosno povećanjem kapaciteta baterija), te smanjivanjem fizičkih dimenzija elektronskih komponenti. Savremeni uređaji potrošačke elektronike danas poseduju procesore sa jednim ili više jezgara koji rade na brzinama takta iznad 1 GHz. Kapacitet memorije za privremeno smeštanje podataka prevazilazi 1 GB, dok memorija za trajno smeštanje podataka ima kapacitet od 16 GB i više. Pojedini mobilni uređaji troše manje od 0.5 W energije u jednom satu korišćenja. Fizičke dimenzije elektronskih komponenti omogućavaju realizaciju uređaja potrošačke elektronike na štampanim pločama veličine svega nekoliko milimetara. Tehnološke mogućnosti podstakle su razvoj velikog broja funkcionalnosti današnjih uređaja potrošačke elektronike, od reprodukcije multimedije, multimedijalnih komunikacija, navigacije, preko pametnih kuća, kontrole i automatizacije sistema, do medicinskih uređaja i aplikacija i sl.

Iako se mnogi proizvođači deklarativno zalažu za pažljiv dizajn uređaja u smislu njegove upotrebljivosti (*usability*), i dalje je primetna tendencija realizacije uređaja po odavno preispitivanom principu *if we build it they will come* – ako napravimo (uređaj) oni (korisnici) će ga koristiti (Markus, 1994) – te da će korisnici biti zadovoljni sve

Uvod

dok su proizvodi zanimljivi, dok donose makar nešto novo u odnosu na prethodnike i ukoliko im je cena prihvatljiva. Na ovo treba dodati i trend objavljivanja novih verzija uređaja u sve kraćim ciklusima vremena, koji često nisu duži od nekoliko meseci. Izgleda da se za slučaj potrošačke elektronike princip odlučivanja o prihvatanju od strane korisnika može dopuniti u *if we build it first, they will come* – ako prvi napravimo (uređaj) oni (korisnici) će ga koristiti. Dakle, naglasak pri razvoju je i dalje postavljen na uređaj, njegovu funkcionalnost i cenu, dok je pažnja prema krajnjem korisniku i dalje nedovoljna. Pa čak i kada se posebno ističu lakoća upotrebe i kvalitet korisničke spege, rešenja i dalje pate od suštinskog problema zanemarivanja korisnika – korisnička sprega je često suviše „samocentrična“, odnosno ne uzima u obzir činjenicu da su korisnici različiti, koriste uređaj u različitim prilikama i na različitim mestima, kao i da veoma verovatno koriste mnoge druge uređaje uporedo.

Današnji korisnik uređaja potrošačke elektronike ima na raspolaganju različite funkcionalnosti, često uz maksimalnu dostupnost tih funkcionalnosti sa mobilnih uređaja koje nosi sa sobom. Iako se trend povećavanja broja funkcionalnosti nastavlja, sve češće se postavlja pitanje gde je granica do koje će korisnici po definiciji prihvatati nove izume u novim uređajima. Portret prosečnog korisnika uređaja potrošačke elektronike uključuje hodanje ulicom, pogleda fiksiranog za ekran mobilnog telefona i slušalicama u ušima. Kod kuće, na godišnjem odmoru, u kancelariji, proveravanje internet stranica, elektronske pošte, socijalnih mreža i sl. postaju opsesija mnogih korisnika, podstičući ih da usmere fokus ka virtuelnom svetu, zanemarujući stvarni kvalitet života i njegove merljive, fizičke parametre. Današnji uređaji potrošačke elektronike podstiču ovakvo ponašanje korisnika. Uređaji sve više upravljaju korisnikom, njegovim vremenom i pažnjom umesto da mu služe usklađeni sa ritmom njegovog svakodnevnog života (Davidow, 2012).

Samocentričnost novih uređaja i novih funkcionalnosti, odnosno njihov nemar prema osobinama korisnika, osobinama okruženja i širim prilikama u kojima se uređaji koriste, predstavljaju tempiranu bombu za tržište potrošačke elektronike. Globalni rast prodaje uređaja potrošačke elektronike, u kome sve veći udeo imaju mobilni uređaji (Research and Markets, 2010) otkriva da je danas primarni potrošač pojedinac, a ne društvena grupa, ili domaćinstvo. Osnovni resursi pojedinca u današnjem užurbanom tempu života su njegovo vreme, kao i količina stresa koju svakodnevno konzumira.

Tako je u najboljem interesu proizvođača novih uređaja potrošačke elektronike da povedu računa o svakom pojedinačnom korisniku – tako da funkcije svojih uređaja prilagode osobinama korisnika, njegovim aktivnostima i njegovom okruženju, odnosno drugim uređajima sa kojima dolazi u dodir. Nespremnost proizvođača da razmišljaju u ovom smeru može doprineti sve većem odbijanju korisnika da prihvate novi proizvod – na čemu se inače zasniva uspeh na tržištu potrošačke elektronike. Ukoliko korisnika posmatramo sa stanovišta ograničenih resursa koje poseduje (vreme, pažnja, koncentracija, raspoloženje i sl), onda svaki novi uređaj i funkcionalnost troši dodatnu količinu tih resursa. Optimalna upotreba korisničkih resursa može biti presudna u budućnosti, čak ispred optimalne upotrebe internih resursa uređaja, poput procesorskog vremena i memorije, kojih je sve više na raspolaganju. Naprotiv, korisničkih resursa je na raspolaganju sve manje.

Odgovor na problem optimalne upotrebe korisničkih resursa može se tražiti u oblasti kontekstualnog računarstva (*context-aware computing*), uz pažljivo proučavanje koncepata te oblasti, uobličavanje njihovih osnovnih aspekata i primenu na oblast potrošačke elektronike. Kontekstualno računarstvo omogućava kreiranje aplikacija uz uvažavanje konteksta korišćenja. Kontekst predstavlja fizičko okruženje sistema i korisnika (vreme, aktivnost korisnika, osobine korisnika – pol, godine, zanimanje i sl) kao i stanje korisnika (psihičko i emocionalno stanje, prethodne aktivnosti, uobičajene aktivnosti, omiljeni režim rada, korisničke postavke i sl). Na osnovu prikupljenih parametara konteksta moguće je doneti zaključke od koristi za ciljnu aplikaciju i uključiti te zaključke u obliku dodatnih ulaznih promenljivih u rad aplikacije. Na ovaj način moguće je podesiti nove funkcionalnosti uređaja potrošačke elektronike tako da one budu u harmoniji sa korisnikom.

PREDMET ISTRAŽIVANJA

U okviru disertacije biće istražena oblast kontekstualnog računarstva, korišćeni algoritmi u toj oblasti i primena rešenja na različite aplikacije i uređaje u potrošačkoj elektronici. Posebna pažnja biće posvećena istraživanju arhitekture svih nivoa programske podrške kontekstualne platforme, njihovih međusobnih veza, načina funkcionisanja, kao i mehanizama komunikacije platforme sa okruženjem.

Iako su segmenti oblasti kontekstualnog računarstva obuhvaćeni dosadašnjim istraživanjima, ne postoji istraživanje koje sveobuhvatno analizira oblast i predlaže arhitekturu i algoritme primenljive na uređaje i sisteme potrošačke elektronike. Takođe, ne postoji integralan pogled na kontekstualno računarstvo, uređaje potrošačke elektronike i neke od osnovnih problema uređaja i aplikacija u potrošačkoj elektronici, poput problema asinhronih zahteva. Nedostaju i rezultati ispitivanja kontekstualnih funkcija u okruženjima svojstvenim potrošaču, kao što je okruženje unutar domaćinstva.

CILJ ISTRAŽIVANJA

Cilj istraživanja je da se pregledom dostupnih modela, mehanizama i alata primenljivih na uređaje i aplikacije potrošačke elektronike, uspostavi model arhitekture kontekstualne platforme i osnovni principi izrade kontekstualnih korisničkih sprega koji će poslužiti kao referenca pri budućem razvoju kontekstualnih funkcija za nove uređaje potrošačke elektronike. Iz tog razloga biće data detaljna specifikacija modela arhitekture kontekstualne platforme, uključujući specifikaciju zahteva i slučajeva korišćenja, potrebnu fizičku arhitekturu i detaljan model za sve potrebne blokove programske podrške. Verifikacija modela biće obavljena kroz niz eksperimentalnih scenarija i kroz primenu rešenja u nekoliko realnih aplikacija.

Istraživanje takođe ima za cilj da na osnovu postojećeg stanja u oblasti predloži jednostavne kontekstualne algoritme za primenu u uređajima potrošačke elektronike uzimajući u obzir osobine uređaja potrošačke elektronike, kao i da unapredi performanse i tačnost algoritama u poređenju sa trenutno dostupnim rešenjima.

S obzirom na cilj primene rešenja u oblasti potrošačke elektronike, prilikom razvoja rešenja jedan od osnovnih principa bila je mogućnost komercijalizacije.

NAUČNI DOPRINOS

Osnovni naučni doprinos disertacije jeste u oblasti kontekstualnog računarstva i u oblasti potrošačke elektronike. Disertacija donosi novi, unapređeni model programske podrške kontekstualne platforme koji je eksperimentalno i praktično potvrđen i koji može da služi kao osnova za buduća istraživanja u oblasti kontekstualnog računarstva. Takođe, date su ideje i načini primene nekih jednostavnih

algoritama za zaključivanje koji daju dobre eksperimentalne rezultate. Konačno, realizovano je nekoliko inovativnih kontekstualnih korisničkih sprega kao primera primene kontekstualne platforme, koje koriste svetlosne izvore kao medijume za interakciju sa korisnikom, što može biti dobra polazna osnova za dalja istraživanja i proširenja u oblasti kontekstualnog računarstva.

Doprinos u oblasti potrošačke elektronike je u donošenju sveobuhvatnog, sistematičnog i primenljivog pristupa kontekstualnim funkcionalnostima u uređajima potrošačke elektronike. Namera autora je da ova disertacija bude polazni materijal pri kreiranju novih kontekstualnih funkcionalnosti za uređaje potrošačke elektronike, odnosno pri kreiranju samostalnih kontrolera opremljenih kontekstualnim funkcijama. U tom svetlu u okviru disertacije dati su primeri i eksperimentalni rezultati koji u fokusu imaju uređaje potrošačke elektronike i primenu kontekstualne platforme u njihovom radu.

U oblasti ove disertacije u okviru istraživanja objavljena su tri rada u međunarodnim naučnim časopisima, od kojih je jedan vodeći u oblasti potrošačke elektronike i više od 20 radova na međunarodnim naučnim konferencijama.

ORGANIZACIJA DISERTACIJE

Disertacija je organizovana u 6 poglavlja.

U Poglavlju 1 izlaže se trenutno stanje u oblasti kontekstualnog računarstva i primene kontekstualnog računarstva na oblast potrošačke elektronike, i definišu se osnovni problemi koji su trenutno prisutni u istraživanjima i osnove na kojima se gradi konkretan doprinos disertacije. U okviru poglavlja data je i analiza trenutnog stanja u oblastima od posebnog značaja za disertaciju, kao što je oblast upravljanja asinhronim zahtevima.

U Poglavlju 2 izneti su detalji algoritama za semantizaciju korisničkog konteksta, kao osnovnih mehanizama za razvoj kontekstualne platforme. U okviru poglavlja dati su i predlozi rešenja algoritama za problem upravljanja asinhronim zahtevima. Takođe, dati su mehanizmi realizacije algoritama u vidu kontekstualnih skriptova, uključujući i detaljnu evaluaciju postojećih skript jezika u cilju izbora optimalnog rešenja.

Uvod

U Poglavlju 3 detaljno je izložen model arhitekture kontekstualne platforme. Modelom je obuhvaćen detaljan spisak zahteva visokog nivoa, specifikacija korisnika i slučajeva korišćenja, potrebna fizička arhitektura za realizaciju platforme i detalji svih nivoa, blokova i modula programske podrške kojima se realizuje kontekstualna platforma. Takođe, date su i važne sekvence rada kontekstualne platforme u cilju ilustracije komunikacije između različitih entiteta. U svrhu prezentacije modela korišćen je inženjerski pristup koji se primenjuje prilikom dizajna sistema, uz zadržan akademski pogled na moguće probleme i davanje smernica za rešenja. U tom smislu, gde god je to bilo moguće, korišćen je jezik za modelovanje UML.

U Poglavlju 4 dat je primer realizacije kontekstualne platforme u vidu platforme UAK (*User Awareness Kit*). Dati su osnovni detalji arhitekture platforme, retrospektivni pregled razvoja i konkretni blokovi fizičke arhitekture i programske podrške korišćeni za realizaciju.

U Poglavlju 5 dati su rezultati eksperimenata u cilju verifikacije i validacije rada realizovane platforme. Date su analize performansi, odziva sistema i tačnosti korišćenih algoritama, i obavljeno je poređenje rezultata sa osnovnim zahtevanim metrikama i sa nekim vodećim rešenjima u oblasti.

U Poglavlju 6 dati su primeri primene realizovanog rešenja u različitim ciljnim aplikacijama, uz odgovarajuće ilustracije.

Poglavlje 7 sadrži zaključak i smernice za dalje istraživanje.

Kao dodatak priložena je analiza trenutnog stanja komercijalnih senzorskih tehnologija dostupnih za integraciju sa kontekstualnom platformom.

POGLAVLJE 1.

STANJE U OBLASTI

1.1 KONTEKSTUALNE PLATFORME

Začetak ideje kontekstualnog računarstva nalazimo kod autora M. Weiser-a ranih devedesetih godina, koji uvodi pojam *računarstvo svuda oko nas (ubiquitous computing)*. Taj pojam obuhvata uređaje zasnovane na računaru čije je zaduženje da nam pomognu u svakodnevnom životu, ali tako da je korisnik u prvom planu, a ne uređaji, odnosno, uređaji moraju da budu neuočljivi za korisnika (Weiser, 1991). Pojam kontekstualnog računarstva uveli su Schilit, Adams i Want nedugo nakon Weiser-ove vizije budućnosti. Po njihovoj početnoj definiciji, kontekstualno računarstvo omogućuje otkrivanje *konteksta*, odnosno zaključivanje zasnovano na tri najvažnija aspekta: (1) gde je korisnik; (2) u čijem društvu je korisnik i (3) koji resursi su na raspolaganju korisniku (Schilit, 1994).

Novija istraživanja u oblasti posebno ističu razdvojenost uređaja za otkrivanje konteksta i njihovih funkcionalnosti, od korisnika i njegovih aktivnosti – navodeći da svet uređaja i svet korisnika jesu u međusobnoj sprezi ali da svet korisnika nije uslovljen svetom uređaja (Dey, 2001; Hill, 2004; Kwon, 2005a; Kwon, 2005b). Da bi se pružila adekvatna usluga korisnicima, aplikacije i servisi se moraju automatski prilagođavati promenama konteksta. Ovo prilagođavanje nazivamo *kontekstualnost*

Stanje u oblasti

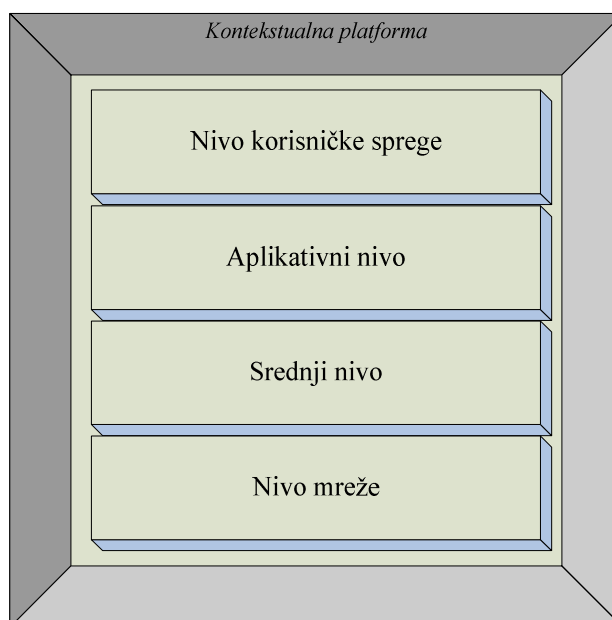
(*context-awareness*) (Bolchini, 2007; Dey, 2001; Zhu, 2005). Kontekstualnost obuhvata prilagođavanje promeni trenutnog stanja *entiteta* u okruženju: ljudi, mesta, stvari i uređaja (Korpiää, 2003a; Kwon, 2004). Da bi se pojednostavio mehanizam zaključivanja, biraju se samo oni entiteti koji su od interesa za spregu korisnika i ciljne aplikacije. Pod *stanjem* entiteta, podrazumeva se njegova trenutna lokacija, vreme, trenutno obavljana aktivnost i željene postavke. Sistem, odnosno platforma, se smatra kontekstualnom ukoliko je u stanju da sazna, tumači i koristi informacije o kontekstu, u cilju prilagođavanja svoje funkcionalnosti tom kontekstu (Byun, 2004). Današnja istraživanja u oblasti kontekstualnog računarstva smatraju kontekstualnost presudnom u naporima da se računarska tehnologija nevidljivo usadi u naše životno okruženje.

U nastavku poglavlja izlaže se trenutno stanje u oblasti realizacije kontekstualnih platformi, u pogledu tipičnih rešenja njihove arhitekture i tipičnih slučajeva upotrebe. Predstavljene paradigme zasnivaju se na dominantnim istraživanjima u oblasti u periodu od 2000-2012. godine. U okviru pregleda stanja u oblasti odabran je pristup uočavanja i razjašnjavanja zajedničkih koncepata i metodologija, nasuprot pristupu taksativnog navođenja i opisivanja mnogobrojnih dostupnih pojedinačnih rešenja, jer autor smatra da je samo takav vid pregleda svrsishodan i koristan kako za izlaganje u okviru disertacije, tako i za polazni pregledni materijal koji se može iskoristiti za buduća istraživanja. Za detalje o svakoj pojedinačnoj platformi priložene su reference na radove drugih autora.

1.1.1 Arhitektura kontekstualne platforme

Jedan od osnovnih ciljeva mnogih autora u oblasti kontekstualnog računarstva je da pronađu najbolji apstraktni model arhitekture kontekstualne platforme. Različiti modeli su do sada predstavljeni, kao što su CyberDesk (Dey, 2000), CAPIA (Plaza, 2001), Hydrogen (Hofer, 2002), HyCon (Bouvin, 2003), CoBrA (Chen, 2003), Gulliver's Genie (O'Hare, 2003; O'Grady, 2005), CORTEX (Biegel, 2004), X-CAF (Kao, 2004), CAS (Fahy, 2004), SOCAM (Gu, 2005), ACAI (Khedr, 2005), NAMA (Kwon, 2005a), Gaia (Roman, 2002; Gaia Project, 2005), PeCAN (Jutla, 2006), model za kreiranje rasprostranjenih (*pervasive*) servisa (Achilleos, 2010) i drugi. Zajednička karakteristika većine apstraktnih modela je slojevita arhitektura kontekstualne platforme, koja najčešće sadrži sledeća četiri nivoa: (1) nivo mreže, koji se sastoji od mreže komponenti (najčešće senzora) za prikupljanje informacija o korisničkom

kontekstu; (2) srednji nivo, koji upravlja svim procesima kontekstualne platforme i skladišti informacije o kontekstu; (3) aplikativni nivo, koji pruža konkretne usluge korisnicima zavisno od slučaja korišćenja i (4) nivo korisničke sprege, za omogućavanje interakcije sa korisnicima. Prikaz slojevite arhitekture kontekstualne platforme dat je na Slici 1.



Slika 1. Prikaz slojevite arhitekture kontekstualne platforme

Nivo mreže kontekstualne platforme se sastoji od: (1) definicije senzorskih komponenti za prikupljanje parametara konteksta i definicije njihove sprege sa kontekstualnom platformom; (2) definicije mrežnog protokola za povezivanje senzorskih komponenti i kontekstualne platforme i (3) realizacije mrežnog protokola na strani kontekstualne platforme (poslužioca).

Srednji nivo kontekstualne platforme zadužen je za obezbeđivanje semantike na osnovu parametara konteksta i omogućavanje aplikativnom nivou da koristi tu semantiku za realizaciju konkretnih aplikacija. Izvedbe srednjeg nivoa kontekstualne platforme su raznolike, počev od izvedbi zasnovanih na inteligentnim agentima (Soldatos, 2007), prostoru uređenih n-torki (Murphy, 2004), metapodacima (Bellavista, 2003a), OSGI okruženju (Choi, 2005) itd.

Aplikativni nivo sastoji se od različitih aplikacija koje kao podršku svom radu koriste informacije o kontekstu. Veliki broj istraživanja pokriva primenu kontekstualnih platformi u tzv. inteligentnim okruženjima (*smart space*), uključujući

Stanje u oblasti

domaćinstva, zdravstvene i obrazovne ustanove, kao i poslovna okruženja. Dalje, nailazimo na primene u informacionim i komunikacionim sistemima, mobilnim sistemima za trgovinu (*M-commerce*), kao i servisima na internetu.

Nivo korisničke sprege sastoji se od definicije korisničke sprege u smislu njenog izgleda i funkcionalnosti. Poseban naglasak pri realizaciji korisničke sprege je njena upotrebljivost (*usability*), koja se odnosi na prirodno korišćenje koje se oslanja na prethodno znanje korisnika, sve u cilju ispunjenja Weiser-ove paradigme o „nevidljivom“ računarskom okruženju.

1.1.1.1 Nivo mreže

Kontekstualnost zahteva povezivanje izvora informacija o kontekstu sa kontekstualnom platformom, te samim tim i sa krajnjom korisničkom aplikacijom. Povezivanje tih izvora u zajedničku mrežu, čime se omogućava prikupljanje informacija o kontekstu, je ključan zahtev koji se realizuje na nivou mreže. Za potrebe povezivanja koriste se različiti mrežni protokoli. Osnovni zahtevi mrežnog protokola navedeni u većini dosadašnjih istraživanja obuhvataju sledeće: (1) iniciranje i kontrolu komunikacije između tačaka; (2) kreiranje mobilne ad-hok mreže niske potrošnje energije za povezivanje velikog broja uređaja malih dimenzija i (3) automatsko otkrivanje uređaja u mreži (Friday, 2003; Yau, 2003; Kellerer, 2004; Chen, 2005; Lin, 2005; Tyan, 2005; Zhu, 2005; Benedetto, 2006; Boukerche, 2006; Giaffreda, 2007; Wood, 2008; Bjelica, 2012).

Za potrebe povezivanja izvora kontekstualnih informacija u mrežu s obzirom na gore navedene zahteve najčešće se koriste bežične senzorske mreže sa protokolima zasnovanim na IEEE 802.15.4, kao što su Zigbee (Jung, 2007), RF4CE (Yu-Ren, 2010), 6LoWPAN i ROLL (Yung-Bok, 2008), WirelessHART (Bhattacharyya, 2010), mada se koriste i bežične senzorske mreže sa protokolima zasnovanim na Bluetooth Low Energy (Gonzalez-Castano, 2005; Cano, 2006) i Z-Wave (Ghanam, 2009). Za povezivanje većih uređaja stalno priključenih na izvor napajanja koristi se žično ili bežično povezivanje korišćenjem protokola zasnovanih na IEEE 802.3 (Ethernet) i IEEE 802.11 (WiFi). Mnogobrojna istraživanja u oblasti kontekstualnih platformi ne proučavaju u dovoljnoj meri protokole veze i njihove implikacije na funkcionalnost kontekstualne platforme, te se stoga veoma često koriste i uobičajene PC periferije poput USB ili RS-232.

S obzirom na raznolikost potencijalnih izvora kontekstualnih informacija u smislu protokola za povezivanje sa kontekstualnom platformom, a uslovljeno zahtevom za uspostavljanje jedinstvene (*seamless*) mreže, veliki broj istraživanja predlaže mehanizme za povezivanje u uslovima heterogenih mreža. Neki od predloženih mehanizama uključuju vertikalno prebacivanje sa mreže na mrežu (Balasubramaniam, 2004; McNair, 2004), kada izvor kontekstualnih informacija dinamički menja način veze u zavisnosti od mogućnosti. Alternativni način za suočavanje sa heterogenošću je apstrakcija mrežnih primitiva u jedan jedinstven mrežni servis, koji povezuje pošiljaoca i odredište putem optimalne mrežne sprege bez potrebe da bilo pošiljalac ili odredište znaju koja se fizička mrežna sprega koristi (Rocha, 2006).

Nivo mreže takođe obuhvata definisanje senzorskih komponenti kao izvora informacija o kontekstu i definisanje njihove sprege sa kontekstualnom platformom. Dakle, svaka senzorska komponenta mora da zadovoljava uslove propisane od strane kontekstualne platforme, u pogledu podržanih primitiva, odnosno servisa. Senzorske komponente mogu se podeliti u tri velike grupe: (1) fizičke senzore; (2) virtuelne senzore i (3) logičke senzore (Indulska, 2003).

Fizički senzori su najrasprostranjenija grupa senzora koja se koristi u okviru kontekstualnih platformi. Ovi senzori predstavljaju zasebne uređaje ili blokove fizičke arhitekture, koji prikupljaju fizičke parametre okruženja i obaveštavaju kontekstualnu platformu o tim parametrima. Okvirni spisak merljivih fizičkih parametara i pripadajućih fizičkih senzora koji su danas u upotrebi dat je u Tabeli 1.

Virtuelni senzori dobavljaju informacije o kontekstu od strane različitih servisa ili aplikacija. Ovo uključuje npr. analizu pokrenutih aplikacija na računaru, zakazanih sastanaka u elektronskom kalendaru, aktivnosti korisnika za računarom (korišćenje miša, tastature i sl), analizu podataka sa interneta (vesti, vremenske prilike i sl).

Logički senzori koriste informacije o kontekstu koje dolaze sa jednog ili više različitih izvora (fizičkih i/ili logičkih senzora), i izvode zaključak na osnovu tih informacija. Neretko, logički senzori zaključke izvode i na osnovu već prikupljenih informacija koje arhiviraju u bazu podataka. Osnova rada logičkih senzora su algoritmi na osnovu kojih se izvode zaključci višeg nivoa semantike (semantički događaji). Ovi algoritmi su često zasnovani na obradi slike i zvuka, a neretko i na obradi ostalih

Stanje u oblasti

događaja, kao što su npr. podaci sa akcelerometra na osnovu kojih se prepoznaju gestovi korisnika. Veliki broj istraživanja se odnosi na konkretne algoritme obrade događaja sa fizičkih senzora. Zbirni pregled dela mogućih algoritama i pripadajućih istraživanja dat je u Tabeli 2.

<i>Parametar koji se meri</i>	<i>Fizički senzor</i>
Nivo osvetljenosti	Fotodiode, senzori za boju, IR i UV senzori itd.
Fotografija okruženja	Optičke kamere
Fotografija okruženja sa dubinom prostora (3D)	TOF (<i>time-of-flight</i>) kamere, 3D kamere struktuiranog svetla, stereo kamere
Zvuk	Mikrofoni
Kretanje, ubrzanje	Akcelerometri, magnetna polja, živini prekidači
Prisustvo	PIR (<i>passive infrared</i>) senzori, ultrazvučni senzori, optičke kamere, Zigbee RSSI senzori, senzori opterećenja itd.
Prisustvo sa identifikacijom	RFID (<i>radio frequency ID</i>)
Stanje vrata, ormara i sl.	Kontaktne prekidači
Lokacija	Globalni pozicioni sistem (GPS), Globalni sistem za mobilnu vezu (GSM), sistem aktivne značke (<i>Active badge</i>)
Temperatura	Termometri
Lične bioinformacije	Biosenzori za merenje krvnog pritiska, srčanog ritma i sl.

Tabela 1. Tipovi fizičkih senzora u upotrebi

Distribuirane mrežne arhitekture, u kojima lokalne jedinice komuniciraju sa lokalnim senzorskim komponentama korišćenjem jednog fizičkog protokola, a povezane su sa centralnom jedinicom preko širokopojsne mreže (npr. Internet ili GSM) najčešće se realizuju u srednjem nivou kontekstualne platforme, koji je opisan u poglavlju 1.1.1.2. Zbog toga arhitektura nivoa mreže predstavljena u ovom poglavlju sadrži isključivo sprege ka lokalnim senzorskim komponentama.

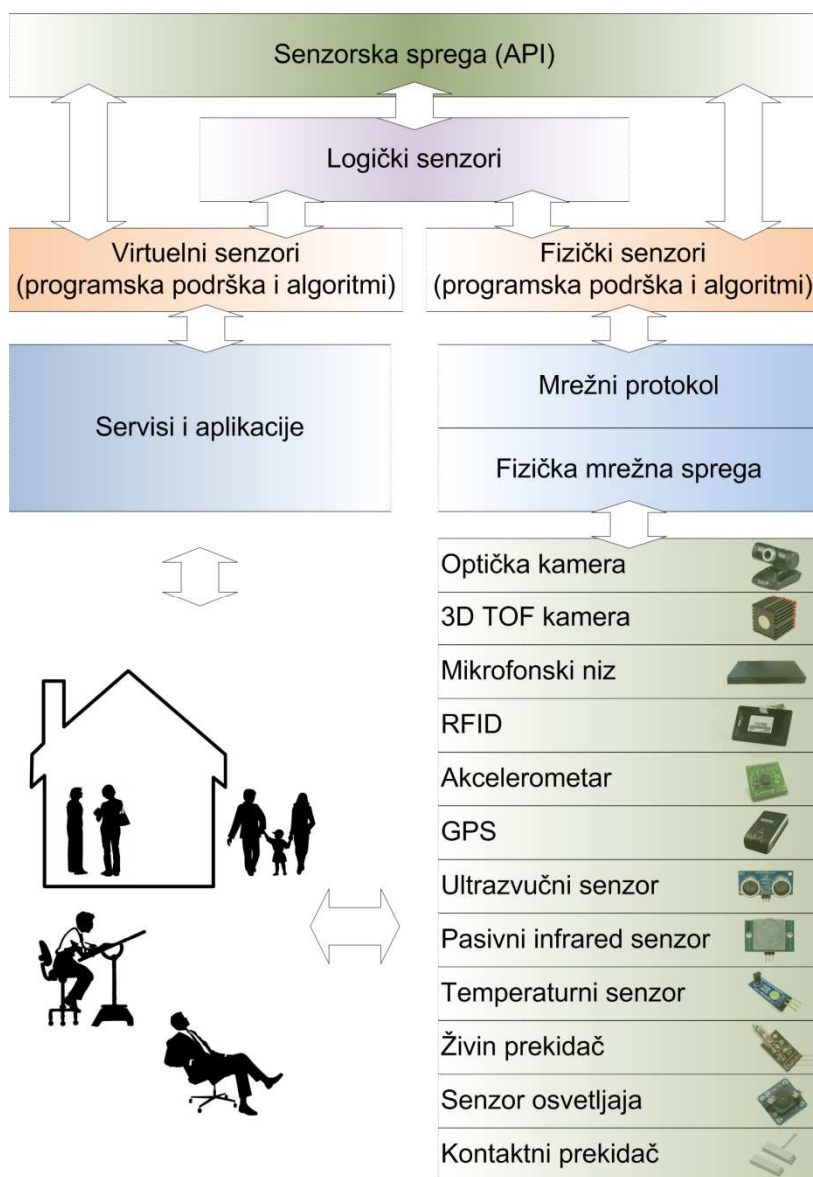
Detaljna specifikacija, korišćenje i integracija pojedinih senzorskih tehnologija na nivou mreže nije primarni fokus disertacije, te se u tom smislu izbegava navođenje

suvišnih detalja o tehnologijama u okviru ovog poglavlja. Ipak, s obzirom na praktičnu prirodu oblasti kojom se disertacija bavi (potrošačka elektronika), specifikacija senzorskih tehnologija sa mogućim aplikacijama je od interesa za buduće projekte i istraživanja u ovoj oblasti. U tom smislu, više detalja o senzorskim tehnologijama dato je u prilogu disertacije na kraju teksta.

<i>Fizički parametri / senzori</i>	<i>Semantički događaj</i>	<i>Istraživanje</i>
GPS	Obrazac kretanja korisnika, najčešće korišćene putanje	Brilingaite, 2007
Osvetljenje, ubrzanje, temperatura, zvuk	Stanje određenog uređaja ili okruženja uređaja (šolja za kafu, mobilni telefon)	Gellersen, 2002
Ubrzanje (akcelerometar)	Pokreti delova tela, procena usvojene telesne veštine	Knight, 2007
Optička kamera	Broj ljudi na nekom masovnom događaju	Chan, 2008
Optičke kamere	Trajektorije kretanja osoba	Fleuret, 2008
Optička kamera	Prepoznavanje lica	Wright, 2009
3D fotografija (TOF kamera)	Detekcija gesta u prostoru	Holte, 2010
Ubrzanje (akcelerometar)	Detekcija gesta pri pomeranju uređaja	Liu, 2009
Zigbee RSSI varijacije	Detekcija prisustva po zonama	Mrazovac, 2012
RFID	Pozicija osobe u zatvorenom prostoru	Guang-yao, 2006
Zvuk (mikrofonski niz)	Pozicija (ugao) osobe koja trenutno govori	Papp, 2007
Zvuk (mikrofoni)	Prepoznavanje govora	Gales, 2008

Tabela 2. Istraživanja u kojima se realizuju algoritmi sa mogućnošću primene u logičkim sensorima

Blok dijagram tipične arhitekture nivoa mreže dat je na Slici 2.



Slika 2. Tipična arhitektura nivoa mreže kontekstualne platforme

1.1.1.2 Srednji nivo

Osnovni mehanizmi funkcionisanja kontekstualne platforme sadržani su u okviru implementacije srednjeg nivoa (*middleware*). Osnovna funkcija srednjeg nivoa je da prikuplja informacije o kontekstu sa svih dostupnih izvora, sortira i kategoriše te informacije, izvede zaključak o stanju i potrebama okruženja, te da aktivira određene akcije, odnosno da omogući aplikativnom nivou da obavi željene akcije na osnovu pristupa zaključcima objavljenim od strane srednjeg nivoa.

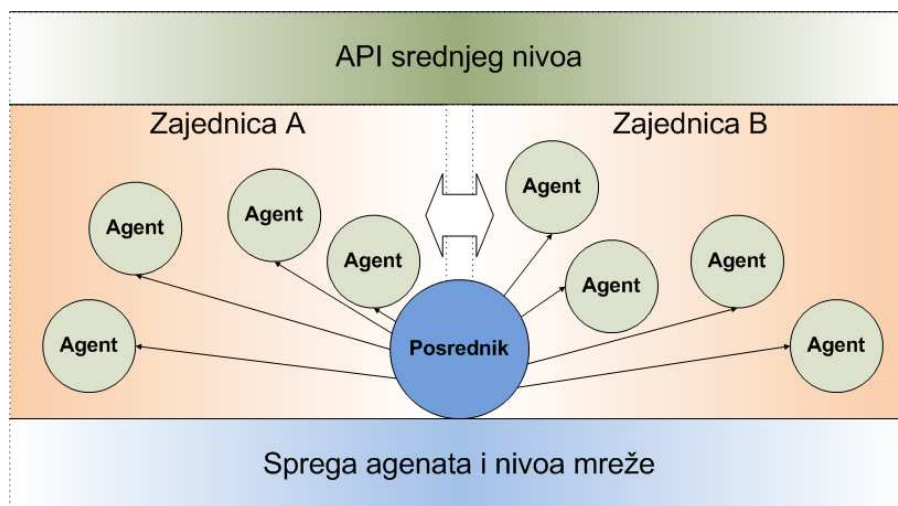
Srednji nivo mora da omogući izvođenje različitih tipova zaključaka, i u skladu sa tim mehanizmi zaključivanja moraju biti podešljivi. U skladu sa ovim zahtevom mnoge realizacije srednjeg nivoa sastoje se od modula sa sopstvenom logikom zaključivanja, koji u nekim realizacijama međusobno komuniciraju. Svaki od modula ima pristup informacijama o kontekstu, poseduje sopstvenu logiku zaključivanja i prilagođava se promenama. Većina realizacija srednjeg nivoa ima za cilj da olakša razvoj i podešavanje modula u cilju njihovog prilagođenja različitim slučajevima upotrebe.

S obzirom na veliki broj različitih arhitektura srednjeg nivoa koje su prisutne u dosadašnjim istraživanjima, one se mogu podeliti u šest kategorija (Hong, 2009): (1) srednji nivo zasnovan na agentima; (2) reflektivni srednji nivo; (3) srednji nivo zasnovan na metapodacima; (4) srednji nivo zasnovan na prostoru uređenih n-torki; (5) adaptivni srednji nivo i (6) srednji nivo zasnovan na OSGI okruženju.

Srednji nivo zasnovan na agentima sastoji se od inteligentnih agenata koji nezavisno prikupljaju informacije o kontekstu sa mrežnog nivoa, zaključuju na osnovu tih informacija i skladište zaključke (znanje) u bazu podataka srednjeg nivoa. Agenti mogu biti distribuirani, što znači da se mogu nalaziti na različitim prostorno udaljenim lokacijama, gde prikupljaju informacije korišćenjem lokalnog nivoa mreže (Satoh, 2010; Ryzko, 2010). Velika pažnja u istraživanjima posvećena je razvoju aplikacija koje koriste mobilne agente. Mobilni agenti imaju mogućnost da odluče o nastavku svog izvršavanja u različitom programskom okruženju na različitoj platformi. Ova „selidba“ odnosi se na premeštanje informacija (zaključaka, znanja) i samog algoritma (programa) do drugog zainteresovanog entiteta, na zahtev ili po potrebi. Realizacije srednjeg nivoa koje koriste mobilne agente ovako obezbeđuju mehanizam za prikupljanje kontekstualnih informacija sa različitih lokacija tako što omogućavaju da mobilni agenti menjaju lokaciju, dok zadržavaju algoritam i prethodno znanje (Burstein, 2005). Većina realizacija srednjeg nivoa zasnovanog na agentima organizuje agente u *zajednice*, i poseduje koncept *posrednika* između zajednica, koji je takođe realizovan u obliku agenta. Ovaj posrednik upravlja bazom znanja i stara se o objavljivanju tog znanja svim zainteresovanim stranama (Roman, 2002; Riekkii, 2003; Chen, 2004; Khedr, 2004; Khedr, 2005; Bellavista, 2006; Julien, 2006; Alahuhta,

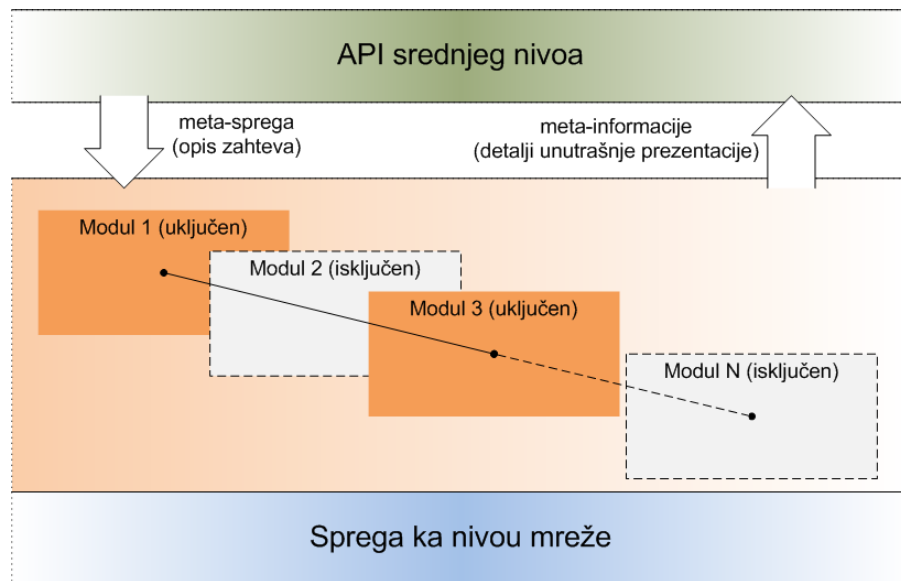
Stanje u oblasti

2007; Soldatos, 2007; Feng 2009; Yaghmaie, 2011). Tipična arhitektura srednjeg nivoa zasnovanog na agentima prikazana je na Slici 3.



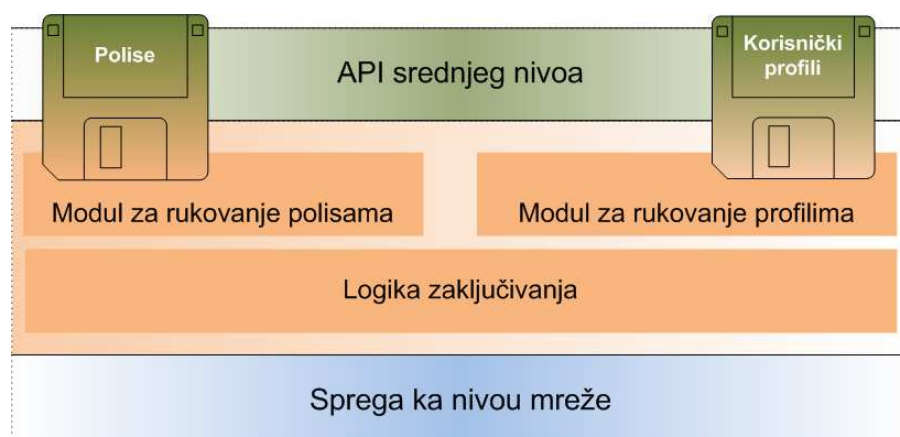
Slika 3. Arhitektura srednjeg nivoa zasnovanog na agentima

Reflektivni srednji nivo odlikuje se po tome što API srednjeg nivoa omogućava aplikativnom nivou da podešava rad srednjeg nivoa, odnosno da ima delimičan uvid u detalje njegove implementacije. U tom smislu aplikacija može da deklariše, npr. za koju vrstu informacija o kontekstu je zainteresovana, i to na način da srednji nivo zaista vrši obradu isključivo tih informacija. U terminologiji realizacije reflektivnog srednjeg nivoa, srednji nivo dostavlja aplikaciji meta-informacije (*meta-information*) o detaljima svoje unutrašnje prezentacije čime omogućuje aplikaciji da uspostavi zahteve. Aplikacija dostavlja srednjem nivou definiciju meta-sprege (*meta-interface*), kojim se opisuju aplikativni zahtevi i definišu sprege za isporuku informacija (Capra, 2003; Chan, 2003; Chan, 2004; Meng, 2009; Papakos, 2010; Liu, 2011). Arhitektura reflektivnog srednjeg nivoa prikazana je na Slici 4.



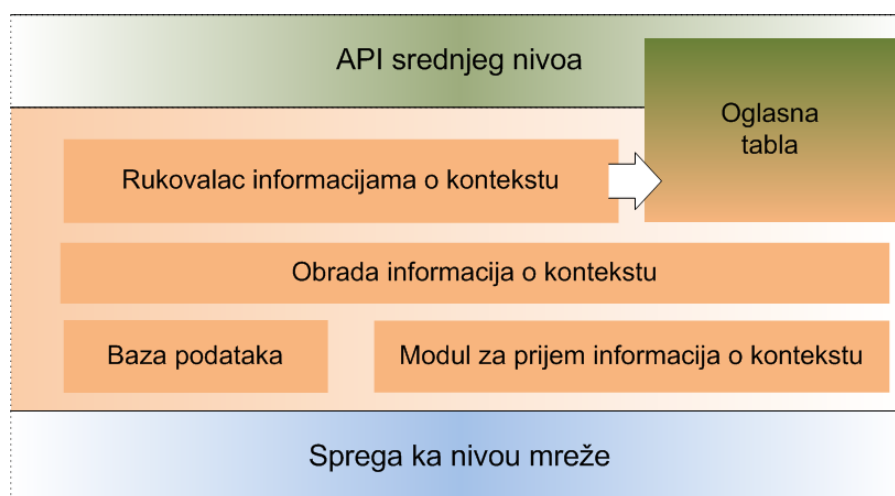
Slika 4. Arhitektura reflektivnog srednjeg nivoa

Srednji nivo zasnovan na metapodacima kao osnovu za svoj rad koristi metapodatke, odnosno podatke koji opisuju osnovne osobine korisnika, predmeta i pojava u okruženju, kao i skup pravila na osnovu kojih se izvode zaključci. U skladu sa ovom paradigmom, metapodaci su podeljeni na *profile* i *polise*. Profilima se zadaju karakteristike, mogućnosti i zahtevi vezani za korisnike, uređaje i servise. U okviru istraživanja još uvek nije usaglašen jedinstven format zapisa ovih profila. Polise obuhvataju sistem odabiranja zasnovan na pravilima (korišćenjem npr. stabla odlučivanja). Polise se opisuju notacijom visokog nivoa semantike, tako da one budu jednostavno izmenljive od strane administratora, menadžera servisa ili, čak, krajnjeg korisnika (Bellavista, 2003a; Bellavista, 2003b; Wood, 2008; Strassner, 2009; Malandrino, 2010; Bjelica, 2011a; Magableh, 2011). Arhitektura srednjeg nivoa zasnovanog na metapodacima prikazana je na Slici 5.



Slika 5. Arhitektura srednjeg nivoa zasnovanog na metapodacima

Srednji nivo zasnovan na prostoru uređenih n-torki (*tuple space*), realizovan je u skladu sa istoimenom paradigmom koja omogućava konkurentni pristup uređenim n-torkama podataka od strane više zainteresovanih entiteta (procesa ili aplikacija). Informacije o kontekstu, dobijene od strane senzorskih komponenti sa nivoa mreže, srednji nivo organizuje u n-torke prema unapred poznatoj definiciji. Ove n-torke informacija srednji nivo smešta u jedinstven memorijski prostor, kome se može pristupiti direktno i istovremeno od strane različitih aplikacija na aplikativnom nivou. Dakle, srednji nivo zasnovan na prostoru uređenih n-torki poseduje jednostavan API od koga se jedino zahteva da omogući pristup jedinstvenom memorijskom prostoru za smeštanje informacija o kontekstu (Cabri, 2003; Murphy, 2004; Curino, 2005; Balzarotti, 2007; Matthews, 2011). Ovaj jedinstveni memorijski prostor se u pojedinim istraživanjima naziva *oglasna tabla (blackboard)* (Korpijaa, 2003b; Suh, 2011). U ovom modelu arhitekture srednjeg nivoa informacije o kontekstu su organizovane *ontološki*, odnosno, poseduju oznaku koja jasno ukazuje na nivo hijerarhije određene informacije, kao i na sve oznake matičnih informacija, sa kojima je ta informacija povezana. Na primer, *Okruženje->Zvuk->Govornik1->PozicijaGovornika* predstavlja oznaku informacije koja saopštava poziciju govornika sa oznakom 1 na osnovu zvuka iz neposrednog okruženja. Arhitektura srednjeg nivoa zasnovanog na prostoru uređenih n-torki prikazana je na Slici 6.

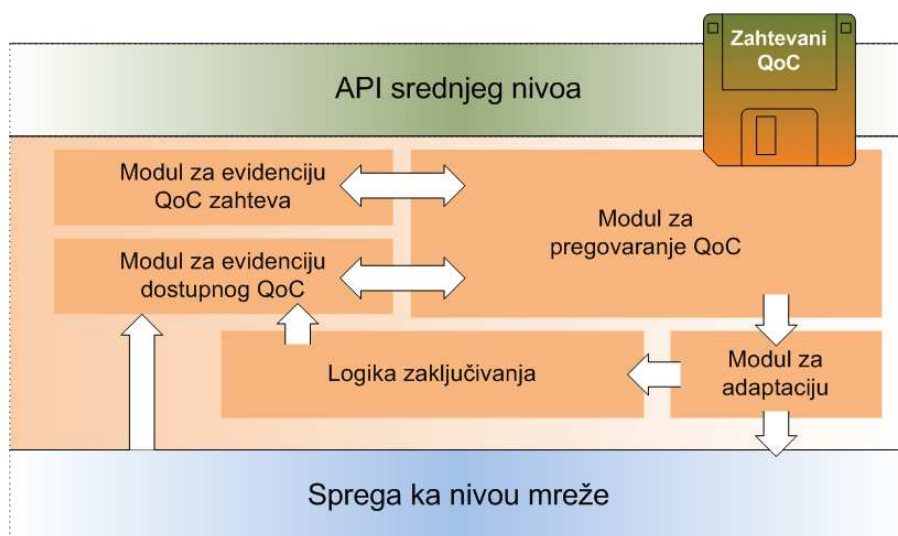


Slika 6. Arhitektura srednjeg nivoa zasnovanog na prostoru uredenih n-torki

Adaptivni srednji nivo zasniva se na mogućnosti srednjeg nivoa da prilagodi svoj rad u skladu sa parametrima kvaliteta informacija o kontekstu (QoC – *quality of context*). Kvalitet informacija o kontekstu je pojam koji označava skup atributa pridruženih svakoj informaciji o kontekstu, a koji uslovljavaju kvalitet, odnosno upotrebljivost te informacije i ne zavise od konkretnog slučaja korišćenja ili krajnje korisničke aplikacije. Kao osnovni deo QoC paradigme najčešće navode se četiri atributa, uz promenljivu terminologiju: prostorno poreklo (*spatial origin*), zastarelost (*age*), pouzdanost (*reliability, trustworthiness*) i stepen slaganja (*degree of relationship*) (Zimmer, 2006; McKeever, 2009). Prostorno poreklo označava lokacijsku pripadnost entiteta za koji je vezana informacija o kontekstu posmatranoj lokaciji, odnosno udaljenost od nje. Prostorno poreklo se najčešće obeležava zonalno, te informacija o prostornom poreklu nosi oznaku zone, kao i celokupno stablo hijerarhije zona (npr. *Kompanija->Sprat 2->Kancelarija 3->Markov sto->Računar*) (Beigl, 2002; Satoh, 2007). Zastarelost se označava kao relativno vreme proteklo od prijema najstarije informacije o kontekstu istog tipa koja se nalazi u bazi, do prijema date informacije. Pouzdanost se najčešće označava kao kombinacija pouzdanosti izvora informacija o kontekstu, zasnovano na specifikaciji senzorske komponente i prethodnih eksperimentalnih studija (Neisse, 2008). Stepem slaganja odnosi se na odnos dva odvojena konteksta, i uključuje meru u kojoj se informacije o kontekstu u tim kontekstima podudaraju (Zimmer, 2006).

Stanje u oblasti

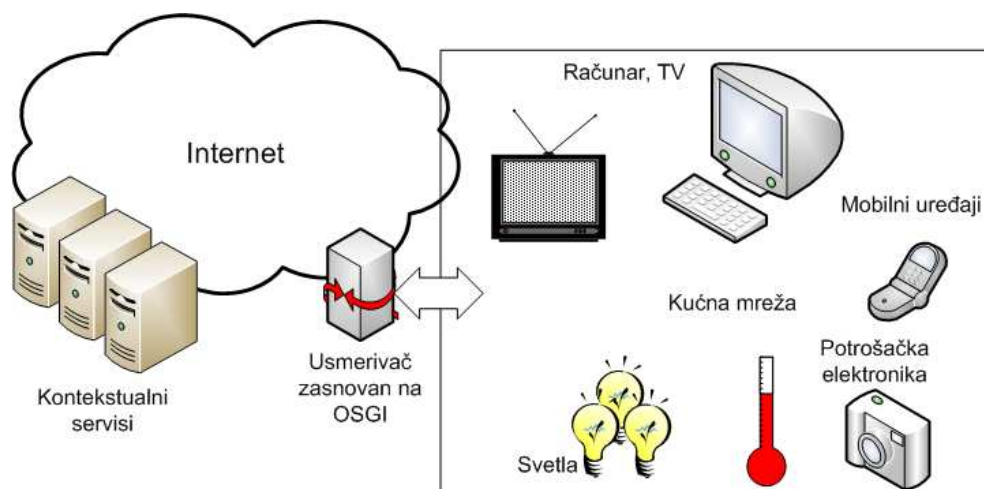
Prilagođavanje rada adaptivnog srednjeg nivoa zasniva se na težnji za poklapanjem QoC zahteva zadatim od strane aplikacije, sa QoC mogućnostima obezbeđenih od strane mrežnog nivoa kontekstualne platforme i moduli za zaključivanje iz srednjeg nivoa. Ukoliko postoji dovoljan nivo QoC u okviru platforme, srednji nivo se samostalno kalibriše tako da optimizuje upotrebu resursa u cilju obezbeđivanja traženog nivoa QoC (Alex, 2005; Wu, 2007b; Abid, 2009; Lopes, 2009; Zheng, 2011). Arhitektura adaptivnog srednjeg nivoa prikazana je na Slici 7.



Slika 7. Arhitektura adaptivnog srednjeg nivoa

Srednji nivo zasnovan na OSGI (*Open Services Gateway Initiative*) okruženju izdvojen je u posebnu kategoriju iz razloga što veliki broj istraživanja na temu kontekstualnih platformi koristi upravo OSGI okruženje kao osnovu. OSGI okruženje je platforma zasnovana na programskom jeziku Java, koja omogućava realizaciju dinamičkog modela programske podrške (*software*) koji se sastoji od različitih komponenti (servisa). Posebna odlika tog okruženja je što omogućava pokretanje, zaustavljanje, dodavanje i uklanjanje servisa bez potrebe za ponovnom inicijalizacijom okruženja, i svi prisutni servisi dobijaju obaveštenja o dinamičkoj izmeni strukture servisa. Osnovna motivacija istraživačima da arhitekturu kontekstualne platforme zasnuju na OSGI okruženju, ogleda se u tome što OSGI poseduje veliki broj dodataka za različite protokole, uključujući automatsko pronalaženje uređaja (npr. UPnP), komunikaciju (žično, bežično, korišćenjem električne mreže), širokopoljnu vezu i sl. Stoga realizacija kontekstualne platforme zasnovane na OSGI u najvećem broju

slučajeva ne vodi računa o realizaciji mrežnog nivoa, koji već postoji i obezbeđuje komunikaciju sa velikim brojem uređaja (Slika 8).



Slika 8. Tipična arhitektura kontekstualne platforme zasnovane na OSGI okruženju

Iako arhitektura srednjeg nivoa zasnovanog na OSGI može biti različita i u suštini realizovati bilo koju od ranije navedenih arhitektura, tipično je da se moduli organizuju u obliku servisa. U skladu sa tim kao sastavni deo srednjeg nivoa zasnovanog na OSGI nalazimo servise poput dostavljača informacija o kontekstu (*context provider*), servisa za zaključivanje (*context reasoner*), servisa za pronalaženje servisa (*service-locating service*) i sl (Gu, 2004; Choi, 2005; Gu, 2005; Yu, 2006; Huang, 2009; Vilas, 2010; Zhang, 2011).

1.1.1.3 Algoritmi za zaključivanje

Osnovu rada svih inteligentnih sistema, pa tako i kontekstualnih platformi, čine algoritmi zaključivanja, koji kao ulaz koriste informacije o kontekstu sa mrežnog nivoa, objavljene od strane senzorskih komponenti. Cilj ovih algoritama je da dovedu nivo informacije o kontekstu do visokog nivoa semantike (što ćemo nazvati *semantizacijom konteksta*), tako da te informacije budu direktno upotrebljive od strane aplikacija. Na primer, izlaz algoritma zaključivanja može da bude korisnička aktivnost (spava, sprema ručak, čita novine, gleda TV program, na sastanku je i sl.), odnosno stanje nekog predmeta ili pojave (npr. otkrivanje tipa okruženja – učionica, kuća, ulica, stadion). Srednji nivo omogućava, pored ranije navedenih osobina, pristup zaključcima visokog nivoa semantike od strane aplikacija. U slučaju srednjeg nivoa zasnovanog na

Stanje u oblasti

agentima, aplikacija direktno komunicira sa odgovarajućim agentom, i pri tome svaki od agenata realizuje neki od algoritama zaključivanja. Reflektivni srednji nivo omogućava preuzimanje zaključaka korišćenjem prosledene meta-sprege, dok su algoritmi zaključivanja realizovani po modulima, sa mogućnošću definisanja koji se moduli, odnosno algoritmi koriste a koji ne. Srednji nivo zasnovan na metapodacima deo algoritama zaključivanja realizuje u posebnom modulu, a deo se realizuje u okviru prosledene polise, koja definiše i isporuku zaključaka aplikativnom nivou. Srednji nivo zasnovan na prostoru uređenih n-torki realizuje algoritme u okviru modula za obradu informacija o kontekstu, dok se zaključci smeštaju u zajednički memorijski prostor (na oglasnu tablu) dostupno aplikacijama. Adaptivni srednji nivo realizuje algoritme u okviru posebnog modula, dok su ulazi u algoritme, osim kontekstualnih informacija, i željeni nivoi parametara QoC. Dakle, u svim realizacijama srednjeg sloja algoritmi čine važan i neodvojivi deo.

Veliki broj istraživanja u oblasti kontekstualnih platformi bavi se koncipiranjem, realizacijom i unapređivanjem algoritama za zaključivanje. Kako je oblast algoritama veštačke inteligencije (*AI – artificial intelligence*) izuzetno široka, i nije predmet ove doktorske disertacije, pregled koji sledi zasnovan je isključivo na istraživanjima koja realizuju algoritme namenjene ili posebno definisane za oblast kontekstualnih platformi. Dobar pregled šire oblasti veštačke inteligencije i fundamentalnih algoritama daju Russel i Norvig u knjizi *Artificial Intelligence – A Modern Approach* (Russel, 2010). Zbirni pregled značajnijih istraživanja algoritama za primene u kontekstualnim platformama dat je u Tabeli 3.

1.1.1.4 Aplikativni nivo

Najveći broj skorijih istraživanja bavi se primenama kontekstualnih platformi za rešavanje različitih problema današnjice, odnosno za podršku svakodnevnom životu i radu. Prema broju istraživanja u oblasti primene kontekstualnih platformi, izdvajaju se sledeće oblasti primene: (1) inteligentna okruženja (domaćinstvo, bolnica, učionica, kancelarija); (2) turistički vodiči; (3) informacioni sistemi; (4) komunikacioni sistemi; (5) sistemi za trgovinu i marketing, uključujući mobilnu trgovinu (*M-commerce*) i (6) internet servisi.

<i>Tip zaključka</i>	<i>Ulazni parametri</i>	<i>Korišćeni algoritam</i>	<i>Istraživanje</i>
korisnička aktivnost (situacija)	Različite senzorske komponente	Fazi logika za aproksimativno zaključivanje	Anagnastopoulos, 2007
geografska lokacija	Snaga radio signala sa različitih pristupnih tačaka	Teorija vektora podrške (<i>Support Vector Machine</i>)	Brunato, 2005
lokacija unutar zgrade	Snaga signala sa WiFi uređaja	Bajesovo zaključivanje (<i>Bayesian inference</i>)	Ladd, 2005
aktivnost korisnika, stanje prostorije	RFID, pokrenute aplikacije, stanje svetla u prostoriji i sl.	Bajesove mreže (<i>Bayesian networks</i>)	Ranganathan, 2004
predviđanje lokacije korisnika u prostoriji	RFID	Teorija igara, ekvilibrijum Nash-a (<i>Nash H-learning</i>)	Sajal, 2006
predviđanje geografske lokacije korisnika	GSM mreža	Dempster-Shafer teorija	Samaan, 2005
fizičke aktivnosti korisnika	Akcelerometri	Stabla odlučivanja	Bao, 2004
tip okruženja korisnika	Akcelerometri, WiFi mreže i snaga signala, mikrofoni	AdaBoost, k-najbliži sused	Xu, 2011
predlaganje servisa	korisnički profili i postavke, povratna sprega sa korisnikom	Veštačke neuronske mreže	Al-Masri, 2009
aktivnost korisnika u domaćinstvu	digitalni kontaktni senzori	Skriveni Markovljevi lanci	Kasteren, 2008
nivo zauzetosti korisnika	korišćenje miša i tastature, aktivne aplikacije na računaru, audio, WiFi	Stabla odlučivanja	Tullio, 2007
tip fizičke vežbe koju korisnik izvodi	akcelerometri	Naivni Bajesov klasifikator	Chang, 2007

Tabela 3. Pregled istraživanja algoritama zaključivanja sa primenom u kontekstualnim platformama

Stanje u oblasti

Primena kontekstualnih platformi u inteligentnim okruženjima, odnosi se na povećanje ugodnosti ili produktivnosti ljudi koji pripadaju datom okruženju. Na primer, jedan od velikih problema današnjih sistema za automatizaciju domaćinstva, jeste što više aplikacija ili procesa može upravljati istim skupom uređaja, odnosno može pristupiti različitim resursima, kao što su izvori kontekstualnih informacija, konkurentno. Kontekstualna aplikacija za rešavanje ovog problema ponaša se kao posrednik u upravljanju resursima, i određuje pravo pristupa na osnovu prioriteta, potreba korisnika i mogućnosti sistema (Baek, 2005). Takođe, kontekstualne aplikacije poboljšavaju sisteme za automatizaciju domaćinstva, tako što uvode čvršću spregu sa korisnicima, u smislu prikupljanja informacija korišćenjem dijaloških formi i učenja na osnovu prethodnih izbora (Intille, 2002). Veoma često, kontekstualnost u domaćinstvu se ostvaruje aktiviranjem režima rada od strane korisnika, odnosno definisanjem i izvršavanjem skriptova koji opisuju ponašanje domaćinstva, za šta su predviđene korisničke kontekstualne aplikacije. Za opis ponašanja sistema korišćeni su CPL i CGI skriptovi (Schulzrinne, 2003), Lua (Bjelica, 2011b), XML (Miori, 2006), Perl (Kusznir, 2010) i dr.

Kontekstualne aplikacije za primene u inteligentnim okruženjima u zdravstvu, poput ordinacija i bolnica, služe da medicinskom osoblju olakšaju i unaprede rad, odnosno da poboljšaju ugodnost i iskustvo pacijenata. Na primer, kontekstualna aplikacija može da kreira detaljan izveštaj o toku operacije, prateći stanje pacijenta, aktivnosti i kretanje osoblja, i upotrebu instrumenata. Pristup ovim informacijama nakon operacije može da se koristi za ispitivanje efikasnosti i obučavanje (Agarwal, 2007). Kontekstualna aplikacija koja prati zdravstveno stanje pacijenata na daljinu, pruža pomoć timovima hitne pomoći za pravovremenu organizaciju i intervenciju (Bottazzi, 2006; Sridevi, 2010). Takođe, moguće je realizovati aplikacije koje omogućavaju uvid u trenutno stanje (aktivnost) kolege medicinskog radnika, kako bi se minimizovali neželjeni prekidi od strane drugih kolega u kritičnim momentima rada, ili, aplikacije koje prikazuju zdravstveni karton pacijenta na način podesan za osoblje koje dolazi u vizitu (Favela, 2007).

Inteligentne učionice zasnovane na kontekstualnim aplikacijama omogućavaju jednostavniji uvid predavačima u nivo zainteresovanosti studenata za izlaganu materiju, koji je zaključen kao kombinacija zvučnog zapisa i unosa od strane studenata

(„Aplauz“, „Dosadno“ ili „Izgubljen“), kao i automatsku procenu kvaliteta i dinamike studentske diskusije na zadatu temu (Sung, 2005). Učenje na daljinu se takođe može poboljšati korišćenjem kontekstualnih aplikacija, tako da se sadržaj i intenzitet lekcija zasniva na sposobnostima, lokaciji, dostupnom vremenu i zainteresovanosti polaznika kursa (Chen, 2010).

Kontekstualne aplikacije za primenu u kancelarijskom okruženju, imaju prevashodno za cilj povećanje produktivnosti zaposlenih. Na primer, aplikacija može imati za cilj smanjenje broja prekida u radu zaposlenog od strane drugih kolega, telefona ili notifikacija iz aplikacija na računaru (Fogarty, 2005; Bjelica, 2011c). Aplikacija virtuelne telefonske sekretarice omogućuje automatsko obaveštavanje osobe koja poziva zaposlenog u kancelariji o njegovoj trenutnoj dostupnosti (Danninger, 2008). Okruženje za rad, uključujući svetla, grejanje, prikaz informacija na ekranu i sl, automatski se podešava na osnovu istorije ponašanja korisnika i njegovih postavki (Cheverst, 2005).

Sve veća primena kontekstualnih platformi nalazi se u različitim aplikacijama za podršku turizmu, tzv. virtuelnim turističkim vodičima. Naime, na osnovu informacija o kontekstu, koje najčešće uključuju lokaciju turista, trenutni pogled na određenu znamenitost, njegova interesovanja i postavke, aplikacija prikazuje najvažnije i najzanimljivije informacije o posećenom turističkom mestu, koristeći adekvatne multimedijalne izraze. Ovakve aplikacije suočene su sa posebnim izazovima, kao što su podeljena pažnja turista ukoliko vozi automobil i istovremeno obilazi znamenitosti – aplikacija turističkog vodiča na njegovom mobilnom telefonu ili tablet računaru mora biti specifično projektovana da ne ugrozi aktivnost vožnje automobila (Bellotti, 2005). Virtuelni turistički vodič omogućava poboljšano iskustvo posetilaca muzeja, kada prikazuje dodatne informacije o eksponatima zasnovano na poziciji posetioca unutar muzeja i njegovim interesovanjima (Cano, 2006; Bihler, 2011).

Mnogi autori predlažu poboljšanja informacionih sistema kontekstualnim funkcijama. Važnu primenu kontekstualnih aplikacija nalazimo u ekspertskim sistemima za podršku odlučivanju (*decision support systems*), koji imaju za cilj da na osnovu velikog broja pokazatelja pomognu rukovodiocima da donesu najbolju odluku koja se tiče njihovog poslovnog sistema (Kwon, 2005b). Takođe, veoma važna grupa kontekstualnih aplikacija su aplikacije za unapređeno rukovanje podacima i

Stanje u oblasti

prezentaciju podataka, u zavisnosti od dostupnih prezentacionih kanala, korisničkog profila, lokacije i sl. Zanimljiv primer je realizacija sistema koji povezuje različite načine prezentacije i izbora informacija, uključujući tradicionalne papirne mape i adresare sa kojima se interakcija obavlja korišćenjem digitalne olovke s jedne strane, i digitalne mobilne uređaje sa malim ekranima sa druge (Norrie, 2007; Signer, 2007). Kontekstualne funkcije nalaze primenu i u geoinformacionim sistemima, gde se mogu koristiti za kreiranje personalizovanih geografskih mapa koje prikazuju tačke od interesa za određenog korisnika, na osnovu prikupljenih informacija o kontekstu i korisničkih postavki (Wilson, 2007). Primene u poslovnim sistemima su takođe od značaja, uključujući aplikacije za oporavak od izuzetaka (*exceptions*), odnosno grešaka u aplikacijama za poslovne procese (Adams, 2007), reinženjering poslovnog procesa otkrivanjem prirode procesa iz kontekstualnih informacija (Bessai, 2008), ili optimalno upravljanje poslovnim procesima (Bucchiarone, 2011).

Kontekstualne aplikacije za primenu u komunikacionim sistemima uključuju komunikacione klijente opremljene kontekstualnim funkcijama, aplikacije za mobilnu telefoniju i aplikacije za preporučivanje komunikacionih servisa. Jedna od osnovnih ideja proširenja komunikacionih aplikacija koja se sreće u većem broju istraživanja, jeste uključivanje kontekstualnih informacija iz okoline svake od strana u komunikaciji, u komunikacionu sesiju, najčešće neposredno pre njenog uspostavljanja. Na ovaj način, na primer, strana koja inicira komunikaciju može da zna da li je trenutak za komunikaciju, odnosno za neke teme u komunikaciji, podesan u smislu zauzetosti i raspoloženja potencijalnog sagovornika. Do sada su realizovana, na primer, poboljšanja za tekstualne komunikacione (*chat*) klijente (Ranganathan, 2002; Fogarty, 2004; Hong, 2010) i audio pozive u mobilnoj telefoniji (Schmidt, 2000; Bardram, 2004; Danninger, 2008). Aplikacije za mobilnu telefoniju najčešće se suočavaju sa problemom izbora najbolje tehnologije veze ili pristupne tačke, u cilju očuvanja neprekidnosti usluge, pa u tu svrhu koriste kontekstualne informacije poput lokacije, osobina dostupnih pristupnih tačaka i sl (Udugama, 2007). Aplikacije za preporučivanje komunikacionih servisa obuhvataju, na primer, automatsko ubacivanje reklama u program interaktivne televizije zasnovano na trenutnim i prošlim izborima korisnika i ostalim kontekstualnim informacijama (Thawani, 2004), kao i preporučivanje TV servisa korisniku (Tsunoda, 2008).

Aplikacije za trgovinu i marketing sa kontekstualnim proširenjima omogućavaju procenu interesovanja korisnika za kupovinu određenog proizvoda, i u tom smislu omogućavaju optimizaciju marketinških strategija u cilju što boljeg prijema proizvoda od strane korisnika (Kwon, 2003). Takođe, u svetu mobilnih uređaja kontekstualne funkcije omogućavaju kreiranje aplikacija koje uživo prenose informacije o kontekstu vezane za određene mogućnosti kupovine (npr. popusti u određenoj radnji) do ostalih zainteresovanih korisnika, korišćenjem *servisnih anotacija* (Riva, 2007, str. 1961). Direktna primena kontekstualnih aplikacija u prodavnicama realizovana je u vidu kolica za kupovinu sa ekranom osetljivim na dodir (Slika 9), koja navode kupca na željene proizvode i omogućavaju uvid u trenutni iznos kupovine na osnovu RFID oznake (Black, 2009; Kahl, 2011). Aplikacije za mobilne uređaje koje omogućavaju kupovinu i informisanje o proizvodima uz pomoć čitača QR kodova sa proizvoda (Fu, 2011), ili zasnovano samo na optičkom senzoru (Xu, 2008), neka su od najnovijih postignuća u oblasti kontekstualnih aplikacija za trgovinu.



Slika 9. Aplikacija za prodavnice: ekran na osnovu lokacije ukazuje na najbliže proizvode i omogućava pregled i poređenje (Black, 2009, str. 2)

Kontekstualne aplikacije za podršku internet servisima najčešće se odnose na generisanje profila korisnika u cilju odabiranja optimalnog web servisa za rešavanje traženog problema, kako bi se poboljšali iskustvo i performanse korisnika na internetu (Blake, 2007; Ceri, 2007). Personalizacija web servisa, odnosno podešavanje režima rada tako da on bude podesan trenutnoj lokaciji korisnika, kontekstu njegove pretrage interneta, strukturi njegovih saradnika i sl. takođe je zanimljiv predmet za razvoj kontekstualnih aplikacija (Gandon, 2004). Realizacija aplikacija i okruženja za

Stanje u oblasti

podršku razvoju kontekstualnih internet servisa je aktuelan istraživački problem, sa predloženim rešenjima poput aplikacija za grafičko modelovanje kontekstualnih internet servisa korišćenjem UML jezika (Sheng, 2009), ili modelovanje zavisnosti događaj->akcija korišćenjem deklarativnog opisa u obliku XML (Beach, 2010).

1.1.1.5 Nivo korisničke sprege

Nivo korisničke sprege kontekstualne platforme realizuje metode prezentacije informacija korisnicima, odnosno omogućava korisnicima da na određeni način saopšte ulazne informacije sistemu. Način interakcije korisnika sa kontekstualnom platformom može biti *eksplicitan*, kada korisnici unose informacije u sistem korišćenjem korisničke sprege koja je za to predviđena i koja je poznata korisnicima, odnosno *implicitan*, kada se interakcija obavlja indirektno, bez nužne potrebe da korisnik zna za interakciju, posredstvom senzorskih komponenti (Rehman, 2007). Eksplicitan način interakcije najčešće se obavlja korišćenjem ekrana, često osetljivih na dodir, koji su pričvršćeni na unapred određene površine u okviru inteligentnog okruženja (Kahl, 2011), ili su deo mobilnih uređaja kojima korisnici rukuju (mobilni telefoni, PDA i tablet računari i sl) (Shin, 2010). Neki drugi načini interakcije uključuju glasovne komande i zvučna obaveštenja (Hatala, 2005), klasične računarske sprege (miš i tastaturu) ili neke druge netipične sprege poput svetlosnih indikatora (Bjelica, 2011d), zadavanja akcija gestovima rukom (Mrazovac, 2011) ili pokretima tela (Bigdelou, 2012) i sl.

Iako je osnovna težnja kontekstualnih platformi da se minimizuje vidljivost interakcije sa korisnicima, i da se informacije o korisnicima generišu od strane mrežnog nivoa i senzorskih komponenti, odnosno da se koristi isključivo implicitan način interakcije, često je neophodno praćenje različitih parametara rada sistema, i konfigurisanje sistema od strane korisnika. U nekim inteligentnim okruženjima, poput domaćinstva, korisnici koriste korisničku spregu za postavljanje režima rada sistema, ili za zadavanje eksplicitne komande nekom od uređaja. Nakon pažljive analize istraživanja na ovom nivou, izdvajaju se sledeći osnovni razlozi za postojanje korisničkih sprega u okviru kontekstualne platforme i eksplicitnu interakciju sa korisnikom: (1) nesavršenost kontekstualne platforme, u smislu da platforma nije u mogućnosti da samo na osnovu senzorskih komponenti sazna o osobinama i navikama

korisnika, predmeta ili pojava, te je za pravilan rad neophodno zadati određena pravila i postavke; (2) aspekt kontrole za koji se vezuje potreba da korisnik u svakom trenutku može na zahtev da izmeni način rada sistema, bez obzira na to što, u idealnom slučaju, sistem radi optimalno; (3) konvergencija korisničkih radnji, čime kontekstualna platforma omogućava da korisnik aktivira određene radnje posredstvom kontekstualne platforme, za koje je inače koristio jedan ili više drugih sistema, ili obavljao neke manualne akcije i (4) prikaz dodatne vrednosti, koji zahteva postojanje korisničke sprege kako bi se korisniku saopštila informacija generisana od strane kontekstualne platforme, a koja nije podesna za prikaz korišćenjem nekog od postojećih komunikacionih medijuma u okruženju (Hatala, 2005; Hong, 2007; Korhonen, 2007; Lieberman, 2007; Dey, 2009; Bjelica, 2011e).

Jedan od osnovnih problema u istraživanjima grafičkih korisničkih sprega sa aspekta kontekstualnih platformi, jeste potreba da se istovetan sadržaj prikaže na različitim ekranima različitih osobina, rezolucija i performansi. U cilju rešenja ovog problema predložena su okruženja koja obavljaju prilagođavanje korisničke sprege u zavisnosti od medijuma za prikaz i nekih informacija o kontekstu (Calvary, 2003; Butter, 2007; David, 2011). Neka istraživanja u oblasti korisničkih sprega problem posmatraju sa šireg aspekta, te se pronalaze rešenja za realizaciju korisničkih sprega koje su prilagodljive različitim ulaznim i izlaznim mogućnostima uređaja, odnosno *multimodalne* korisničke sprege (Dumas, 2009).

Kontekstualne i adaptivne korisničke sprege pokriva posebna oblast istraživanja i one su od posebnog interesa za temu ove disertacije, te su detaljno obrađene u poglavlju 1.2. Za razliku od korisničkih sprega koje služe za prikaz informacija i interakciju sa korisnicima u smislu eksplicitne kontrole i upravljanja radom opštenamenske kontekstualne platforme, kontekstualne korisničke sprege *u širem smislu* obuhvataju mehanizme upotrebe kontekstualne platforme u cilju kreiranja dodatne vrednosti radu same korisničke sprege. Tako kontekstualne korisničke sprege treba posmatrati dvojako: kao jednu od aplikacija razvijenu na osnovu kontekstualne platforme, ali i kao medijum za saradnju kontekstualne platforme i korisnika.

1.2 KONTEKSTUALNE KORISNIČKE SPREGE

Istraživanja u oblasti kontekstualnih korisničkih sprega zasnivaju se na *adaptivnosti* korisničkih sprega, odnosno mogućnosti prilagođavanja kontekstu. Opšti utisak stečen pregledom oblasti jeste da postoje tri velike grane različitih istraživanja u oblasti kontekstualnih korisničkih sprega, pa se može napraviti sledeća klasifikacija kontekstualnih korisničkih sprega: (1) *sprega sa izmenljivim režimima rada*, koje obuhvataju klasične korisničke sprega poboljšane tako da podržavaju više različitih izgleda, odnosno načina rada, koji se mogu menjati na osnovu informacija dobijenih od strane kontekstualne platforme; (2) *asistentske sprega*, koje na osnovu kontekstualnih informacija uče i predviđaju akcije korisnika u cilju pružanja pomoći i automatizacije radnji; i (3) *transparentne sprega*, koje se utapaju u korisničko okruženje i teže da budu nevidljive za korisnike.

Sprege sa izmenljivim režimima rada zasnivaju se na izmeni načina rada i izgleda korisničke sprega, najčešće u zavisnosti od detektovane korisničke aktivnosti. Aplikacija *Nokia Situations* za mobilne telefone omogućava zadavanje različitih situacija (*Spavam, Na koncertu, Igram se sa decom, Na sastanku* i sl) na osnovu parametara poput vremena ili GPS lokacije, te definisanje režima rada u određenoj situaciji (ton i jačina zvona, automatski odgovor na poruke i sl) (Nokia Situations, 2011). Okruženje *Mylyn* za *Eclipse* omogućava izmenu korisničkog okruženja tako da odgovara trenutnoj aktivnosti programera, što u velikoj meri olakšava rad s obzirom na kompleksnost korisničke sprega i veliki broj različitih mogućnosti (Mylyn, 2012). Koncept ovih sprega najpre je preciziran u radu A. Schmidt-a, koji je predložio arhitekturu za omogućavanje kreiranja režimske korisničke sprega, koja menja režime rada na osnovu izmene lokacije i/ili aktivnosti korisnika, sa primenom u mobilnim aplikacijama (Schmidt, 1999). Na osnovu te arhitekture kreirane su različite aplikacije za podešavanje korisničke sprega. Na primer, *TaskTracer* aplikacija omogućava otkrivanje režima rada korisnika za računarom, i automatsko podešavanje korisničke sprega da najbolje odgovara određenom režimu (postavljanje putanje do odgovarajućeg radnog direktorijuma, kreiranje prečica na desktopu, pokretanje aplikacija potrebnih za rad i sl.) (Dragunov, 2005). Još jedno istraživanje realizuje sistem za automatsko podešavanje profila na mobilnom telefonu u zavisnosti od prethodnog načina korišćenja upotrebom fazi logike (Valtonen, 2009).



Slika 10. Adaptivna korisnička sprega za odabir muzičkog sadržaja, istaknutog u zavisnosti od sličnosti albuma i afiniteta korisnika (Strober, 2010, str. 388)

Asistentske sprega najčešće se poklapaju sa definicijom adaptivnih korisničkih sprega (*adaptive user interfaces*), koje se gotovo isključivo odnose na sprega za praćenje ponašanja korisnika i predviđanje njegovih akcija i interesovanja. Veliki broj realizovanih adaptivnih korisničkih sprega koristi se u komercijalne i marketinške svrhe, za izbor i prezentaciju informacija korisniku koje su u skladu sa njegovim prethodnim ponašanjem i interesovanjima. Najbolji primer za to je internet pretraživač *Google*, koji rezultate pretrage predstavlja korisniku u redosledu koji je u skladu sa njegovim postavkama i profilom koji je zaključen na osnovu ranijeg ponašanja korisnika (Google Search, 2009). *Google AdSense* omogućava izbor i prikazivanje tekstualnih reklama na internet stranicama koje najbolje odgovaraju posetiocu stranice (AdSense, 2012). *Amazon* preporučuje posetiocima internet stranice one proizvode za koje su bili zainteresovani i koje su kupovali drugi posetioci, korišćenjem tehnike kolaborativnog filtriranja informacija (Linden, 2003). Novija naučna istraživanja u oblasti adaptivnih korisničkih sprega su brojna. Na primer, korisnička sprega koja obavlja ulogu asistenta na radnom mestu poboljšava organizaciju posla tako što automatizuje repetitivne aktivnosti i ukazuje na moguće greške, a sve na osnovu praćenja aktivnosti korisnika (Myers, 2007). Na institutu MIT razvijena je korisnička sprega koja na osnovu aktivne povratne sprega od strane studenata, kao i dužine pregleda i tipa pregledanih materijala, određuje materijale koji su najrelevantniji za određeni kurs i predlaže ih studentima (Umapathy, 2011). Korisnička sprega za

Stanje u oblasti

pretragu muzičkih sadržaja koristi inovativan način prezentacije omota albuma u obliku „galaksije“ (Slika 10), tako da najistaknutiji albumi budu povezani po međusobnoj sličnosti i prema afinitetima korisnika (Stober, 2012).

Oblast transparentnih korisničkih sprega obuhvata sprege čija je suština dizajna i interaktivnosti takva, da se one uklapaju u korisničko okruženje, i najčešće se odnosi na sprege za interakciju sa inteligentnim okruženjima. Primena ovakvih korisničkih sprega nalazi se u aplikacijama augmentirane realnosti (*augmented reality*). Pod transparentne korisničke sprege mogu se svrstati *opipljive* korisničke sprege (*tangible user interfaces*), kao i *nepostojane* korisničke sprege (*ephemeral user interfaces*), mada i klasične korisničke sprege, poput ekrana osjetljivih na dodir, mogu biti transparente ukoliko su, na primer, ugrađene u postojeće uređaje koji su prihvaćeni od strane korisnika, na način koji ne zahteva aktiviranje posebnog režima rada tih uređaja (dakle, moraju da budu realizovani u sklopu postojećih korisničkih sprega).

Postoji nekoliko osnovnih premisa za realizaciju opipljivih korisničkih sprega. Najpre, preplitanje fizičkog (opipljivog) i digitalnog sveta zahteva dizajniranje sprega za korišćenje fizičkih objekata koji najbolje odgovaraju željenoj digitalnoj informaciji koja se saopštava putem sprega (Ullmer, 2002). Dalje, opipljive korisničke sprege su u suštini kontekstualne, što znači da rezultat njihove upotrebe zavisi od konteksta u kojem se koriste, odnosno od interakcije sa drugim opipljivim objektima i/ili korisničkim spregama. Na tržištu ne postoje standardizovane opipljive korisničke sprege, već one nastaju kao kombinacija više ulazno-izlaznih medijuma i kontekstualnih informacija. Korisnici ovih sprega očekuju kontinualnu interakciju, bez eksplicitnog zahteva za njom, kao i mogućnost da više korisnika uporedo koriste spregu (Shaer, 2009, str. 4-5). Na primer, obrazovna korisnička sprega *Tern* omogućava učenicima da kreiraju računarski program za prolazak robota kroz lavirint korišćenjem drvenih slagalica (Horn, 2007). Opipljivi video editor *TVE* omogućava montažu filmskih snimaka povezivanjem fizičkih komponenti sa ekranima na kojima se reprodukuju odgovarajući delovi snimaka (Zigelbaum, 2007). Opipljive sprege za inteligentna okruženja, poput domaćinstva, kreirane su u formi demonstratora sa utikačima koji prihvataju, npr. muzički plejer, sistem ambijentalnog osvetljaja i sl. i na osnovu priključenih uređaja kreiraju ambijentalni scenario za domaćinstvo (Vlist, 2010).



Slika 11. Opipljive korisničke sprege: Video editor (levo) (Zigelbaum, 2007, str. 2) i Tern, slagalica za programiranje (desno) (Horn, 2007, str. 2)

1.3 OBLAST POTROŠAČKE ELEKTRONIKE

Oblast potrošačke elektronike, kao oblast primene rezultata ove doktorske disertacije, poseduje specifičnosti koje kreiraju nove izazove za primenu kontekstualnih platformi. Istraživanja u oblasti kontekstualnih platformi najčešće su opšteg tipa, odnose se na algoritme, metode i tehnike otkrivanja i saopštavanja konteksta, sa malo ili ni malo osvrta na *izvodljivost (feasibility)* njihove primene u svetu potrošača. Da bi se razumeo širi kontekst za primenu kontekstualnih platformi u potrošačkoj elektronici, najpre treba razumeti oblast potrošačke elektronike u celini.

Oblast potrošačke elektronike odnosi se na elektronske uređaje namenjene svakodnevnoj upotrebi, u sferi zabave, komunikacija i produktivnosti. Jednim od prvih uređaja potrošačke elektronike može se smatrati analogni radio prijemnik, proizveden početkom 20-og veka. Danas, uređaji potrošačke elektronike obuhvataju lične računare, mobilne telefone i tablete, MP3 plejere, audio opremu, TV prijemnike i STB (*set-top box*), GPS navigacione uređaje, digitalne fotoaparate i kamere, video plejere i rekordere, brojne uređaje za domaćinstvo itd. U poslednjoj deceniji trend povećanja broja prodatih uređaja potrošačke elektronike je izuzetno izražen, što čini oblast veoma atraktivnom za istraživanja. Prema istraživanjima tržišta, ukupan prihod od prodaje uređaja potrošačke elektronike raste uprkos nastavku svetske ekonomske krize, sa prihodom od preko 300 milijardi dolara u 2009, do 350 milijardi dolara u 2011. godini, i predviđenim rastom od oko 5% godišnje do 2014 (Selburn, 2011).

Personalizacija uređaja potrošačke elektronike i povećanje njihove mobilnosti predstavlja takođe nezaustavljiv trend. Na primer, broj prodatih pametnih telefona

Stanje u oblasti

(*smartphone*) u 2011. godini iznosi 488 miliona, što je porast od 62.7% u odnosu na 2010. godinu, i to tako da prvi put u istoriji broj prodatih pametnih telefona nadmašuje broj svih prodatih računara, uključujući desktop, laptop, netbuk i ručne (*pad*) računare, kojih je zajedno u 2011. godini prodato 415 miliona komada (Canalys, 2012). Takođe, broj korisnika tablet računara, koji su takođe namenjeni pojedinačnim korisnicima, povećava se izuzetno velikom brzinom, sa preko 60 miliona prodatih uređaja u 2011 (IDC, 2011), sa prognozama prodaje gotovo 100 miliona uređaja u 2012, što je povećanje za preko 55% (JP Morgan, 2011). U ovakvoj situaciji od velikog je interesa za proizvođače da ponude maksimalan kvalitet i funkcionalnost svojih proizvoda korisnicima.

Za razliku od opštenamenskih računarskih sistema, najveći broj uređaja potrošačke elektronike može se definisati korišćenjem sledećih grupa zahteva: (1) tehničko-tehnološki zahtevi, koji uključuju zahteve za niskom potrošnjom energije i sertifikacione zahteve; (2) zahtevi za povezivanjem i integracijom, koji proizilaze iz opšteg trenda digitalne konvergencije; (3) zahtevi za visokom konkurentnošću i kratkim vremenom revizije, i (4) specifični korisnički zahtevi, koji proizilaze iz raznovrsnosti i širine ciljne grupe korisnika.

Tehničko-tehnološki zahtevi koji se postavljaju pred uređaje potrošačke elektronike su brojni, i uslovljeni istovremeno i državnim regulativama na prostoru na kome se prodaju, kao i zahtevima tržišta. Zahtevi za niskom potrošnjom energije su prirodni za mobilne uređaje sa stanovišta relativno niskog kapaciteta današnjih baterija, tako da ove zahteve diktira tržište. Na primer, potrošnja jednog modernog pametnog telefona u pripravnom (*standby*) režimu iznosi oko 70 mWh, a za vreme razgovora oko 1 Wh (Carroll, 2010, str. 4,8), dok je gustina energije jedne prosečne litijum-jonske baterije, poput Nokia BL-5C, ispod 4 Wh. Dakle, autonomija rada ovakvih uređaja je i dalje prilično skromnog trajanja. S druge strane, u toku je žestoka kampanja za globalnu uštedu energije i podizanje svesti korisnika o potrošnji energije uređaja u domaćinstvu, u cilju smanjenja globalne potrošnje energije a samim tim i emisije štetnih supstanci prilikom njene proizvodnje. Na primer, inicijativa *Jedan vat* (*One Watt Initiative*) Međunarodne agencije za energiju ima za cilj smanjenje potrošnje enegije uređaja, dok je u pripravnim režimu rada, na manje od 1 W do 2010, odnosno 0.5 W do 2013 (Meier, 1999). U skladu sa ovom inicijativom donete su

regulative o potrošnji energije u pripravnim režimu rada koje su obavezujuće za proizvođače u Sjedinjenim Državama, (Executive Order 13221, 2001) i Evropi (EC Regulation 1275, 2008). Ostale važne regulative koje se primenjuju na uređaje potrošačke elektronike su, na primer, regulativa o zabrani korišćenja određenih štetnih supstanci za izradu elektronske opreme Saveta Evrope (EC Directive 2002/95/EC, 2003), kao i specifikacija zahteva elektromagnetske kompatibilnosti uređaja (EN 55014-2, 2009). Ostali tehničko-tehnološki zahtevi se odnose na dostupnu procesorsku moć, količinu memorije i ulazno/izlazne sprege, u smislu njihove raspoloživosti za aplikacije, kao i njihove raznolikosti u različitim uređajima. Ispunjenje ovih zahteva direktno utiče na fizičku arhitekturu i programsku podršku uređaja.

Oblast potrošačke elektronike odlikuje se utrivanjem proizvođača da ponude potrošačima sve veći broj novih funkcionalnosti uređaja. Smanjenjem dimenzija integrisanih kola i njihovom izuzetno visokom integracijom, te nastankom SoC (*System on Chip*) komponenti, mnogi uređaji potrošačke elektronike su dobili centralni procesor i memorijske podsisteme, što je omogućilo povećavanje broja funkcija koje jedan uređaj poseduje. Tome svedoči i trend digitalne konvergenije (*digital convergence*), koji označava objedinjavanje informacionih tehnologija, telekomunikacija (interneta), potrošačke elektronike i industrije zabave, odnosno, svi uređaji potrošačke elektronike teže da imaju pristup mreži (internetu) putem koje preuzimaju i dele multimedijalne sadržaje, komuniciraju sa drugim uređajima i sl (Strasser, 2010). Međutim, specifičnosti poput zahteva za smanjenom potrošnjom energije, malih dimenzija uređaja i sl. određuju prepreke za protokole veze koji su na raspolaganju. Upravo iz tih specifičnosti proizilaze i brojna, različita rešenja protokola veze koja se mogu naći u današnjim uređajima potrošačke elektronike. Suočavanje sa heterogenošću protokola veze je izazovan, ali ključan zadatak za omogućavanje transparentne integracije uređaja (Bottaro, 2008).

S obzirom da je potrošačka elektronika okrenuta prevashodno potrošaču, podvrgnuta je strogim ekonomskim zahtevima današnjeg uglavnom kapitalistički uređenog društva, odnosno, osnovna premisa je kontinualna prodaja i kontinualni prihod. Žestoka tržišna utakmica utiče da pritisak na kompanije da objavljuje nove ili revidirane verzije uređaja u što kraćim vremenskim intervalima bude sve veći. Trenutno prosečno vreme od trenutka početka dizajna uređaja do početka prodaje je

Stanje u oblasti

skraćeno na 15-18 meseci. Pritisak na proizvođače integriranih kola je još intenzivniji, pošto je ciklus u njihovom slučaju nešto veći (18-21 mesec) (KPMG, 2008, str. 17). Na važnost vremenske komponente pri razvoju integriranih kola ukazuje i intervju sa direktorima velikih kompanija, gde je na pitanje ako bi imali mogućnost da na magičan način reše jedan problem, koji bi to problem bio, 80% direktora odgovorilo: „Vreme“ (CES, 2008). Zbog toga se za razvoj uređaja potrošačke elektronike koriste različite savremene metodologije, poput istovremenog razvoja fizičke arhitekture i programske podrške (*Hardware-Software Co-Design*) (Balarin, 1997), agilnog razvoja programske podrške (Hazzan, 2008), *Scrum* metodologije upravljanja projektom (Schwaber, 2009) i dr.

Potrošačka elektronika odlikuje se nadasve specifičnim i raznovrsnim korisničkim zahtevima. Ciljna grupa korisnika je izuzetno široka, dok je tržište prema korisnicima segmentirano najčešće samo po godinama starosti. Dakle, pri dizajniranju uređaja potrošačke elektronike retko se može računati na prethodno iskustvo, znanje ili veštine koje korisnici poseduju. Osnovni kriterijum uspeha uređaja potrošačke elektronike je njegova prihvaćenost od strane korisnika. Dojam koji uređaj ostavlja na korisnika, u smislu kompleksnosti korišćenja, kompatibilnosti sa korisničkim potrebama i inovativnosti u odnosu na konkurenciju, važniji je od stvarnog, merljivog kvaliteta i vrednosti uređaja (Huang, 2011). Kompleksnost korišćenja se često navodi kao jedan od presudnih razloga za neprihvatanje uređaja i njegov neuspeh. Čak polovina od svih uređaja potrošačke elektronike koji su odneti na reklamaciju zbog kvara, ispostavilo se da su potpuno ispravni – potrošači samo nisu razumeli način njihovog korišćenja (Ouden, 2006). S obzirom da kontekstualne platforme i kontekstualne korisničke sprege omogućavaju korišćenje informacija o kontekstu za unapređenje korisničke sprege, primena kontekstualnih platformi u okviru uređaja potrošačke elektronike je od sve većeg značaja.

Oblast potrošačke elektronike je veoma izazovna u smislu integracije i primene kontekstualnih funkcija u uređajima. Osim zahteva u pogledu potrošnje, sertifikata, raspoloživih resursa itd. od izuzetnog je interesa integracija kontekstualnih funkcija bez narušavanja izgleda i funkcionalnosti koje su tradicionalno prihvaćene za određeni uređaj (Bjelica, 2010a). Zbog toga se metode realizacije kontekstualnih platformi i kontekstualnih korisničkih sprega za primene u uređajima potrošačke elektronike

razlikuju od metoda navedenim u opštim istraživanjima u oblasti, primenjujući samo podskup predloženih rešenja i usvajajući nova rešenja, koja ispunjavaju najveći broj zahteva vezanih za potrošačku elektroniku.

1.4 KONTEKSTUALNE PLATFORME U POTROŠAČKOJ ELEKTRONICI

Primena kontekstualnih platformi u potrošačkoj elektronici još uvek nije doživela potreban uzmah, a postojeća rešenja poseduju ograničen skup mogućnosti koji je često veoma uslovljen eksplicitnim podešavanjem od strane korisnika, a veoma malo ili ni malo implicitnim prikupljanjem kontekstualnih informacija. S obzirom da oblast potrošačke elektronike i istraživanja u ovoj oblasti imaju smisla samo ako se realizovano rešenje konačno plasira potrošaču, pod postojećim rešenjima smatraju se samo ona rešenja koja se mogu, ili su se u nekom trenutku mogla kupiti u prodavnici (*off the shelf* – *OTS*). U nastavku će biti prikazana takva rešenja.

1.4.1 Kontekstualne platforme namenjene inteligentnim okruženjima

Rešenja namenjena inteligentim okruženjima odnose se na sisteme za automatizaciju domaćinstva, sa mogućnošću lake integracije većeg broja senzorskih komponenti u cilju otkrivanja informacija o kontekstu. Većina postojećih sistema za automatizaciju domaćinstva nije u potpunosti kontekstualna, međutim, mnoga se mogu na odgovarajući način proširiti i/ili programirati da ponude kontekstualne funkcije korisnicima, s obzirom na podudarnost arhitekture nivoa mreže, dela srednjeg nivoa i nivoa korisničke sprege sa opštom arhitekturom kontekstualne platforme. U Tabeli 4 dat je spisak vodećih svetskih sistema za automatizaciju domaćinstva, uz kategorizaciju po podršci za osnovne zahteve kontekstualnih platformi. Prikazani su samo oni sistemi koji u određenoj meri podržavaju kontekstualne funkcije u okviru srednjeg nivoa, odnosno oni sistemi, na osnovu kojih su realizovane kontekstualne aplikacije u prethodnim istraživanjima, uz date odgovarajuće reference.

<i>Naziv sistema</i>	<i>Osobine sistema</i>		<i>Referenca</i>
Control4	Mrežni nivo	Zigbee, Ethernet/WiFi, RS232	Bjelica, 2011e
	Srednji nivo	Srednji nivo zasnovan na metapodacima, polise (drajveri) u Lua jeziku	
	Aplikativni nivo	Aplikacije za 4Store: <i>Wakeup, Vacation</i>	
	Nivo korisničke sprege	TV ekran, ekrani osetljivi na dodir, tasteri, daljinski upravljač	
Adhoco	Mrežni nivo	Zigbee, Radio, Ethernet	Guillemin, 2003
	Srednji nivo	Srednji nivo zasnovan na OSGI okruženju	
	Aplikativni nivo	Inteligentna kontrola roletni sa učenjem	
	Nivo korisničke sprege	Daljinski upravljač, mobilni telefon	
AMX	Mrežni nivo	Ethernet-WiFi, Zigbee, RS232	Franklin, 2001
	Srednji nivo	Srednji nivo zasnovan na metapodacima, polise u jeziku NetLinx	
	Aplikativni nivo	Inteligentna učionica	
	Nivo korisničke sprege	Prezentacioni ekrani, ekrani osetljivi na dodir, tasteri, daljinski upravljač, mobilne aplikacije (Android, iOS) i sl.	

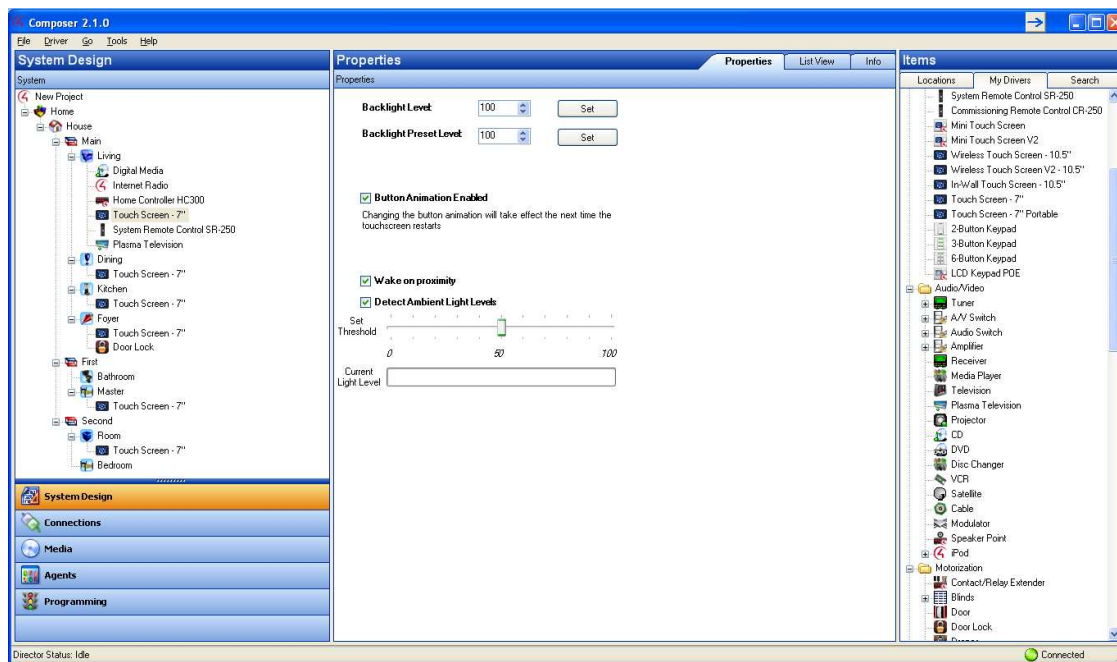
CRESTRON	Mrežni nivo	Ethernet, Cresnet, radio veza (infiNET), WiFi, Zigbee, RS232	Vergnes, 2005
	Srednji nivo	Nema. U realizovanom rešenju (Vergnes, 2005) koristi se hibridni srednji sloj zasnovan na agentima i na metapodacima (korisničkim profilima)	
	Aplikativni nivo	Asistencija osobama sa posebnim potrebama	
	Nivo korisničke sprege	Ekрани osetljivi na dodir, prekidači, daljinski upravljači, mobilne aplikacije, PC aplikacije. Programabilan UI korišćenjem Core3UI paketa.	
OBLO	Mrežni nivo	Ethernet, WiFi, Zigbee, Z-Wave	Mrazovac, 2011
	Srednji nivo	Srednji nivo zasnovan na metapodacima, skriptovi u Lua jeziku	
	Aplikativni nivo	Inteligentno upravljanje potrošnjom energije	
	Nivo korisničke sprege	Daljinski upravljač, Web interfejs, prekidači, mobilna aplikacija (Android)	
Philips Retail Solutions	Mrežni nivo	Ethernet, WiFi, RS232	Monaci, 2011
	Srednji nivo	Srednji nivo zasnovan na metapodacima, polise u Lua jeziku	
	Aplikativni nivo	Prodavnica sa svetlosnim efektima koji se prilagođavaju posetiocima.	
	Nivo korisničke sprege	Prezentacioni ekrani, ekrani osetljivi na dodir, web interfejs, senzorne zone	

Tabela 4. Sistemi za automatizaciju domaćinstva kao osnova kontekstualnih platformi

Stanje u oblasti

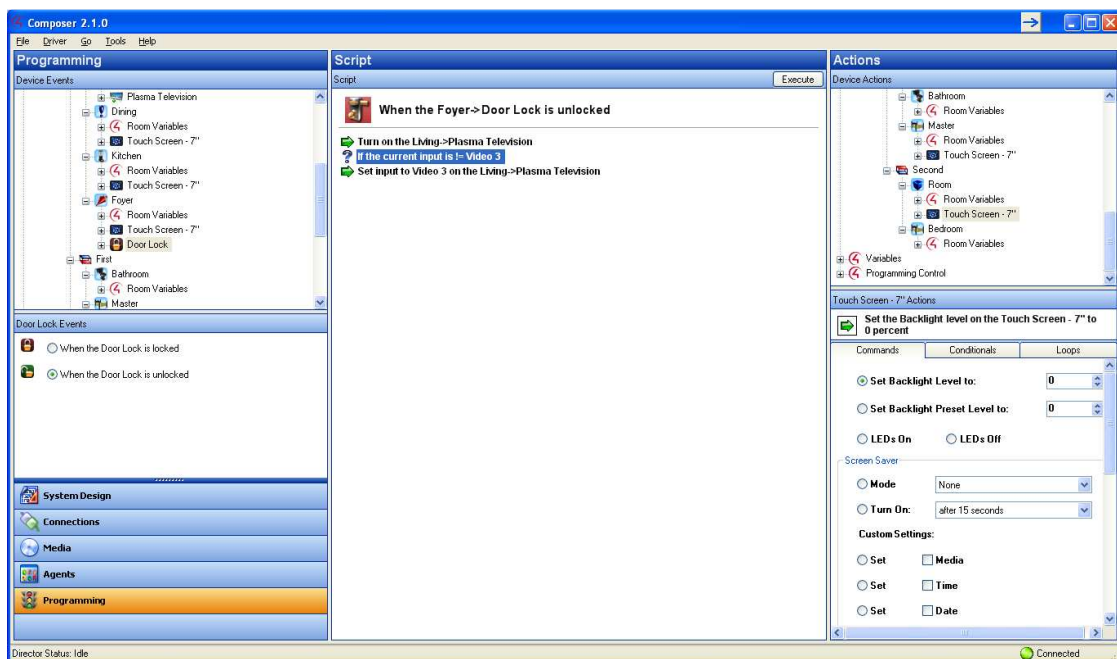
Princip funkcionisanja kontekstualnih platformi u potrošačkoj elektronici, koje su zasnovane na sistemu za automatizaciju domaćinstva, biće detaljnije razmotren na primeru aplikacija *Wakeup* i *Vacation*, realizovanih za *Control4* sistem.

Osnovna konfiguracija *Control4* sistema sadrži sledeće gradivne blokove: (1) centralni kontroler (HC-300), (2) ekran osetljiv na dodir veličine dijagonale 7", (3) Zigbee prekidače za kontrolu svetla, (4) infracrvene senzore za kontrolu TV uređaja i klimatizacione jedinice. Osnovna programska podrška centralnog kontrolera zasniva se na aplikaciji *Director*, koja upravlja celokupnim domaćinstvom. Za konfiguraciju i programiranje ponašanja *Director* aplikacije, obezbeđena je korisnička sprega u obliku PC aplikacije za Windows operativni sistem – *Composer*. Nakon pokretanja, *Composer* aplikacija obavlja povezivanje sa centralnim kontrolerom (odnosno, sa odgovarajućom *Director* aplikacijom), te preuzima i prikazuje trenutno aktivni projekat domaćinstva. Projekat domaćinstva se sastoji od ontološkog prikaza domaćinstva podeljenog u sobe, u kojima se nalaze različiti uređaji. *Composer* omogućava izmenu projekta domaćinstva, kao npr. dodavanje soba i uređaja u sobe iz velike dostupne biblioteke podržanih uređaja. Nakon dodavanja, potrebno je obaviti programsko povezivanje uređaja (koje korespondira fizičkoj vezi), kao i konfigurisati parametre uređaja. Sve ovo omogućava aplikacija *Composer* (Slika 12).



Slika 12. Izgled *Control4 Composer* aplikacije

Iako se samim konfigurisanjem sistema omogućava kontrolabilnost (kontrola instaliranih uređaja posredstvom ekrana osjetljivih na dodir ili daljinskog upravljača), dodatna inteligencija se postiže programiranjem. *Composer* omogućava jednostavno programiranje korišćenjem odeljka *Programming*. Ovaj odeljak se sastoji od tri osnovne grupe: događaja (*events*), naredbi (*commands*) i programskog opisa. Korisnik odabira događaj iz panela sa leve strane, koji služi kao pokretač sekvence naredbi koje korisnik može da odabere iz panela sa desne strane. Osim ove osnovne mogućnosti, takođe je moguće uslovno izvršavanje komandi, korišćenje varijabli i sl (Slika 13).



Slika 13. Odeljak *Programming* u *Composer* aplikaciji

Control4 sistem se sastoji od funkcionalnih blokova koji se u velikoj meri podudaraju sa arhitekturom kontekstualne platforme, stoga je Control4 sistem moguće koristiti za realizaciju funkcija kontekstualne platforme.

Na mrežnom nivou, Control4 omogućava interakciju sa okruženjem korišćenjem Ethernet, WiFi, Zigbee i RS232 protokola. Zajedno sa postojanjem analognih i digitalnih ulaza, ovaj skup protokola omogućava dobru interakciju sa senzovima, što je ključan zahtev za realizaciju kontekstualnih aplikacija. Centralni kontroler poseduje odgovarajuće blokove fizičke arhitekture i pripadajuće Linux rukovaoce koji omogućavaju korišćenje navedenih protokola. *Director* aplikacija poseduje biblioteke koje omogućavaju korišćenje mrežnih primitiva od strane srednjeg nivoa.

Stanje u oblasti

Srednji nivo je gotovo u potpunosti zasnovan na konceptu *drajvera*, realizovanih u Lua jeziku. Control4 drajveri omogućavaju realizaciju logike između fizičkog uređaja i korisničke sprege, odnosno modula za programiranje. Svaki uređaj, a često i korisnička aplikacija, poseduju Lua drajver koji se pokreće pri njihovoj inicijalizaciji. Ovaj drajver ima pristup mrežnim primitivama korišćenjem posebno razvijenog Lua API-ja za tu namenu. Takođe, postoji API za objavljivanje podataka korisničkim spregama, odnosno prijem određenog unosa iz korisničke sprege, kao i API za objavljivanje događaja i prijem naredbi za korišćenje uređaja prilikom programiranja u *Composer*-u. Ovakav srednji nivo, koji je po arhitekturi podudaran srednjem nivou zasnovanom na metapodacima, pri čemu su polise ekvivalentne Lua drajverima, omogućava realizaciju različite kontekstualne logike potrebne za različite kontekstualne aplikacije.

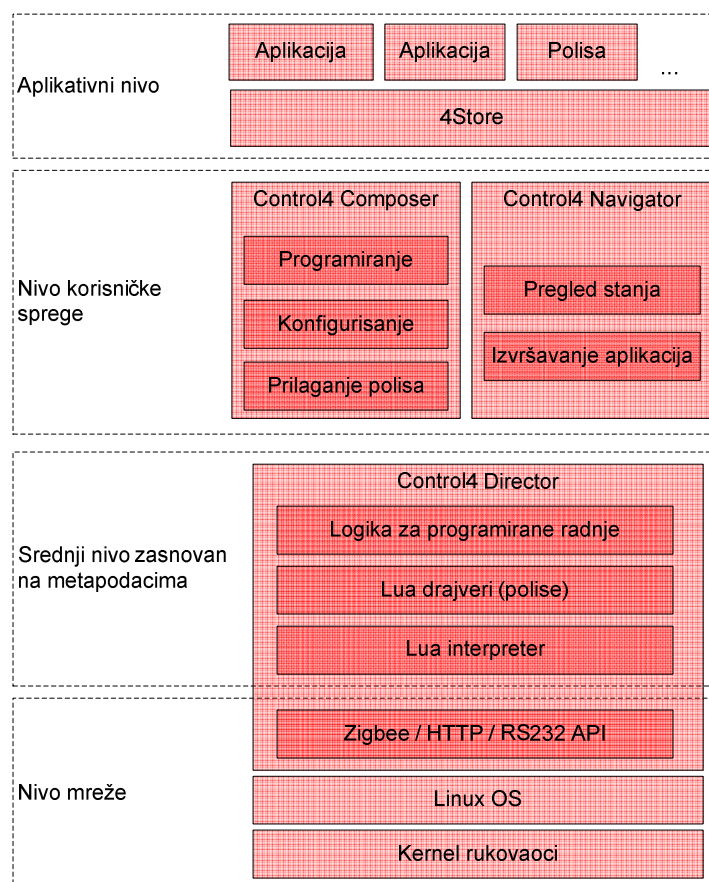
Nivo korisničke sprege realizovan je kao *Adobe Flash* aplikacija pod nazivom *Navigator*, koja korišćenjem TCP protokola razmenjuje XML objekte sa *Director* aplikacijom. Ova aplikacija se izvršava na uređajima sa ekranom osetljivim na dodir, na PC računaru, mobilnim telefonima ili kao web aplikacija. Ovako je omogućeno korisnicima da korišćenjem iste korisničke sprege obavljaju interakciju sa sistemom sa različitih pristupnih tačaka (Slika 14).



Slika 14. Izgled Control4 Navigator aplikacije

Realizacija aplikacija na aplikativnom nivou za kontekstualne platforme sa Control4 sistemom kao osnovom, može se obaviti na dva načina: (1) korišćenjem *Programming* odeljka *Composer* aplikacije, i (2) realizacijom posebne aplikacije za

Control4 *Navigator*. S obzirom da su mogućnosti *Programming* odeljka ograničene na sekvence *događaj->naredba*, što otežava realizaciju kompleksnijih aplikacija, kao i da *Programming* odeljak ima ograničene mogućnosti interakcije sa korisničkim spregama, razmotrićemo drugu mogućnost. Naime, Control4 sistem omogućava jednostavno proširivanje *Navigator* aplikacije različitim korisničkim *Flash* aplikacijama koje su pisane korišćenjem *Navigator SDK* paketa biblioteka za *Flash Actionscript*. Sve takve aplikacije mogu se priložiti korišćenjem servisa *4Store*, odakle ih korisnici, po potrebi, mogu preuzimati (plaćajući naknadu ili ne) na svoje lokalne uređaje. Aplikacija, nakon pokretanja, ima pristup kompletnom Control4 sistemu i ima vezu sa internetom. Ovim se omogućava gotovo neograničena sloboda realizacije aplikacija koje kao osnovu koriste *Control4* sistem i internet servise. Dodatno, aplikacija, pri instalaciji, može da priloži sopstveni *Lua* drajver (polisu) u Control4 sistem, čime je praktično omogućeno neprekidno izvršavanje logike, bez potrebe za stalnom aktivnošću aplikacije.



Slika 15. Arhitektura Control4 sistema preslikana na zahteve kontekstualnih platformi

Stanje u oblasti

Na Slici 15 dat je dijagram na kome je prikazana arhitektura Control4 sistema i preslikavanje funkcionalnosti na zahteve za realizaciju kontekstualne platforme.

U nastavku biće dat primer realizacije dve kontekstualne aplikacije objavljene u *4Store* biblioteci. U pitanju su aplikacije *Wakeup* i *Vacation* (C4Central, 2011). Obe aplikacije sastoje se od grafičke korisničke sprege za konfigurisanje realizovane u obliku *Flash* aplikacije, i odgovarajuće polise pisane u *Lua* jeziku. Aplikacija *Wakeup* omogućava kreiranje naprednih scenarija jutarnjeg buđenja, korišćenjem postepenog uključivanja svetla, uključivanjem omiljene muzike ili TV kanala, zagrevanjem prostorije na željenu temperaturu i automatskim otvaranjem roletni. Korisnik ima mogućnost da odabere vreme i efekte koje želi da primeni na scenario buđenja. Aplikacija *Vacation* omogućava simulaciju prisustva ukućana u domaćinstvu, tako što, nakon aktiviranja, nastavlja da koristi TV, uključuje/isključuje svetla, kontroliše roletne i sl, u zavisnosti od korisničkih postavki. Izgled ekrana *Wakeup* i *Vacation* aplikacija dat je na Slici 16.



Slika 16. Izgled ekrana *Vacation* aplikacije (gornji red) i *Wakeup* aplikacije (donji red) na uređaju sa ekranom osetljivim na dodir u okviru Control4 Navigator aplikacije

Autor ove disertacije je imao priliku da u dužem vremenskom periodu koristi Control4 sistem i da učestvuje u razvoju različitih aplikacija. Na primeru Control4 sistema mogu se uočiti osnovni nedostaci kontekstualnih platformi u potrošačkoj elektronici namenjenih inteligentnim okruženjima, koji su gotovo bez izuzetka uočljivi i u svim ostalim opštenamenskim kontekstualnim platformama za inteligentna okruženja: (1) prenatrpan programski pristup, umesto učenja; (2) nedovoljno oslanjanje na senzorske ulaze; (3) nedovoljno korišćenje povratne sprege od strane korisnika za fino podešavanje i (4) samocentričnost aplikacija. Osnovna napomena je da pri navođenju ovih mana u obzir se uzima kompletan potencijal platforme, ne osvrćući se pritom isključivo na postojeće kontekstualne aplikacije, kojih je veoma malo i koja u minimalnoj meri podražavaju kontekstualne funkcije.

Prenatrpan programski pristup ne odnosi se isključivo na činjenicu da se logika ponašanja zadaje na osnovu metapodataka (polisa) – ovo se najčešće ne može izbeći i potrebno je kako bi se osigurala fleksibilnost – osnovni problem je način na koji se polise zadaju. Većina sistema ne poseduje brz i intuitivan način za zadavanje polisa, već se one moraju programirati u nekom od skript jezika. Čak i u slučaju da se obezbedi alat za programiranje, kao što je *Control4 Composer*, način programiranja i dalje veoma podseća na programski jezik, dok se za bilo kakve ozbiljnije namere mora koristiti logika razmišljanja koja je potpuno jednaka logici programiranja. Osnovno pribežište korisnika je u ovom slučaju pomoć tehničke podrške. *Control4* se zbog toga isključivo oslanja na svoju servisno-distributivnu mrežu da instalira i konfiguriše sistem u domaćinstvu korisnika – čak za ovo postoje i posebni treninzi za osobe koje obavljaju instalaciju, te se izdaju odgovarajući sertifikati.

Nedovoljno oslanjanje na senzorske ulaze je činjenica u sistemima za automatizaciju domaćinstva. Iako je osećaj kontrole korisnika važan, mogućnosti platforme u smislu prikupljanja senzorskih informacija su blago rečeno nedovoljno iskorišćene. Osim merenja temperature i sistemskog vremena, ostali senzori se retko ili nikako ne koriste. Iako se polise koje rade sa sensorima mogu programirati, nedostaje podrška koja bi olakšala programiranje, u vidu API-ja koji poseduje logiku zaključivanja ne nivou semantike bližem korisniku (ili bar nešto jasnijem od „prisustvo signala na digitalnom ulazu 3“, npr). Nedovoljno korišćenje senzora još uvek je prisutno u uređajima potrošačke elektronike, pa čak i u onima koji pretenduju na veći

Stanje u oblasti

nivo kontekstualnosti – sa maksimalnim dometom koji se može uporediti sa detekcijom prisustva korisnika u blizini uređaja (npr. Control4 ekrani osetljivi na dodir se uključuju kada korisnik zastane ispred njih).

Nedovoljno korišćenje povratne sprege od strane korisnika za fino podešavanje uslovljava posezanje za korisničkim servisom gotovo svaki put kada je potrebno izmeniti ponašanje sistema. Ono što nedostaje je praćenje ponašanja korisnika, odnosno načina na koji korisnik i sistem interaguju, te predlaganje izmena režima rada ili automatsko fino podešavanje parametara. Nedovoljno se koriste čak i mogućnosti koje su izuzetno jednostavne za realizaciju i ne zahtevaju složene senzorske sprege – kao što je npr. odgovor korisnika na upit koji se pojavljuje na ekranu osetljivom na dodir. Dokaz za nemar prema korisničkoj povratnoj sprezi predstavlja i API za sprezanje Control4 Lua polisa sa korisničkim spregama – polisa može da prikaže, npr. dijalog u okviru *Navigator* aplikacije, ali akcija korisnika u tom dijalogu ostaje misterija za polisu!

Samocentričnost aplikacija je globalni problem u potrošačkoj elektronici, a u prilog njegovom rešavanju mogu se koristiti upravo kontekstualne aplikacije. Samocentričnost se odnosi na nemogućnost aplikacija za suživot sa drugim aplikacijama. Čak i u slučaju kontekstualnih aplikacija, ne postoji način da one inteligentno dele kontekst sa nekim drugim aplikacijama, van svoje oblasti primene. Samocentričnost kontekstualnih aplikacija je, dakle, transcendentalan problem, koji zahteva platformu koja ne samo da omogućava kreiranje kontekstualnih funkcija na osnovu parametara okruženja ili ugrađenih algoritama, već i podelu znanja između svih entiteta koji dele platformu, odnosno koji su na raspolaganju korisnicima.

1.4.2 Kontekstualne aplikacije za mobilne uređaje

Kontekstualne aplikacije za mobilne uređaje su značajne sa stanovišta potrošačke elektronike, s obzirom na udeo mobilnih uređaja u potrošačkoj elektronici. Drugi faktor koji doprinosi značaju kontekstualnih aplikacija za mobilne uređaje je personalizovanost mobilnih uređaja, odnosno njihovo prirodno svojstvo pripadnosti tačno jednom korisniku, što je činjenica koja direktno ukazuje na korisnički kontekst i omogućava jednoznačnu realizaciju kontekstualnih aplikacija. Iako kontekstualne aplikacije za mobilne uređaje nisu brojne, postoji nekoliko primera. Postoje mobilne aplikacije za Android operativni sistem koje omogućavaju automatsko postavljanje

profila na osnovu trenutne pozicije telefona (Intelligent Profile, 2011), odnosno na osnovu prostorne lokacije korisnika (ProfileRules, 2011). Ovakve aplikacije su veoma jednostavne, i oslanjaju se na jednostavne senzorske ulaze sa nivoa mreže (akcelerometar, mrežni lokacijski servis) i pravila koja postavi korisnik, za određivanje profila i aktiviranja zadate korisničke akcije (promena postavki telefona kao što su melodija i jačina tona zvona, vibracija, uključivanje/isključivanje WiFi veze i sl). iPhone aplikacija Hollrback omogućava kontekstualno razmenjivanje biznis kartica korišćenjem QR kodova, i to tako da su kontakt detalji anotirani vremenom i mestom gde su prikupljeni, što olakšava kasniju pretragu kontakata po prilici u kojoj su ostvareni (Hollrback, 2011).

Opšti zaključak vezan za kontekstualne aplikacije za mobilne uređaje u potrošačkoj elektronici je da su u najvećem broju slučajeva zasnovane na ad-hoc rešenjima, uz samo delimičnu i slabo proširivu realizaciju funkcionalnosti kontekstualne platforme. Potreba za jedinstvenim prilazom problemu realizacije kontekstualnih aplikacija za mobilne uređaje, posredstvom kontekstualne platforme i odgovarajuće programske sprege veoma je izražena.

1.5 KONTEKSTUALNE KORISNIČKE SPREGE U POTROŠAČKOJ ELEKTRONICI

U okviru analize kontekstualnih korisničkih sprega u potrošačkoj elektronici, biće obuhvaćene dve grupe sprega čija je komercijalna upotreba intenzivirana, a koje po svojim osobinama spadaju u *transparentne* sprega (vidi 1.2): (1) sprega za prirodnu komunikaciju i (2) sprega za augmentiranu realnost.

Sprege za prirodnu komunikaciju zasnivaju se na korišćenju prirodnih atributa ljudske komunikacije, kao što su govor i gest, za upravljanje kontekstualnom platformom. Na nivou mreže, za prepoznavanje govora koriste se mikrofonski nizovi (Papp, 2007), dok se za prepoznavanje gestova korisnika koriste kamere, i to optičke kamere i metoda stereovizije, ili infracrvene kamere sa Time-Of-Flight pristupom (Hussman, 2008). Kompanija *Microsoft* među prvima je komercijalizovala spregu za prirodnu komunikaciju u vidu uređaja *Kinect*, kao dodatka za popularnu igračku konzolu *Xbox*. *Kinect* omogućava upravljanje video igrama korišćenjem telesnih gestova i glasovnih komandi, bez potrebe za korišćenjem posebnog kontrolera niti

Stanje u oblasti

optičkih ili aktivnih markera. *Kinect* kombinuje upotrebu RGB optičke kamere i infracrvenog 3D senzora za otkrivanje korisničkih gestova, kao i mikrofonski niz za prepoznavanje govora (Slika 17).



Slika 17. *Kinect* uređaj sa označenim senzorima (slajd sa Microsoft E3 konferencije)

Potencijal za rešenja nalik *Kinect*-u u svetu potrošačke elektronike, ogleda se i u činjenici da je ovaj uređaj upisan u Ginisovu knjigu rekorda kao najprodavaniji uređaj potrošačke elektronike ikada, sa prodajom od 8 miliona jedinica u prvih 60 dana od pojave u prodavnicama (Guinness World Records, 2011). Upotreba u kontekstualnim aplikacijama omogućena je zahvaljujući programskom SDK paketu koji se isporučuje uz *Kinect*, a koji je dostupan za *Windows 7* počev od juna 2011. godine (Orland, 2011). Iako izuzetno dobro prihvaćen od strane korisnika (osvojio je nagradu za najbolji uređaj za video igre po izboru roditelja – Parent's Choice Award) (Parent's Choice, 2011), poseduje i određene nedostatke. Naime, većina korisnika i dalje nije zadovoljna brzinom odziva ovog uređaja, odnosno, vreme potrebno za registrovanje gesta i reagovanje aplikacije na osnovu gesta je dovoljno dugo da se može percipirati, što donekle remeti lagodno korišćenje. Takođe, korisnici prijavljuju pojedine gestove koji se zahtevaju za izvršavanje određenih radnji u aplikacijama kao neprirodne (npr. postavljanje ruke pod uglom od 45 stepeni pri stajanju za pauziranje video igre). Konačno, jedan od važnih nedostataka uređaja je i zahtevani prostor za rad, odnosno, potrebno je obezbediti prostor veličine oko 2x3 m za udobno korišćenje. Iako ova disertacija nema za cilj da analizira ograničenja senzorskih tehnologija, izveštaji koje prijavljuju korisnici a vezani su za uređaje potrošačke elektronike koji se koriste u kontekstualnim aplikacijama su od izuzetnog značaja. Na primer, problem koji

ispoljava *Kinect* nastaje prevashodno iz nefleksibilnosti sistema, odnosno nemogućnosti finog podešavanja prema ciljnom okruženju i ciljnom korisniku, a taj problem se može ublažiti pažljivim dizajniranjem kontekstualne platforme i uvođenjem nivoa apstrakcije koji bi omogućili korišćenje više različitih kontekstualnih parametara u postupku odlučivanja.

Alternativa *Kinect*-u je *Playstation Move*, čija je osnovna razlika u odnosu na *Kinect* što koristi vizualne markere koji korisnici drže u rukama. Ipak, pokreti korisnika su prirodni i usklađeni sa potrebama aplikacije. Ovaj dodatak isporučuje se za *Playstation 3* igračku konzolu. Iako omogućava nešto veću brzinu odziva u odnosu na *Kinect*, potpuno ispunjavanje transparentnog koncepta nije moguće s obzirom da su korisnici svesni postojanja uređaja za kontrolu koji drže u rukama (Slika 18).



Slika 18. Playstation 3 igračka konzola i Playstation Move uređaj

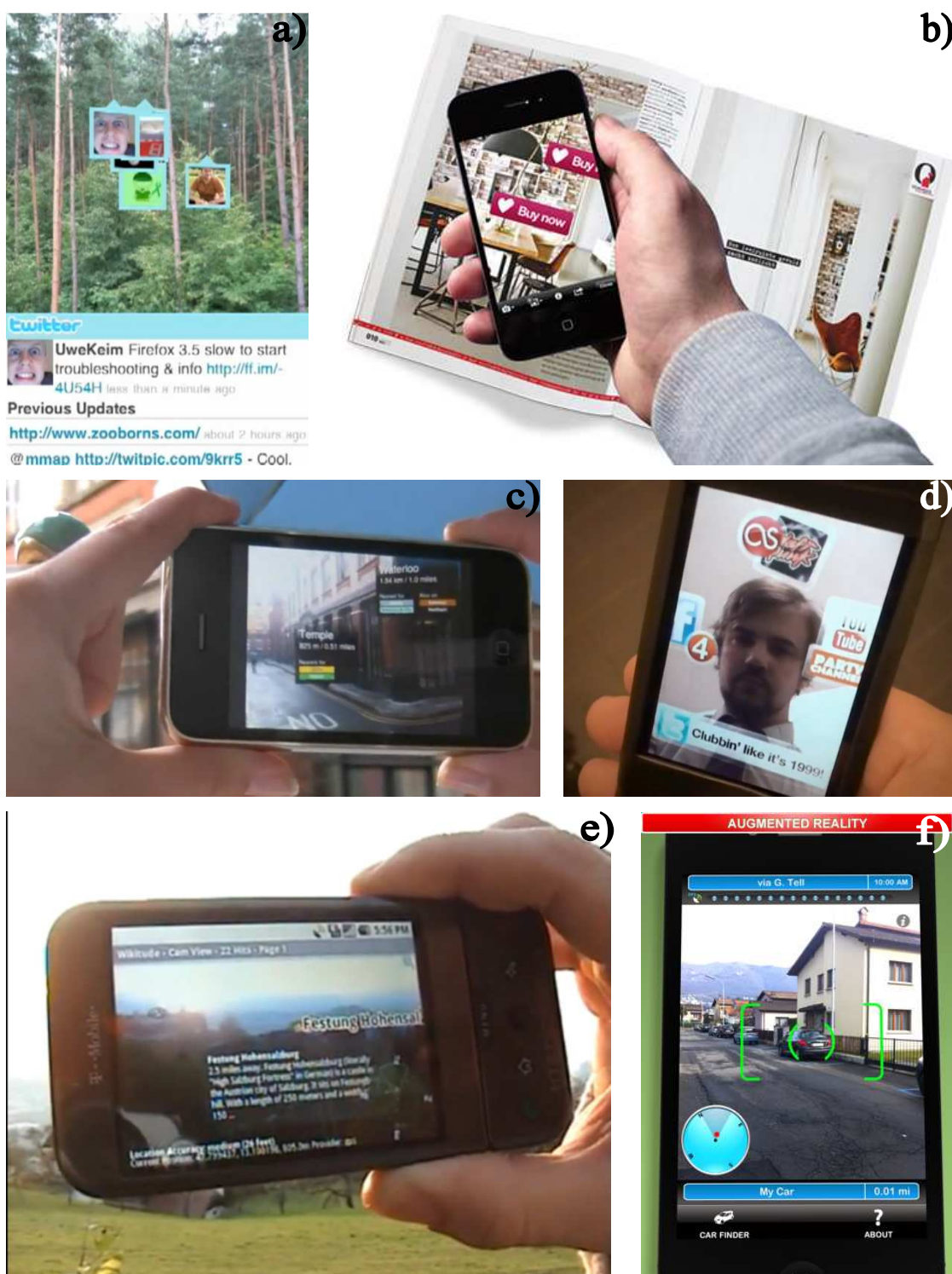
U odnosu na opšta istraživanja u oblasti transparentnih korisničkih sprega koje ilustruju koncept i prikazuju demonstracione slučajeve korišćenja prema kojima su sistemi unapred podešeni, u oblasti potrošačke elektronike naglasak se prebacuje na inženjersku realnost, u kojoj problemi nižeg prioriteta u istraživanju mogu znatno da naruše prihvaćenost proizvoda od strane potrošača. U slučaju transparentnih korisničkih sprega, poput *Kinect*-a i *Playstation Move*-a, osnovni problem je zahtev za kalibracijom uređaja, s obzirom na različitost veličine prostora u kome se koristi (*Kinect*) i na različitu osvetljenost prostorije (*Move*). Kalibraciju je, stoga, potrebno obavljati pre svakog pokretanja aplikacije. Problem kalibracije i rekalkibracije je izuzetno izražen, tako da kontekstualna platforma za uređaje potrošačke elektronike mora da omogući mehanizme za jednostavnu, ili čak automatsku kalibraciju.

Stanje u oblasti

Veliki broj korisničkih sprega za augmentiranu realnost namenjen je mobilnim uređajima, i to tako da se u okviru aplikacije prikazuje video koji snima ugrađena kamera, preko koga se postavlja grafička sprega za različite namene, koja „augmentira“ snimljenu scenu, odnosno dodaje joj dodatne kontekstualne funkcije. Tako, na primer, iPhone aplikacija *TwittARound* (Cashmore, 2009) prikazuje *tweet*-ove korisnika u okviru video snimka, pozicionirane tako da odgovaraju geografskoj lokaciji korisnika koji šalje poruku i njegovu udaljenost u odnosu na korisnika aplikacije. Aplikacija *Layar* za Android, omogućava kombinovanje fizičkog i digitalnog sveta na osnovu prepoznavanja slike, nakon čega omogućava određene akcije, poput kupovine, preuzimanja sa interneta, dobavljanja dodatnih informacija i sl. (Schroeder, 2009). Aplikacija za iPhone *NearestTube* omogućava korisniku da korišćenjem kamere pronade najbližu metro stanicu (Acrossair, 2012). Dalje, *TAT Augmented ID* je aplikacija koja u video snimak lica osobe ugrađuje informacije prikupljene sa socijalnih mreža vezane za tu osobu, na osnovu prepoznavanja lica (TAT, 2009). *Wikitude AR Travel Guide* obogaćuje video snimak neke turističke lokacije informacijama o toj lokaciji, pronađenih na *Wikipedia* sajtu (Wikitude, 2012). Ukoliko ne možete da pronađete automobil na velikom parkingu ispred supermarketa, aplikacija *CarFinder* omogućava da ga pronađete korišćenjem kamere (CarFinder, 2011). Ove aplikacije su ilustrovane na Slici 19.

Opšti trend koji je prisutan pri kreiranju kontekstualnih korisničkih sprega za potrošačku elektroniku, je izvesna doza senzacionalizma koja prati predstavljanje ovih proizvoda. Ovo uslovljava veoma veliku zainteresovanost potrošača pri samom objavljivanju određenog uređaja ili aplikacije, dok se zainteresovanost smanjuje intenzitetom srazmernom senzacionalizmu kojem je potrošač bio prvobitno izložen. Opšte uzev, ukoliko se uspehom uređaja potrošačke elektronike smatra njegova prihvaćenost nedugo nakon početka prodaje, ali i kontinuitet prodaje i razvoja, onda je jasna složenost realizacije korisničkih sprega za potrošačku elektroniku koje su u najvećem broju slučajeva presudne za prihvatanje uređaja. Nije za očekivati da ova činjenica zaobiđe kontekstualne korisničke sprege, s obzirom na senzacionalizam koji im po definiciji pripada jer uvode nove koncepte. S obzirom na to, dizajn kontekstualne korisničke sprege, odnosno kontekstualne platforme koja rukovodi

spregom, poseduje kritične zahteve koji se odnose na upotrebljivost (*usability*) i korisničko iskustvo (*user experience*).



Slika 19. Korisničke sprege i aplikacije za augmentiranu realnost: a) TwittARound, b) Layar, c) NearestTube, d) TAT Augmented ID, e) Wikitude AR Travel Guide i f) CarFinder.

1.6 PROBLEM ASINHRONIH ZAHTEVA

Osim opštih karakteristika kontekstualne platforme, koja su obrađena u prethodnim poglavljima, mišljenje autora ove disertacije je da postoji dodatni izazov pri realizaciji kontekstualne platforme namenjene uređajima potrošačke elektronike, koji će biti nazvan *problem asinhronih zahteva*. Ovaj problem se u većem delu odnosi na mobilne uređaje potrošačke elektronike, čiji je broj u neprekidnom porastu, ali i na uređaje sa neprekidnim napajanjem, poput TV prijemnika, zatim sistema za automatizaciju domaćinstva ali i svih ostalih uređaja potrošačke elektronike koji poseduju neki vid korisničke sprege. Priroda uređaja je takva da u određenim trenucima vremena emituju grafičko ili zvučno obaveštenje, odnosno zahtev, namenjene korisniku, nakon kojeg se očekuje akcija korisnika i njegova pažnja. Ova osobina uređaja potrošačke elektronike je sve izraženija, s obzirom na količinu informacija dostupnih svakom uređaju u svakom trenutku posredstvom globalne mreže. Zahvaljujući postojanju računara i njegovih derivata (laptop, tablet, pad računar i sl), kao i savremenih tehnologija povezivanja i razmene podataka, dostupnost informacija i povezanost među ljudima dostigli su veoma visok nivo (Agre, 2001). Danas je vreme pristupa znanju uslovljeno gotovo isključivo brzinom reakcije korisnika. U ovoj situaciji asinhroni zahtevi, koji su predmet ovog poglavlja, postaju osnovni alat za pridobijanje pažnje korisnika, u cilju održavanja visokog nivoa njegove produktivnosti u radnom okruženju (Appelbaum, 2008). Kao nusprodukt, asinhroni zahtevi se intenzivno koriste i u okruženjima koja nisu namenjena obavljanju posla, poput okruženja u dnevnoj sobi u kome se uglavnom obavljaju zabavne i multimedijalne aktivnosti. U situacijama u kojima uređaji ne omogućavaju dovoljno brzu interakciju među ljudima, ljudi samostalno usvajaju sličan stil komunikacije, odnosno prekidaju osobe od interesa svojim ličnim prisustvom i zahtevima za pažnjom. Ovu poslednju pojavu nazivamo *socijalnim asinhronim zahtevima*, koje takođe uvrstavamo u problem asinhronih zahteva, iako poreklo takvih zahteva ne dolazi od strane uređaja potrošačke elektronike, već od samih ljudi. Razlog za uvrštavanje socijalnih asinhronih zahteva u razmatranje jeste činjenica da se ovakvi zahtevi prepliću sa asinhronim zahtevima uređaja potrošačke elektronike i konkurišu za istovetan resurs – korisničku pažnju. S druge strane, kontekstualna platforma koja omogućuje medijaciju, odnosno upravljanje asinhronim zahtevima uređaja potrošačke

elektronike, može iskoristiti znanje za usmeravanje socijalnih asinhronih zahteva što je, po mišljenju autora, značajan doprinos.

U današnjem okruženju izuzetno brze razmene informacija, koje je veoma mentalno zahtevno, broj asinhronih zahteva je potrebno minimizovati u što većoj meri (Renaud, 2008), te je jedan od zahteva kontekstualne platforme da obezbedi mehanizme za tu svrhu. Ranija istraživanja obuhvataju različite tipove neželjenih asinhronih zahteva (Czerwinski, 2004), i oni se mogu grupisati u sledeći nepotpun spisak:

- komunikacioni zahtevi (dolazni poziv, nova poruka, obaveštenje o promeni statusa osobe i sl);
- statusno-tehničke notifikacije (stanje baterije, završetak obrade, prekid ili uspostava rada servisa i sl);
- upitni zahtevi (isticanje sesije, unos korisničkog imena i lozinke, asinhroni anketni unos i sl);
- socijalni asinhroni zahtevi (ulazak kolege u kancelariju sa pitanjem, ometanje od strane ukućana pri gledanju filma, kontinuirana posredna interakcija sa kolegom u toku rada i sl) (Cube Smart, 2002).

Osnovni problem asinhronih zahteva jeste što izazivaju zaustavljanje aktivnosti koja je bila u toku u trenutku kada je zahtev uspostavljen. Vreme povratka na prethodno obavljanu aktivnost nakon asinhronog zahteva je značajno i utiče na konačnu produktivnost (Altmann, 2007). Dodatno, ponovljeni asinhroni zahtevi značajno utiču na povećanje stresa (Butterfield, 1964), kao i na povećanu kognitivnu aktivnost osobe kojoj se zahtevi upućuju (Kirsh, 2000; Bailey, 2008). Povećanje stresa usled socijalnih asinhronih zahteva takode je potvrđeno u ranijim istraživanjima (Carton, 2009). S obzirom na to da je zadovoljstvo korisnika presudno u prihvatanju uređaja potrošačke elektronike, jedan od osnovnih ciljeva kontekstualne platforme je da doprinese smanjenju stresa i povećanju produktivnosti korisnika, koji su ugroženi asinhronim zahtevima. Da bi se broj neželjenih asinhronih zahteva smanjio, potrebno je omogućiti mehanizme za otkrivanje neželjenih asinhronih zahteva i upravljanje njima na osnovu procene trenutne korisničke aktivnosti, kao i mehanizme za dostavljanje rezultata rada zainteresovanim stranama (drugim aplikacijama, uređajima i osobama), kao što je npr. nivo zauzetosti osobe kojoj bi zahtev mogao biti upućen. U

Stanje u oblasti

nastavku je dat pregled stanja u oblasti otkrivanja neželjenih asinhronih zahteva i upravljanja njima. Zatim su izloženi problemi pri objavljivanju stanja korisnika u cilju smanjenja broja asinhronih zahteva, kao i smer u kojem se traži rešenje u okviru ove disertacije.

1.6.1 Otkrivanje i upravljanje asinhronim zahtevima

U okviru oblasti kontekstualnog računarstva i interakcije čovek-računar (*Human-Computer Interaction – HCI*) istraživanja u vezi sa asinhronim zahtevima započeta su nakon rada *McFarlane*-a, koji je predložio i razmotrio četiri tehnike za koordinisanje asinhronih zahteva: (1) neposredno; (2) pregovaranjem; (3) upravljanjem i (4) zakazivanjem (*McFarlane*, 2002). Veliki broj teoretskih i eksperimentalnih istraživanja, uglavnom zasnovanih na anketama i samoproceni učesnika u eksperimentima navodi negativne efekte asinhronih zahteva. Na primer, *Bailey* je sproveo eksperiment na grupi korisnika koja je obavljala osnovne poslove na računaru, poput sabiranja, brojanja, čitanja i sl. Na osnovu samoprocene korisnika zaključeno je značajno povećanje vremena potrebnog za obavljanje posla, povećanje uznemirenosti, nervoze i percepcije težine posla (*Bailey*, 2001). U drugom istraživanju postavljen je cilj da se proceni performansa korisnika u toku igranja računarske igre *Tower of London*, ukoliko se igrači ometaju zahtevima za samoprocenu u iskačućim (*popup*) dijalozima koji su povremeno delom zaklanjali ekran sa igrom. Otkriveno je da postoji značajno povećanje vremena potrebnog za rešavanje problema u igri kod grupe tako ometanih igrača, u odnosu na kontrolnu grupu (*Hodgetts*, 2007). Nasuprot ovome, pojedina istraživanja navode mogućnost povećanja brzine rada i efikasnosti ukoliko je korisnik izložen asinhronim zahtevima, usled optimizacije načina rada koji je time izazvan. Međutim, i u ovom slučaju zabeleženo je veoma izraženo povećanje mentalne opterećenosti, frustracije i stresa (*Mark*, 2008). Na radnom mestu, u okruženjima u kojima je veoma izražena potreba za produktivnošću i postizanjem cilja, negativni efekti asinhronih notifikacija su takođe prisutni (*Harman*, 2006). Čak i među rukovodiocima koji su izloženi asinhronim zahtevima dolazi do povećanja hazarderskog ponašanja – njihov fokus na izvodljivost neke odluke opada i često se u odlučivanju opredeljuju za opcije visokog rizika i visokog profita, odnosno za opcije visokog kvaliteta i visoke cene (*Liu*, 2008). Kao što se može videti, efekti asinhronih

zahteva nisu bezazleni i veoma je važno ustanoviti mehanizme za upravljanje tim zahtevima u cilju umanjavanja njihovih negativnih efekata (Gievska, 2005).

Većina istraživanja koja se bavi upravljanjem asinhronim zahtevima u obzir uzima metodu posredovanja pri uspostavljanju zahteva, kao i vremensko raspoređivanje zahteva. Na primer, zahtevi koji su uspostavljeni neposredno nakon ili neposredno ispred prekida u nekoj aktivnosti ili radu pomažu smanjenju negativnih efekata tih zahteva (Adamczyk, 2004). Mehanizam za otkrivanje ovih prekida u radu kao idealnih trenutaka za asinhronu zahtevu predložili su Iqbal i Bailey, koji su u svom radu pokazali da takav pristup smanjuje frustracije korisnika (Iqbal, 2007). Još jedan način za otkrivanje stanja visoke opterećenosti poslom da bi se izbegli neželjeni asinhroni zahtevi je putem analize fizioloških indikatora, kao što je veličina zenice oka (Adamczyk, 2005). Posredovanje pri uspostavljanju zahteva ispitano je na primerima posredovanja pre obavljanja telefonskog poziva, tako što su pozivani korisnici obavljali samoprocenu negativnih efekata prethodnih poziva te se na osnovu toga generisala odgovarajuća poruka za buduće pozive (Avrahami, 2007; De Guzman, 2007). Merenje nivoa zauzetosti korisnika korišćenjem fiksnih senzora u kancelarijskom okruženju obavljeno je u istraživanjima koja su sproveli Hudson i Fogarty. Njihov pristup je značajan za ovu disertaciju, s obzirom da je najbliži opštenamenskoj realizaciji koja je donekle podudarna sa arhitekturom kontekstualne platforme. Korišćenjem kontaktnih senzora, praćenja stanja aplikacija na PC računaru i mikrofona, te korišćenjem prediktivnih algoritama poput stabala odlučivanja, Bajesovih mreža, Vektora podrške i *Ada Boost*-a, postignuta je prosečna preciznost procene zauzetosti korisnika na nivou od 75% (Fogarty, 2005). U pojedinim istraživanjima predstavljena su i rešenja detekcije korisničkog konteksta u cilju procene zauzetosti korišćenjem senzora podesnih za nošenje (Bernstein, 2007; Ho, 2005).

Iako prethodna istraživanja u oblasti otkrivanja i upravljanja asinhronim zahtevima omogućuju osnovu za realizaciju sličnih funkcionalnosti u okviru kontekstualnih platformi, nekoliko važnih rešenja još uvek nedostaje. Na primer, najčešće se razmatraju poslovna okruženja poput radne kancelarije, dok se izuzetno malo pažnje posvećuje okruženju domaćinstva. S obzirom na cilj ove disertacije da primeni mehanizme na uređaje potrošačke elektronike, očigledno je potreban napredak

Stanje u oblasti

u tom pravcu. Takođe, potrebna su dodatna unapređenja robustnosti i tačnosti algoritama za procenu zauzetosti, u slučajevima u kojima je potrebno izraziti zauzetost na skali sa većim brojem diskretnih nivoa.

1.6.2 Objavljivanje procenjenog stanja zauzetosti korisnika

Upravljanje asinhronim zahtevima u okviru korisničke sprege je od izuzetnog značaja, što posredno uzrokuje značaj objavljivanja procenjenog stanja korisnika u smeru ka aplikacijama koje realizuju korisničku spregu, kao i ka drugim korisnicima kao potencijalnim inicijatorima socijalnih asinhronih zahteva. Količina asinhronih zahteva koje korisnička sprega generiše, svrstana je kao jedna od tri važne karakteristike bilo koje korisničke sprege, zajedno sa razumljivošću i odzivom, što su druge dve karakteristike (McCrickard, 2003). Iz tog razloga koriste se tehnike posredovanja u asinhronom zahtevu pri realizaciji različitih korisničkih sprega, od kojih su mnoge veoma inovativne. Na primer, predložena je korisnička sprega za uspostavljanje asinhronih zahteva koji angažuju samo periferni vid, odnosno zahteva koji se prenose taktilnim putem (Hameed, 2009). Korisničke sprege koje saopštavaju informacije na način prilagođen stanju korisnika nazivaju se *Atentivne Korisničke Sprege (Attentive User Interfaces)* (Vertegaal, 2003).

U radnom okruženju korisnici čak i nesvesno obavljaju upravljanje socijalnim asinhronim zahtevima, tako što koriste programe za razmenu tekstualnih poruka da postave kratko pitanje korisniku („Da li si slobodan?“), pre ličnog kontakta (Garrett, 2008). U ovu svrhu takođe služe tekstualni ili grafički indikatori u istim programima, koji se nazivaju *oznake stanja (busy flags)*. Iako ove oznake jasno svedoče o stanju korisnika, sagovornici se sve češće na njih ne obaziru, što je potvrđeno u nekoliko istraživanja (Fogarty, 2004; Begole, 2004). Razlog tome je najverovatnije iskustvo koje korisnici imaju vezano za ažurnost oznaka stanja, koje se postavljaju ručno npr. pre početka sastanka ili napuštanja radnog mesta pa se kasnije usled zaboravnosti ili nemara ne vrata na prethodnu vrednost (Milewski, 2000).

S obzirom na to da je nivo korisničke sprege deo arhitekture kontekstualne platforme, objavljivanje informacija o zauzetosti korisnika korišćenjem nekog oblika korisničke sprege spada u zahteve realizacije. Iz tog razloga su u okviru ove disertacije istražene nove mogućnosti za prikaz zauzetosti korisnika korišćenjem svetlosnih efekata, što se smatra posebno pogodnim za upravljanje socijalnim asinhronim

zahtevima. Ovaj smer realizacije motivisan je novijim istraživanjima u oblasti kognitivnih nauka, gde se pokazalo da različite boje imaju različit uticaj na ljude, uključujući vreme reakcije korisnika na događaje, odnosno njihovu brzinu zaključivanja. U okviru studije koju su sprovedi *Elliott* i dr. istraživani su efekti različitih osnovnih boja, poput crvene, zelene ili sive na performanse ljudi u obavljanju zadataka i na njihovo opšte raspoloženje. Osnovni zaključak koji je izveden je da crveni tonovi boje motivišu ljude da izbegavaju problem ili akciju i utiču negativno na način na koji rade, na kvalitet rezultata koje postižu i na vreme reakcije (*Elliot*, 2007). Sa aspekta korišćenja svetlosnih izvora, ustanovljena je korelacija između raspoloženja ljudi i različitih postavki osvetljenja u prostoriji (*Knez*, 1997). Dodatno, svetlosni izvor postavljen kao deo ambijentalnog osvetljenja, može poslužiti u cilju privlačenja pažnje korisnika na važna obaveštenja (*Vastenburger*, 2009). Ove studije predstavljaju teoretsku osnovu za razvoj korisničke sprege koja koristi svetla u boji za objavljivanje informacije o zauzetosti korisnika, a što je tehnički izvodljivo u okviru kontekstualne platforme u potrošačkoj elektronici, koja je opremljena rukovaocima za kontrolu svetla, poput DMX, na nivou mreže. Na primer, ukoliko se svetlosni izvor usmeri ka pogodnim površinama u zatvorenom prostoru, poput plafona ili uglova u predsoblju ili hodniku ispred prostorije u kojoj boravi osoba čija se zauzetost prati, moguće je osvetliti te površine bojom koja odgovara trenutnom stepenu zauzetosti osobe. Na primer, ukoliko je osoba unutar prostorije zauzeta, svetlosni tonovi biće postavljeni na crvenu boju, što će služiti kao indikacija osobama koje bi mogle da upute neželjeni socijalni asinhroni zahtev da takav zahtev u tom trenutku nije poželjan. Narandžasti i žuti tonovi označavaju umerenu dostupnost (na primer, samo za neko kratko pitanje), dok plavi i zeleni tonovi simbolizuju visoku dostupnost, i označavaju pravi trenutak za uspostavljanje asinhronih zahteva. Zahvaljujući mogućnostima kontekstualne platforme, boja svetla bi se menjala automatski, bez manualne akcije posmatrane osobe.

1.6.3 Realizovana okruženja i primene u potrošačkoj elektronici

U dosadašnjim istraživanjima realizovan je veći broj različitih okruženja za procenu zauzetosti korisnika, te samim tim i za upravljanje asinhronim zahtevima, potpuno ili delimično. U okviru rada na disertaciji istražene su mogućnosti sledećih okruženja i aplikacija: *MyVine* (*Fogarty*, 2004), *Lilsys* (*Begole*, 2004), *AuraOrb*

Stanje u oblasti

(Altosaar, 2006), PETDL (Bailey, 2006), Subtle (Fogarty, 2007), BusyBody (Horvitz, 2004), i Life Log Collector (Abe, 2009). Ipak, nijedno od ovih okruženja i prototipova nije blizu primene na tržištu, u uređajima potrošačke elektronike. Iako je komercijalna primena gore navedenih rešenja nešto teža u obliku u kojem su realizovana, potencijal je znatno veći usled integracije sličnih koncepata u opštenamensku kontekstualnu platformu, čija je primena veoma jednostavna i izvodljiva u okviru, npr. sistema za automatizaciju domaćinstva, što je oblast koja beleži izuzetan rast (Abi Research, 2012). Stoga je jedan od ciljeva predstavljenih u ovom istraživanju da omogući mehanizme integracije kontekstualne platforme u druge sisteme, što bi omogućilo prodor novih kontekstualnih funkcionalnosti, poput upravljanja asinhronim zahtevima, u svet potrošačke elektronike.

1.7 DOPRINOSI DISERTACIJE U ODNOSU NA OBLAST

U ovom poglavlju biće dat pregled problema uočenih na osnovu analize stanja u oblasti, te smer u kome će biti traženo rešenje u okviru disertacije za svaki od navedenih problema. Spisak je naveden po važnosti.

Problem 1. *Neprilagođenost postojećih rešenja zahtevima potrošačke elektronike.* Da bi se kontekstualna platforma integrisala u uređaje potrošačke elektronike, potrebno je zadovoljiti veći broj pretpostavki (vidi 1.3). Kontekstualna platforma realizovana u okviru ove disertacije uzima u obzir većinu navedenih pretpostavki u okviru arhitekture programske podrške i alata korišćenih za razvoj. Takođe, u okviru disertacije naveden je skup heuristika i zahtevi na osnovu kojih je moguće obaviti ranu procenu integrabilnosti kontekstualne platforme unutar nekog uređaja potrošačke elektronike, uz date odgovarajuće primere i eksperimente.

Problem 2. *Nepostojanje mehanizma integracije kontekstualnih platformi sa postojećim sistemima u njenoj okolini.* S obzirom da je mogućnost povezivanja sa ostalim, raznorodnim uređajima od presudnog značaja za današnje moderne uređaje, kontekstualna platforma realizovana u okviru disertacije poseduje mehanizme veze sa okolinom korišćenjem jednostavne HTTP programske sprege. Posredstvom te sprege omogućeno je objavljivanje rezultata rada kontekstualne platforme svim zainteresovanim entitetima s jedne strane, kao i pristup različitim senzorskim događajima generisanih na drugim uređajima, s druge.

Problem 3. *Nedovoljna skalabilnost i primenljivost postojećih rešenja i suviše veliki uticaj ciljnih aplikacija na dizajn.* Arhitektura kontekstualne aplikacije treba da bude jednostavna i opštenamenska, kako bi se platforma brzo i jednostavno mogla integrisati u ciljno okruženje ili uređaj. Arhitektura srednjeg sloja zasnovanog na metapodacima, poboljšani algoritmi zaključivanja zasnovani na stablima odlučivanja, kao i mogućnost korišćenja brojnih protokola veze na mrežnom nivou i HTTP programske sprege za opštu interakciju sa sistemima i uređajima u okruženju predstavljaju polaznu tačku za realizaciju skalabilne i proširive kontekstualne platforme jednostavne za korišćenje i održavanje.

Problem 4. *Automatsko prilagođavanje kontekstu treba da bude poboljšano, sa ciljem da se umani potreba za intervencijom korisnika.* Posebna pažnja mora biti posvećena interakciji sa korisnikom – i ona treba da bude svedena na minimalnu meru, onoliko koliko je potrebno da korisnik zadrži osećaj kontrole, ali da izbegne potrebu za stalnim konfigurisanjem sistema. U okviru disertacije predložen je mehanizam prilagođavanja rada platforme koji je primenljiv na različite klase problema.

Problem 5. *Revolucionarizam i senzacionalizam su često izraženi, dok su aspekti korisničke psihologije nedovoljno razmatrani sa stanovišta prihvatanja rešenja.* Osnovni razlog neprihvatanja uređaja od strane korisnika leži u komplikovanom korišćenju i nedopadljivom izgledu. Stoga, novi dizajn mora da uključi dovoljan stepen postojećeg znanja korisnika, kako bi korišćenje bilo jednostavno, a opet, mora da poseduje dovoljno inovativnosti u cilju omogućavanja diferencijacije u odnosu na konkurenciju. Prilaz realizaciji kontekstualne platforme u okviru ove disertacije, način izbora rešenja i odabrana rešenja, kao i skup različitih heuristika doprinose razumevanju metodologije razvoja koja ima veće šanse za bolji plasman uređaja na tržište.

Problem 6. *Dosadašnje kontekstualne platforme ne poseduju ugrađene mehanizme upravljanja asinhronim zahtevima.* Kontekstualna platforma realizovana u okviru disertacije je, po mišljenju autora, prva koja omogućuje ugrađene mehanizme upravljanja asinhronim zahtevima, poput određivanja stepena zauzetosti korisnika i preporučivanja režima rada zainteresovanih aplikacija.

Problem 7. *Dosadašnja istraživanja ne uspevaju da obezbede mehanizme za adekvatno upravljanje socijalnim asinhronim zahtevima.* U ovom pogledu

Stanje u oblasti

prevazilaženje *busy flag* problema je od velike važnosti za ispravno upravljanje asinhronim zahtevima (vidi 1.6.2). U okviru disertacije predstavljeni su primeri korišćenja predložene kontekstualne platforme u aplikacijama za objavljivanje stanja zauzetosti korisnika korišćenjem svetlosnih efekata, što je inovativno rešenje koje može poslužiti za poboljšanje upravljanja socijalnim asinhronim zahtevima, ili poslužiti kao osnova za dalja istraživanja u tom pravcu.

Problem 8. *Težište primene kontekstualnih platformi je suviše na poslovnom okruženju, dok su okruženja poput domaćinstva nedovoljno istražena.* Današnji skup uređaja potrošačke elektronike naklonjeniji je primeni za multimediju i zabavu, odnosno ciljna grupa korisnika van poslovnog okruženja je veća. Najbolji primer za ovo je skorašnji neuspeh kompanije RIM (*Research in Motion*) i njihovog proizvoda *Blackberry*, zbog njegove prevelike orjentisanosti ka upotrebi u poslovnom okruženju (Munarriz, 2012). U istraživanjima u oblasti kontekstualnih platformi je takođe favorizovano poslovno okruženje. Jedan od ciljeva disertacije je da fokus pri realizaciji i evaluaciji kontekstualne platforme usmeri ka okruženjima kao što je domaćinstvo.

Problem 9. *Kompleksnost arhitekture programske podrške.* Programska podrška predloženih rešenja je često prekompleksna, i oslanja se na virtualizovana okruženja zasnovana na Java programskom jeziku, poput OSGI. Iako ovaj pristup nije pogrešan, utisak autora disertacije je da nije u potpunosti podesan za primenu u uređajima potrošačke elektronike. Fokus disertacije jeste upravo na jednostavnim i efikasnim realizacijama, koje su skalabilne i proširive, te primenljive na veći broj različitih slučajeva upotrebe.

Problem 10. *Nepostojanje sveobuhvatnog rešenja na mrežnom nivou koje obuhvata veći broj senzorskih i aktuatorskih protokola u jednoj platformi.* Većina istraživanja u oblasti kontekstualnih platformi ima specifičan fokus, koji se odnosi na opšte algoritme, arhitekturu, simulacije ili pojedine tehnike otkrivanja konteksta. Do sada nije predstavljena sveobuhvatna realizacija mrežnog nivoa, koja bi omogućila korišćenje većeg broja različitih logičkih i fizičkih senzora. Iako nije cilj disertacije da poboljša algoritme na mrežnom nivou, biće predstavljena arhitektura i primeri primene tehnologija čiji će cilj biti da budu sveobuhvatni za mnogobrojne primene u kontekstualnim aplikacijama.

POGLAVLJE 2.

ALGORITMI ZA OTKRIVANJE

KONTEKSTA

Osnovno razmatranje u okviru metodologije realizacije kontekstualne platforme odnosi se na algoritme za otkrivanje konteksta, odnosno identifikaciju stanja konteksta, što se naziva *semantizacijom*. S obzirom na cilj disertacije da se realizuje kontekstualna platforma koja može da se integriše u veliki broj uređaja potrošačke elektronike uz što manje izmene, naglasak pri izboru algoritama mora da bude na brzini i jednostavnosti realizacije. Iz ovog razloga osnovni polazni algoritam zasnivaće se na stabilima odlučivanja i sistemu pravila. Ipak, predložene su brojne nove varijacije na osnovni algoritam, kao i osnovna evaluacija.

U ovom poglavlju najpre će biti izloženi detalji algoritma za semantizaciju konteksta, kao što je otkrivanje korisničkih aktivnosti. Dalje, biće dat algoritam za procenu zauzetosti korisnika, u svetlu omogućavanja kontekstualnoj platformi da upravlja asinhronim zahtevima.

2.1 OSNOVE ALGORITMA ZA SEMANTIZACIJU KONTEKSTA

Pod procesom *semantizacije* konteksta podrazumeva se identifikacija kognitivno prepoznatljivih stanja na osnovu brojnih kognitivno neprepoznatljivih kontekstualnih ulaza. Kao izlaz iz postupka semantizacije naslovljava se trenutno *stanje okruženja* i/ili

Algoritmi za otkrivanje konteksta

korisnika, na osnovu kog se direktno mogu izvesti određeni zaključci. Na primer, stanje „Noć“, stanje „Sastanak“ ili stanje „Ulica“ odgovaraju semantizovanim naslovima zaključenim na osnovu kontekstualnih ulaza. U zavisnosti od toga da li je semantizovane naslove potrebno predstaviti u čitljivom obliku u okviru neke grafičke korisničke sprege ili ne, oni mogu i ne moraju biti izraženi na malopre navedeni način. Podjednako je validna enumeracija stanja prirodnim brojevima. Ono što je važno je da kontekstualna platforma obezbedi API ka entitetima zainteresovanim za izlaz postupka semantizacije, u kojem će biti jasna upotreba identifikatora u odnosu na ciljnu radnju. Tako identifikator može da se označava i nivojski (npr. Likert 1-10) i da na taj način određuje stepen slaganja okruženja sa nekom očekivanom pretpostavkom.

U nastavku biće razmotrena dva osnovna načina semantizacije konteksta, i to: *semantizacija zasnovana na polisama* i *automatska semantizacija*.

2.1.1 Semantizacija konteksta zasnovana na polisama

Neka je S_j stanje okruženja, gde je j redni broj stanja u tabeli svih mogućih stanja. Neka je E vektor svih kontekstualnih parametara dostupnih kontekstualnoj platformi. Neka je E_i vektor kontekstualnih parametara e prikupljenih od strane izvora kontekstualnih informacija i .

$$E_i = [e_{1_i} \quad e_{2_i} \quad \dots \quad e_{n_i}] \quad (1)$$

Tada se verovatnoća da je okruženje u stanju S_j može izraziti kao:

$$p(S_j | E) = \sum_i p(S_j | E_i) p(E_i) \quad (2)$$

Kontekstualna platforma mora da obezbedi uslov kompletnosti prostora kontekstualnih informacija:

$$\sum_{i=1}^n p(E_i) = 1 \quad (3)$$

U prethodnoj jednačini n označava ukupan broj izvora kontekstualnih informacija. Uz pretpostavku da izvori kontekstualnih informacija imaju podjednak uticaj na konačnu verovatnoću da je okruženje u stanju S_j , dobijamo uprošćen izraz:

$$p(S_j | E) = \frac{1}{n} \sum_{i=1}^n p(S_j | E_i) \quad (4)$$

Uređenu $n+m$ -torku $\{\{S_1, S_2, \dots, S_n\}, \{E_1, E_2, \dots, E_m\}\}, n, m \in \mathbb{N}$ treba posmatrati kao skup podataka o kontekstu koji su na raspolaganju algoritmu za semantizaciju, koji odražavaju trenutno stanje okruženja u vremenu.

Neka je g_i prekidačka funkcija (pravilo) definisano na sledeći način:

$$g_i(E_i) = \begin{cases} 0 \\ 1 \end{cases} \quad (5)$$

Osnovu algoritma za semantizaciju predstavlja skup pravila, koja definišu preslikavanje između prostora kontekstualnih parametara, u prostor verovatnoća da je okruženje u određenom stanju:

$$g_i(E_i) \xrightarrow{f_i} p(S_j | E_i) \quad (6)$$

Funkcije preslikavanja f_i nazivaćemo *prostim pravilima semantizacije*. Neka je t određeni trenutak vremena u kome se posmatra okruženje. Tada možemo usvojiti:

$$E_i = E_i(t) \quad (7)$$

S obzirom na to da stanje okruženja zavisi kako od trenutnih kontekstualnih parametara, tako i od kontekstualnih parametara prikupljenih iz prošlosti, definiše se vektor kontekstualnih parametara iz prošlosti za izvor i kao konkatencija vektora kontekstualnih parametara za izvor i dobavljenih u diskretnim trenucima u prošlosti:

$$\overline{E}_i = E_i(t - \Delta t_1) \| E_i(t - \Delta t_2) \| \dots \| E_i(t - \Delta t_n) \quad (8)$$

$$\overline{E}_i^T = \begin{bmatrix} e_{i_1}(t - \Delta t_1) \\ \dots \\ e_{i_p}(t - \Delta t_1) \\ e_{i_1}(t - \Delta t_2) \\ \dots \\ e_{i_q}(t - \Delta t_2) \\ \dots \\ e_{i_1}(t - \Delta t_n) \\ \dots \\ e_{i_r}(t - \Delta t_n) \end{bmatrix} \quad (9)$$

Neka je g_i' binarna prekidačka funkcija definisana na sledeći način:

$$g_i'(E \| \overline{E}) = \begin{cases} 0 \\ 1 \end{cases} \quad (10)$$

Algoritmi za otkrivanje konteksta

Sada je moguće definisati složeno pravilo semantizacije f' definisano kao:

$$g'_i(E||\bar{E}) \xrightarrow{f'_i} p(S_j | E||\bar{E}) \quad (11)$$

Vektor preslikavanja Ω sastavljen je od složenih pravila semantizacije f' :

$$\Omega = [f'_1 \quad f'_2 \quad \dots \quad f'_n] \quad (12)$$

Ovaj vektor preslikavanja definiše *kontekstualnu polis* na osnovu koje se obavlja semantizacija konteksta. U praksi, kontekstualne polise određuju se eksperimentalnim putem, i definišu za svaki slučaj korišćenja ponaosob.

Najčešće, kontekstualna polisa se vezuje za jedno inteligentno okruženje. Određivanje kontekstualne polise obavlja se posmatranjem ciljnog okruženja od strane obučениh lica ili korišćenjem izveštaja samoprocene popunjenih od strane učesnika u sceni, kombinovano sa video snimkom i log datotekom koja sakuplja sve kontekstualne parametre dostupne u diskretnim trenucima vremena. Tada se pristupa kreiranju kontekstualne polise na sledeći način: (1) definiše se skup stanja na osnovu izveštaja obučениh posmatrača i samoprocene; (2) odrede se vremenski markeri trajanja stanja uvidom u video snimak okruženja; (3) uvidom u log datoteku sa kontekstualnim parametrima uočavaju se pravilnosti; (4) generišu se složena pravila semantizacije za svaku aktivnost; (5) ponavlja se eksperiment sa različitim učesnicima i postupak se ponavlja u cilju poboljšanja kvaliteta kontekstualne polise. Određivanje kvalitetnih kontekstualnih polisa na gore predloženi način je izuzetno složen zadatak, i potrebno ga je poveriti kompanijama posebno obučениm za izvođenje studija ljudskog faktora. Ovim se otvara prostor za usluge namenskog razvoja kontekstualnih polisa za primene u uređajima potrošačke elektronike. Prednosti ovog pristupa su u sledećem: (1) potreba za fazom učenja i konfigurisanjem sistema je minimalna ili ne postoji; (2) realizacija novih kontekstualnih funkcija za nove uređaje je veoma jednostavna. Mane ovog pristupa su: (1) složen (i skup) postupak kreiranja kontekstualnih polisa; (2) mogući problemi sa tačnošću u slučaju slabog poklapanja testnog i ciljnog inteligentnog okruženja.

2.1.2 Automatska semantizacija konteksta

Alternativa semantizaciji konteksta zasnovanoj na polisama je da se obezbedi automatsko podešavanje pravila semantizacije i da se izbegne potreba za korišćenjem kontekstualnih polisa. Da bi se ovo postiglo, primeniće se algoritam naveden u nastavku, koji kao osnovu za rad koristi prosta pravila semantizacije (Jednačina 6).

Prosta pravila semantizacije f_i preslikavaju vektor kontekstualnih parametara E_i izvora i na verovatnoće da je okruženje u stanju S_j . Da bi se ispravno odredile verovatnoće $p(S_j | E)$ potrebno je obezbediti mogućnost treniranja sistema od strane korisnika, odnosno nadzirano treniranje od strane posmatrača. U vremenskom trenutku t_r , korisnik, odnosno posmatrač, deklariše trenutno stanje j . Deklaracija trenutnog stanja je asinhrona, odnosno ne usleđuje kao odgovor na upit, niti je unapred raspoređena, već je isključivo ustanovljena po nahodanju korisnika, odnosno posmatrača. Kvalitet predloženog algoritma ogleda se u činjenici da je deklaracija stanja potpuno binaran unos, koji može ili ne mora da nosi oznaku stanja. Postupak treniranja može biti privremen ili kontinualan, s obzirom na to da ne utiče na rad kontekstualne platforme. UML dijagram aktivnosti za algoritam koji opisuje osnovni tok postupka treniranja dat je na Slici 20. Nakon što korisnik ili posmatrač deklarišu trenutno stanje, potrebno je ažurirati verovatnoće vezane za to stanje u svim pravilima semantizacije koja zadovoljavaju prekidačku funkciju g_i za kontekstualne parametre E_i izvora i . To se obavlja tako što se najpre proveriti, da li je okruženje u trenutku deklaracije u nekom od poznatih stanja S_j , odnosno da li je neka od verovatnoća $p(S_j|E)$ na visokom nivou :

$$p(S_j | E) > P_{thr} \quad (13)$$

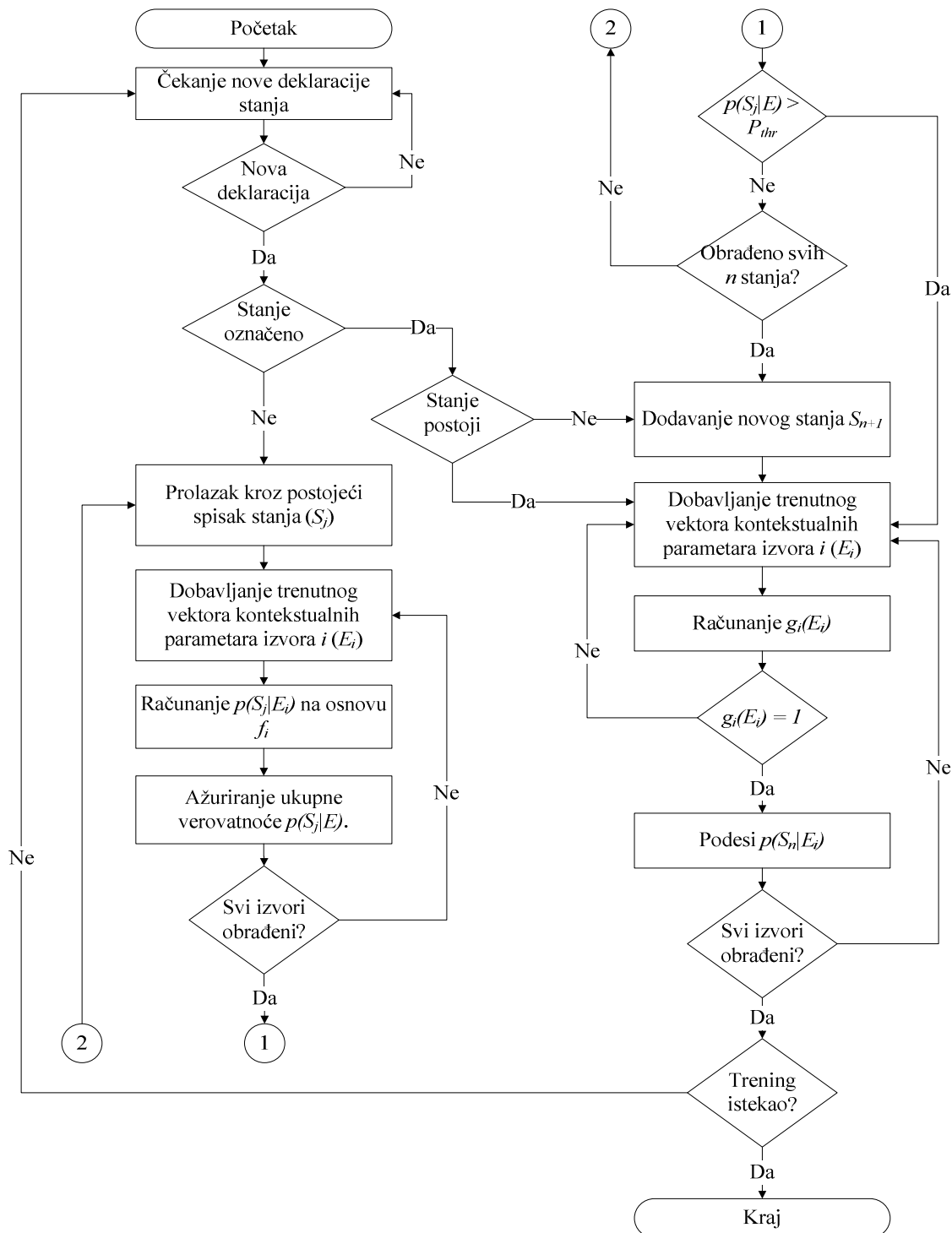
gde se P_{thr} odabira tako da vrednosti kontekstualnih parametara budu u $1-\alpha$ intervalu poverenja, odnosno:

$$P_{thr}(|E| < t) = 1 - \alpha \quad (14)$$

Zavisno od ciljnog okruženja, vrednost α se empirijski određuje, i za slučaj detekcije korisničkih aktivnosti margina tačnosti može biti prilično široka, npr. $\alpha=[0.05, 0.3]$.

Algoritmi za otkrivanje konteksta

Ukoliko je neko od već postojećih stanja dovoljno verovatno, obavlja se podešavanje pravila semantizacije f_i tako da uključe najnoviju deklaraciju.



Slika 20. UML dijagram aktivnosti algoritma za automatsko podešavanje verovatnoća stanja okruženja

Predlog mehanizma kojim se obavlja podešavanje pravila semantizacije f_i obuhvata iterativni postupak, gde jedna iteracija odgovara novoj deklaraciji stanja. Nulto stanje algoritma podrazumeva nepostojanje nijednog stanja okruženja u tabeli mogućih stanja. Nakon prijema nove deklaracije stanja, novo stanje se dodaje u tabelu, dok se pravila podešavaju na sledeći način:

$$g_i(E_i) = \begin{cases} 0 \xrightarrow{f_i} p_0(S_j | E_i) = 0 \\ 1 \xrightarrow{f_i} p_0(S_j | E_i) = 1 \end{cases} \quad (15)$$

U skladu sa jednačinom 15, vidi se da su verovatnoće da je okruženje u stanju S_j u prvoj iteraciji $p_0(S_j|E_i)$ maksimalne u svakom pravilu semantizacije f_i koje je zadovoljeno prekidačkom funkcijom g_i , odnosno, za vrednosti kontekstualnih parametara sa izvora i prikupljenih u trenutku uspostavljanja deklaracije koje zadovoljavaju prekidačku funkciju g_i . U svakoj sledećoj iteraciji, najpre se izračunava verovatnoća $p(S_j|E)$, i ukoliko je je uslov iz jednačine 13 zadovoljen, podešavanje pravila semantizacije u k -toj iteraciji obavlja se kao:

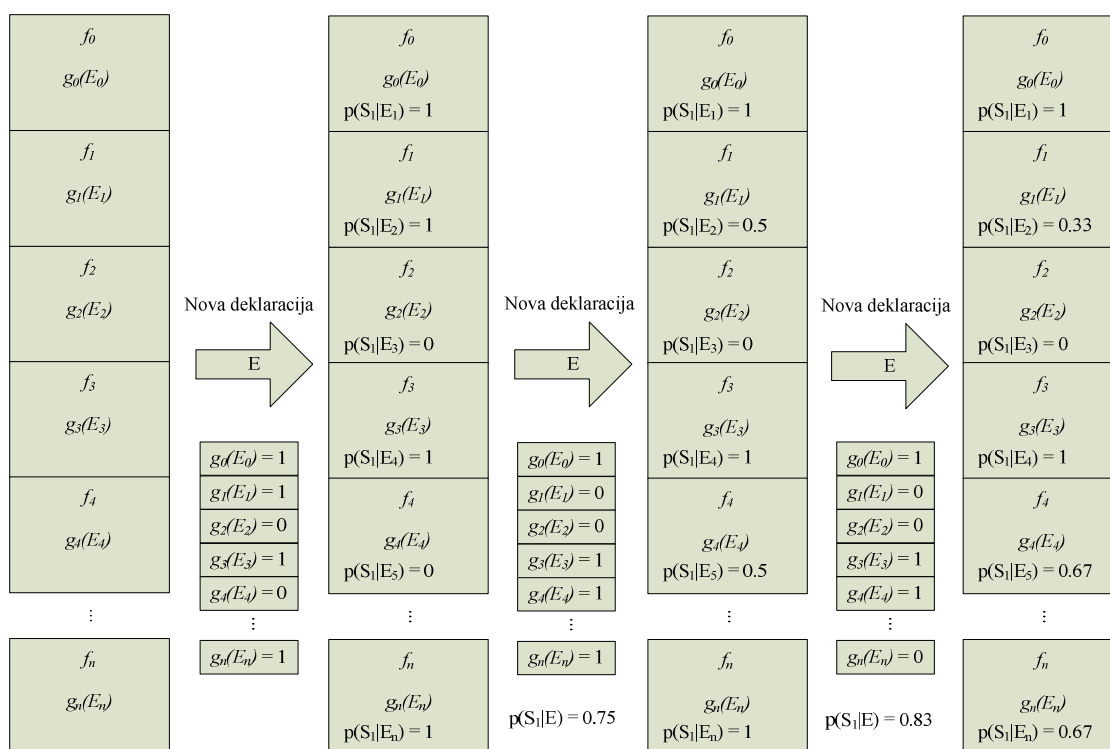
$$g_i(E_i) = \begin{cases} 0 \xrightarrow{f_i} p_k(S_j | E_i) = \frac{kp_{k-1}(S_j | E_i)}{k+1} \\ 1 \xrightarrow{f_i} p_k(S_j | E_i) = \frac{kp_{k-1}(S_j | E_i) + 1}{k+1} \end{cases} \quad (16)$$

Na Slici 21 data je ilustracija ažuriranja verovatnoća stanja okruženja korišćenjem gore opisanog iterativnog postupka, ako je $\alpha=0.3$. Nakon svake iteracije pravila semantizacije su sve tačnija. Jedna od mogućih tehnika za ispravno određivanje parametra α je na osnovu broja kontekstualnih parametara koji su na raspolaganju kontekstualnoj platformi – što je broj kontekstualnih parametara veći, α je manje i obratno. Tako je moguće odrediti α u funkciji broja kontekstualnih parametara, i to tako da je α dovoljno veliko da promena vrednosti prekidačke funkcije na jednom od izvora kontekstualnih parametara u drugoj iteraciji, tako da ona više ne bude zadovoljena, ne izazove izlazak parametara iz intervala poverenja:

$$1 - \frac{\sum_{i=0}^n g_i(E_i) p(S_j | E_i)}{\sum_{i=0}^n g_i(E_i)} < \alpha \quad (17)$$

Tako, u primeru sa Slike 21, prekidačka funkcija g_2 je bila zadovoljena u prvoj iteraciji, ali nije u drugoj. Ukoliko usvojimo da je broj izvora kontekstualnih parametara, te broj pravila semantizacije $n=5$, dobijamo, na osnovu jednačine 17:

$$1 - \frac{3}{4} < \alpha \Rightarrow \alpha > 0.25 \quad (18)$$



Slika 21. Ilustracija ažuriranja verovatnoće stanja okruženja nakon tri uzastopne deklaracije (iterativni postupak)

2.2 PROŠIRENJA ALGORITMA ZA SEMANTIZACIJU KONTEKSTA

Iako je osnovni algoritam predstavljen u poglavlju 2.1 dovoljan za podršku semantizaciji konteksta u smislu detekcije stanja okruženja, problem nastaje ukoliko su verovatnoće stanja okruženja međuzavisna. Međuzavisnost verovatnoća stanja izaziva rezonantne efekte koji utiču da se izlazne verovatnoće u algoritmu za treniranje

ponašaju nalik belom šumu, što čini tako izvedene zaključke neupotrebljivim. Da bi se ovaj problem prenebregao, u okviru ove disertacije uvode se modeli verovatnoće, odnosno *P-modeli*.

2.2.1 Jednodimenzioni P-model

P-model je neprekidna funkcija (ili funkcija neprekidna po intervalima), koja definiše promenu verovatnoće ciljnog stanja okruženja u zavisnosti od drugog stanja okruženja, ili nekog određenog kontekstualnog parametra e_i . Na primer, verovatnoća izražena P-modelom $P_{ab}(S_j|e_i)$ odgovara verovatnoći da je okruženje u stanju S_j , za opseg vrednosti kontekstualnih parametara $[a,b]$, odnosno:

$$P_{ab}(S_j|e_i) = \frac{1}{b-a} \int_a^b p(e) de \quad (19)$$

Sada je moguće proširiti jednačinu 4 tako da se uključe rezultatne verovatnoće dobijene P-modelima:

$$p(S_j|E) = \frac{1}{2n} \sum_{i=1}^n p(S_j|E_i) + \frac{1}{2m} \sum_{k=1}^m P(S_j|e_k) \quad (20)$$

Međuzavisnost stanja okruženja izražava se tako što izlazna verovatnoća koju daje P-model za određeno stanje u obzir uzima verovatnoće više različitih P-modela. Na primer, verovatnoća preseka stanja S_A i S_B računa se korišćenjem P-modela kao:

$$P_{ab}(S_A \cap S_B | e_i) = \frac{1}{b-a} \int_a^b p_A(e) p_B(e) de \quad (21)$$

Način funkcionisanja P-modela može se najbolje ilustrovati sledećim primerom. Neka se posmatra okruženje dnevne sobe kao referentno okruženje za realizaciju kontekstualne platforme. U dnevnoj sobi nalazi se moderni IP TV prijemnik, sa instaliranom aplikacijom za komunikaciju koja omogućava da se direktno sa TV prijemnika obavljaju glasovni i video pozivi, i razmenjuju tekstualne poruke. Kontekstualna platforma mora da obavi semantizaciju konteksta i da proceni stanje okruženja „korisnik koristi komunikacionu aplikaciju na TV prijemniku“. Kompleksnost ove determinacije leži u činjenici da komunikaciona aplikacija može da bude pokrenuta i sa mobilnog telefona i PC računara, pa je veoma razložna pretpostavka da korisnik neće koristiti TV verziju komunikacione aplikacije u slučaju da inače ne posmatra TV program, odnosno, da neće uključiti TV prijemnik da bi

Algoritmi za otkrivanje konteksta

koristio aplikaciju, već će je koristiti samo ukoliko je TV prijemnik već uključen i korisnik sedi ispred ekrana. Potrebno je ustanoviti verovatnoću stanja „korisnik koristi komunikacionu aplikaciju na TV prijemniku“, obeleženog sa S_{ctv} . Ova verovatnoća dobija se na osnovu verovatnoća dva stanja, i to S_{tv} , što je stanje „korisnik posmatra TV program“, kao i S_c , što je stanje „korisnik ima potrebu za korišćenjem komunikacione aplikacije“. Korišćenjem P-modela možemo dalje pisati:

$$P_{ab}(S_{ctv} | e_i) = P_{ab}(S_{tv} \cap S_c | e_i) = \frac{1}{b-a} \int_a^b p_{tv}(e) p_c(e) de \quad (21)$$

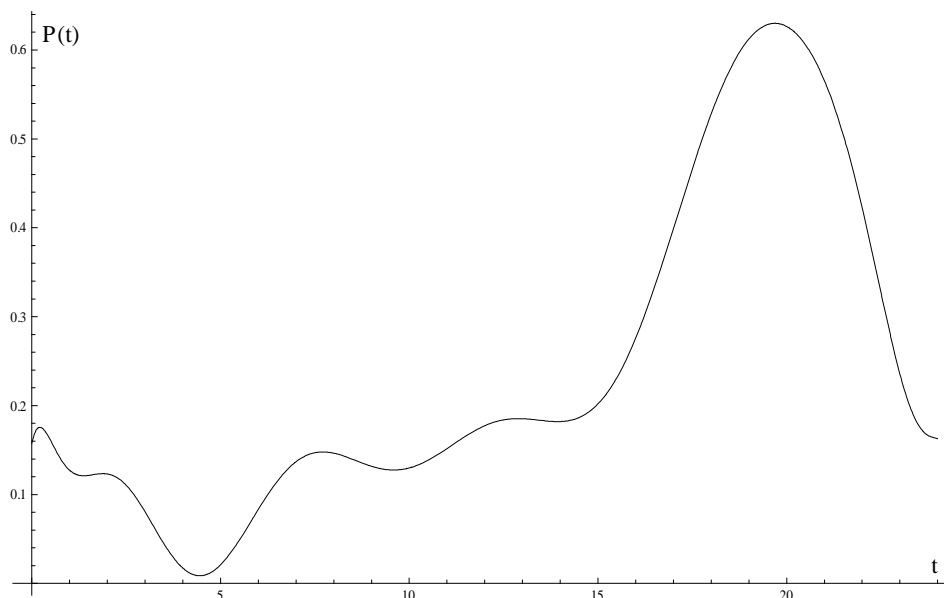
Da bi se primer materijalizovao, kao izvor kontekstualnih parametara za P-model biće uzet najjednostavniji izvor, kao što je sistemski sat. Posmatraćemo vreme u satima, kao realan broj izražen u intervalu $[0, 24]$, kao skup mogućih vrednosti kontekstualnih parametara $e_i (e_i = t)$. Sada možemo izvesti P-model za stanje okruženja „korisnik posmatra TV program“, na osnovu statističkih podataka o posmatranju TV programa u zavisnosti od doba dana (Australian Government Research and Statistics, 2009). Da bismo dobili kontinualan izračunljiv model, korišćenjem linearne regresije nad statističkim podacima dobijen je polinom 14-og stepena kao polinom najnižeg stepena sa dovoljno dobrim opisom statistike:

$$P(S_{tv} | t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \dots + \alpha_{14} t^{14} \quad (22)$$

Koeficijenti ovog polinoma dati su u Tabeli 5, dok je grafik P-modela za stanje okruženja „korisnik posmatra TV program“ u zavisnosti od kontekstualnog parametra sistemskog vremena t , dat na Slici 22.

Koeficijent	Vrednost	Koeficijent	Vrednost
α_0	$\frac{452535470}{2873103700}$	α_8	$\frac{-16997689}{36465652900}$
α_1	$\frac{108593841}{551482700}$	α_9	$\frac{41800111}{1278186895300}$
α_2	$\frac{-358845822}{537223700}$	α_{10}	$\frac{-15857461}{9881752890500}$
α_3	$\frac{675571380}{889537100}$	α_{11}	$\frac{384694}{7128252379500}$
α_4	$\frac{-490198798}{1114908500}$	α_{12}	$\frac{-14755}{12445665612200}$
α_5	$\frac{91033027}{610764900}$	α_{13}	$\frac{3285}{214308148717900}$
α_6	$\frac{-138132775}{4294183200}$	α_{14}	$\frac{-269}{3040371986555200}$
α_7	$\frac{35051583}{7527201700}$		

Tabela 5. Koeficijenti polinoma koji predstavlja P-model za stanje okruženja „korisnik posmatra TV program“.



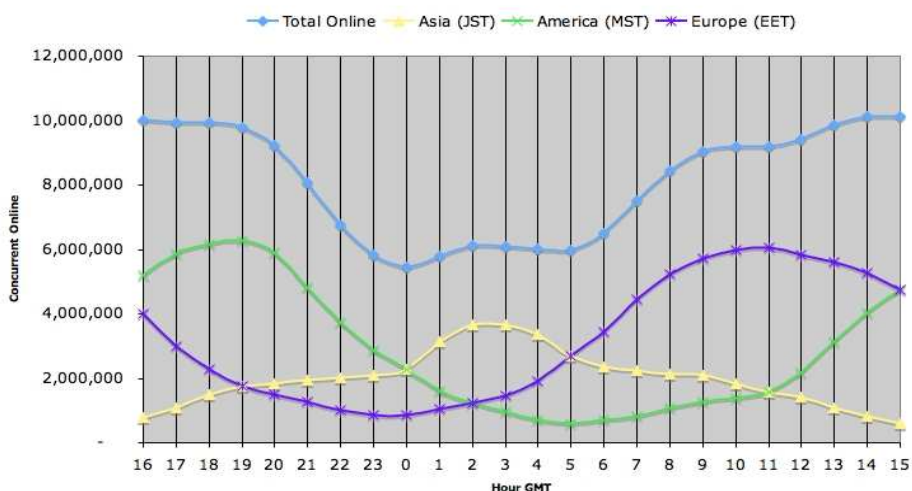
Slika 22. Grafik P-modela za stanje okruženja „korisnik posmatra TV program“ u zavisnosti od kontekstualnog parametra sistemskog vremena t

P-model verovatnoće stanja S_c „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ možemo odrediti aproksimiranjem potrebe za korišćenjem komunikacione aplikacije na osnovu statističkih podataka o korišćenju najpopularnijeg komunikacionog klijenta *Skype* (Skype, 2012). Ukoliko koristimo procenu ukupnog broja korisnika sa uključenim *Skype* klijentom u okviru svakog sata u toku dana u evropskoj časovnoj zoni EET (Borderless, 2008; Slika 23), kao i informaciju da je prosečno trajanje komunikacionih sesija u toku jednog sata oko 13 minuta, odnosno $t = 0.22$ (Kuan-Ta, 2006; Guha, 2006), možemo izvesti P-model za stanje S_c . Linearnom regresijom dobija se zadovoljavajuća aproksimacija polinomom osmog stepena:

$$P(S_c | t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \dots + \beta_8 t^8 \quad (23)$$

Koeficijenti ovog polinoma dati su u Tabeli 6, dok je grafik P-modela za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ u zavisnosti od kontekstualnog parametra sistemskog vremena t , dat na Slici 24.

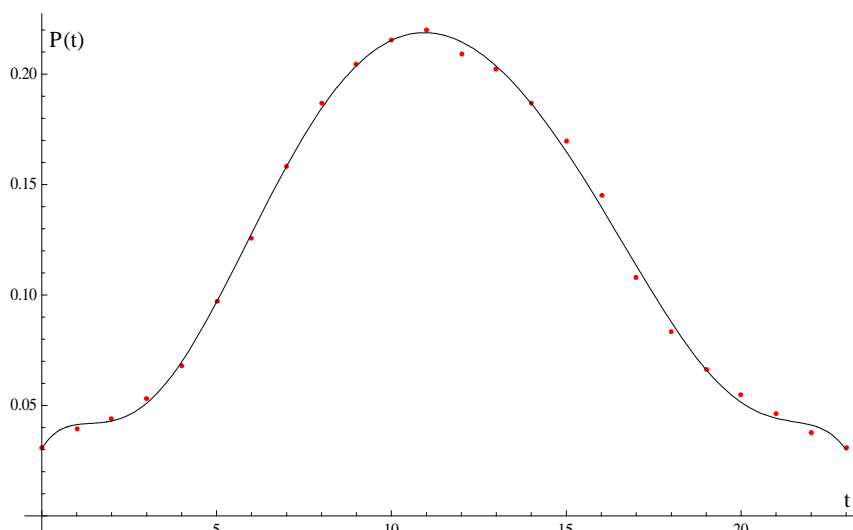
Algoritmi za otkrivanje konteksta



Slika 23. Ukupan broj korisnika sa priključenim *Skype* klijentom u toku dana (Borderless, 2008).

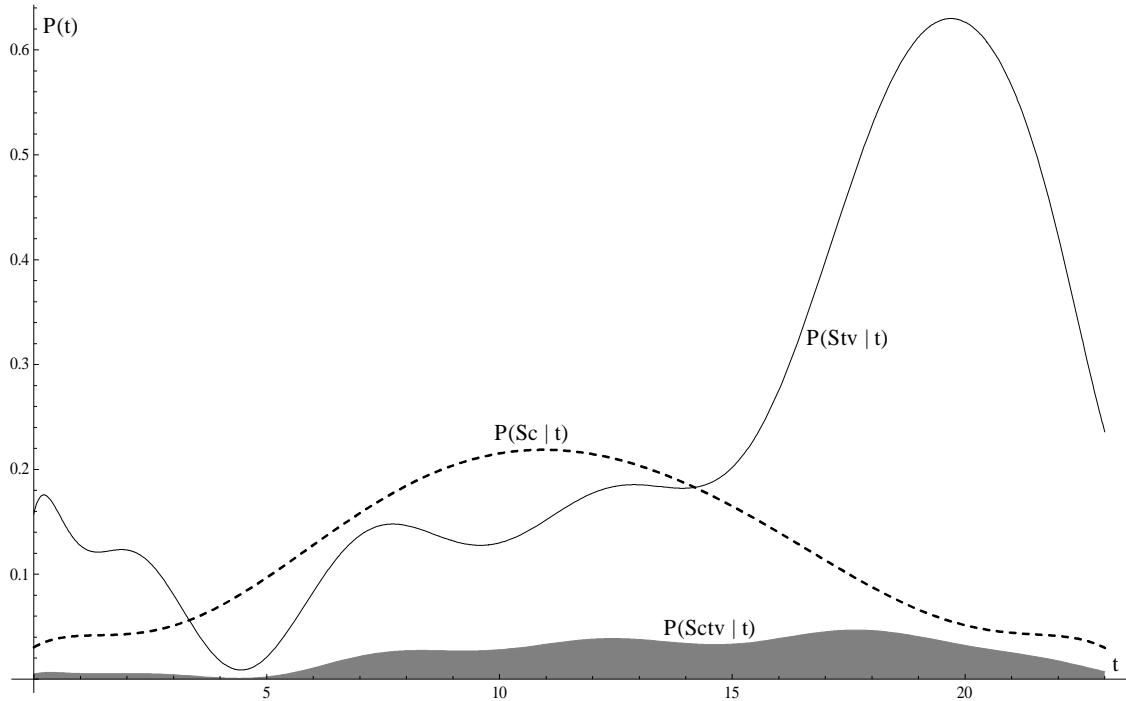
Koeficijent	Vrednost	Koeficijent	Vrednost
β_0	$3.034246776563 \cdot 10^{-2}$	β_5	$1.041898220023 \cdot 10^{-4}$
β_1	$2.527458628698 \cdot 10^{-2}$	β_6	$-4.899394729310 \cdot 10^{-6}$
β_2	$-2.113168518352 \cdot 10^{-2}$	β_7	$1.235991684201 \cdot 10^{-7}$
β_3	$7.941834465751 \cdot 10^{-3}$	β_8	$-1.299428549405 \cdot 10^{-9}$
β_4	$-1.252213926860 \cdot 10^{-3}$		

Tabela 6. Koeficijenti polinoma koji predstavlja P-model za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije”.



Slika 24. Grafik P-modela za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ u zavisnosti od kontekstualnog parametra sistemskog vremena t , sa verovatnoćama za svaki sat korišćenja

Korišćenjem jednačine 21 za navedeni primer, dobija se P-model koji odgovara stanju S_{ctv} , „korisnik koristi komunikacionu aplikaciju na TV prijemu“ označen sivom površinom na Slici 25.



Slika 25. Grafik P-modela za stanje okruženja „korisnik koristi komunikacionu aplikaciju na TV prijemu“ (siva površina na slici), generisan na osnovu međuzavisnosti P-modela za stanje okruženja „korisnik ima potrebu za korišćenjem komunikacione aplikacije“ (isprekidana linija) i P-modela za stanje okruženja „korisnik posmatra TV program“ (puna linija).

2.2.2 Višedimenzioni P-model

Višedimenzioni P-model je funkcija dva ili više argumenata (odnosno kompletnog vektora kontekstualnih parametara E_i , izvora kontekstualnih parametara i), koja kao rezultat daje vjerovatnoću stanja okruženja. Tako se ukupna vjerovatnoća da je okruženje u stanju S_j može računati daljim proširenjem jednačine 4:

$$p(S_j | E) = \frac{1}{3n} \sum_{i=1}^n p(S_j | E_i) + \frac{1}{3m} \sum_{k=1}^m P(S_j | e_k) + \frac{1}{3r} \sum_{l=1}^r P(S_j | E_l) \quad (24)$$

Za ilustraciju višedimenzionog P-modela nadovezujemo se na primer iz poglavlja 2.2.1, koji se odnosi na P-model vezan za stanje okruženja „korisnik posmatra TV program“, koji zavisi od jednog kontekstualnog parametra – sistemskog

Algoritmi za otkrivanje konteksta

vremena. Ovaj P-model se može realizovati kao višedimenzioni, dodavanjem dodatnih kontekstualnih parametara. Često se više dimenzije P-modela odnose na osobine korisnika koji se posmatraju. Tako, moguće je u obzir uzeti i godine starosti korisnika (x), što bi realizovalo sledeći P-model:

$$P_{tv_age} \left(S_j \mid \begin{matrix} t \\ x \end{matrix} \right) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} p(t) \left(\frac{1}{x_2 - x_1} \int_{x_1}^{x_2} p(x) dx \right) dt \quad (25)$$

Godine starosti korisnika utiču na verovatnoću na način da prave verovatnosni otklon (*probability offset*) u odnosu na polaznu verovatnoću. Ovaj otklon se može izvesti na osnovu statističkih podataka (UK Office for National Statistics, 2003), koji posebno daju zavisnost gledanosti TV programa u odnosu na godine i pol. Ova statistika se može najbolje aproksimirati polinomom šestog stepena za slučaj muškog pola, odnosno polinomom sedmog stepena za slučaj ženskog pola:

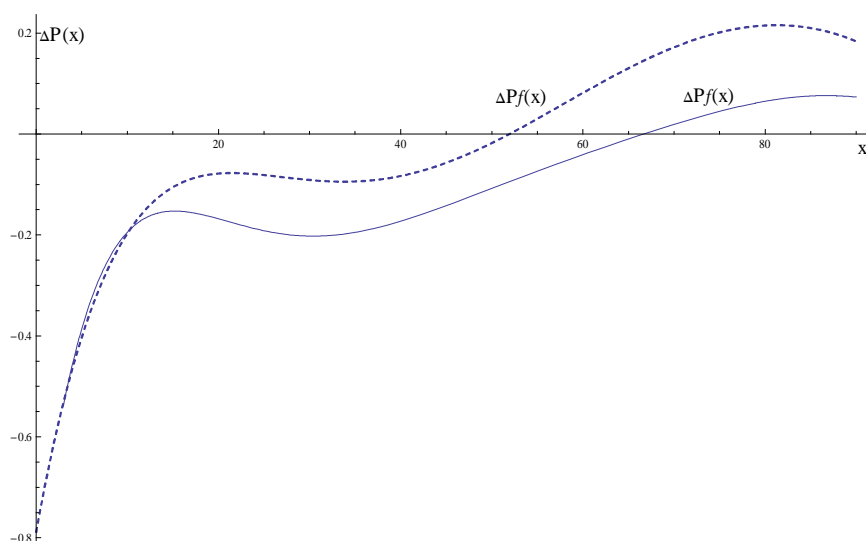
$$\Delta P_m (S_{tv} \mid x) = \gamma_0 + \gamma_1 x + \gamma_2 x^2 + \gamma_3 x^3 + \dots + \gamma_6 x^6 \quad (26)$$

$$\Delta P_f (S_{tv} \mid x) = \delta_0 + \delta_1 x + \delta_2 x^2 + \delta_3 x^3 + \dots + \delta_7 x^7 \quad (27)$$

U Tabeli 7 dati su koeficijenti polinoma iz jednačina 26 i 27, dobijeni linearnom regresijom na osnovu statističkih podataka. Na Slici 26 dati su grafici tih polinoma.

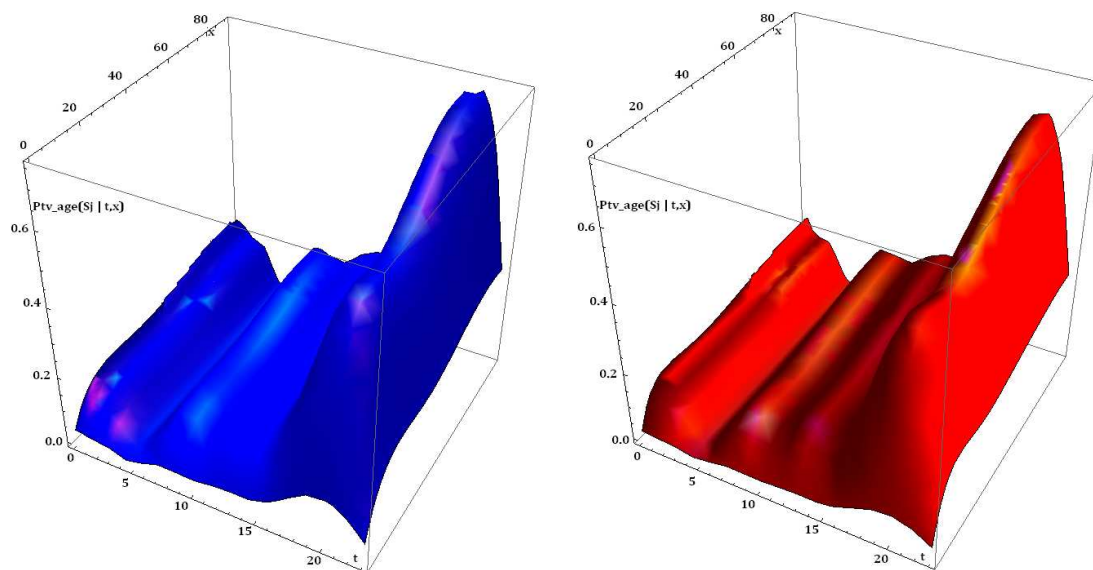
Koeficijent	Vrednost	Koeficijent	Vrednost
γ_0	$\frac{-784170653}{994662400}$	δ_0	$\frac{-113081463}{128331100}$
γ_1	$\frac{215200202}{2141194900}$	δ_1	$\frac{120741694}{840324500}$
γ_2	$\frac{-21358235}{4037422700}$	δ_2	$\frac{-67090167}{6277493900}$
γ_3	$\frac{11060057}{85091570300}$	δ_3	$\frac{18972706}{49089967500}$
γ_4	$\frac{-752600}{476893512900}$	δ_4	$\frac{-2030778}{267011954900}$
γ_5	$\frac{20269}{2156245618300}$	δ_5	$\frac{247594}{2940048349100}$
γ_6	$\frac{-2758}{122715942192500}$	δ_6	$\frac{-133131}{69563828600000}$
		δ_7	$\frac{906}{759825057232900}$

Tabela 7. Koeficijenti polinoma za verovatnosni otklon verovatnoće stanja okruženja “korisnik posmatra TV program” za korisnike muškog pola (γ) i ženskog pola (δ).



Slika 26. Grafici polinoma za verovatnosni otklon verovatnoće stanja okruženja „korisnik posmatra TV program“ u zavisnosti od godina starosti korisnika, za korisnike muškog pola (isprekidana linija) i za korisnike ženskog pola (puna linija)

Konačno, moguće je generisati višedimezioni P-model čiji je grafički prikaz dat na Slici 27.



Slika 27. Grafik višedimenzionog P-modela za stanje okruženja „korisnik posmatra TV program“ u zavisnosti od kontekstulnih parametara sistemskog vremena (t) i godina starosti korisnika (x), za korisnike muškog pola (levo) i za korisnike ženskog pola (desno)

2.2.3 Personalizacija semantizacije konteksta

Za primenu u kontekstualnim platformama, izuzetno je važna mogućnost personalizacije algoritma za semantizaciju. To znači da, pri proceni stanja okruženja, stanje nije jedinstveno i objektivno za celokupno okruženje, već se stanje može razlikovati u zavisnosti od korisnika ili korisničkih grupa. Otkrivanje stanja okruženja pri čemu se uzima u obzir identifikacija korisnika, odnosno njegova pripadnost određenoj grupi korisnika, može se obaviti *statistički* ili *dinamički*.

Statistička procena zavisi od konfiguracije parametara okruženja, odnosno od definicije korisničkih grupa i pripadnosti korisnika korisničkim grupama. Najpre je potrebno definisati ukupan broj korisnika koji pripadaju okruženju (N), kao:

$$N = \sum_{i=1}^q n_i \quad (28)$$

U ovoj jednačini q predstavlja ukupan broj različitih korisničkih grupa, a n_i broj korisnika koji pripada jednoj od grupa G_i pri čemu se pretpostavlja da je:

$$\bigcap_{i=1}^q G_i = \emptyset \quad (29)$$

Sada može biti od značaja da se izračuna statička verovatnoća, da bar jedan korisnik iz grupe G_k pripada celokupnom posmatranom okruženju Ψ , tj. utiče na stanje okruženja ili je stanje okruženja je od njegovog interesa:

$$p(X \subseteq G_k, X \subseteq \Psi) = 1 - \frac{\sum_{j=1}^N \binom{\sum_{i=1}^{k-1} n_i + \sum_{i=k+1}^q n_i}{j}}{\sum_{j=1}^N \binom{N}{j}} \quad (29)$$

U prethodnoj jednačini n_i , $i \neq k$, jeste broj korisnika koji pripadaju ostalim korisničkim grupama.

Takođe, može se izračunati verovatnoća da samo korisnici koji pripadaju grupi G_k pripadaju posmatranom okruženju Ψ , tj. uzimaju se u obzir samo ona stanja okruženja na koja utiču članovi grupe G_k :

$$p(X \subseteq G_k, Y \subseteq G_{m \neq k}, X \subseteq \Psi, Y \not\subseteq \Psi) = \sum_{i=1}^{n_k} \frac{\binom{n_k}{i}}{\sum_{j=1}^N \binom{N}{j}} \quad (30)$$

U prethodnoj jednačini, n_k je broj korisnika koji pripadaju grupi G_k .

U nekim primenama od značaja je i statistička procena pojedinačnog uticaja na stanje okruženja, kao i uticaja grupe, odnosno, potrebno je izračunati verovatnoću da jedan i samo jedan korisnik utiče na stanje okruženja (jednačina 31), odnosno da više od jednog korisnika utiče na stanje okruženja (jednačina 32).

$$p(x \in \Psi, |\Psi| = 1) = \frac{N}{\sum_{j=1}^N \binom{N}{j}} \quad (31)$$

$$p(X \subseteq \Psi, |\Psi| > 1) = 1 - \frac{N}{\sum_{j=1}^N \binom{N}{j}} \quad (32)$$

Uvrštavanjem statističke procene personalizacije u jednačinu 4, verovatnoća da je okruženje Ψ u stanju S_j u odnosu na korisničku grupu G_k može se izraziti kao:

$$p(S_j | E) = \frac{P(X \subseteq G_k, X \subseteq \Psi)}{n} \sum_{i=1}^n p(S_j | E_i) \quad (33)$$

Osim gore navedenih, naravno, moguće je navesti različite statističke i verovatnosne procene koje se odnose na korisnike, ili na bilo koji drugi način personalizacije okruženja koji može zavistiti od određenih kontekstualnih parametara. Pozitivna strana statističkih procena je što su u mogućnosti da daju procenu bez obzira na tačnost na nivou mreže kontekstualne platforme, odnosno bez obzira na povratnu spregu sa senzora. Time se obezbeđuje minimum potrebne funkcionalnosti u uslovima u kojima kontekstualna platforma otežano funkcioniše (u slučaju otkaza), odnosno u uslovima kada kontekstualna platforma nije kalibrisana na odgovarajući način. S druge strane, oslanjanje isključivo na statističke procene izuzetno smanjuje ukupnu tačnost semantizacije konteksta, s obzirom na to da svako okruženje ima specifična fizička svojstva i svaka grupa korisnika se suštinski razlikuje od neke druge grupe.

Dinamička procena oslanja se na kontekstualne parametre koji se dobavljaju od strane nekog relevantnog izvora koji je u mogućnosti da obavi prepoznavanje i

Algoritmi za otkrivanje konteksta

prebrojavanje korisnika. Tako bi se, na primer, mogla zahtevati procena verovatnoće stanja okruženja „korisnik posmatra TV program“ za poznatog korisnika. Jedno od pravila semantizacije u kontekstualnoj polisi može se definisati kao:

$$g'(E^T \parallel \bar{E}^T) = g' \left(\begin{bmatrix} e(t) \\ e(t - \Delta t_1) \\ \dots \\ e(t - \Delta t_n) \end{bmatrix} \right) = \begin{cases} 0, \frac{1}{2n} \sum_{i=1}^n e(t - \Delta t_i) + \frac{1}{2} e(t) \neq 1 \\ 1, \frac{1}{2n} \sum_{i=1}^n e(t - \Delta t_i) + \frac{1}{2} e(t) = 1 \end{cases} \quad (34)$$

$$f'(E^T \parallel \bar{E}^T) = p(S_j | E^T \parallel \bar{E}^T) = g'(E^T \parallel \bar{E}^T)$$

U prethodnom primeru, podrazumeva se da su kontekstualni parametri e binarne vrednosti, koje imaju vrednost 1 ukoliko je detektovan korisnik od interesa, odnosno vrednost 0 ukoliko korisnik nije detektovan. Tipičan senzor koji može da pruži ovakvu informaciju je, npr. optička kamera sa algoritmom za prepoznavanje lica. Ukoliko je u sadašnjem trenutku t , kao i u svim prethodnim trenucima odmeravanja počev od $t - \Delta t_n$, korisnik detektovan, prekidačka funkcija omogućuje postavljanje verovatnoće da korisnik posmatra TV program, dok pravilo semantizacije dodeljuje maksimalnu verovatnoću za pozitivan izlaz prekidačke funkcije.

2.3 ALGORITAM ZA PROCENU ZAUZETOSTI KORISNIKA

S obzirom na istaknuti značaj rešenja problema asinhronih zahteva, kontekstualna platforma je opremljena algoritmima koji omogućuju procenu stanja, odnosno zauzetosti korisnika, na osnovu koje je moguće efikasno upravljati asinhronim zahtevima. Algoritam za procenu zauzetosti korisnika, koji je predstavljen u ovom poglavlju, zasniva se na algoritmu za semantizaciju konteksta predstavljenom u poglavlju 2.1.

Algoritam za procenu zauzetosti korisnika za polaznu osnovu koristi cenu asinhronog zahteva (*cost of interruption*) (Horvitz, 2004). Cena asinhronog zahteva procenjena je kao vrednost koju je korisnik voljan da plati da bi izbegao negativne efekte koji proizilaze iz asinhronog zahteva. Algoritam je personalizovan na način da svaki korisnik određuje cenu asinhronog zahteva uspostavljenog u toku stanja okruženja S_j . Ukupna procenjena cena asinhronog zahteva (*estimated cost of interruption – ECI*) se dalje računa kao zbir svih definisanih cena ponderisanih

verovatnoćom određenog stanja okruženja. Ukoliko se cena asinhronog zahteva uspostavljenog u toku stanja okruženja S_j obeleži kao $C(S_j)$, može se pisati:

$$ECI = \sum_j p(S_j | E) C(S_j) \quad (35)$$

Da bi se obezbedila uporediva procena cene asinhronog zahteva, potrebno je ograničiti mogući opseg vrednosti ECI . Iz tog razloga može se definisati standardizovana procenjena cena asinhronog zahteva (\overline{ECI}) koja može uzimati vrednosti iz opsega $[0,1]$, odnosno:

$$\overline{ECI} = \frac{1}{1 + e^{-ECI}} \quad (36)$$

Dalje, definisaćemo zauzetost korisnika (*availability score* – AS) kao vrednost koja je podesnija za primene u posredovanju pri uspostavljanju asinhronih zahteva, sa vrednostima u opsegu $[0,1]$, gde 1 označava maksimalnu zauzetost korisnika, dok 0 označava da korisnik uopšte nije zauzet. AS možemo izraziti preko ECI kao:

$$AS = 1 - \overline{ECI} = \frac{e^{-ECI}}{1 + e^{-ECI}} \quad (37)$$

S obzirom da je određivanje $p(S_j|E)$ problem koji je razrađen u poglavljima 2.1 i 2.2, jedini preostali problem je određivanje cene $C(S_j)$ i njeno saopštavanje kontekstualnoj platformi. U nastavku biće razrađene dve metode za saopštavanje cene: *tabelarni* metod i *direktni* metod.

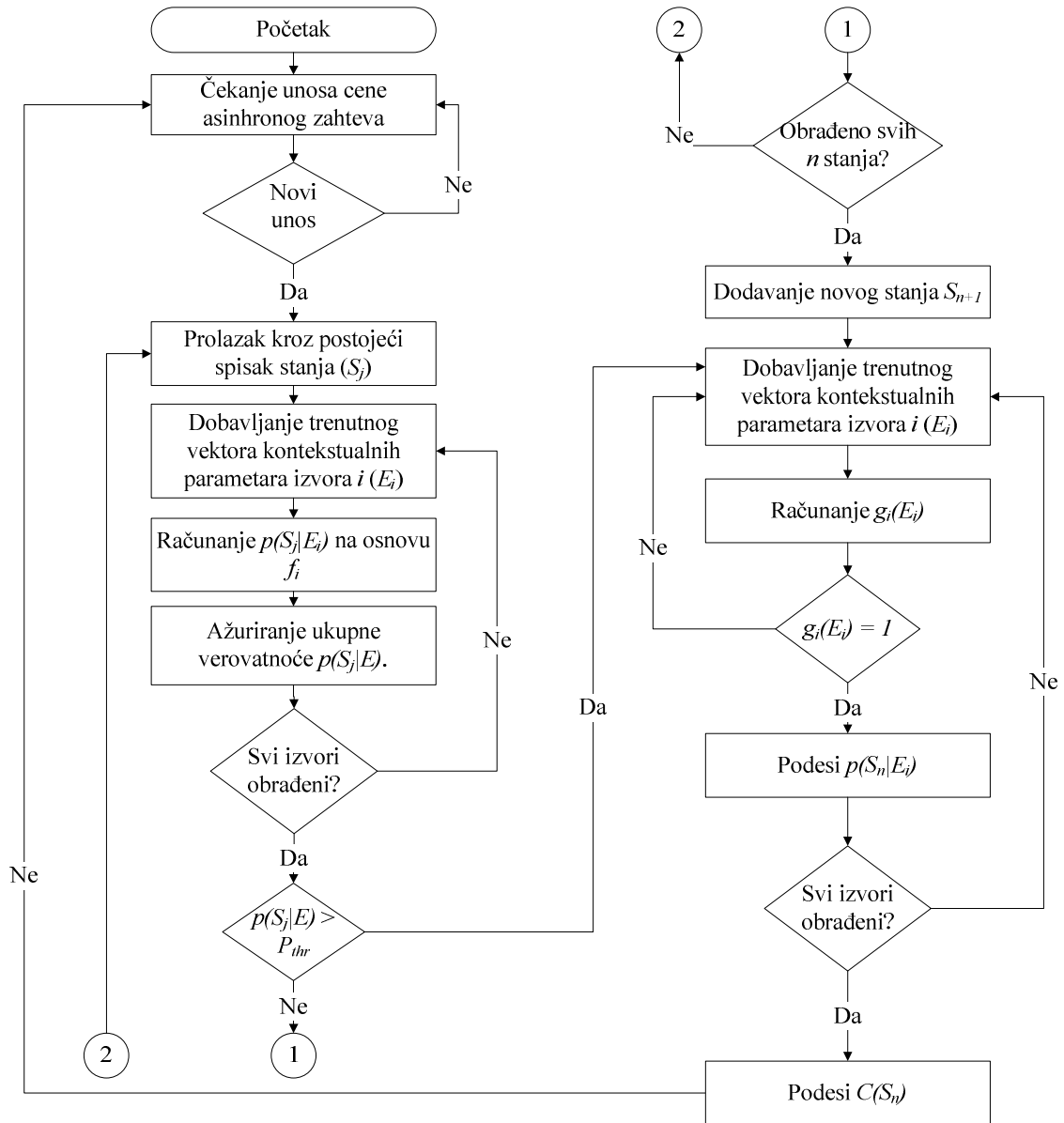
Tabelarni metod za saopštavanje cene asinhronog zahteva kao osnovu koristi semantizaciju konteksta zasnovanu na polisama, ali koristi delom i rezultate automatske semantizacije konteksta. Osnova ovog algoritma zasniva se na periodičnim upitima korisniku (ili nasumičnom pristupu stranici sa upitom od strane korisnika), u kojima je predstavljena tabela za određivanje cene sa dve kolone – u prvoj koloni nalaze se sva stanja okruženja, a u drugoj koloni korisnik definiše korespondentne cene asinhronih zahteva, uspostavljenih dok je okruženje u odgovarajućem stanju. Primer tabele za određivanje cene asinhronog zahteva dat je kao Tabela 8.

<i>Stanje okruženja (aktivnost)</i>	<i>Cena asinhronog zahteva</i>
Provera elektronske pošte	50
Sastanak	100
Telefonski razgovor	150
Gledanje TV programa	30
Pretraživanje interneta	40

Tabela 8. Primer tabele za određivanje cene asinhronog zahteva u zavisnosti od stanja okruženja

Spisak stanja okruženja definisan je na osnovu kontekstualnih polisa, u kojima su stanja jasno numerisana i semantički označena tako da mogu biti prepoznata od strane korisnika. Takođe, spisak stanja se može proširiti i stanjima generisanim u okviru automatske semantizacije, ali samo za ona stanja koja su označena u postupku deklaracije. Oznaka stanja u ovom slučaju mora biti semantička, kako bi stanje moglo da bude prepoznato od strane korisnika, te da mu bude dodeljena adekvatna cena.

Direktni metod za saopštavanje cene asinhronog zahteva kao osnovu koristi automatsku semantizaciju konteksta. Proširenja i izmene postupka automatske semantizacije odnose se na sledeće: (1) deklaracija stanja okruženja zamenjena je deklaracijom cene, u trenutku po sopstvenom nahođenju korisnika, posebno u slučajevima neposredno nakon uspostavljanja asinhronog zahteva; (2) nije potrebna deklaracija stanja koja nisu numerisana od strane korisnika, već se stanja kreiraju automatski, korespondirajući deklaracijama cene; i (3) dodaje se korak ažuriranja cene asinhronog zahteva za odgovarajuće stanje okruženja. Dijagram aktivnosti za ovako izmenjen algoritam dat je na Slici 28.



Slika 28. UML dijagram aktivnosti algoritma za direktni metod saopštavanja cene asinhronog zahteva

2.4 FIZIČKA REALIZACIJA ALGORITAMA

S obzirom na osobenosti predloženih algoritama, od izuzetnog je značaja pažljiv odabir i projektovanje njihove fizičke realizacije. Očekivano je da se realizacija osnovnog dela algoritma odnosi na ciljno okruženje u kojem se razvija kontekstualna platforma, što je predmet razmatranja u poglavlju 3. Međutim, veoma je važno ispuniti i sledeće zahteve: (1) kreiranje kontekstualnih polisa se može obavljati poluautomatski, što zahteva obezbeđivanje mogućnosti za podešavanje polise; (2) instalacija

Algoritmi za otkrivanje konteksta

kontekstualne platforme i njena primenljivost na više različitih ciljnih okruženja zahteva jednostavan mehanizam podešavanja rada algoritama i (3) robustnost kontekstualne platforme je izuzetno važna, te je potrebno minimizovati mogućnost da neispravna ili neprimenljiva kontekstualna polisa izazove otkaz kontekstualne platforme u celini.

Analizom gore navedenih zahteva nameće se zaključak da bi bilo dobro da se kontekstualne polise, odnosno delovi algoritma koji se direktno odnose na semantizaciju konteksta, izvršavaju u nekom obliku virtuelnog okruženja, i da posredstvom odgovarajuće sprege komuniciraju sa ostatkom kontekstualne platforme. Dodatno, s obzirom na potrebu za podešavanjem, a neretko i pisanjem kontekstualnih polisa od strane čoveka, mehanizam za obavljanje tih izmena mora biti jednostavan, i ne sme zahtevati ponovno prevođenje programske podrške za kontekstualnu platformu, pa čak ne sme izazvati ni njeno ponovno pokretanje (*restart*). Konačno, semantika izmena mora biti jednostavna i prirodna – izuzetno je važno da nije potrebno imati fakultetsko poznavanje alata za razvoj programske podrške da bi se obavila izmena kontekstualne polise.

U okviru ove disertacije biće razmotreno i realizovano korišćenje skript jezika ili meta jezika za realizaciju kontekstualnih polisa i dela algoritama u okviru kontekstualne platforme. Korišćenjem skript jezika odgovara se na sve postavljene zahteve u sledećem: (1) izmena skript jezika zahteva zaustavljanje izvršavanja skripta, otvaranje skripta u nekom tekst editoru i jednostavno unošenje novog ili izmenu postojećeg teksta; (2) skriptovi se mogu dodavati i uklanjati sa kontekstualne platforme, te skladištiti i organizovati bez potrebe za proverom kompatibilnosti sa programskom podrškom kontekstualne platforme, i bez potrebe za bilo kakvim izmenama programske podrške kontekstualne platforme, što izuzetno olakšava instalaciju i konfiguraciju za ciljno okruženje, i (3) ukoliko dođe do nepravilnosti u izvršavanju skripta, njegovo izvršavanje se automatski prekida, zamenjuje novim skriptom ili ponovo startuje, čime je praktično onemogućen otkaz kontekstualne platforme u celini. Alternativa skriptovima je realizacija programa za virtuelne mašine (kao što je, npr. Java), međutim, osim potrebe za prevođenjem aplikacija u bajt kod, ovakve realizacije ne zadovoljavaju ključni kriterijum rada u realnom vremenu,

odnosno brzina izvršavanja nije odgovarajuća da bi realizacija bila primenljiva na široj klasi uređaja potrošačke elektronike, što ova disertacija ima za cilj.

S obzirom na to da je broj skript jezika i meta jezika koji mogu da dođu u obzir za korišćenje u okviru kontekstualne platforme veliki, u nastavku će biti izložena analiza skript jezika i dati razlozi za konačan izbor jezika koji će biti korišćen.

2.4.1 Skript jezici za realizaciju kontekstualnih polisa

Pri izboru skript jezika za realizaciju kontekstualnih polisa, uzima se u obzir veliki broj različitih kriterijuma, specifičnih za uređaje potrošačke elektronike i za primene u kontekstualnim platformama. Pre detaljne analize izbor će najpre biti sužen, i to na tri skript jezika koja su najbliže mogućoj primeni.

Najstrožiji kriterijum izvodljivosti upotrebe skript jezika u ciljnom sistemu, je zahtev rada u realnom vremenu. Iz tog razloga, jezik, odnosno odgovarajući interpreter, moraju da podrže sledeće:

- mala veličina biblioteka, kao i zauzeće procesora i memorije;
- mogućnost jednostavnog sprezanja sa C/C++ okruženjem;
- potvrdu upotrebe u različitim komercijalnim projektima, najbolje u uređajima potrošačke elektronike.

Polazni skup jezika koji se uzimaju u razmatranje dobijen je uvidom u novije trendove korišćenja skript jezika za različite aplikacije, te za aktivnost zajednice u održavanju i ažuriranju jezika. To su sledeći jezici: Javascript, PHP, Perl, Python, Lua, XML, SMIL. Javascript i PHP su namenjeni za izvršavanje na klijentskom, odnosno serverskom delu *web* aplikacije, respektivno. Iako je njihova upotreba za druge namene moguća, sprezanje sa jezicima C/C++, iako moguće, nije dobro dokumentovano ili nije predviđeno. Python i Perl se mogu integrisati u C/C++, ali ne zadovoljavaju kriterijum male veličine biblioteka. Iako postoje minimalističke implementacije interpretera za oba jezika (CPython, Microperl itd) upotreba ovih rešenja u uređajima potrošačke elektronike još nije zaživela. Upotreba Perl-a se nalazi u nekim PC aplikacijama, poput programa otvorenog koda Mister House (MisterHouse, 2008) koji koristi Perl jezik za opis ponašanja uređaja u kući. Lua skript jezik (Ierusalimschy, 1996) zadovoljava gore navedene kriterijume i već je postao de fakto standard za opis ponašanja logičkih entiteta na visokom nivou apstrakcije. Na primer, on se uveliko koristi za definisanje igračkih nivoa u video igrama poput World

of Warcraft (Emmerich, 2009) i CryEngine (CryDev, 2010), kao i za definisanje kontekstualnih polisa u sistemima za automatizaciju domaćinstva, poput Control4 (Control4, 2012). Nekoliko različitih autora predlaže XML jezik (Harold, 2004) za opis ponašanja sistema u zavisnosti od senzorskih ulaza. Na primer, predstavljen je prototip koji koristi vremenske i senzorske događaje koji su definisani atributima matičnih čvorova u XML sintaksi. Po objavi ovih događaja, ugnježdeni čvorovi se interpretiraju redom i po hijerarhiji, nalik programskim jezicima visokog nivoa. Ovi čvorovi opisuju naredbe koje se izdaju aktuatorima (uključujući svetlo, pokreni reprodukciju snimka i sl) (Bjelica, 2010c). Projekat otvorenog koda Open Remote koristi XML jezik za opis svih uređaja u kući i njihovih međusobnih interakcija (OpenRemote, 2012). Poput XML jezika, SMIL se koristi za primene u oblasti digitalnih oglasa (*digital signage*) za opis interaktivnih multimedijalnih prezentacija (Hoschka, 1998). Postoje nastojanja da se SMIL usvoji kao standard u oblasti digitalnih oglasa.

Lua, XML i SMIL jezik koriste se u većem broju istraživanja u oblasti kontekstualnih platformi. Gaia metaoperativni sistem predložen je kao srednji sloj programske podrške za opis ponašanja uređaja u senzorskim mrežama (Roman, 2002). Za programiranje i koordinaciju svih entiteta u okviru sistema, Gaia koristi LuaOrb jezik, koji je zasnovan na Lua jeziku. Takođe, predloženo je i proširenje OSGi okruženja u obliku mobilnih agenata. Ovi agenti se koriste za opis akcija i šablona u kućnoj automatici, a programiraju se MASML skript jezikom u potpunosti zasnovanom na XML sintaksi (Wu, 2007b). SMIL jezik, odnosno njegova ekstenzija IPML, predložen je za opis distribuiranog multimedijalnog okruženja za primenu u pametnim kućama (Hu, 2006). Iz tog razloga, kao skup jezika koji će dalje biti evaluirani uzimaju se Lua, XML i SMIL.

2.4.1.1 Lua, XML i SMIL

U ovom poglavlju dat je primer upotrebe navedenih skript jezika za definisanje jednostavnih pravila semantizacije u okviru kontekstualnih polisa. Dat je kratak opis svakog jezika, kao i primer skripta za opis jednostavnog pravila semantizacije: ukoliko je veće ili noć, i u prostoriji ima detektovanih osoba, postavi verovatnoću stanja okruženja „večernje druženje“ sa oznakom S_8 na 0.8.

Lua jezik je već u upotrebi u komercijalnim sistemima kućne automatike (Control4, 2012). Osnovna odlika ovog jezika je jednostavnost ugradnje skript interpretera unutar bilo koje aplikacije pisane u jezicima C/C++, ali i onima pisanim u jezicima Java, C#, Smalltalk, Fortran, Ada, Erlang, Perl i Ruby. Sintaksa Lua jezika nalik je sintaksi Javascript-a. Da bi se koristio u okviru kontekstualne platforme, Lua jezik mora biti proširen novim funkcijama, kojim se omogućava mehanizam reagovanja na događaje, postavljanje verovatnoće stanja okruženja i paralelni rad. Lua jezik podržava uključivanje modula pisanih u odvojenim datotekama, što omogućava dodatnu apstrakciju programske sprege (API) u okviru Lua koda, čime se olakšava programiranje scenarija. Primerni skript napisan u Lua jeziku dat je na Slici 29.

```
require("STATES")
require("PIR")
evid = events_register("PIR Sensor", PIR.MOTION_IN)
while true do
  events_wait(evid, 0)
  if (tonumber(os.date("%H")) > 19) or
      (tonumber(os.date("%H")) < 6) then
    STATES.s8(0.8)
  end
end
end
events_unregister(evid)
```

Slika 29. Primerni skript pisan u jeziku Lua

XML jezik predstavlja specifikaciju oznaka kojom se prikazuju podaci i njihovi hijerarhijski odnosi (matični čvorovi imaju čvorove potomke). Samim tim XML jezik se ne može direktno koristiti za pisanje skriptova, jer ne poseduje skript interpreter koji bi ga „izvršavao“. Međutim, XML se može proširiti tako da se interpretiranje omogući (Bjelica, 2010c). Ukoliko se čvorovi iz XML opisa interpretiraju kao simboli svojstveni programskim jezicima višeg nivoa (npr. iteracija *for*, selekcija *if*, grananje *switch-case*, naredbe) moguće je koristiti XML jezik u okviru kontekstualne platforme. Podrška za module i paralelni rad nije svojstvena XML jeziku. Ipak, scenario koji se sastoji od izvršavanja niza naredbi nakon prijema određenog događaja može se jednostavno programirati. Primerni skript napisan u XML jeziku dat je na Slici 30.

```
<macro>
  <driver type="signal" modulename="PIR Sensor" status="MOTION_IN">
    <if>
      <case namea="@time_hour" val="19" rel="GR">
        <instruct modulename="Bedroom Light" function="on" />
      </case>
      <case namea="@time_hour" val="6" rel="LS">
        <instruct modulename="S8" function="0.8" />
      </case>
    </if>
  </driver>
</macro>
```

Slika 30. Primerni skript pisan u jeziku XML

SMIL jezik se vodi logikom vremenskih osa – moguće je definisati niz naredbi, ili nekoliko nizova naredbi koji se uporedo izvršavaju, kao i programirati sa zahtevima realnog vremena uz definisanje jasnih vremenskih određenja (trajanje komandi, vreme početka i završetka važnosti bloka i sl). Ovo omogućava jednostavno programiranje različitih efekata u sprezi kontekstualne platforme sa nekim aktuatorskim sistemom, poput naizmenične reprodukcije multimedijalnih sadržaja, svetlosnih efekata i sl. Zahvaljujući postojanju *ref* oznake SMIL sintaksu je moguće proizvoljno proširiti, dok modularizacija nije podržana. SMIL omogućava jednostavnije programiranje reakcije na događaj nego sekvencijalno programiranje. Primerni skript napisan u SMIL jeziku dat je na Slici 31.

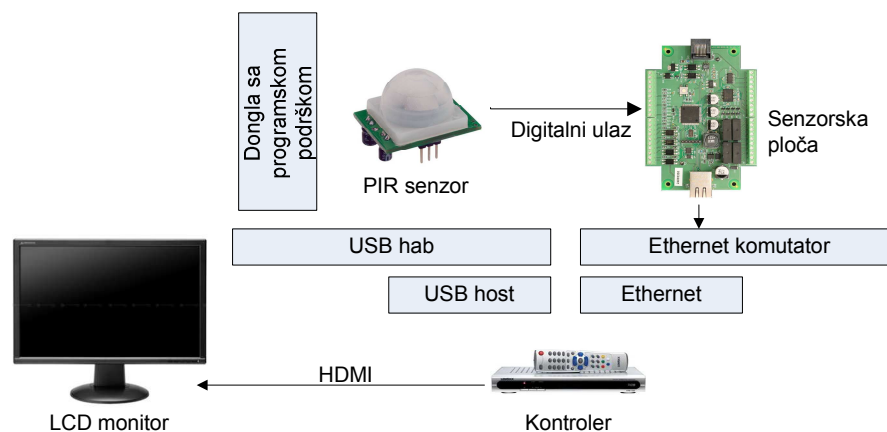
```
<body>
  <par>
    <ref id="pir2" begin="wallclock(19:00)" end="wallclock(06:00)">
      <param name="ctx:sensor" value="PIR Sensor"/>
      <param name="ctx:event" value="MOTION_IN"/>
    </ref>
    <ref begin="pir2.activateEvent">
      <param name="ctx:act" value="S8"/>
      <param name="ctx:command" value="0.8"/>
    </ref>
  </par>
</body>
```

Slika 31. Primerni skript pisan u jeziku SMIL

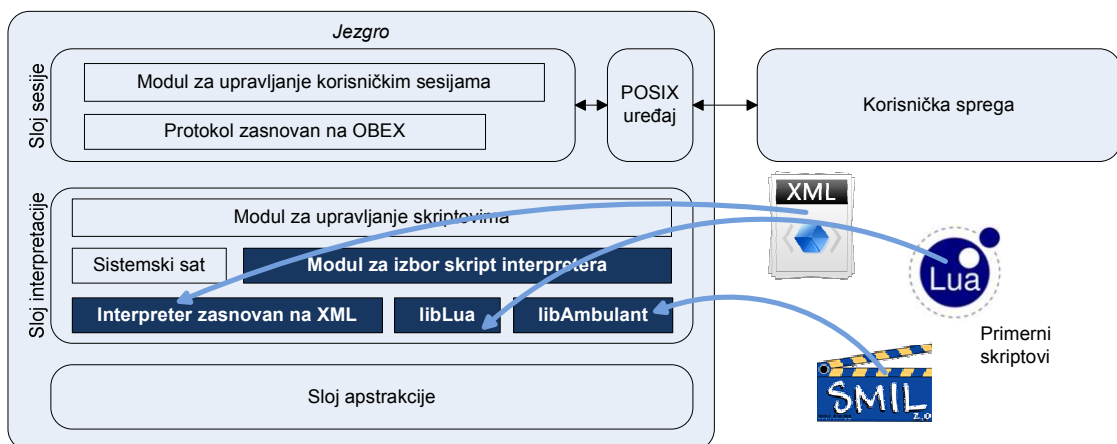
2.4.1.2 Uporedna analiza primene skript jezika u okviru kontekstualne platforme

Za potrebe analize performansi, upotrebljivosti i ostalih važnih faktora koji odlučuju o izboru određenog jezika za korišćenje u okviru kontekstualne platforme, odgovarajući skript interpreteri integrisani su u realni sistem razvijen u okviru

prethodnog istraživanja (Bjelica, 2010c). Komponente koje su korišćene u okviru procene su kontroler, PIR senzor za detekciju pokreta i senzorska ploča. Ovim je omogućeno jedinstveno okruženje za ispitivanje, kojim se izbegavaju greške uslovljene upotrebom različitih okruženja programske podrške za svaki od jezika. Korišćenjem posebne prevodilačke direktive omogućena je izmena korišćenog skript interpretera u okviru programske podrške kontrolera. Komponente fizičke arhitekture korišćene u proceni prikazane su na Slici 32. Izmene u programskoj podršci kontrolera u cilju omogućavanja izbora skript interpretera prikazane su na Slici 33.



Slika 32. Arhitektura sistema korišćenog za procenu upotrebljivosti skript jezika



Slika 33. Moduli programske podrške korišćenog sistema sa izmenama potrebnim za procenu upotrebljivosti skript jezika

Korišćene su sledeće biblioteke skript interpretera: (1) Zvanična Lua biblioteka v5.1 (Lua Download, 2012); (2) Parser XML jezika sa TinyXML bibliotekom v2.6.1

(TinyXML, 2012); (3) Ambulant biblioteka za SMIL v2.2 (Ambulant Player, 2012). Primerni skriptovi korišćeni za procenu dati su u poglavlju 2.4.1.1.

Procenom su obuhvaćene sledeće karakteristike: (1) kompleksnost sintakse; (2) brzina izvršavanja, pokretanja i odziva skripta; (3) utrošak memorije; (4) primenljivost; (5) vreme potrebno za pisanje skripta i (6) mogućnost upotrebe u komercijalne svrhe.

Kompleksnost sintakse je računata kao kompleksnost strukture primernih skriptova. Ocenjene su tri različite kategorije: (1) ukupan broj linija koda; (2) broj poziva nestandardnih funkcija; (3) Halstead metrike kompleksnosti, uključujući dužinu, rečnik, volumen, težinu i vreme potrebno za pisanje skripta (Halstead, 1977). Ukupan broj linija koda određuje se na osnovu broja redova koji skript zauzima u okviru tekstualne datoteke. Broj poziva nestandardnih funkcija je u srazmeri sa brojem potrebnih proširenja jeziku kako bi se koristio za opis ponašanja sistema kućne automatike, i računa se prebrojavanjem poziva funkcija koje predstavljaju ta proširenja. Halstead metrike se računaju na osnovu sledećih elemenata koji se prebrojavaju u okviru skripta: broj različitih operatora (n_1), broj različitih operandada (n_2), ukupan broj operatora (N_1) i ukupan broj operandada (N_2). Nazivi funkcija i oznake jezika smatraju se operatorima, dok se svi parametri funkcija i atributi čvorova smatraju operandima. Dalje se računaju pojedinačni parametri: dužina skripta (N) i rečnika (n) (Jednačina 38); volumen (V) i težina pisanja skripta (D) (Jednačina 39), vreme potrebno za pisanje skripta (E , izraženo u sekundama E_t) (Jednačina 40).

$$N = N_1 + N_2, n = n_1 + n_2 \quad (38)$$

$$V = N \cdot \log_2 n, D = \frac{n_1}{2} \cdot \frac{N_2}{n_2} \quad (39)$$

$$E = D \cdot V, E_t = \frac{E}{18} \quad (40)$$

Brzina izvršavanja merena je uvidom u broj izvršenih ciklusa procesora u odgovarajućim tačkama interpretiranja skripta. Na odgovarajuća mesta umetnuti su pozivi funkcije `__rdtsc` za dobavljanje broja izvršenih ciklusa pre i nakon merene operacije. Za merenje brzine izvršavanja i odziva, postavljene su dve merne tačke: (1) neposredno pre objavljivanja MOTION_IN događaja od strane PIR senzora, nakon što je događaj registrovan (ispred poziva povratne funkcije za objavu događaja u okviru rukovaoca PIR sensorom); (2) neposredno nakon poziva naredbe

SET_STATE_PROBABILITY. Ovim je omogućeno da brzina zavisi isključivo od korišćenog skript jezika, čime se pokazuje koji skript je obavio najbržu reakciju na detekciju pokreta i poslao komandu za uključivanje svetla. Za merenje brzine pokretanja određene su sledeće merne tačke: (1) pre pokretanja interpretacije primernog skripta (izuzimajući operacije učitavanja datoteke); (2) neposredno nakon ulaska u obradu prve linije skripta. U slučaju Lua jezika, merna tačka 1 postavljena je neposredno pre poziva funkcije `lua_pcall` kojim se započinje interpretiranje skripta, dok je merna tačka 2 postavljena na prvu liniju implementacije funkcije `event_register`. U slučaju XML jezika, merna tačka 1 postavljena je neposredno nakon poziva metode `doc.loadFile`, dok je merna tačka 2 postavljena na prvu liniju obrađivača `driver` elementa, kojim se obavlja prijavljivanje na događaj definisan atributima. U slučaju SMIL jezika, merna tačka 1 postavljena je neposredno pre poziva metode `smilPlayer->start()`, dok je merna tačka 2 postavljena na prvu liniju obrađivača oznake `ref` (`custom_parser_handler::start()`). Pošto je uočeno da jedino Lua interpreter dostiže obe merne tačke u okviru jedne niti, ispitan je i slučaj gde je zadato interpretiranje Lua skripta iz posebne niti (merna tačka 1 postavljena neposredno pre odgovarajućeg `pthread_create` poziva).

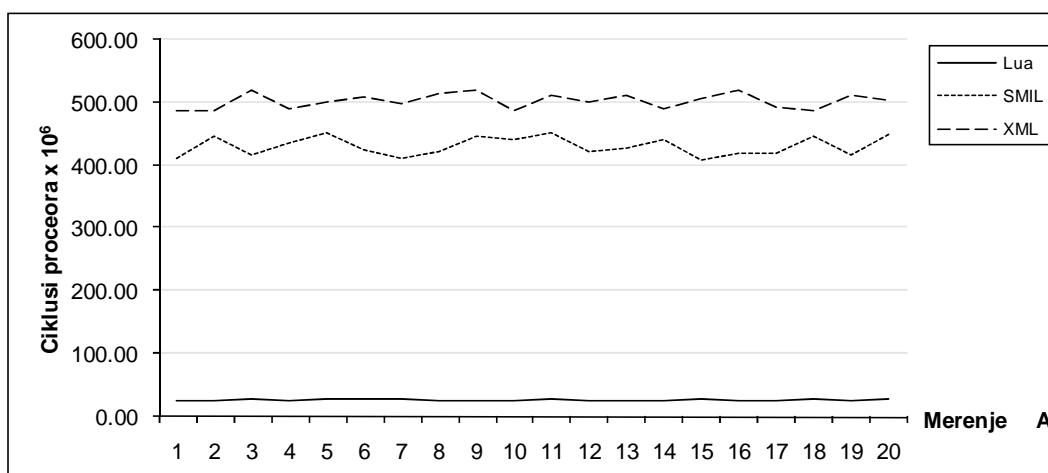
Primenljivost se izražava po zahtevima za primenu u kontekstualnoj platformi, u smislu odgovora na zahtev sa *podržava*, *ne podržava* ili *podržava uz izmene*. Dodatni skup izdvojenih zahteva je: (1) podrška za paralelni rad, u smislu omogućavanja startovanja paralelnih nizova naredbi, u slučaju da kontekstualna polisa aktivira niz određenih radnji; (2) proširivost, u smislu dodavanja novih naredbi svojstvenih kontekstualnim polisama; (3) podrška za vremenske događaje, odnosno, da li jezik podrazumevano omogućava pristup sistemskom datumu i vremenu; (4) strukturno programiranje, odnosno, da li je moguće programirati nizove naredbi i standardne konstrukte programskih jezika koji se sekvencijalno izvršavaju; (5) programiranje reakcije na događaj, odnosno, da li je moguće jednostavno programirati oblike uzrok-posledica. Dodatno, ispitan je ukupan utrošak memorije u Kb za svaki od interpretera.

Vreme potrebno za pisanje skripta dato je kao rezultat ankete pet inženjera sa iskustvom u pisanju skriptova u Lua, XML i SMIL jezicima. Postavljeno pitanje odnosi se na potrebno vreme izrade skripta za postavljeni problem. Odgovori su predstavljeni na Likert skali 1-10, gde 10 označava najveće potrebno vreme.

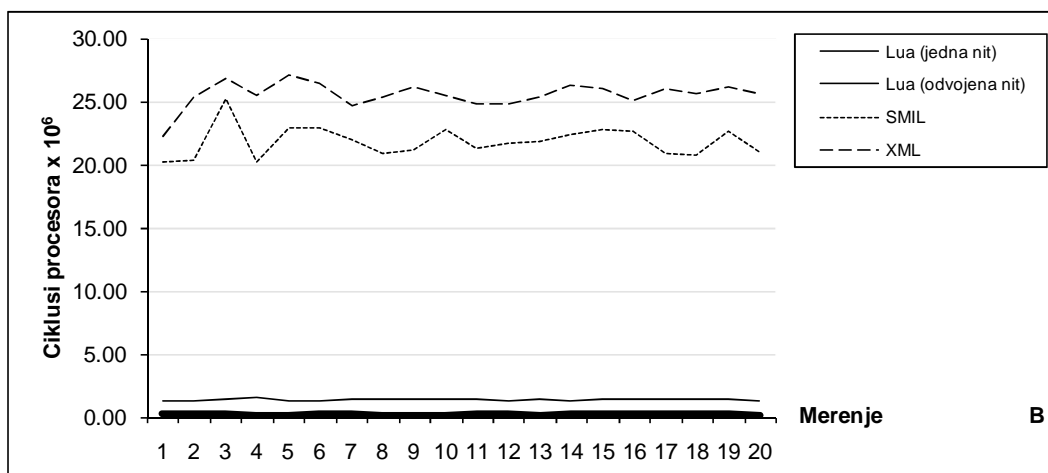
Algoritmi za otkrivanje konteksta

Mogućnost upotrebe u komercijalne svrhe odgovara na pitanje da li licenca interpretera dozvoljava komercijalizaciju rešenja u koje je interpreter uključen.

Poređenje brzine izvršavanja i odziva za jezike Lua, XML i SMIL dato je na Slici 34. Poređenje brzine pokretanja skripta dato je na Slici 35. Lua nadmašuje konkurente u oba merenja - čak i kada se namerno izvršava iz posebne niti ostvaruje deset puta bolji rezultat. SMIL i XML koriste sličan skup oznaka koje zahtevaju kreiranje stabla dokumenta pre parsiranja, što može biti delimično objašnjenje za sporiji odziv.



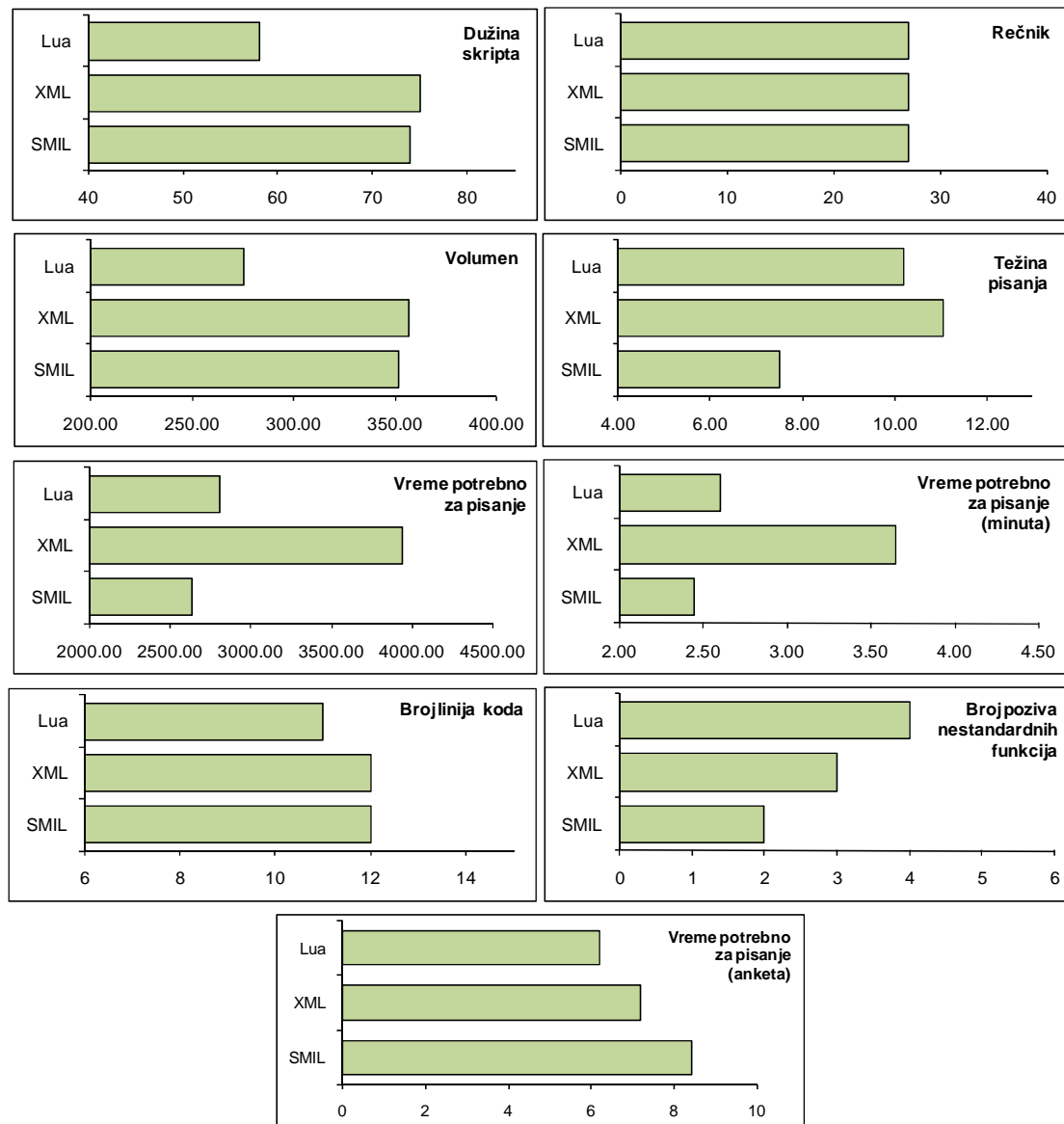
Slika 34. Brzina odziva na događaj (u ciklusima procesora)



Slika 35. Brzina pokretanja skripta (u ciklusima procesora)

Kompleksnost sintakse za različite parametre data je na Slici 36. Rezultati pokazuju da su XML i SMIL jezici duži i sintaksno zahtevniji. Ipak, SMIL je najmanje komplikovan za pisanje sa stanovišta logike. Ipak, programeri koji su navikli na

proceduralno programiranje daju prednost jeziku Lua, iz razloga što SMIL zahteva nešto drugačiju logiku razmišljanja koju najpre treba usvojiti. Važan parametar je i broj poziva nestandardnih funkcija – ovde je Lua najuopšteniji jezik koji zahteva najviše proširenja. Ipak, proširivanje Lua jezika je jednostavno zahvaljujući mehanizmu za proširivanje.



Slika 36. Kompleksnost sintakse za skripte pisane u jezicima Lua, XML i SMIL (prva tri reda – Halstead metrike)

Rezultati primenljivosti jezika u okviru kontekstualne platforme dati su u Tabeli 9. Ukoliko kontekstualna platforma predviđa mogućnosti definisanja nizova akcija direktno uz određivanje stanja okruženja, odnosno poseduje zahteve za proceduralnim

Algoritmi za otkrivanje konteksta

programiranjem i paralelnim radom, Lua jezik se nameće kao najbolji izbor. Za visoku jednostavnost realizacije i upotrebe, sa zahtevima programiranja reakcije na događaje, preporučuje se XML. SMIL ima najbolju vremensku sinhronizaciju i takođe omogućava programiranje reakcije na događaje.

	<i>Lua</i>	<i>XML</i>	<i>SMIL</i>
<i>Paralelni rad</i>	Uz izmene	Ne podržava	Ne podržava
<i>Proširivost</i>	Podržava	Uz izmene	Uz izmene
<i>Vremenski događaji</i>	Podržava	Uz izmene	Podržava
<i>Strukturalno programiranje</i>	Podržava	Uz izmene	Ne podržava
<i>Programiranje reakcije na događaj</i>	Uz izmene	Podržava	Podržava
Utrošak memorije	~143 Kb	~482 Kb	~724 Kb

Tabela 9. Primenljivost skript jezika u okviru kontekstualne platforme

S obzirom na prirodu kontekstualnih polisa navedenih u okviru algoritama u ovom poglavlju, Lua, XML i SMIL načelno zadovoljavaju postavljene kriterijume. Ključ odluke u izboru jezika leži u zahtevu za podrškom sekvencijalnom, proceduralnom programiranju, kao reakciji na određeni događaj ili zaključak, odnosno zadovoljenje pravila semantizacije. Naše mišljenje je da kontekstualna polisa nije pravo mesto za ovakve aktivacije, s obzirom na njenu generičku i izmenljivu prirodu. Umesto toga, može se predvideti poseban deo kontekstualne platforme, koji obavlja određenu aplikativnu akciju u zavisnosti od stanja okruženja. Takođe, kontekstualna polisa poseduje fiksni format i jednostavnu sintaksu – podešavanje ove polise više se svodi na popunjavanje njenih delova, nego na strukturalno programiranje. Ove činjenice ujednačavaju izbor i poništavaju prirodnu prednost Lua jezika, s obzirom na ciljnu primenu. Ipak, Lua jezik je znatno brži i znatno podesniji za upotrebu u realnom vremenu. S druge strane, vreme reakcije od najviše ~300ms koje je svojstveno XML i SMIL jezicima je često dovoljno za većinu kontekstualnih aplikacija. SMIL jezik je namenjen za pisanje vremenom vođenih događaja, i njegov format je donekle nepraktičan za realizaciju fiksnih kontekstualnih polisa, a troši i više memorije od druga dva jezika. Gore navedene činjenice i rezultati uporedne analize nameću zaključak da je upotreba XML jezika za definisanje kontekstualnih polisa najprirodnija. Dalje, upotreba Lua jezika se preporučuje za aplikativno programiranje u okviru kontekstualne platforme, te će Lua biti osnov aplikativnog nivoa kontekstualne

platforme. SMIL jezik gubi trku u odnosu na XML, te se smatra nedovoljno primenljivim za primene u kontekstualnoj platformi. Zaključak je, ipak, da SMIL jezik poseduje dovoljno osobenosti da se može iskoristiti u određenim kontekstualnim aplikacijama, koje, ipak, neće biti deo razmatranja u okviru ove disertacije.

2.4.1.3 Kontekstualne polise u XML jeziku

U ovom poglavlju biće naveden primer realizacije jedne jednostavne kontekstualne polise u jeziku XML, u skladu sa razmatranjima izloženim u poglavljima 2.1 i 2.2.

Kontekstualna polisa se sastoji od pravila semantizacije, na osnovu kojih se postavljaju verovatnoće stanja okruženja ukoliko prekidačka funkcija, koja je sastavni deo pravila, bude zadovoljena za ulazni skup kontekstualnih parametara. Ne uzimajući u obzir potpis, zaglavlje i ostale meta-podatke o kontekstualnoj polisi, telo kontekstualne polise ima sledeću sintaksu:

- Na 0-tom nivou hijerarhije nalaze se elementi pod nazivom *rule* koji definišu pravila semantizacije;
- Elementi imaju jedinstveni identifikator označen atributom *id*;
- Na 1-om nivou hijerarhije nalaze se elementi kojima se definiše određeno pravilo semantizacije, i to elementi prekidačke funkcije, i elementi podešavanja verovatnoće stanja okruženja;
- Elementi prekidačke funkcije obuhvataju element pod nazivom *check*, kojim se dobavljaju vrednosti kontekstualnih parametara od interesa, i proverava njihova vrednost u odnosu na zadati opseg, kao i elemente koji predstavljaju logičke operacije (*and*, *not*, *or*), čime se omogućuje realizacija prekidačke funkcije;
- Elementi podešavanja verovatnoće stanja okruženja pod nazivom *activity*, kojima se podešava verovatnoća jedne ili više aktivnosti u zavisnosti od zadovoljenja prekidačke funkcije.
- Dodatni elementi kojima se omogućuje kontrola i podešavanje, kao na primer kontrolni ispis *print*.

Primer jedne jednostavne kontekstualne polise definisane na gore navedeni način dat je na Slici 37.

Algoritmi za otkrivanje konteksta

```
<rule id="rule_0">
  <and>
    <check sensorID = "12" event = "1"
      within = "1000" param1 = "[-180:0]" />
    <check sensorID = "12" event = "1"
      within = "5000" param1 = "[0:180]" />
  </and>
  <activity id="1" probability="0.8" />
  <activity id="2" probability="0.2" />
</rule>
<rule id="rule_1">
  <check sensorID = "13" event = "1"
    within = "500" param1 = "[2:]" />
  <activity id="1" probability="0.6" />
  <activity id="2" probability="0.4" />
</rule>
<rule id="rule_2">
  <not>
    <check sensorID = "12" event = "2"
      within = "10000" param1 = "1" />
  </not>
  <activity id="2" probability="0.9" />
</rule>
<rule id="rule_3">
  <check sensorID = "10" event = "1"
    within = "1000"/>
  <activity id="3" pmodel="7" />
</rule>
```

Slika 37. Primer jedne jednostavne kontekstualne polise

Kontekstualna polisa sa Slike 37 se sastoji od četiri pravila semantizacije, od kojih svako zavisi od izlaza iz prekidačke funkcije. Ulazi u prekidačku funkciju su kontekstualni parametri sa izvora sa identifikatorom 12, koji odgovara nizu mikrofona sa nivoa mreže, i izvora 13, koji odgovara izlazu algoritama obrade slike dobijene od strane optičke kamere.

Prekidačka funkcija pravila semantizacije sa identifikatorom *rule_0* postaje zadovoljena u slučaju da kontekstualni parametar sa brojem 1 (dogadjaj, odnosno *event* 1), izvora 12 (mikrofonski niz), koji odgovara detektovanom uglu trenutnog govornika u odnosu na centar mikrofonskog niza, u dva diskretna trenutka u prošlosti (1000 ms pre obrade kontekstualne polise, odnosno 5000 ms pre obrade kontekstualne polise), ima vrednost koja je ispod, odnosno iznad nule. Dakle, prekidačka funkcija je zadovoljena ukoliko je došlo do nagle promene ugla govornika u nedavnoj prošlosti. U tom slučaju, kontekstualna polisa postavlja verovatnoću stanja okruženja sa identifikatorom 1, što odgovara stanju okruženja „sastanak je u toku“ na 0.8. Takođe,

verovatnoća stanja okruženja „nema aktivnosti u prostoriji“, sa identifikatorom 2, biva postavljena na vrednost 0.2.

Prekidačka funkcija pravila semantizacije sa identifikatorom *rule_1* postaje zadovoljena u slučaju da kontekstualni parametar sa brojem 1 (događaj, odnosno *event* 1), izvora 13 (optička kamera i algoritmi), koji odgovara prijavljenom broju osoba u prostoriji izračunatom na osnovu detekcije izmena u slici i segmentacije slike, ima vrednost veću ili jednaku broju 2. Dakle, prekidačka funkcija je zadovoljena ukoliko je u prostoriji prisutno 2 ili više ljudi. U tom slučaju, kontekstualna polisa postavlja verovatnoću stanja sa identifikatorom 1, što odgovara stanju okruženja „sastanak je u toku“ na 0.6. Takođe, verovatnoća stanja okruženja „nema aktivnosti u prostoriji“, sa identifikatorom 2, biva postavljena na vrednost 0.4.

Prekidačka funkcija pravila semantizacije sa identifikatorom *rule_2* postaje zadovoljena u slučaju da kontekstualni parametar sa brojem 2 (događaj, odnosno *event* 2), izvora 12 (mikrofonski niz), koji odgovara detekciji govora, nije bio postavljen na vrednost 1 u poslednjih 10000ms. Dakle, prekidačka funkcija je zadovoljena ukoliko u poslednjih 10 sekundi nije bilo detekcije govora u prostoriji. U tom slučaju, kontekstualna polisa postavlja verovatnoću stanja sa identifikatorom 2, što odgovara stanju okruženja „nema aktivnosti u prostoriji“, na 0.9.

Prekidačka funkcija pravila semantizacije sa identifikatorom *rule_3* postaje zadovoljena u slučaju da je vrednost kontekstualnog parametra sa brojem 1 (događaj, odnosno *event* 1), izvora 10 (sistemsko vreme), koji odgovara trenutnom vremenu u toku današnjeg dana (u sekundama), osvežena u poslednjih 1000ms. S obzirom na to da ovde nema provere parametara, prijem odgovarajućeg kontekstualnog parametra u zadatom vremenskom intervalu zadovoljiće prekidačku funkciju. U tom slučaju, kontekstualna polisa postavlja verovatnoću stanja sa identifikatorom 3, što odgovara stanju okruženja „korisnik posmatra TV program“ na vrednost dobijenu P-modelom sa identifikatorom 7 (što je P-model predstavljen na Slici 22). P-modelu su automatski prosleđene vrednosti onih kontekstualnih parametara koji su navedeni u okviru prekidačke funkcije, po redosledu navođenja (što je, u slučaju ove prekidačke funkcije, trenutna vrednost sistemskog vremena).

U narednom poglavlju izložena je detaljna formalna gramatika kontekstualne polise koja je predložena u okviru ove disertacije.

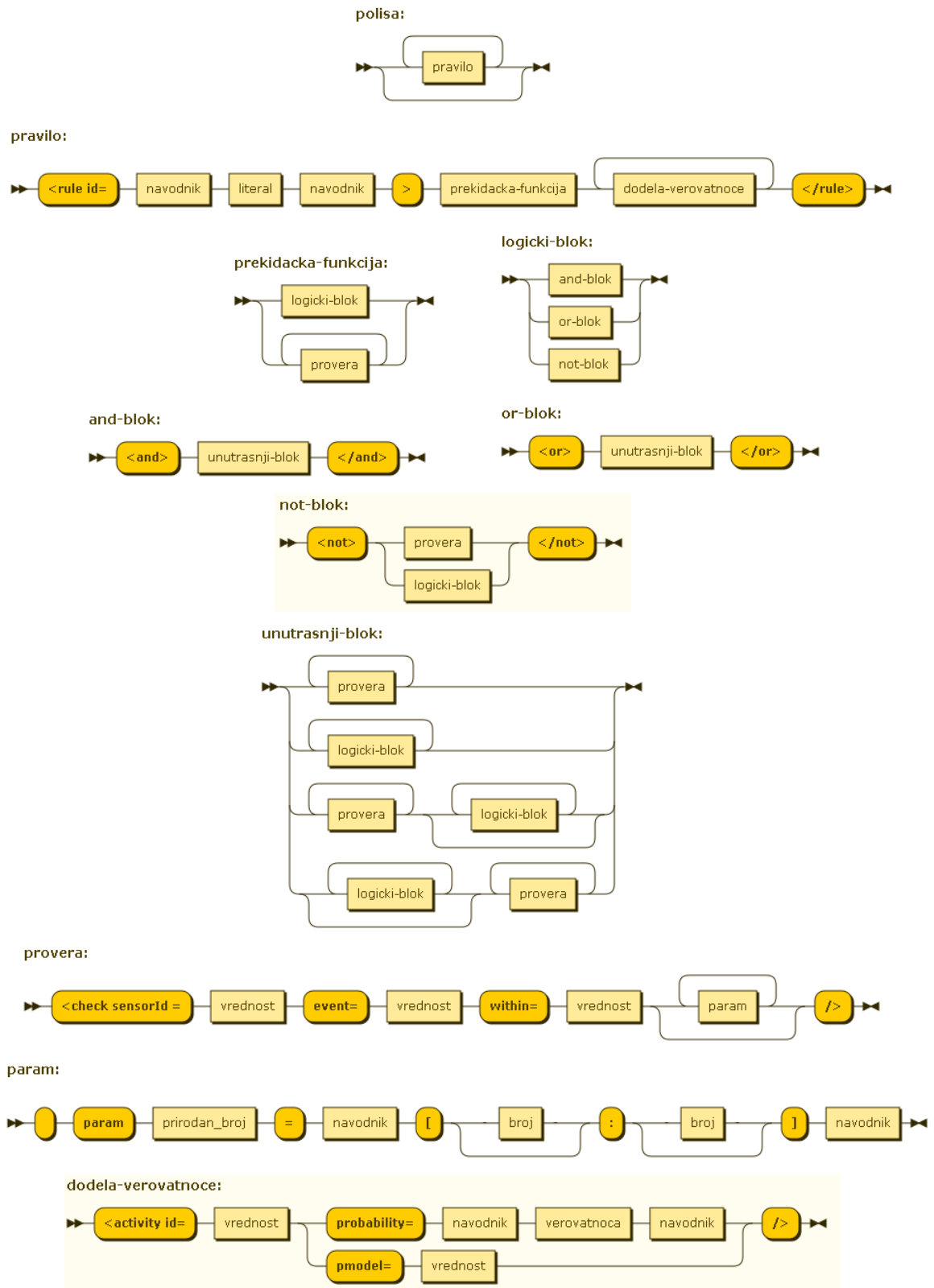
2.4.1.4 Gramatika kontekstualne polise

Gramatika kontekstualne polise navedena je korišćenjem proširene Bakus-Naurove notacije (*Extended Backus-Naur Form* - EBNF) (Slika 38).

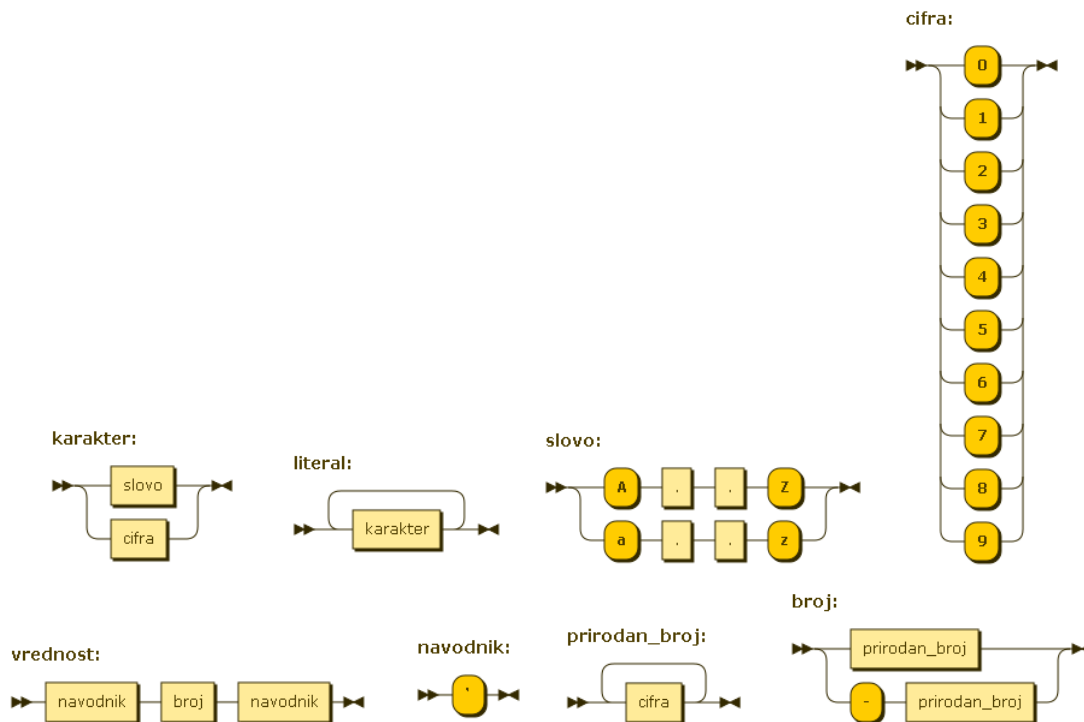
```
polisa ::= pravilo*
pravilo ::= "<rule id=" navodnik literal navodnik ">"
           prekidacka-funkcija dodela-verovatnoce+ "</rule>"
prekidacka-funkcija ::= logicki-blok | provera+
logicki-blok ::= and-blok | or-blok | not-blok
and-blok ::= "<and>" untrasnji-blok "</and>"
or-blok ::= "<or>" untrasnji-blok "</or>"
not-blok ::= "<not>" (provera | logicki-blok) "</not>"
untrasnji-blok ::= provera+ | logicki-blok+ |
                 (provera+ logicki-blok*) |
                 (logicki-blok* provera+)
provera ::= "<check sensorId =" vrednost " event=" vrednost "
           within=" vrednost param* "/>"
param ::= " " "param" prirodan_broj "=" navodnik "[" [broj] ":"
          [broj] "]" navodnik
dodela-verovatnoce ::= "<activity id=" vrednost ("probability="
                      navodnik verovatnooca navodnik) |
                      ("pmodel=" vrednost)) "/>"
literal ::= karakter+
karakter ::= slovo | cifra
slovo ::= "A" .. "Z" | "a" .. "z"
cifra ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
vrednost ::= navodnik broj navodnik
navodnik ::= " ` "
prirodan_broj ::= cifra+
broj ::= prirodan_broj | ("-" prirodan_broj)
```

Slika 38. Gramatika kontekstualne polise u EBNF notaciji

Sintaksni dijagram kontekstualne polise generisan je iz EBNF notacije korišćenjem alata *Railroad Diagram Generator* (Rademacher, 2012). Sintaksni dijagram za kontekstualnu polisu definisanu u okviru ove disertacije dat je na Slici 39 i Slici 40.



Slika 39. Sintaksni dijagram kontekstualne polise (1)



Slika 40. Sintaksni dijagram kontekstualne polise (2)

2.4.1.5 Semantika kontekstualne polise

Opis semantike kontekstualne polise dat je u Tabeli 10.

Naziv konstrukta		
rule	Atributi	id – identifikator pravila Ugnježđuju se prekidačka funkcija i dodele verovatnoće.
	Upotreba	Navodi se za enkapsulaciju jedne prekidačke funkcije. Identifikator pravila mora da bude jedinstven na nivou kontekstualne polise.

check	Atributi	<p><code>sensorID</code> – identifikator izvora kontekstualnih informacija (senzora)</p> <p><code>event</code> – identifikator kontekstualnog parametra izvora</p> <p><code>within</code> – vreme Δt u milisekundama, tako da vrednost kontekstualnog parametra važi u intervalu $[t-\Delta t, t]$, gde je t sadašnji trenutak.</p> <p><code>param1</code> – <code>paramN</code>: opcione vrednosti parametara koje se ispituju, u formatu $[a:b]$, gde su a i b minimalna, odnosno maksimalna vrednost kontekstualnog parametra za koju će provera biti zadovoljena. Ukoliko se neka od vrednosti ne navede, podrazumeva se $-\infty$, odnosno $+\infty$.</p>
	Upotreba	<p>Provera vrednosti kontekstualnog parametra sa traženog izvora. Ukoliko je u intervalu $[t-\Delta t, t]$, gde je t sadašnji trenutak, kontekstualni parameter bio osvežen i imao vrednost u intervalu $[a,b]$, provera se smatra tačnom. U suprotnom, provera je netačna. Ukoliko se parametri ne navedu, provera se smatra tačnom u slučaju da je kontekstualni parametar osvežen u intervalu $[t-\Delta t, t]$, bez obzira na njegovu vrednost. Nakon obavljanja provere zadate intervalne vrednosti kontekstualnih parametara, u slučaju da je prekidačka funkcija zadovoljena, prosleđuju se kao ulazi u evaluaciju P-modela, ukoliko je to specificirano, u redosledu u kojem su navedeni.</p>
and	Atributi	Argumenti su izrazi koji se ugnježđuju.
	Upotreba	Blok koji izračunava logičko “I” između svih ugnježđenih blokova, koji mogu biti provere check ili drugi logički blokovi and, or ili not. Ukoliko se svi ugnježđeni blokovi smatraju tačnim, i nadređeni and blok se smatra tačnim.

or	<i>Atributi</i>	Argumenti su izrazi koji se ugnježđuju.
	<i>Upotreba</i>	Blok koji izračunava logičko “ILI” između svih ugnježđenih blokova, koji mogu biti provere <i>check</i> ili drugi logički blokovi <i>and</i> , <i>or</i> ili <i>not</i> . Ukoliko se bar jedan ugnježđeni blok smatra tačnim, i nadređeni <i>or</i> blok se smatra tačnim.
not	<i>Atributi</i>	Argument je jedan i samo jedan izraz koji se ugnježđuje.
	<i>Upotreba</i>	Blok koji izračunava logičko “NE” nad ugnježđenim blokom, koji može biti provera <i>check</i> ili drugi logički blokovi <i>and</i> , <i>or</i> ili <i>not</i> . Ukoliko se ugnježđeni blok smatra tačnim, nadređeni <i>not</i> blok se smatra netačnim.
activity	<i>Atributi</i>	<i>id</i> – identifikator stanja okruženja <i>probability</i> – verovatnoća koja se dodeljuje za navedeno stanje okruženja <i>pmodel</i> – identifikator P-modela koji se alternativno koristi za dodeljivanje verovatnoće
	<i>Upotreba</i>	Ukoliko se svi blokovi istog nivoa hijerarhije, navedeni ispred blokova <i>activity</i> , smatraju tačnim, samo tada se ovaj blok evaluira. Verovatnoća stanja okruženja za određeni skup kontekstualnih parametara evaluiranih prekidačkom funkcijom, postavlja se kao izlaz iz ovog bloka. Alternativno, umesto atributa <i>probability</i> , može se navesti atribut <i>pmodel</i> , što označava da će stanje okruženja dobiti verovatnoću na osnovu P-modela, kome se prosleđuju vrednosti kontekstualnih parametara (sa zadatim intervalima) koji su zadati u svim proverama <i>check</i> , u redosledu navođenja.
print	<i>Atributi</i>	Ugnježđuje se tekst koji treba ispisati u log.
	<i>Upotreba</i>	Tekst koji treba ispisati u log.

Tabela 10. Opis semantike kontekstualne polise

2.4.1.6 *Fizička realizacija algoritama*

Pretpostavke za fizičku realizaciju algoritma za parsiranje kontekstualne polise i semantizaciju konteksta su sledeće:

- s obzirom na zahteve namenskih sistema i uređaja potrošačke elektronike, jezik odabran za optimalno izvršavanje algoritma je C++;
- algoritam za parsiranje kontekstualne polise i semantizaciju konteksta isporučuje se u obliku dinamičke biblioteke (.so) i datoteke sa zaglavljem (.h) namenjene izvršavanju u Linux okruženju, i u potpunosti je realizovan korišćenjem jezika C++;
- biblioteka za parsiranje kontekstualnih polisa u XML jeziku, zasnovanih na gramatici predstavljenoj u 2.4.1.4 i s obzirom na jednostavnost XML sintakse, je TinyXML (TinyXML, 2012);
- parser kontekstualne polise je realizovan kao poseban programski modul;
- realizovan je modul za prihvatanje i izvršavanje kontekstualnih polisa;
- realizovan je modul za dobavljanje kontekstualnih parametara, koji se oslanja na usluge srednjeg sloja kontekstualne platforme;
- realizovan je modul za računanje vrednosti prekidačke funkcije;
- realizovan je modul za ažuriranje verovatnoće stanja okruženja;
- realizovan je modul za računanje verovatnoće na osnovu traženog P-modela;
- realizovan je API na nižem nivou, za dobavljanje kontekstualnih parametara od strane srednjeg sloja;
- realizovan je API na višem nivou, za objavljivanje verovatnoće traženog stanja okruženja i za prijem deklaracije stanja / unosa cene asinhronog zahteva iz korisničke sprege / prilaganje kontekstualnih polisa / konfiguraciju;
- realizovan je modul za automatsko ažuriranje cene asinhronog zahteva za stanje okruženja;
- realizovan je modul za automatsku semantizaciju konteksta;
- realizovan je modul za kontrolu rada biblioteke za parsiranje kontekstualnih polisa i semantizaciju konteksta, što obuhvata postavljanje režima rada i konfigurisanje biblioteke;

Algoritmi za otkrivanje konteksta

Više detalja o realizaciji programske podrške za parsiranje kontekstualnih polisa i semantizaciju konteksta dato je u okviru pregleda arhitekture programske podrške kontekstualne platforme, u poglavlju 3.3.2.

POGLAVLJE 3.

MODEL ARHITEKTURE PLATFORME

U okviru ovog poglavlja biće izložen predloženi model arhitekture kontekstualne platforme, uključujući osnovne smernice za izradu fizičke arhitekture i detaljnu arhitekturu programske podrške platforme. Najpre će biti definisani zahtevi i slučajevi korišćenja predviđeni za jednu opštenamensku kontekstualnu platformu. S obzirom na cilj da platforma bude opštenamenska, odnosno primenljiva na veći broj različitih, unapred nepredvidljivih slučajeva upotrebe, dati su samo zahtevi i slučajevi upotrebe visokog nivoa apstrakcije. Dalje, biće dat prikaz visokog nivoa arhitekture, koji predstavlja osnovni koncept kontekstualne platforme uključujući njenu interakciju sa okolinom. Nakon toga daju se detalji fizičke arhitekture, uključujući sve potrebne blokove za današnju realizaciju kontekstualne platforme. Konačno, detaljno se izlaže arhitektura programske podrške i opis svih potrebnih modula za realizaciju predviđenih funkcionalnosti.

3.1 ANALIZA ZAHTEVA ZA KONTEKSTUALNU PLATFORMU

Cilj ovog poglavlja je da se definišu polazni zahtevi visokog nivoa na osnovu kojih se realizuje arhitektura kontekstualne platforme. Navedeni zahtevi dobijeni su analizom stanja u oblasti kontekstualnih platformi, analizom oblasti potrošačke elektronike, kontekstualnih korisničkih sprega i problema asinhronih zahteva, kao i uzimanjem u obzir zahteva koje se mogu razumeti kao „dobra praksa“ na osnovu

iskustva autora disertacije tokom razvoja prototipa kontekstualne platforme. Pre definisanja funkcionalnih zahteva, navedene su dve grupe opštih zahteva. Prva grupa uključuje zahteve kontekstualnih platformi po Dey, koji se uzimaju kao polazna osnova u akademskom smislu (Dey, 2001). Druga grupa obuhvata skup kriterijuma za odlučivanje u smislu prihvatanja, odnosno ne prihvatanja uređaja sa kontekstualnim funkcijama od strane korisnika. Definisanje heuristika u tom smislu, te uvid u te heuristike pre izlistavanja funkcionalnih zahteva je ključan u fazi analize zahteva, s obzirom na jasan kriterijum rešenja u oblasti potrošačke elektronike – prihvatanja od strane korisnika.

3.1.1 Zahtevi kontekstualne platforme po Dey

Dey definiše sledećih šest opštih zahteva za realizaciju kontekstualnih platformi, koje treba inkorporirati u dalju razradu zahteva (Dey, 2001):

- Razdvojenost fizičkog nivoa (senzora) od nivoa semantizacije (*separation of concerns*);
- Mogućnost interpretacije konteksta na višem nivou semantike (*context interpretation*);
- Mogućnost transparentne i distribuirane komunikacije sa sensorima (*transparent, distributed communications*);
- Trajno prikupljanje parametara konteksta, nezavisno od pokretanja aplikacija (*constant availability of context acquisition*);
- Mogućnost skladištenja kontekstualnih parametara i uvid u istorijat (*context storage and history*);
- Mogućnost pretraživanja uređaja i servisa (*resource discovery*).

3.1.2 Heuristike za prihvatanje rešenja od strane korisnika

Definišu se sledeće heuristike za prihvatanje rešenja od strane korisnika, koje moraju da budu zadovoljene i iskazane kroz kasnije funkcionalne zahteve. Ove heuristike mogu se smatrati kao „deset zapovesti“ pri razvoju uređaja potrošačke elektronike sa kontekstualnim funkcijama, u smislu obezbeđivanja prihvatanja uređaja od strane korisnika.

HEUR1: *Usklađenost sa aktivnostima korisnika.* Novo rešenje u oblasti potrošačke elektronike sa kontekstualnim funkcijama, koje pretenduje na resurs

vremena korisnika, mora svoje funkcionisanje uskladiti sa dotadašnjim aktivnostima, odnosno navikama korisnika. Stoga, pristup funkcionalnostima mora biti omogućen kroz minimalan skup koraka, odnosno, funkcionalnost treba pravovremeno ponuditi korisniku, tako da sklad ostalih aktivnosti nije narušen. Na primer, ukoliko se digitalni fotoaparati opremi kontekstualnim funkcijama koje omogućavaju korisniku da napravi najbolju fotografiju na osnovu lokacije, doba dana i uslova osvetljenja, onda te funkcionalnosti ne smeju onemogućiti korisnika da fotografisanje obavi onom brzinom kojom je to do tada činio.

HEUR2: *Vlasništvo*. Ovaj aspekt odgovara na pitanje da li je uređaj, odnosno aplikacija, namenjena jednom korisniku ili je deljeni resurs između više korisnika. Potrebno je izbegavati konflikt između više korisnika u ovom pogledu.

HEUR3: *Privatnost*. Svi podaci prikupljeni od strane kontekstualne platforme moraju biti maksimalno sigurni, ili najbolje, ne smeju se čuvati u okviru platforme osim u minimalnom obimu potrebnom za algoritme semantizacije. Jedan od glavnih razloga neprihvatanja uređaja ili servisa je usled percepcije korisnika da im je ugrožena privatnost. Korisnici ne smeju imati osećaj da uređaj, sistem ili njegovi delovi imaju tehničku mogućnost da narušavaju privatnost korisnika prikupljanjem podataka iz okruženja, pa čak i u slučajevima kada oni to zaista mogu. Takođe, ukoliko uređaj deli više korisnika, privatni podaci moraju biti maksimalno sigurni.

HEUR4: *Osećaj kontrole*. Korisnik u svakoj situaciji mora da ima osećaj da je uređaj, odnosno sistem sa kontekstualnim funkcijama u potpunosti pod njegovom kontrolom. U tom smislu mora se praviti balans između zahteva za korisničkom interakcijom sa sistemom, kada se postojanje sistema otkriva, i automatskih funkcionalnosti. Jedan od najboljih načina da se osećaj kontrole uspostavi je kroz dozvoljavanje korisniku da odabira režim rada. Ovako korisnik ima mogućnost da postavi željeni režim, koji nadalje funkcioniše automatski. Da bi se dodatno pospešio osećaj kontrole, potrebno je omogućiti jednostavnu izmenu režima rada, za one korisnike kojima je to značajno.

HEUR5: *Poverenje*. Korisnik mora da ima poverenja u funkcionisanje uređaja ili sistema, pogotovo u situacijama kada mu poverava podatke. Poverenje u stabilnost i robustnost uređaja su takođe od izuzetnog značaja. Poverenje se izgrađuje u kritičnim slučajevima kada je uređaj ili sistem uspeo da ispuni očekivanja, npr. baterija

Model arhitekture platforme

mobilnog uređaja je izdržala dovoljno dugo da korisnik obavi važnu operaciju, pristup podacima je omogućen na pravovremen način, skladištenje podataka je sigurno i bezbedno, i ne može se dogoditi brisanje usled starenja medijuma bez prethodnog upozorenja korisnika i sl.

HEUR6: *Osećaj poznavanja.* Vreme potrebno da se korisnici upoznaju sa načinom funkcionisanja uređaja ili sistema treba da bude minimalno, a najčešće veoma blizu nultog. Potrebno je da koncepti funkcionisanja budu bliski prethodnom iskustvu korisnika. Revolucionarne, odnosno skokovite izmene potrebno je izbegavati. Preporučuje se maksimalna količina izmena između verzija od 30%.

HEUR7: *Korisnička sprega zasnovana na iskustvu.* Uređaj ili sistem sa kontekstualnim funkcijama treba da predvidi korisničku spregu koja se koristi na način blizak opštem životnom iskustvu korisnika, odnosno njegovim prirodnim biološkim svojstvima. Iz ovog razloga sve je popularnija interakcija korišćenjem prirodnih gestova. Takođe treba izbegavati notifikacije i upite tada kada je namera korisnika očigledna, odnosno treba predložiti izmenu režima rada onda kada to korisnik očekuje. Opšte smernice vezano za ovaj heuristički zahtev odnose se i na sledeće konkretne osobine korisničke sprege:

- Grafički elementi treba da budu podesni za uređaj na koji su primenjeni. Ovo znači da koncepti grafičke korisničke sprege treba da budu u duhu sličnih koncepata.
- Vreme prilagođavanja korisničkoj sprezi treba da bude minimalno, s obzirom na izuzetno široku ciljnu grupu korisnika.
- Unos podataka mora biti prilagođen fizičkom ulaznom uređaju, npr. ukoliko se koristi numerička tastatura, navigaciju je poželjno obavljati strelicama, tok se unos teksta može obavljati korišćenjem *T9* ili *multi-tap* načina unošenja teksta.
- Grafička korisnička sprega treba da bude vidljiva sa aspekta prosečne udaljenosti korisnika pri korišćenju grafičke sprege.
- Korisnička sprega mora biti u skladu sa prirodom uređaja, odnosno njegovom namenom. Ukoliko je uređaj namenjen za pasivnu zabavu, poput TV prijemnika ili multimedijalnog plejera, korisničke kontrole treba da budu maksimalno pojednostavljene. Stoga je veoma opasno preslikavati koncepte

grafičke korisničke sprege iz sveta PC računara i PC aplikacija, u svet potrošačke elektronike, s obzirom na često potpuno drugačiju prirodu korišćenja uređaja.

HEUR8: *Doživljaj upotrebljivosti.* Važniji aspekt od stvarne upotrebljivosti, robustnosti i funkcionalnosti uređaja ili sistema je doživljaj upotrebljivosti od strane korisnika. Uređaj ili sistem treba da za korisnika realizuju takvo okruženje, u kome je dojam funkcionalnosti besprekoran u situacijama kada pozadinski procesi ne postižu maksimalnu performansu. Na primer, izuzetno je važan pažljiv dizajn grafičke korisničke sprege, te obavljanje asinhronih operacija u pozadini, koje bi trebalo obavljati tako da korisnik bude malo ili ni malo svestan dužine njihovog trajanja, što se najčešće postiže u dovoljno kvalitetnoj meri omogućavanjem brzog odziva korisničke sprege u svim situacijama.

HEUR9: *Nevidljiva integracija u okruženje.* Potrebno je omogućiti da posle izvesnog vremena korisnik zaboravi na postojanje uređaja ili sistema, odnosno da fokus bude premešten sa uređaja ili sistema, na postignuća koja se obavljaju njegovim korišćenjem. Ta postignuća, vremenom, postaju sastavni deo života korisnika.

HEUR10: *Veliki izbor servisa, proširivost i dobra korisnička podrška.* Iako je jedan od ciljeva razvoja uređaja potrošačke elektronike brza prodaja nove, ažurirane verzije, u skorije vreme podjednako važno za prihvatanje od strane korisnika je mogućnost izbora servisa i proširivost mogućnosti uređaja bez potrebe za fizičkim menjanjem uređaja. Na primer, koncept aplikacija koje se preuzimaju posredstvom interneta omogućavaju korisniku da proširi skup servisa koji su mu na raspolaganju, te da na taj način delom osavremeni uređaj koji koristi. Takođe, dobra korisnička podrška, u situacijama kada neki od heurističkih zahteva jednostavno ne mogu da budu zadovoljeni, može da spase situaciju i da korisnik nastavi da koristi uređaj. Kvalitet usluge koja se pruža korisniku a koja se odnosi na tehničku podršku, dokumentaciju i mogućnost besplatne popravke su, iako među najstarijim, i dalje među najefikasnijim metodama zadržavanja korisnika.

3.1.3 Funkcionalni zahtevi

U ovom poglavlju navedeni su funkcionalni zahtevi koji se moraju poštovati pri razvoju kontekstualne platforme sa mogućnošću primene u uređajima potrošačke elektronike.

3.1.3.1 Tehnički zahtevi

- Z1** Kontekstualna platforma se sastoji od fizičke arhitekture koja uključuje procesor na kome se izvršava Linux operativni sistem.
- Z2** Kontekstualna platforma koristi skup Linux rukovalaca za ostvarivanje veze sa okruženjem.
- Z3** Funkcionalnost kontekstualne platforme realizuje se u obliku Linux aplikacije u korisničkom prostoru (*user space*).
- Z4** U sklopu realizacije kontekstualne platforme, za potrebe korišćenja mogućnosti operativnog sistema, koriste se *bash* skriptovi za konfiguraciju i *daemon* procesi za realizaciju mehanizma zaštite (*watchdog*).
- Z5** U slučaju manjeg otkaza programske podrške moguć je oporavak sistema ponovnim pokretanjem aplikacije, odnosno ponovnim pokretanjem uređaja, ukoliko to ne narušava osnovnu funkcionalnost uređaja.
- Z6** Moguće je ažuriranje verzije sistema korišćenjem paketa sa novom verzijom koja koristi jednostavnu transparentnu metodu ažuriranja bez angažovanja korisnika.
- Z7** Nakon integracije kontekstualne platforme u uređaj kome to nije primarna funkcija, performanse uređaja nisu vidno narušene.
- Z8** Nakon integracije kontekstualne platforme u uređaj kome to nije primarna funkcija, vreme trajanja uključivanja uređaja u rad (*boot*) nije značajno povećano.
- Z9** Nakon integracije kontekstualne platforme u uređaj kome to nije primarna funkcija, kontekstualna platforma ne zauzima ekskluzivno fizičke resurse uređaja koji u nekom trenutku mogu biti zahtevane od strane uređaja.
- Z10** Utrošak memorije kontekstualne platforme koja se izvršava kao samostalna aplikacija ili u paraleli sa programskom podrškom osnovnog uređaja, nikada ne premašuje vrednost dobijenu zbirom najveće moguće količine utrošene memorije od strane uređaja i najveće moguće količine utrošene memorije od strane kontekstualne platforme. Ukoliko je platforma opštenamenska te se maksimalna potreba za memorijskim resursima ne može predvideti, obaviti ispitivanje memorije pojedinačno sa svakom od 10 najčešće korišćenih aplikacija u trenutku realizacije platforme.

- Z11** Opterećenje procesora kontekstualne platforme koja se izvršava kao samostalna aplikacija ili u paraleli sa programskom podrškom osnovnog uređaja, nikada ne premašuje vrednost dobijenu zbirom najvećeg mogućeg opterećenja procesora od strane uređaja i najvećeg mogućeg opterećenja procesora od strane kontekstualne platforme. Ukoliko je platforma opštenamenska te se maksimalna potreba za procesorom ne može predvideti, obaviti ispitivanje opterećenja procesora pojedinačno sa svakom od 10 najčešće korišćenih aplikacija u trenutku realizacije platforme.
- Z12** Programska podrška kontekstualne platforme mora da omogući zaštite od dostizanja graničnih vrednosti u pogledu memorijskih resursa pri učitavanju kontekstualnih polisa, zauzetosti Flash memorije pri smeštanju logova (logove je potrebno rotirati), broja maksimalnih konkurentnih objava izmene vrednosti kontekstualnih parametara (uvesti redove čekanja) i sl.
- Z13** Mehanizmi upisa u Flash memoriju moraju obezbediti štednju broja ciklusa pisanja da se značajno ne ugrozi vek trajanja Flash memorijskih modula. Obaviti agregaciju svih zahteva za upisom u radnu memoriju, te u specifičnim intervalima obaviti upisivanje u Flash u jednom upisnom ciklusu.
- Z14** Programska podrška mora da omogući mehanizme podešavanja vremena ažuriranja kontekstualnih parametara, kao i da omogući režime na osnovu kojih se pojedini moduli mogu aktivirati/deaktivirati, u cilju omogućavanja štednje baterije na uređajima sa baterijskim napajanjem.
- Z15** Programska podrška kontekstualne platforme treba da podrži mogućnost zadavanja prioriteta konkurentnim procesima i/ili nitima.
- Z16** Kontekstualna platforma treba da obezbedi mogućnost neprekidnog rada bez isključivanja. Ukoliko je potreban oporavak i/ili automatsko ažuriranje, potrebno je rasporediti takve radnje na intervale u kojima korisnik nije prisutan (npr. noću).

3.1.3.2 Semantizacija konteksta

- Z17** Kontekstualna platforma omogućava obavljanje semantizacije konteksta na osnovu priloženih kontekstualnih polisa, na osnovu mehanizma navedenog u poglavlju 2.1.1.

Model arhitekture platforme

- Z18** Kontekstualna platforma omogućava korišćenje jedne kontekstualne polise za jedno inteligentno okruženje.
- Z19** Kontekstualna platforma omogućava izmenu kontekstualnih polisa, pokretanje i zaustavljanje obrade kontekstualne polise.
- Z20** Kontekstualna platforma omogućava obavljanje automatske semantizacije konteksta na osnovu mehanizma navedenog u poglavlju 2.1.2.
- Z21** Kontekstualna platforma omogućava deklaraciju stanja okruženja korišćenjem API-ja koji se može pozvati sa uređaja koji je u blizini korisnika, kao npr. aplikacije na mobilnom telefonu.
- Z22** Kontekstualna platforma omogućava izračunavanje stanja zauzetosti korisnika na osnovu mehanizma navedenog u poglavlju 2.3.
- Z23** Kontekstualna platforma omogućava definisanje P-modela, njihov uvoz u obliku datoteka i korišćenje na način opisan u poglavlju 2.2.1.
- Z24** Verovatnoće svakog od registrovanih stanja svakog od registrovanih okruženja mogu se dobiti korišćenjem odgovarajućeg API-ja.
- Z25** Zauzetost korisnika za svako od registrovanih okruženja može se dobiti korišćenjem odgovarajućeg API-ja.
- Z26** Skup modula za semantizaciju konteksta obezbeđuje API putem kojeg se može registrovati za događaj, poput izmene verovatnoće traženog stanja okruženja, kao i izmene zauzetosti korisnika za traženo okruženje.
- Z27** Kontekstualna platforma mora da omogući da moduli za semantizaciju konteksta imaju pristup događajima sa senzora i kontekstualnim parametrima, i to posredstvom najbrže moguće sprege (npr. korišćenjem povratnih funkcija).
- Z28** Kontekstualna platforma obezbeđuje API za deklaraciju stanja odnosno objavu cene asinhronog zahteva od strane korisnika.
- Z29** Kontekstualna platforma treba da podrži mogućnost definisanja stanja okruženja u odnosu na ciljnog korisnika u okruženju, ukoliko je identifikacija korisnika moguća.

3.1.3.3 Podrška za senzore

- Z30** Kontekstualna platforma obezbeđuje mehanizam za integraciju fizičkih, virtuelnih i logičkih senzora u vidu dodatnih biblioteka (*plug in*). Ovim se obezbeđuje proširivost kontekstualne platforme.

- Z31** Kontekstualna platforma obezbeđuje API kojim je moguće na generički način dostaviti događaj sa senzora treće strane, i u slučaju da za njega ne postoji registrovana biblioteka u okviru kontekstualne platforme.
- Z32** Senzorske biblioteke su spregnute sa ostalim modulima kontekstualne platforme posredstvom definisane programske sprege.
- Z33** Senzorske biblioteke koriste programsku spregu ka ostatku kontekstualne platforme, da dobave spisak svih ostalih dostupnih senzora.
- Z34** Senzorske biblioteke koriste programsku spregu ka ostatku kontekstualne platforme, da obavljaju registraciju za događaj na nekom od ostalih dostupnih senzora.
- Z35** Senzorske biblioteke se mogu registrovati na link za prijem podataka, posredstvom kojeg mogu da primaju veću količinu podataka od nekih drugih senzora u realnom vremenu.
- Z36** Senzorske biblioteke koriste programsku spregu ka ostatku kontekstualne platforme, da objave određeni događaj sa opcionim navođenjem parametara koji prate taj događaj i njihovih vrednosti.
- Z37** Senzorske biblioteke omogućavaju upravljanje višestrukim instancama senzora koji su preko njih povezani sa kontekstualnom platformom, na način da svaka instanca može biti jedinstveno identifikovana.
- Z38** Senzorske biblioteke obezbeđuju podešljivost svih relevantnih parametara za svoj rad, kao i za rad fizičkih senzora ako njima upravljaju, posredstvom programske sprege iz kontekstualne platforme.
- Z39** Za obavljanje perzistencije podataka, komunikacije sa sensorima i drugih sistemskih poziva, senzorske biblioteke obavezno koriste odgovarajuće mehanizme dostupne u okviru kontekstualne platforme, ukoliko takvi mehanizmi postoje.

3.1.3.4 Prvobitno i naknadno podešavanje

- Z40** Kontekstualna platforma obezbeđuje mehanizam za izvoz i uvoz svih podešavanja, koje uključuju odabrane režime rada, kontekstualne polise, P-modele, aktivne senzorske biblioteke i sl.
- Z41** U slučaju samostalne kontekstualne platforme namenjene za instalaciju u inteligentnom okruženju, potrebno je obezbediti mehanizme za kalibraciju

Model arhitekture platforme

svakog priključenog senzora. Takođe je potrebno obezbediti aplikacije za unos stanja okruženja i za objavu cene asinhronog zahteva koje se mogu izvršavati na uređajima u blizini korisnika (npr. mobilni telefon).

Z42 Kontekstualna platforma obezbeđuje API za naknadno podešavanje parametara rada platforme, i relevantnih parametara rada svih senzora.

3.1.3.5 Korisnička sprega

Z43 Kontekstualna platforma poseduje sistemsku korisničku spregu, posredstvom koje se omogućava uvid u stanje rada platforme, sva podešavanja, aktivne i dostupne kontekstualne polise i njihova izmena/dodavanje, aktivni i dostupni P-modeli i njihova izmena/dodavanje.

Z44 Sistemskoj korisničkoj sprezi se može pristupiti sa velikog broja uređaja (npr. sistemsku korisničku spregu je realizovana kao web aplikacija koju poslužuje web server instaliran u okviru platforme).

Z45 Kontekstualna platforma obezbeđuje module za integraciju sa korisničkim spregama ciljnih aplikacija.

Z46 Kontekstualna platforma obezbeđuje module posredstvom koji se omogućava posredna interakcija sa korisnikom, i to senzorske module koji su u stanju da otkriju poziciju korisnika u prostoriji, njegovo stanje kretanja, govornu aktivnost, poziciju i oblik lica, i sl.

Z47 Kontekstualna platforma obezbeđuje mehanizme za objavljivanje zauzetosti korisnika u traženom okruženju, posredstvom API funkcija platforme, kao i posredstvom drugih situacija (npr. aktiviranjem svetlosnih efekata).

Z48 Sistemsku korisničku spregu i aplikativne korisničke sprege koriste informacije o zauzetosti korisnika da bi podesile sopstvene režime rada i smanjile intruzivnu interaktivnost.

Z49 Kontekstualna platforma omogućuje deaktivaciju kompletnog sistema jednom korisničkom akcijom (poželjno i posredstvom fizičkog tastera/prekidača, ukoliko instalacija kontekstualne platforme to omogućava).

Z50 U okviru korisničke sprege koja se izvršava na uređaju sa većim ekranom, kao što je npr. PC računar ili TV prijemnik, ne treba prikazivati privatne korisničke podatke, ukoliko se to može izbeći. Ukoliko se to ne može izbeći, osetljive

režime rada treba jasno označiti posebnim dizajnom korisničke sprege, kao i odgovarajućim upozorenjem korisniku pri ulasku u takve režime.

- Z51** Izgled korisničke sprege treba da bude prilagođen veličini ekrana, prosečnoj udaljenosti korisnika od ekrana, i gustini piksela na ekranu na kome se prikazuje. Ukoliko se koristi ekran osetljiv na dodir, veličina tastera uvek treba da bude dovoljna za komotnu aktivaciju pritiskom. Obezbediti više prikaza korisničke sprege za kritične ekrane.

3.1.3.6 Podrška za aktuatore

- Z52** Kontekstualna platforma treba da omogući mehanizam izdavanja naredbi aktuatorima u zavisnosti od stanja okruženja, zauzetosti korisnika ili na zahtev aplikacije.
- Z53** Kontekstualna platforma omogućava integraciju novih podržanih aktuatora u vidu aktuatorskih biblioteka (*plug in*).
- Z54** Aktuatorске biblioteke su spregnute sa ostalim modulima kontekstualne platforme posredstvom definisane programske sprege.
- Z55** Aktuatorске biblioteke izvršavaju naredbe dobijene posredstvom programske sprege od strane kontekstualne platforme.
- Z56** Aktuatorске biblioteke omogućavaju upravljanje višestrukim instancama aktuatora koji su preko njih povezani sa kontekstualnom platformom, na način da svaka instanca može biti jedinstveno identifikovana.
- Z57** Aktuatorске biblioteke obezbeđuju podešljivost svih relevantnih parametara za svoj rad, kao i za rad fizičkih aktuatora kojima upravljaju, posredstvom programske sprege iz kontekstualne platforme.
- Z58** Za obavljanje perzistencije podataka, komunikacije sa aktuatorskim uređajima i drugih sistemskih poziva, aktuatorске biblioteke obavezno koriste odgovarajuće mehanizme dostupne u okviru kontekstualne platforme, ukoliko takvi mehanizmi postoje.
- Z59** Kontekstualna platforma mora obavezno da podrži upravljanje izvorima svetla u boji posredstvom nekog od odgovarajućih protokola (npr. DMX), zbog realizacije funkcionalnosti navedene u 1.6.2.

3.1.3.7 Podrška za aplikacije

- Z60** Kontekstualna platforma omogućava izvršavanje korisničkih aplikacija u sopstvenom kontekstu.
- Z61** Kontekstualna platforma omogućava vezu sa korisničkim aplikacijama koje se izvršavaju van konteksta platforme, korišćenjem odgovarajućeg API-ja.
- Z62** Logika potrebna za rad kontekstualne aplikacije koja se izvršava u kontekstu platforme (dalje: ugrađena kontekstualna aplikacija), realizuje se korišćenjem skript jezika Lua.
- Z63** Ugrađene kontekstualne aplikacije imaju mogućnost da korišćenjem predviđenih Lua ekstenzija, u vidu poziva funkcija, menjaju stanje grafičke korisničke sprege, pristupaju događajima sa senzora, izdaju naredbe aktuatorima i pristupaju API funkcijama srednjeg sloja kontekstualne platforme, koje uključuju informacije o trenutnom stanju okruženja i zauzetosti korisnika.
- Z64** Korisnička sprega ugrađenih kontekstualnih aplikacija, ukoliko je potrebna, realizuje se na proizvoljan način (Flash aplikacija, web aplikacija, aplikacija za mobilni telefon).
- Z65** Korisnička sprega ugrađenih kontekstualnih aplikacija prijavljuje se kao klijent na kontekstualnu platformu posredstvom neke od mrežnih sprege, i obavlja upite na osnovu kojih saznaje da li je potrebna promena njene grafičke prezentacije. Takođe, korisničke akcije u okviru korisničke sprege objavljuju se kontekstualnoj aplikaciji, odnosno Lua skriptu koji se izvršava, korišćenjem istog klijent-server mehanizma. Ugrađene kontekstualne aplikacije se registruju za željeni događaj iz korisničke sprege u okviru Lua skripta, korišćenjem odgovarajuće Lua funkcije.
- Z66** Kontekstualna platforma omogućava pokretanje više aplikacija paralelno.
- Z67** Aplikacije je moguće dodati, obrisati, pokrenuti i zaustaviti korišćenjem sistemske korisničke sprege kontekstualne platforme.
- Z68** Aplikacije su opremljene odgovarajućim funkcijama obrađivačima koje se pozivaju u slučaju pokretanja, zaustavljanja, dodavanja i brisanja iz sistema.
- Z69** Kontekstualna platforma omogućava automatsko pokretanje/zaustavljanje aplikacija u zavisnosti od zadate verovatnoće stanja okruženja, zadate zauzetosti korisnika ili u zadato vreme.

3.1.3.8 *Povezivanje sa okruženjem*

- Z70** Kontekstualna platforma omogućava povezivanje sa okruženjem, i to u cilju pribavljanja informacija od fizičkih senzora, izdavanja naredbi fizičkim aktuatorima, objavljivanja stanja okruženja i zauzetosti korisnika zainteresovanim sistemima i aplikacijama, komunikacije sa korisničkim spregama ugrađenih kontekstualnih aplikacija, kao i pristupa sistemskoj korisničkoj sprezi platforme.
- Z71** Kontekstualna platforma teži korišćenju jedinstvenih sprega za komunikaciju, koliko je to moguće.
- Z72** U trenutku pisanja disertacije, potrebno je podržati minimalno sledeće aktuelne protokole: Ethernet, WiFi, Zigbee, USB, RS-232.
- Z73** Komunikacija kojom se isporučuju osetljive informacije treba da bude sigurna. Ovo se primarno odnosi na komunikaciju stanja okruženja, zauzetosti korisnika i drugih informacija upućenih ka korisničkim spregama i od njih.

3.1.3.9 *Sertifikacioni zahtevi*

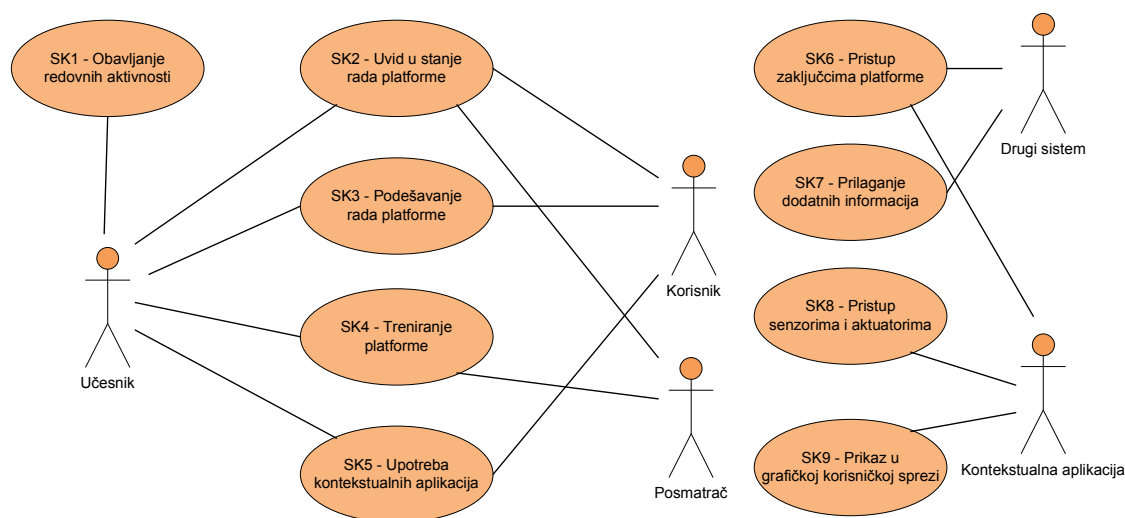
- Z74** Kontekstualna platforma koja se realizuje kao samostalan uređaj, mora da zadovolji sve relevantne sertifikacione zahteve, uključujući elektromagnetnu kompatibilnost (EN 55014, 2009) i materijale korišćene za izradu po direktivi 2002/95/EC Saveta Evrope.
- Z75** Integracija kontekstualne platforme u neki uređaj ne sme da uslovljava gubitak sertifikacije tog uređaja.
- Z76** S obzirom da se kontekstualna platforma sastoji od više integriranih fizičkih komponenti koje međusobno komuniciraju, te od ugrađene programske podrške, za realizaciju biće korišćene komponente koje zadovoljavaju sve potrebne sertifikacione zahteve.

3.1.3.10 *Ostali zahtevi*

- Z77** Minimalni period čuvanja kontekstualnih parametara iz prošlosti je 24 sata.
- Z78** Kontekstualna platforma podržava više nivoa korisničkih naloga, kao i mogućnost određivanja dozvola za pristup određenim funkcionalnostima za pojedine korisnike.

3.2 SLUČAJEVI KORIŠĆENJA

U ovom poglavlju dat je spisak slučajeva korišćenja opštenamenske kontekstualne platforme. S obzirom na generičku prirodu kontekstualne platforme, skup slučajeva korišćenja nije konačan. Na Slici 41 prikazan je skup od 8 grupa slučajeva korišćenja koji su identifikovani za kontekstualnu platformu. Od toga, grupe SK1-SK5 su slučajevi korišćenja od strane ljudi, dok su grupe SK6-SK8 slučajevi korišćenja od strane kontekstualnih aplikacija i drugih sistema za koje su zaključci kontekstualne platforme od interesa.



Slika 41. Grupe slučajeva korišćenja kontekstualne platforme

3.2.1 Strane učesnice (*actors*)

Identifikovane su sledeće strane učesnice u interakciji sa kontekstualnom platformom:

UČESNIK. Učesnik je osoba koja koristi kontekstualnu platformu, i to na način da njeno ponašanje, kao što je obavljanje redovnih aktivnosti, utiče na zaključke kontekstualne platforme. Samim tim, učesnik može da obavlja sve slučajeve korišćenja koji su predviđeni za ljude.

KORISNIK. Korisnik je osoba koja koristi kontekstualnu platformu, ali njeno korišćenje ili prisustvo ne utiče na zaključke kontinualne platforme. S tim u vezi, korisnik može da obavlja sve slučajeve korišćenja koji su predviđeni za ljude, osim slučajeva iz grupe SK1 koji se odnose na obavljanje redovnih aktivnosti, na osnovu kojih platforma izvodi zaključke, i grupe SK4 koja se odnosi na treniranje platforme.

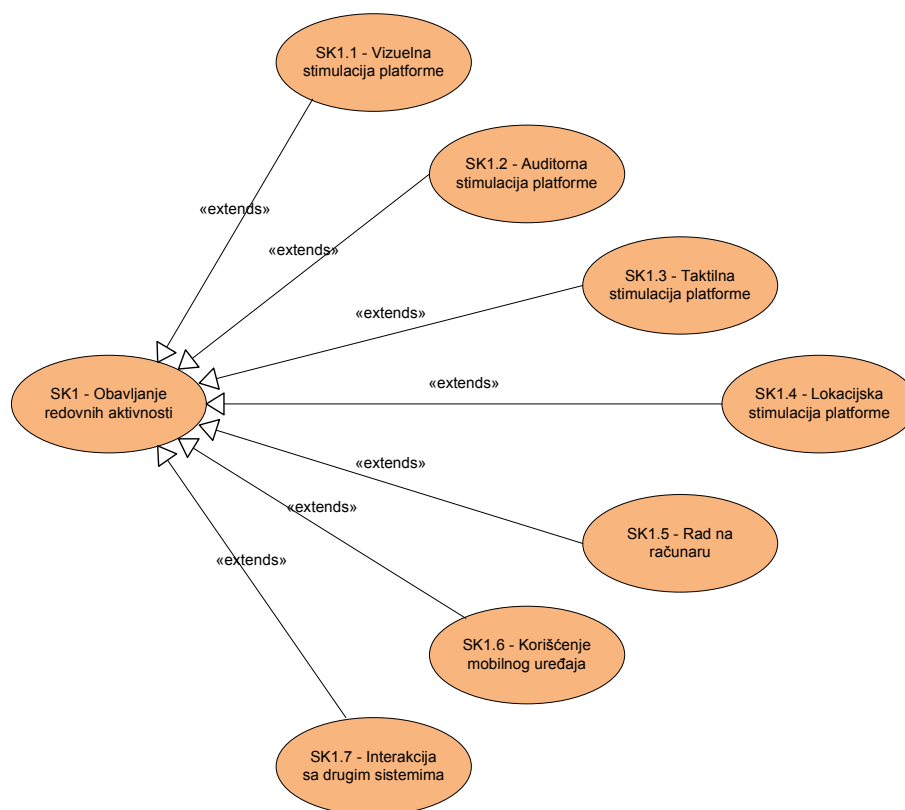
POSMATRAČ. Posmatrač je osoba koja koristi kontekstualnu platformu u fazi treninga i konfiguracije. S tim u vezi, on može da obavlja slučajeve korišćenja SK2 i SK4. Njegov angažman ne utiče na zaključke kontekstualne platforme.

DRUGI SISTEM. Drugi sistem je entitet koji je zainteresovan za zaključke koje donosi kontekstualna platforma, te ima pristup tim zaključcima. Na osnovu ovih zaključaka sistem može da podešava ili prilagođava rad sopstvenih aplikacija, koje se ne izvršavaju u kontekstu kontekstualne platforme.

KONTEKSTUALNA APLIKACIJA. Kontekstualna aplikacija se odnosi na ugrađenu kontekstualnu aplikaciju, koja, osim zaključcima platforme, ima pristup i podacima sa senzora, kao i kontrolu nad aktuatorima. Takođe, kontekstualna aplikacija može da osvežava prikaz u korisničkoj sprezi koji se odnosi na njeno izvršavanje.

3.2.2 Slučajevi korišćenja iz grupe 1 – Obavljanje redovnih aktivnosti

Slučajevi korišćenja iz grupe 1 dalje se proširuju na specifične slučajeve korišćenja, kao što je prikazano na Slici 42.



Slika 42. Slučajevi korišćenja u grupi SK1 – Obavljanje redovnih aktivnosti

Model arhitekture platforme

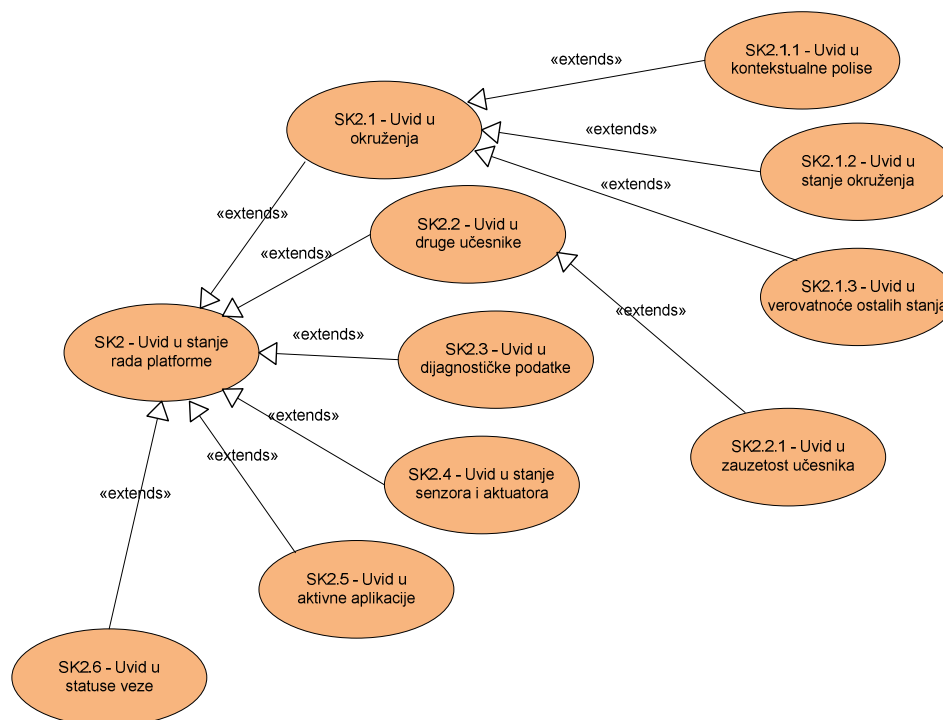
- SK1.1 Vizuelna stimulacija platforme.** Korisnik obavlja aktivnosti koje se mogu identifikovati korišćenjem vizuelnih metoda, koje uključuju snimanje scene optičkim ili 3D kamerama, detekciju prisustva u zoni korišćenjem PIR ili ultrazvučnih senzora i sl. U ove aktivnosti ubrajaju se one aktivnosti kod kojih je važan položaj tela (npr. gledanje TV programa, sedenje, šetanje, i sl), ili koje uzimaju u obzir broj osoba prisutnih u prostoriji.
- SK1.2 Auditorna stimulacija platforme.** Korisnik obavlja aktivnosti koje se mogu identifikovati korišćenjem auditornih metoda, koje uključuju snimanje zvuka u okruženje korišćenjem mikrofona ili mikrofonskih nizova. Ove aktivnosti uključuju aktivnosti kod kojih je važan govor (npr. sastanak, razgovor telefonom i sl.), odnosno za detekciju ostalih zvukova i determinaciju okruženja na osnovu njih (npr. ulica, stadion, zubna ordinacija, zabava, diskoteka i sl).
- SK1.3 Taktilna stimulacija platforme.** Korisnik obavlja aktivnosti koje uključuju interakciju dodirom ili kontaktom. Na primer, senzor težine može da odredi da li osoba stoji na određenoj lokaciji u prostoriji, odnosno da li sedi ili leži. Korisnik takođe može da pritisne tastere koji su povezani sa kontekstualnom platformom, ili da odvajaju kontaktne prekidače (npr. otvaranje vrata, podizanje telefonske slušalice).
- SK1.4 Lokacijska stimulacija platforme.** Korisnik obavlja aktivnosti koje značajnije utiču na promenu lokacije korisnika. Lokacija se otkriva pasivnom ili aktivnom metodom, korišćenjem smetnji u radio signalu (za otkrivanje položaja unutar domaćinstva), odnosno korišćenjem GPS navigacije u kombinaciji sa GPRS za dobavljanje podataka o prostornim koordinatama korisnika u geografskoj sredini.
- SK1.5 Rad na računaru.** Korisnik koristi računar za obavljanje uobičajenih radnji poput pregleda elektronske pošte, pretraživanja interneta, uređivanja dokumenata i sl. Na računaru koji je u upotrebi instalirane su aplikacije za monitoring broja aktivnih procesa, broja otvorenih prozora i pokrenutih aplikacija, aktivnosti na tastaturi/mišu/ekranu osetljivom na dodir, koje obaveštavaju kontekstualnu platformu o korisničkoj aktivnosti.

SK1.6 Korišćenje mobilnog uređaja. Korisnik koristi mobilni uređaj da obavi glasovne pozive, šalje i prima tekstualne poruke ili koristi aplikacije na način sličan kao prilikom rada na računaru. Servis ili pozadinska aplikacija na mobilnom uređaju prati ove aktivnosti i o njima obaveštava kontekstualnu platformu.

SK1.7 Interakcija sa drugim sistemima. Korisnik obavlja interakciju sa drugim sistemima, odnosno uređajima u okruženju. Na primer, korisnik vrši promenu gledanog programa na TV, aktivira aparat za kafu, uključuje klima uređaj i sl. U sistemima za automatizaciju domaćinstva, ovakve akcije se mogu pratiti ili programirati. Sistemi obaveštavaju kontekstualnu platformu o ovim akcijama korišćenjem odgovarajuće sprege.

3.2.3 Slučajevi korišćenja iz grupe 2 – Uvid u stanje rada platforme

Slučajevi korišćenja iz grupe 2 dalje se proširuju na specifične slučajeve korišćenja, kao što je prikazano na Slici 43.



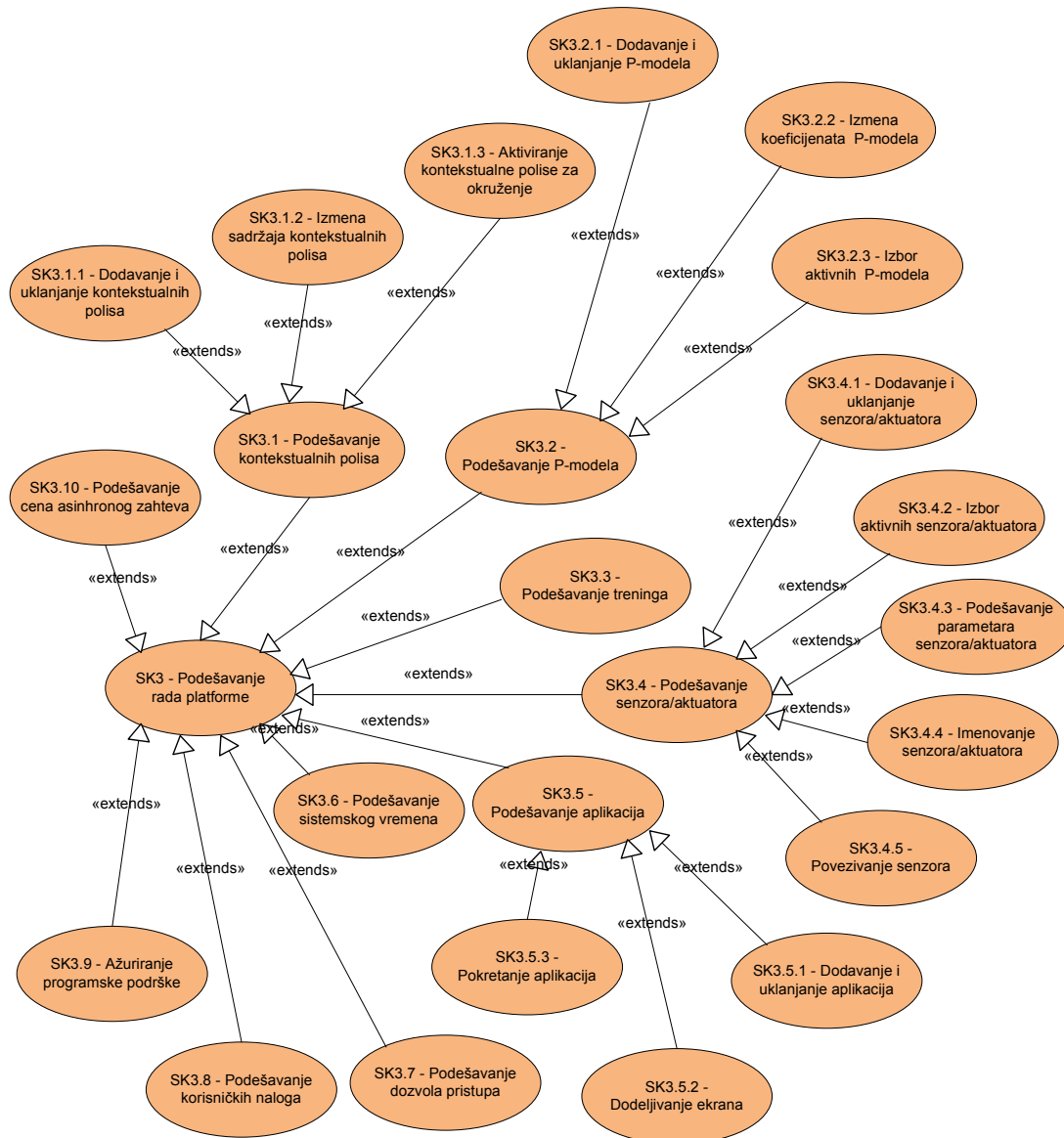
Slika 43. Slučajevi korišćenja u grupi SK2 – Uvid u stanje rada platforme

Model arhitekture platforme

- SK2.1 Uvid u okruženja.** Korisnik posredstvom sistemske korisničke sprege dobija uvid u spisak svih registrovanih okruženja i trenutnu semantizaciju konteksta u tim okruženjima. Konkretno, korisnik ima uvid u naziv i sadržaj kontekstualne polise za svako od okruženja (SK2.1.1), uvid u procenu trenutnog stanja okruženja na osnovu stanja sa najvećom verovatnoćom (SK2.1.2), kao i uvid u verovatnoće svih mogućih stanja definisanih za pregledano okruženje (SK2.1.3).
- SK2.2 Uvid u druge učesnike.** Korisnik ima mogućnost uvida u stanje semantizacije konteksta u okruženjima u kojima se nalaze učesnici. Ove informacije uključuju identifikaciju učesnika i njegovu trenutnu pripadnost određenom okruženju, kao i informaciju o trenutnoj zauzetosti učesnika (SK2.2.1).
- SK2.3 Uvid u dijagnostičke podatke.** Korisnik ima pristup dijagnostičkim podacima koji uključuju opšte stanje komponenti kontekstualne platforme (radi/ne radi, vreme od pokretanja, količina zauzete memorije i sl). Takođe, korisnik ima uvid u važne sistemske greške i ostale informacije od značaja za brzu procenu ispravnosti rada kontekstualne platforme.
- SK2.4 Uvid u stanje senzora i aktuatora.** Korisnik ima mogućnost da izlista sve senzore i aktuatore za svako od okruženja koji su dostupni kontekstualnoj platformi. Takođe, za svaki senzor i aktuator moguć je uvid u njegovo stanje, tj. da li je uključen/isključen. Dodatno, moguće je obezbediti podatke o poslednjoj vrednosti kontekstualnih parametara senzora, kao i prikaz trenutnih podataka koji su senzoru na raspolaganju (npr. slika koju trenutno vidi optička kamera).
- SK2.5 Uvid u aktivne aplikacije.** Korisnik ima mogućnost da pregleda koje su aplikacije aktivne, odnosno koje su mu dostupne, za svako od okruženja.
- SK2.6 Uvid u statuse veze.** Korisnik ima uvid u trenutno stanje veza kontekstualne platforme sa okruženjem, odnosno, da vidi da li kontekstualna platforma ima pristup internetu i koji je dostupan protok, da li je uključen Zigbee ili WiFi, koji je naziv trenutne mreže i koja je IP adresa dodeljena platformi i sl.

3.2.4 Slučajevi korišćenja iz grupe 3 – Podešavanje rada platforme

Slučajevi korišćenja iz grupe 3 dalje se proširuju na specifične slučajeve korišćenja, kao što je prikazano na Slici 44.



Slika 44. Slučajevi korišćenja u grupi SK3 – Podešavanje rada platforme

SK3.1 Podešavanje kontekstualnih polisa. Korisnik ima mogućnost da podešava kontekstualne polise, odnosno, da ima pregled kontekstualnih polisa po svakom registrovanom okruženju. Korisnik može da ukloni kontekstualnu polisu za neko okruženje i da mu dodeli novu, postavljanjem nove datoteke sa kontekstualnom polisom na server kontekstualne platforme, odnosno,

Model arhitekture platforme

kreiranjem nove kontekstualne polise korišćenjem sistemske grafičke sprege i unošenjem njenog sadržaja (SK3.1.1). Takođe, korisnik može da korišćenjem grafičke sprege izmeni sadržaj već aktivne kontekstualne polise (SK3.1.2), kao i da privremeno deaktivira i ponovno aktivira kontekstualnu polisu za određeno okruženje (SK3.1.3).

SK3.2 Podešavanje P-modela. Korisnik ima mogućnost da podešava P-modele, odnosno, da dodaje i uklanja P-modele koji su na raspolaganju kontekstualnoj platformi, kao i da im dodeljuje identifikatore na osnovu kojih se P-modeli adresiraju u okviru kontekstualne polise (SK3.2.1). Takođe, korisnik može da uređuje sadržaj P-modela iz sistemske grafičke sprege, tako što će da unese nove koeficijente P-modela, kao i da pregleda grafik P-modela nakon unošenja koeficijenata (SK3.2.2). Konačno, korisnik može da privremeno suspenduje određene P-modele iz korišćenja, odnosno da ih ponovo omogući, po potrebi (SK3.2.3).

SK3.3 Podešavanje treninga. Korisnik ima mogućnost da deklariše početak, odnosno završetak režima treninga, odnosno da odabere model funkcionisanja kontekstualne platforme – da li se koriste kontekstualne polise (na način naveden u 2.1.1) ili se obavlja automatska semantizacija konteksta (na način naveden u 2.1.2). Dodatno, može se omogućiti vremensko aktiviranje, odnosno deaktiviranje treninga, u skladu sa podešavanjima korisnika.

SK3.4 Podešavanje senzora/aktuatora. Korisnik ima mogućnost da podešava rad senzora i aktuatora, i to: (1) da dodaje i uklanja senzorske i aktuatorске biblioteke, odnosno da dodaje i uklanja instance senzora/aktuatora, sa mogućnošću automatske pretrage za podržanim sensorima i aktuatorima, za protokole u kojima je pretraživanje moguće (SK3.4.1); (2) da omogućuje i onemogućuje pojedine senzore i aktuatore, tako da se definiše njihova dostupnost aktivnim kontekstualnim polisama (SK3.4.2); (3) da detaljno podese sve relevantne parametre rada senzora i aktuatora, uključujući njihovu osnovnu kalibraciju, kao i da ima mogućnost uvida u trenutno stanje senzora, po mogućstvu i grafički, kao što je npr. uvid u sliku koju trenutno snima optička kamera (SK3.4.3); (4) da imenuje instance senzora i aktuatora, odnosno da im zadaje identifikatore, kako bi se isti mogli jednoznačno

identifikovati u okviru kontekstualnih polisa i kontekstualnih skriptova (SK3.4.5); (5) da povezuje međusobno više senzora, s posebnim naglaskom na povezivanje fizičkih i logičkih/virtuelnih senzora, korišćenjem posebnih bafera za razmenu potrebnih podataka (SK3.4.6).

SK3.5 Podešavanje aplikacija. Korisnik ima mogućnost da podešava rad ugrađenih kontekstualnih aplikacija, i to: (1) da dodaje i uklanja dostupne aplikacije, prilaganjem/uklanjanjem pripadajućih kontekstualnih skriptova (SK3.5.1); (2) da dodeljuje podržane ekrane određenim aplikacijama koje koriste grafičku korisničku spregu (SK3.5.2); (3) da pokreće i zaustavlja aplikacije, odnosno da definiše uslove pod kojima se aplikacije pokreću, u smislu stanja okruženja ili zauzetosti korisnika (SK3.5.3).

SK3.6 Podešavanje sistemskog vremena. Korisnik ima mogućnost da podese sistemsko vreme, uključujući datum, sate, minute i sekunde.

SK3.7 Podešavanje dozvola pristupa. Korisnik koji ima najveći stepen dozvola pristupa ima pravo da dodeljuje pojedinačne dozvole pristupa pojedinačnim funkcionalnostima kontekstualne platforme (korisničkim spregama aplikacija, podešavanjima kontekstualne platforme i sl) pojedinačnim korisnicima.

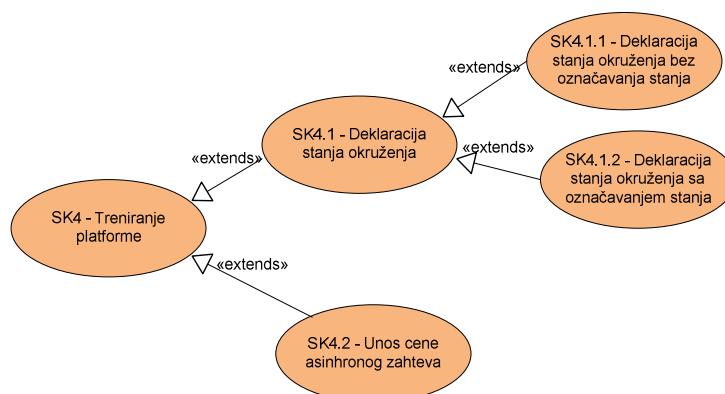
SK3.8 Podešavanje korisničkih naloga. Korisnik koji ima najveći stepen dozvola pristupa ima pravo da dodeljuje nove korisnike koji imaju određena prava pristupa funkcionalnostima kontekstualne platforme.

SK3.9 Ažuriranje programske podrške. Korisnik ima mogućnost da inicijalizuje postupak ažuriranja programske podrške kontekstualne platforme, na predviđeni način (automatski posredstvom interneta, putem USB stick-a, zamenom odgovarajućih datoteka i ponovnim pokretanjem uređaja i dr).

SK3.10 Podešavanje cena asinhronog zahteva. Korisnik ima mogućnost da unese vrednosti cena asinhronog zahteva koji je uspostavljen u stanjima okruženja koja su označena i navedena u okviru ponuđene grafičke sprege.

3.2.5 Slučajevi korišćenja iz grupe 4 – Treniranje platforme

Slučajevi korišćenja iz grupe 4 se dalje proširuju na specifične slučajeve korišćenja, kao što je navedeno na Slici 45.



Slika 45. Slučajevi korišćenja u grupi SK4 – Treniranje platforme

SK4.1 Deklaracija stanja okruženja. Korisnik ima mogućnost da, korišćenjem uređaja u svojoj blizini (mobilni telefon, TV prijemnik i sl) pokrene odgovarajući režim za deklaraciju stanja okruženja. Korisnička sprega kojom se deklariše stanje okruženja ima minimalno jedan taster, čijim pritiskom korisnik deklariše stanje okruženja bez označavanja (SK4.1.1), ili omogućava unos simboličkog naziva stanja i pritiskanje tastera za deklaraciju stanja, čime se stanje okruženja deklariše sa označavanjem (SK 4.1.2).

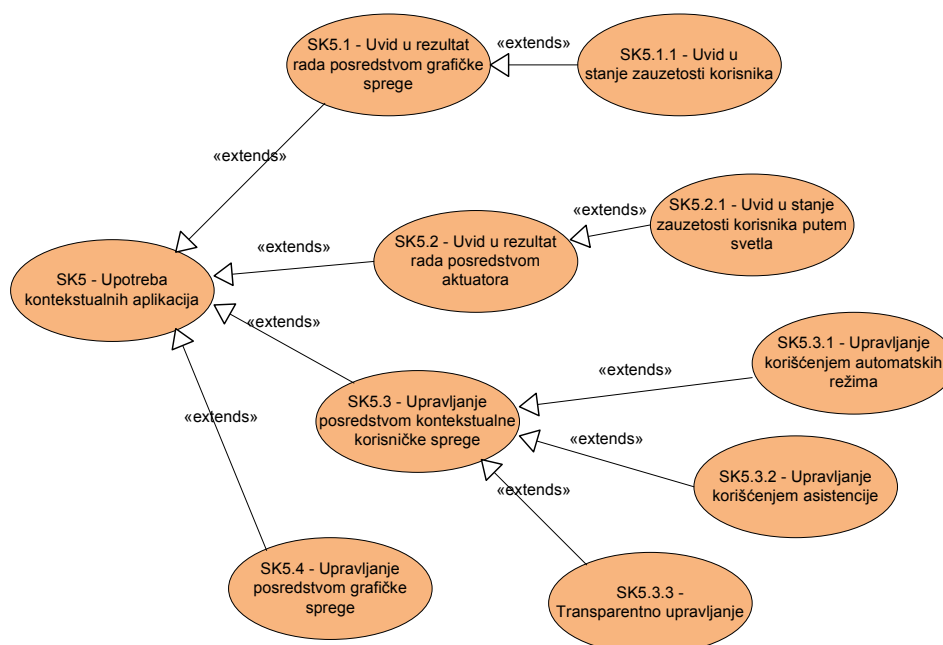
SK4.2 Unos cene asinhronog zahteva. Korisnik ima mogućnost da, korišćenjem uređaja u svojoj blizini (mobilni telefon, TV prijemnik i sl) unese cenu asinhronog zahteva u proizvoljnom trenutku (verovatno neposredno nakon nastupanja asinhronog zahteva). Korisnička sprega omogućava izbor cene asinhronog zahteva (slajderom, spin kontrolom ili izborom iz radio grupe sa Likert skale) i njen unos pritiskom na odgovarajući taster.

3.2.6 Slučajevi korišćenja iz grupe 5 – Upotreba kontekstualnih aplikacija

Slučajevi korišćenja iz grupe 5 se dalje proširuju na specifične slučajeve korišćenja, kao što je navedeno na Slici 46.

SK5.1 Uvid u rezultat rada posredstvom grafičke sprege. Korisnik pristupa grafičkoj korisničkoj sprezi koja se prikazuje na ekranu. Rezultat rada kontekstualne aplikacije predstavljen je korišćenjem grafičkih elemenata, slike i teksta. U zavisnosti od prirode kontekstualne aplikacije, korisniku su na

raspolaganju različite informacije, kao npr. informacija o stanju zauzetosti korisnika u određenom okruženju (SK5.1.1).



Slika 46. Slučajevi korišćenja u grupi SK5 – Upotreba kontekstualnih aplikacija

SK5.2 Uvid u rezultat rada posredstvom aktuatora. Korisnik se informiše o rezultatu rada kontekstualne platforme, na osnovu interakcije sa svojim neposrednim okruženjem. Ovo se odnosi na one kontekstualne platforme koje za cilj rada imaju adekvatno upravljanje aktuatorima, poput kontrole svetla, klimatizacije, multimedijalnih uređaja, roletni, brava na vratima i dr. Na primer, korisnik se informiše o stanju zauzetosti korisnika u određenom okruženju na osnovu uvida u boju svetla kojom je obasjan prostor u blizini tog okruženja, kao što je opisano u 1.6.2 (SK5.2.1).

SK5.3 Upravljanje korišćenjem kontekstualne korisničke sprege. Korisnik upravlja kontekstualnom aplikacijom korišćenjem kontekstualne korisničke sprege. Ovo uključuje izdavanje naredbi posredstvom grafičke sprege koja je postavljena u odgovarajući režim podesan korisničkom kontekstu (SK5.3.1), upravljanje na osnovu upita, odnosno kontekstne pomoći koju izdaje kontekstualna aplikacija (SK5.3.2), ili transparentno upravljanje, stimulacijom senzora na osnovu kojih korisnik saopštava određeni zahtev ili stanje kontekstualnoj aplikaciji, kao što su pozicija korisnika, gestovi, pa i trenutna

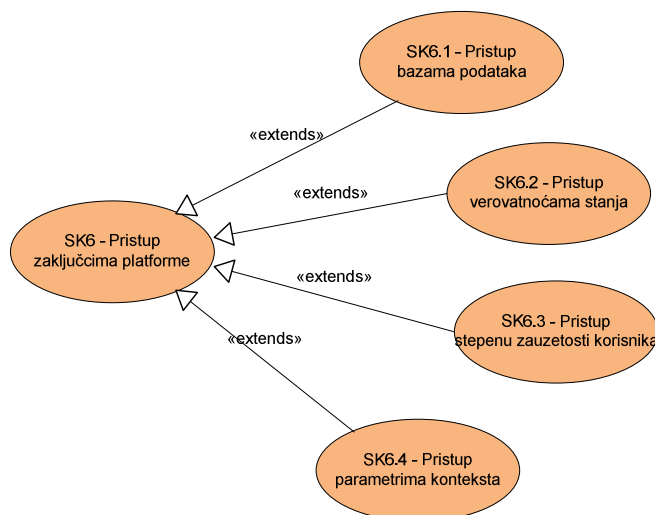
Model arhitekture platforme

aktivnost korisnika (SK5.3.3). Moguće kontekstualne korisničke sprege kojima korisnik može da upravlja navedene su u poglavlju 1.2.

SK5.4 Upravljanje posredstvom grafičke sprege. Korisnik koristi standardne ulazne uređaje (miš, tastatura, ekran osetljiv na dodir i sl.) da saopšti naredbu kontekstualnoj aplikaciji.

3.2.7 Slučajevi korišćenja iz grupe 6 – Pristup zaključcima platforme

Slučajevi korišćenja iz grupe 6 se dalje proširuju na specifične slučajeve korišćenja, navedene na Slici 47.



Slika 47. Slučajevi korišćenja u grupi SK6 – Pristup zaključcima platforme

SK6.1 Pristup bazama podataka. Entitet koristi API kontekstualne platforme da pristupi relevantim bazama podataka. Ovo je osetljiva operacija koja se podrazumevano blokira u većini primena kontekstualne platforme u uređajima potrošačke elektronike. Ipak, njeno postojanje je korisno za interakciju sistema za ispitivanje sa kontekstualnom platformom, ili za interakciju ekstenzija kontekstualne platforme koje potiču od istog proizvođača. U ovom slučaju, informacije koje se prenose API-jem se enkriptuju. Podaci kojima entitet može pristupiti uključuju spiskove senzorskih i aktuatorskih biblioteka, kontekstualnih polisa, P-modela, podešavanja senzora i aktuatora i sl.

SK6.2 Pristup verovatnoćama stanja. Entitet pristupa verovatnoćama stanja okruženja korišćenjem API-ja kontekstualne platforme. Moguće je

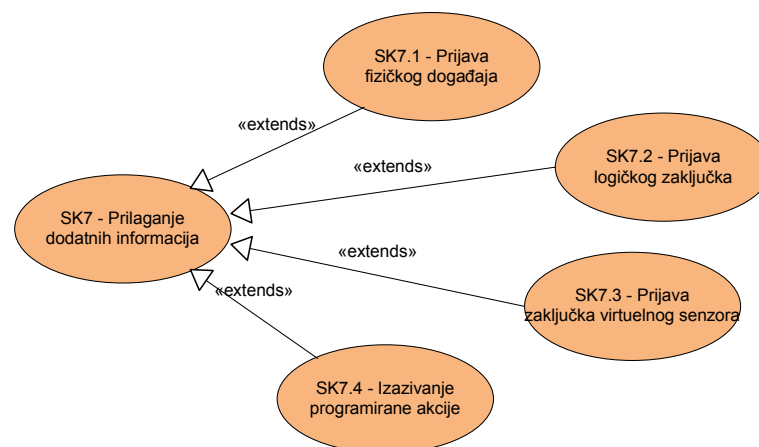
dobavljanje trenutnog stanja okruženja, na osnovu stanja sa najvećom verovatnoćom, kao i dobavljanje verovatnoće svakog od registrovanih stanja. Takođe, moguće je izlistavanje istih informacija za sva dostupna okruženja.

SK6.3 Pristup stepenu zauzetosti korisnika. Entitet pristupa informacijama o stepenu zauzetosti korisnika za svako okruženje ponaosob.

SK6.4 Pristup parametrima konteksta. Entitet pristupa parametrima konteksta koje prikuplja kontekstualna platforma. Ova operacija je takođe osetljiva po pitanju sigurnosti, tako da je njeno korišćenje opravdano gotovo isključivo u slučajevima kada je entitet uređaj ili sistem istog proizvođača, koji pruža usluge dodatne semantizacije konteksta ili virtuelnog senzora, a iz opravdanog razloga nije ga moguće integrisati kao biblioteku kontekstualne platforme (na primer, zbog GPL licence koja zahteva udaljeno sprezanje, po principu *at arm's length* (GPL FAQ, 2012)).

3.2.8 Slučajevi korišćenja iz grupe 7 – Prilaganje dodatnih informacija

Slučajevi korišćenja iz grupe 7 se dalje proširuju na specifične slučajeve korišćenja, navedene na Slici 48.



Slika 48. Slučajevi korišćenja u grupi SK7 – Prilaganje dodatnih informacija

SK7.1 Prijava fizičkog događaja. Entitet prijavljuje fizički događaj koji se dogodio na nekom od senzora kojem entitet ima pristup, korišćenjem API-ja

Model arhitekture platforme

kontekstualne platforme. Na ovaj način kontekstualna platforma proširuje skup dostupnih kontekstualnih parametara u okviru semantizacije konteksta.

SK7.2 Prijava logičkog zaključka. Entitet prijavljuje logički zaključak, u vidu senzorskog događaja, dobijen na osnovu unutrašnje logike entiteta, kao i na osnovu zaključaka kontekstualne platforme kojima entitet ima pristup. Na ovaj način, spoljni entitet može da realizuje dodatnu logiku zaključivanja na osnovu već donetih zaključaka. Zaključak se prijavljuje korišćenjem API-ja kontekstualne platforme.

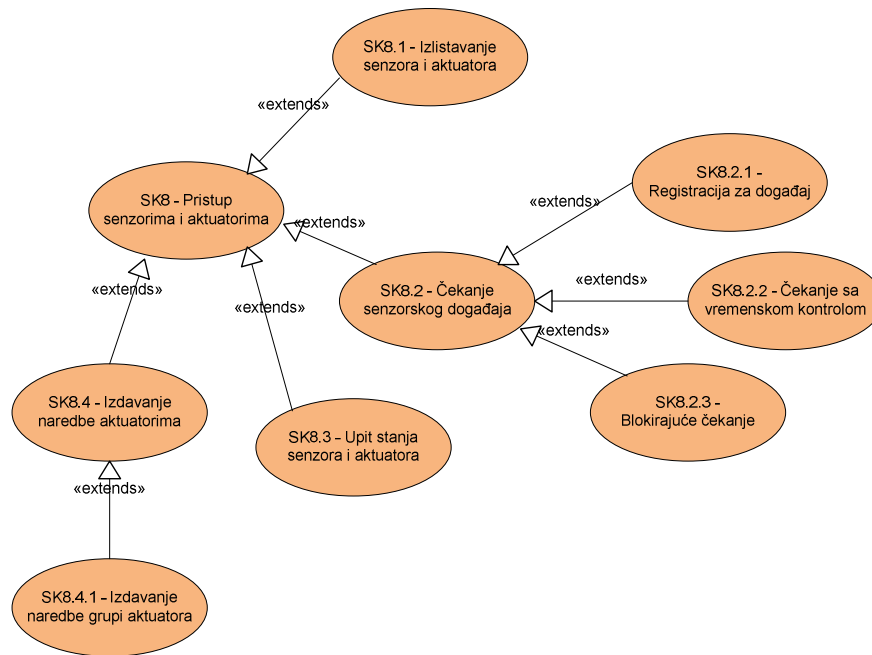
SK7.3 Prijava zaključka virtuelnog senzora. Entitet prijavljuje zaključak u vidu senzorskog događaja, korišćenjem API-ja kontekstualne platforme. Zaključak je izveden na osnovu više ulaznih parametara, koji uključuju fizičke senzore dostupne entitetu, unutrašnju logiku entiteta, kao i već donete zaključke kontekstualne platforme. Ukoliko je entitet neka druga kontekstualna platforma, ovaj slučaj korišćenja opisuje najčešći mehanizam njihovog povezivanja.

SK7.4 Izazivanje programirane akcije. U okviru ovog slučaja korišćenja, opciono, entitetu se omogućuje da izazove proizvoljnu akciju kontekstualne platforme, kao što je, npr. izmena kontekstualne polise, restart platforme, pokretanje aplikativnog skripta i sl. Ova mogućnost odnosi se na integraciju kontekstualne platforme sa entitetima istog proizvođača, kao što je, npr. sistem za automatsko testiranje. Sprega kojom se omogućava izazivanje programirane akcije se pažljivo dizajnira kako bi se osigurala robustnost kontekstualne platforme i izbeglo neželjeno ugrožavanje njenog integriteta.

3.2.9 Slučajevi korišćenja iz grupe 8 – Pristup sensorima i aktuatorima

Slučajevi korišćenja iz grupe 8 se dalje proširuju na specifične slučajeve korišćenja navedene na Slici 49.

SK8.1 Izlistavanje senzora i aktuatora. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta, da izlista sve dostupne senzore i aktuatore u okviru kontekstualne platforme.



Slika 49. Slučajevi korišćenja u grupi SK8 – Pristup sensorima i aktuatorima

SK8.2 Čekanje senzorskog događaja. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta da se registruje za događaje na sensorima za koje je zainteresovana. Za registraciju kontekstualna aplikacija navodi vektor uređenih parova (naziv_senzora, id_događaja), na osnovu kojeg specificira sve događaje od interesa (SK8.2.1). Nakon registracije, kontekstualna aplikacija dobija poseban identifikator registracije, koji potom koristi u pozivima za čekanje događaja, koji mogu biti vremenski ograničeni (SK8.2.2) ili blokirajući (SK8.2.3).

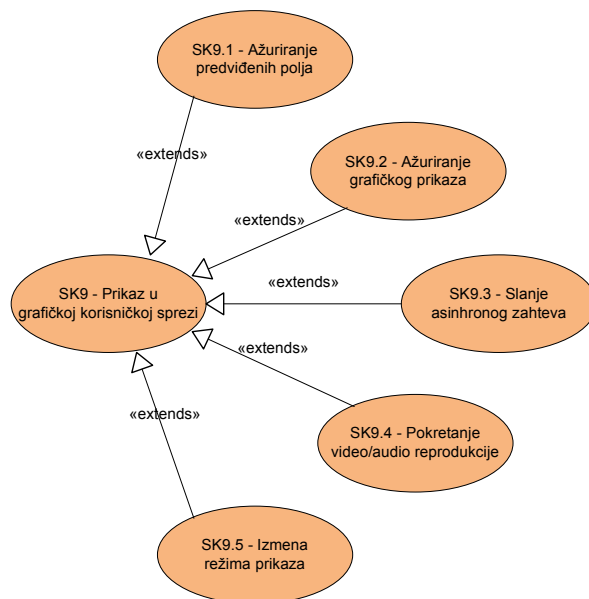
SK8.3 Upit stanja senzora i aktuatora. U bilo kom trenutku kontekstualna aplikacija, posredstvom odgovarajućih poziva iz kontekstualnog skripta, pristupa podacima o stanju bilo kog senzora ili aktuatora. Ova mogućnost je predviđena bez obzira na mehanizam čekanja senzorskog događaja, i ukoliko se koristi u proizvoljnom trenutku vraća poslednje dostupne informacije o stanju.

SK8.4 Izdavanje naredbe aktuatorima. Kontekstualna aplikacija, korišćenjem odgovarajućih poziva iz kontekstualnog skripta, izdaje naredbe dostupnim aktuatorima. Mogući spisak naredbi definisan je dokumentacijom svake od aktuatorskih biblioteka, dok se takođe za svaki tip aktuatora može obezbediti i posebna skripting biblioteka sa odgovarajućim pozivima. Neke od mogućih

naredbi su „Uključi svetlo na akuatoru 15”, ili
 “Postavi ciljanu vrednost hlađenja na klimatizacionoj jedinici 8 na 25° C.

3.2.10 Slučajevi korišćenja iz grupe 9 – Prikaz u grafičkoj korisničkoj sprezi

Slučajevi korišćenja iz grupe 9 se dalje proširuju na specifične slučajeve korišćenja navedene na Slici 50.



Slika 50. Slučajevi korišćenja u grupi SK9 – Prikaz u grafičkoj korisničkoj sprezi

SK9.1 Ažuriranje predviđenih polja. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta, da ažurira vrednosti pojedinih promenljivih kontekstualne platforme, na osnovu kojih sama kontekstualna platforma na odgovarajući način ažurira prikaz u grafičkoj korisničkoj sprezi. U ovom slučaju kontekstualna aplikacija ne utiče direktno na način prikaza date informacije, kao i da li će i kada ta informacija uopšte biti prikazana. Na primer, kontekstualna aplikacija može da priloži određene unose koje je kreirao korisnik.

SK9.2 Ažuriranje grafičkog prikaza. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta, da direktno ažurira sopstveni grafički prikaz. U ovom slučaju, kontekstualna aplikacija je povezana sa grafičkim delom svoje reprezentacije, sa kojim komunicira korišćenjem opštenamenskih komunikacionih poziva iz kontekstualnog skripta. Kontekstualna aplikacija

tako može da utiče na sve osobine prikaza na osnovu sopstvenih konvencija u razmeni podataka sa grafičkim delom. Grafički deo može biti realizovan kao Flash aplikacija, HTML5, SVG, Javascript i sl.

SK9.3 Slanje asinhronog zahteva. Kontekstualna aplikacija upućuje asinhroni zahtev okruženju u kome se nalazi korisnik, korišćenjem odgovarajućih poziva iz kontekstualnog skripta. Asinhroni zahtev je najčešće zvučno ili tekstualno obaveštenje (ili oboje) koje se pojavljuje u grafičkoj korisničkoj sprezi bez obzira na njeno trenutno stanje (npr. kao dijalog ispred trenutne grafike).

SK9.4 Pokretanje video/audio reprodukcije. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta, da inicira početak reprodukcije audio ili video materijala (ili niza statičnih slika) na nekom određenom ekranu. U ovom slučaju se podrazumeva da kontekstualna aplikacija ne poseduje specifičnu grafičku reprezentaciju, nego da koristi inače dostupne ekrane za reprodukciju nekog unapred pripremljenog sadržaja.

SK9.5 Izmena režima prikaza. Kontekstualna aplikacija koristi odgovarajuće pozive iz kontekstualnog skripta da izmeni trenutni režim prikaza. Grafički prikaz se dalje menja pod kontrolom kontekstualne platforme, odnosno, grafički prikaz prima informaciju o zahtevu za promenom režima. Grafički prikaz može biti deo systemske korisničke sprege, ili može biti specifično realizovan za datu kontekstualnu aplikaciju.

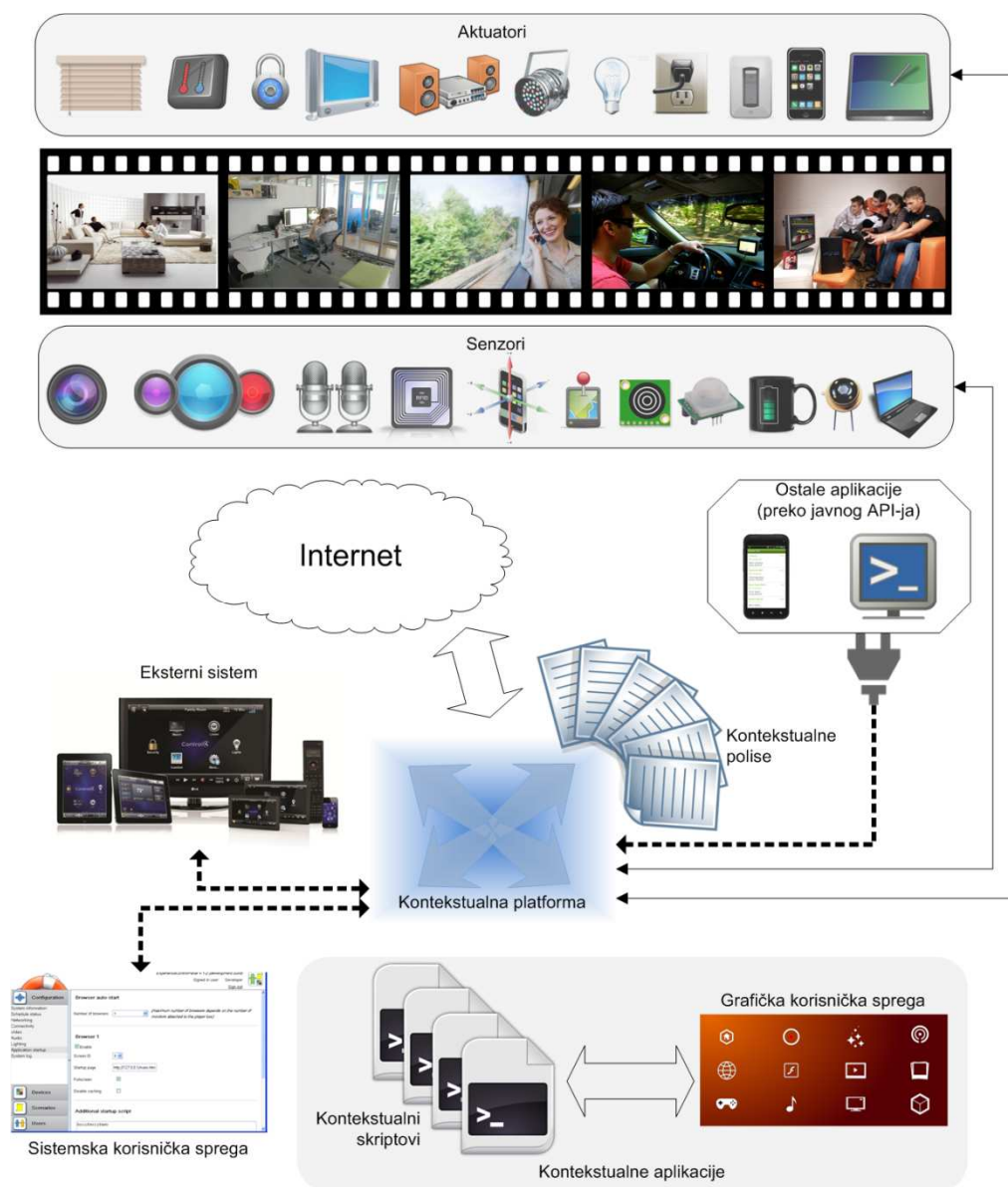
3.3 ARHITEKTURA KONTEKSTUALNE PLATFORME

Cilj ovog poglavlja je da predoči detalje arhitekture kontekstualne platforme, realizovane u skladu sa zahtevima navedenim u poglavlju 3.1 i slučajevima korišćenja iz poglavlja 3.2. Pri prikazu detalja arhitekture biće odražavana generička priroda kontekstualne platforme. Međutim, tamo gde nije moguće zadržati potpuno generičku prirodu prikaza, biće podrazumevana realizacija kontekstualne platforme kao samostalne jedinice u okviru jednog inteligentnog okruženja.

Pogled na kontekstualnu platformu na najvišem nivou hijerarhije dat je na Slici 51. Na ovom nivou uočljivi su blokovi koji su vidljivi pri pogledu na platformu kao na crnu kutiju (*black box*). U nastavku je dato kratko objašnjenje pojedinih elemenata, uz

Model arhitekture platforme

pretpostavku da se podrazumeva realizacija zahteva i slučajeva korišćenja iz poglavlja 3.1 i 3.2.



Slika 51. Pogled na kontekstualnu platformu sa najvišeg nivoa hijerarhije

Kontekstualna platforma je uređaj, odnosno deo programske podrške uređaja u koji je ugrađena, koji omogućava prikupljanje kontekstualnih parametara iz okruženja, njihovu semantizaciju na osnovu kontekstualnih polisa, objavljivanje zaključaka od interesa posredstvom kontekstualnih aplikacija, drugih sistema ili interneta, kao i podešavanje rada platforme korišćenjem sistemske korisničke sprege. Korisnik u najvećem broju slučajeva nije svestan postojanja kontekstualne platforme, izuzev mogućeg uvida u sprege koje ona pruža. Kontekstualna platforma je podesna za

korišćenje u različitim inteligentnim okruženjima, poput domaćinstva (dnevne sobe), radnog mesta (kancelarije), u pokretu kao mobilna aplikacija, kao podrška navigacionom sistemu u automobilu, za podršku video igrama, i konačno, za podršku različitim PC, mobilnim ili sistemskim aplikacijama u cilju približavanja njihovog rada korisničkom kontekstu.

Kontekstualne polise su pisane instrukcije na osnovu kojih kontekstualna platforma obavlja semantizaciju konteksta za određeno inteligentno okruženje, odnosno za druge upotrebe. Ove instrukcije su izmenljive i podešljive od strane korisnika, odnosno tehničkih lica.

Kontekstualne aplikacije su aplikacije koje koriste zaključke kontekstualne platforme, kao i mogućnosti veze sa sensorima i aktuatorima, u cilju omogućavanja određene aplikativne funkcionalnosti. Kontekstualna platforma podržava *ugrađene* kontekstualne aplikacije, koje se izvršavaju u vidu *kontekstualnih skriptova* i opcione *grafičke korisničke sprege*, na nekom od ekrana povezanim sa kontekstualnom platformom. U širem smislu, kontekstualne aplikacije se realizuju kao nezavisne mobilne ili PC aplikacije, koje zaključcima kontekstualne platforme pristupaju posredstvom javnog API-ja.

Sistemska korisnička sprega je aplikativni servis kontekstualne platforme, koji omogućava uvid u stanje njenog rada i pristup svim relevantnim podešavanjima.

Eksterni sistem je sistem koji se najčešće sastoji od sopstvene mreže uređaja, senzora i aktuatora, a koji je povezan sa kontekstualnom platformom u cilju dobavljanja njenih zaključaka, odnosno prilaganja dodatnih kontekstualnih parametara. U okviru eksternog sistema mogu se realizovati dodatne kontekstualne aplikacije i kontekstualne korisničke sprege.

Senzori su skup fizičke opreme ili aplikacija za monitoring koje obezbeđuju informacije o okruženju u vidu kontekstualnih parametara. Kontekstualna platforma je povezana sa sensorima korišćenjem različitih sprežnih podsistema. Najčešći senzori su optičke i 3D kamere, mikrofoni i mikrofonski nizovi, RFID, akcelerometri, GPS lokalizacija, ultrazvučni senzori, PIR senzori, temperaturni senzori i senzori osvetljaja i boje i dr.

Aktuatori su skup fizičke opreme ili aplikacija koje reaguju na naredbu iz kontekstualne aplikacije, u cilju izmene stanja njihovog rada. Najveći deo aktuatora

Model arhitekture platforme

odnosi se na uređaje u inteligentnom okruženju (roletne, klimatizacione jedinice, rasvetu, multimedijalne uređaje i dr). Kontekstualna platforma omogućuje izdavanje naredbi aktuatorima, kako bi omogućila kontekstualnim aplikacijama da u potpunosti preslikavaju izlaz svog rada na stanje inteligentnog okruženja i obezbedi transparentnost funkcionisanja.

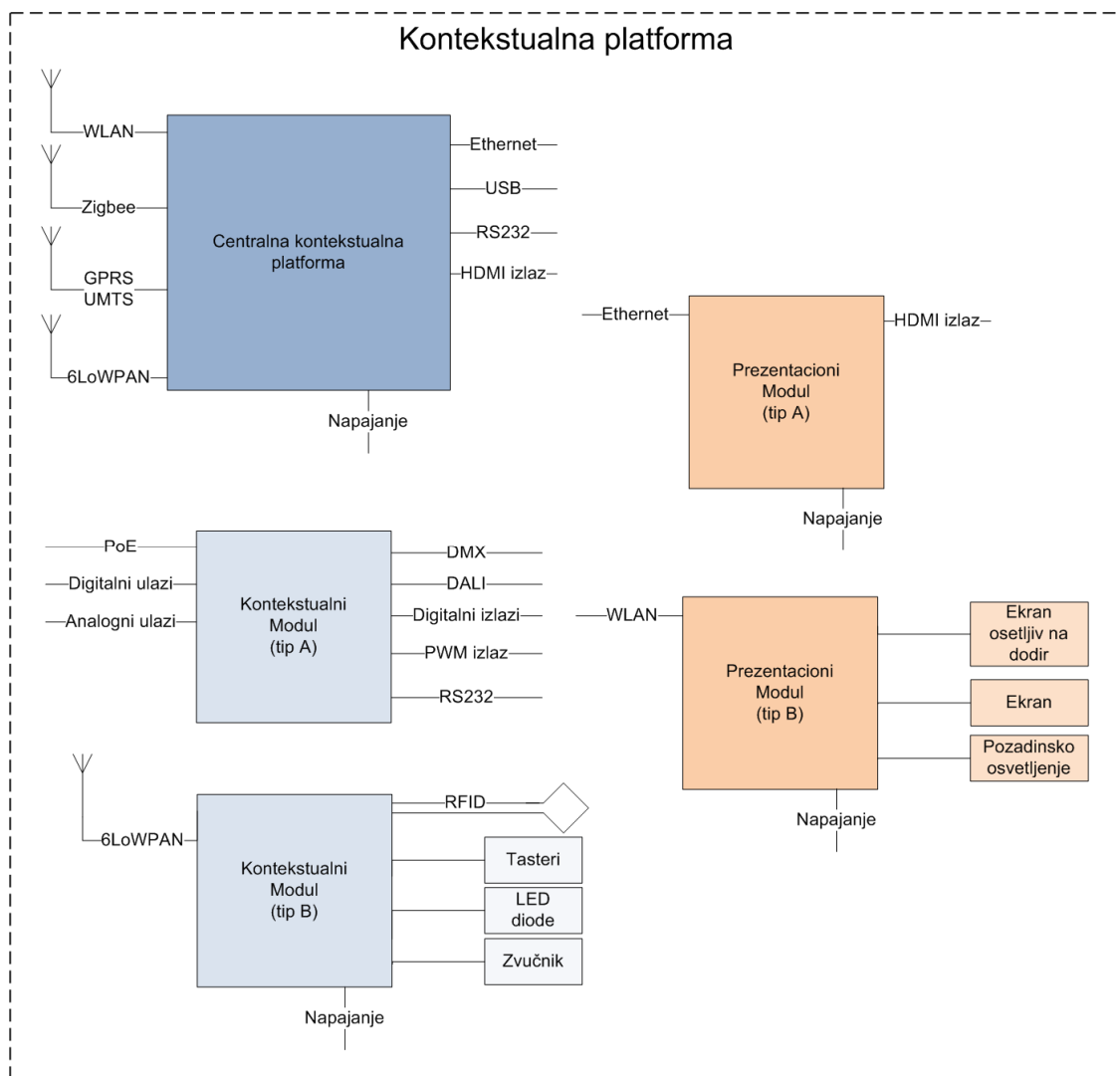
Internet omogućava vezu kontekstualne platforme sa entitetima koji su prostorno udaljeni. Da bi se funkcionalnost deljenja podataka posredstvom interneta u potpunosti realizovala, potrebno je postojanje aplikativnog servera kontekstualne platforme, koji bi se ponašao kao posrednik između zainteresovanih aplikacija koje koriste internet, i kontekstualne platforme.

3.3.1 Pregled fizičke arhitekture

U ovom poglavlju dat je prikaz fizičke arhitekture kontekstualne platforme, kao i kratak opis pojedinih blokova. Ukoliko se kontekstualna platforma ugrađuje u postojeći uređaj potrošačke elektronike, taj uređaj mora da poseduje nadskup navedenih blokova fizičke arhitekture. Na Slici 52 dat je prikaz fizičkih elemenata kontekstualne platforme, korišćenjem pristupa crne kutije (*black box*), i data su osnovna ožičenja i sprege za svaki od elemenata.

Jedini zahtevani modul za minimalnu realizaciju kontekstualne platforme je *centralna kontekstualna platforma*. Ovaj blok realizuje sve potrebne funkcionalnosti i logiku kontekstualne platforme. Interakcija sa okruženjem (senzorima, aktuatorima, drugim sistemima) posredstvom ovog bloka obavlja se žično (Ethernet, RS232) i bežično (WLAN, Zigbee, GRPS/UMTS, 6LoWPAN), za potrebe priključivanja ulaznih uređaja (tastatura, miš), konfiguracije i ažuriranja programske podrške namenjen je USB, dok se HDMI izlaz koristi za priključivanje osnovnog prezentacionog ekrana za grafičku korisničku spregu. S obzirom na to da je platforma zasnovana na TCP/IP komunikaciji, minimalno se zahteva postojanje Ethernet sprege. Opciono, WLAN se koristi u slučajevima kada je kontekstualna platforma realizovana kao samostalan kontroler u kućnom ili kancelarijskom okruženju, ili za korišćenje po potrebi ukoliko je kontekstualna platforma realizovana na mobilnom uređaju. Podrazumevano, u takvom okruženju za pristup internetu mogu se koristiti WLAN ili Ethernet. Za potrebe interoperabilnosti sa velikim brojem prenosivih uređaja sa baterijskim napajanjem koji su komercijalno dostupni, može se opciono koristiti

Zigbee. GPRS/UMTS veza koristi se za pristup internetu i korišćenje mobilnih servisa (npr. SMS) u slučajevima kada se kontekstualna platforma realizuje u sklopu mobilnog uređaja. Konačno, 6LoWPAN se koristi za IP komunikaciju sa uređajima čiji je zahtev izuzetno niska potrošnja energije. U zavisnosti od ciljne izvedbe, centralna kontekstualna platforma se može napajati neprekidnim napajanjem (za primene u inteligentnim okruženjima) ili baterijskim napajanjem (za primene u mobilnim uređajima).



Slika 52. Osnovni elementi fizičke arhitekture kontekstualne platforme i njihova ožičenja

Kontekstualni moduli omogućavaju ostvarenje veze sa sensorima i aktuatorima na fizički udaljenim mestima, dok su moduli bliže njihovoj fizičkoj lokaciji. S obzirom da je tipično okruženje u kojem je upotreba kontekstualne platforme očekivana većih

Model arhitekture platforme

prostornih dimenzija, nije praktično razvođenje fizičke instalacije od centralne kontekstualne platforme do ciljnih senzora. Iako postoje senzori koji omogućavaju bežično ili žično povezivanje standardnim spregama (WLAN, Ethernet), poput, npr. optičkih ili 3D kamera, takođe postoje i sprege kod kojih ovo nije moguće. U okviru kontekstualne platforme predviđena su dva tipa kontekstualnih modula: tip A i tip B.

Modul tipa A je ugradni modul koji se postavlja na odgovarajuće mesto u inteligentnom okruženju, u blizini odgovarajućih senzora i aktuatora. Ovaj modul ostvaruje vezu sa centralnom kontekstualnom platformom korišćenjem *Power Over Ethernet* (PoE) priključka, posredstvom kojeg se i napaja. Modul tipa A poseduje mogućnost priključivanja analognih i digitalnih linija, čime se može priključiti veliki broj senzora koji kao izlaz daju analogne ili digitalne signale. Na aktuatorskoj strani, modul tipa A poseduje digitalne, PWM i RS232 izlaze, posredstvom kojih je moguće upravljati velikim brojem generičkih aktuatorskih jedinica. Dodatno, s obzirom na primene kontekstualne platforme, modul tipa A omogućava kontrolu rasvete korišćenjem DMX i DALI protokola.

Zamisao modula tipa B je da bude manjih dimenzija i da se postavlja neposredno na mesto kontakta senzora sa korisnikom. U ovom smislu, koristi se baterijsko napajanje i odgovarajući protokol niske potrošnje energije 6LoWPAN. Ovaj modul poseduje RFID antenu putem koje može da identifikuje prislonjene objekte, poseduje korisničke tastere, kao i led diode visokog intenziteta za prikaz opšteg stanja rada platforme ili nekog njenog dela.

Prezentacioni moduli omogućavaju realizaciju grafičkih sprega ugrađenih kontekstualnih aplikacija, u slučajevima kada je mesto interakcije sa korisnikom prostorno udaljeno od centralne kontekstualne platforme. S obzirom na trenutno stanje tehnike, predložena su dva tipa prezentacionih modula: tip A i tip B. Prezentacioni modul tipa A komunicira sa centralnom kontekstualnom platformom korišćenjem Ethernet sprege, i prikazuje grafičku korisničku spregu na ekranu priključenom HDMI spregom. Ova prezentaciona jedinica ne poseduje mogućnost unosa podataka od strane korisnika, te se obično koristi u isključivo prezentacionim scenarijima, ili se interakcija sa korisnikom obavlja korišćenjem kontekstualne korisničke sprege posredstvom odgovarajućeg kontekstualnog modula. Modul tipa B se zasniva na sličnom principu, s tim što je ekran njegov sastavni deo. Korisnik obavlja interakciju sa grafičkom

korisničkom spregom dodirrom po ekranu. Prezentacioni moduli su obično sa stalnim napajanjem.

Osim predloženih kontekstualnih i prezentacionih modula, koji su, po mišljenju autora disertacije, najpodesniji za većinu mogućih primena, mogu se realizovati i drugi tipovi modula, koji poseduju drugačiji podskup dostupnih sprega, zavisno od ciljne namene.

U nastavku su dati šematski prikazi fizičke arhitekture pojedinih elemenata kontekstualne platforme, kao i opis njihove realizacije.

3.3.1.1 *Centralna kontekstualna platforma*

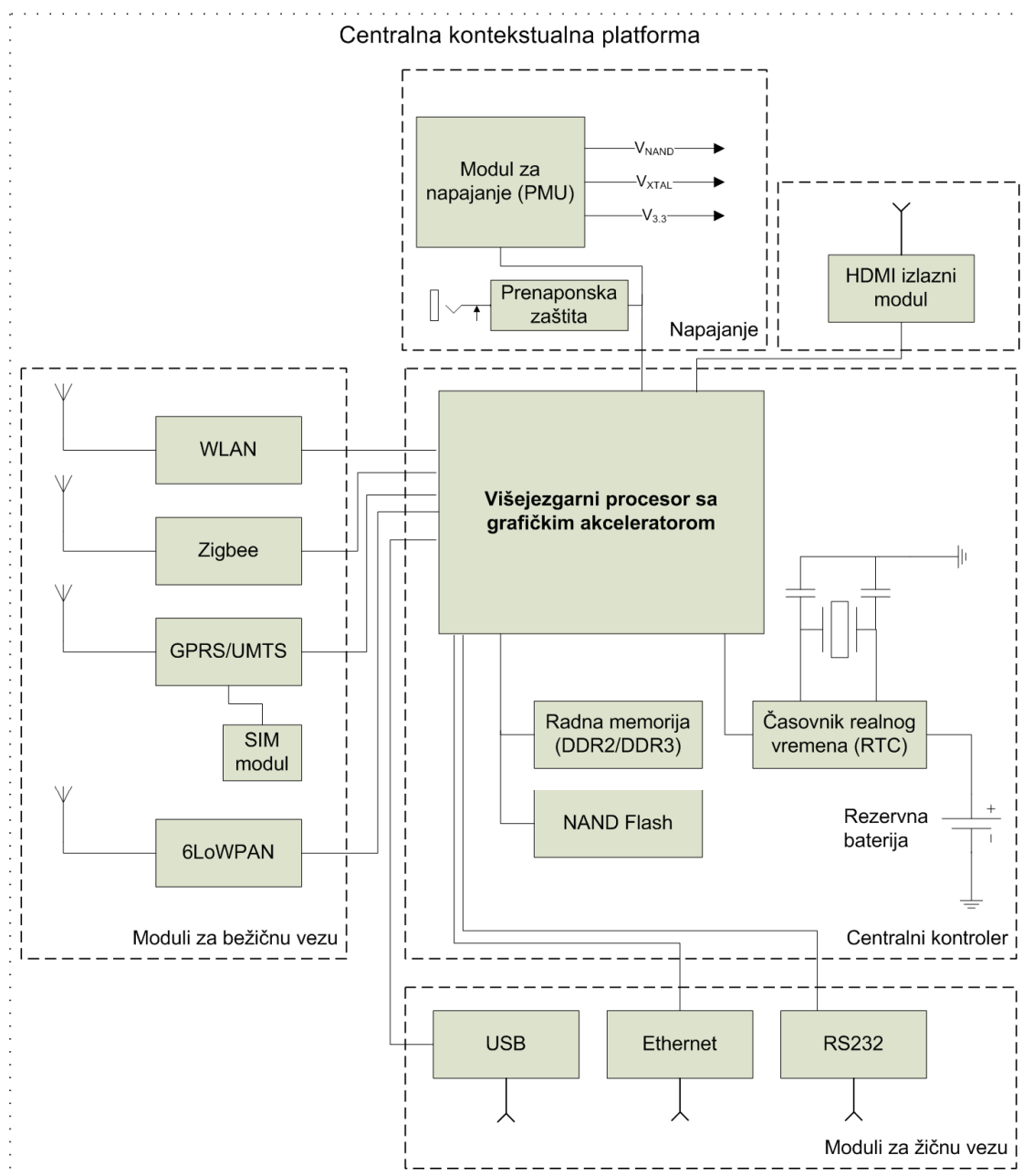
Šematski prikaz fizičke arhitekture centralne kontekstualne platforme dat je na Slici 53.

Centralna kontekstualna platforma realizovana je kao namenski računarski sistem koji se sastoji od centralnog višejezgarnog procesora sa grafičkim akceleratorom, radne memorije, Flash memorije, kao i određenog broja modula koji realizuju potrebne sprežne podsisteme. S obzirom na to da je logika kontekstualne platforme realizovana u njenoj programskoj podršci, koja se izvršava na centralnom kontroleru, za rad u realnom vremenu i ostale potrebne funkcionalnosti ključne su performance centralnog kontrolera. U Tabeli 11 navedeni su minimalni zahtevani parametri komponenti centralnog kontrolera.

<i>Naziv parametra</i>	<i>Vrednost</i>
Broj jezgara procesora	2 ili više
Frekvencija procesora	1 Ghz ili više
Radna memorija	512MB DDR2/DDR3 ili više
NAND Flash	512 MB ili više
Grafički akcelerator	Integrisan sa procesorom
Programska podrška procesora	Linux OS

Tabela 11. Minimalni zahtevani parametri centralnog kontrolera

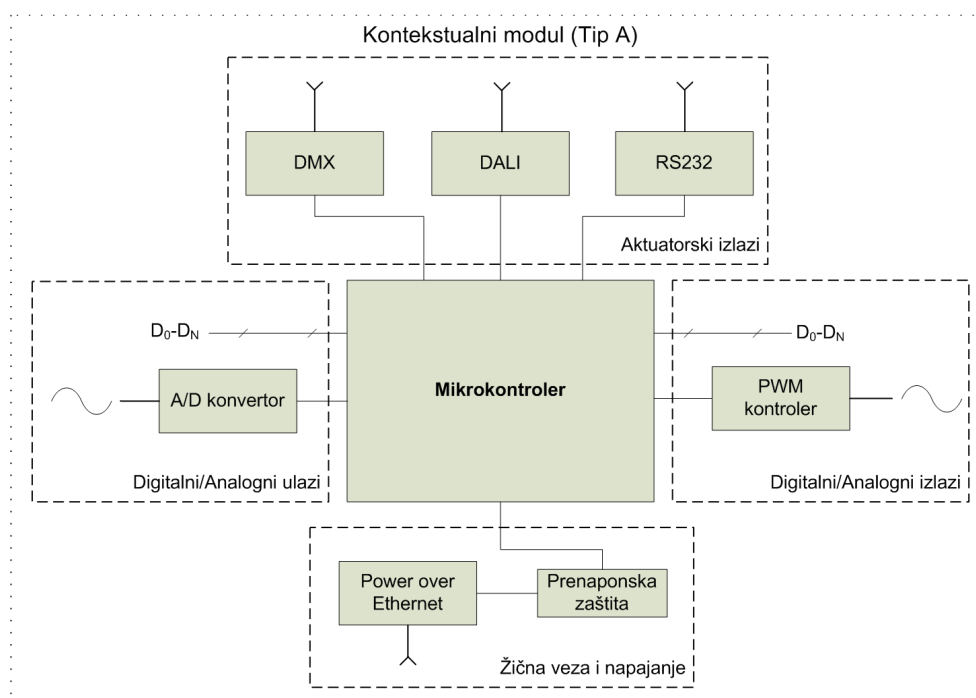
Centralna kontekstualna platforma se, dakle, može realizovati na različitim fizičkim arhitekturama, koje sadrže centralni kontroler i podskup ostalih blokova fizičke arhitekture.



Slika 53. Blok dijagram fizičke arhitekture centralne kontekstualne platform

3.3.1.2 Kontekstualni moduli

Šematski prikaz kontekstualnog modula tipa A dat je na Slici 54.



Slika 54. Blok dijagram fizičke arhitekture kontekstualnog modula tipa A

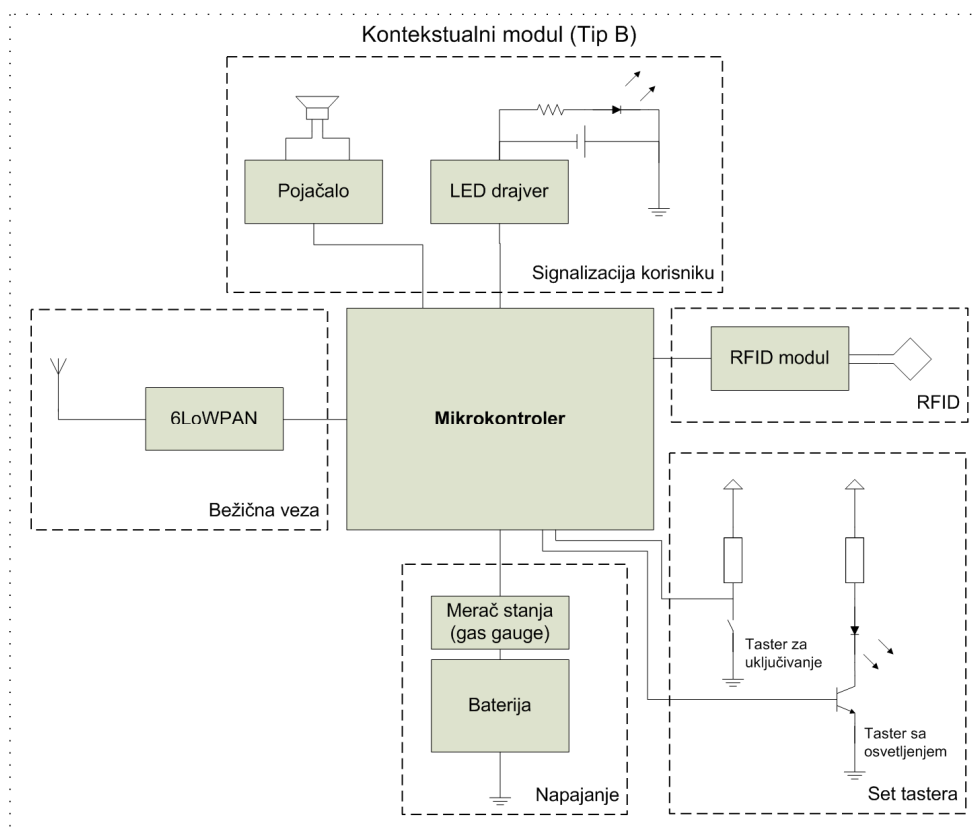
Osnovu kontekstualnog modula tipa A čini namenski mikrokontroler skromnih performansi. Napajanje kompletne arhitekture obavlja se posredstvom Power over Ethernet sprege, putem koje se i ostvaruje veza sa centralnom kontekstualnom platformom. Takođe, integrisani su i svi potrebni blokovi za A/D i D/A konverziju, za kontrolu DMX i DALI rasvetnih tela preko odgovarajuće magistrale, kao i za povezivanje različitih tipova uređaja posredstvom RS232. U Tabeli 12 navedeni su minimalni zahtevi mikrokontrolera koji upravlja kontekstualnim modulom tipa A.

<i>Naziv parametra</i>	<i>Vrednost</i>
Procesor	Minimum 16-bitni 8051 ili ekvivalent
Radna memorija	8 KB ili više
Programabilni Flash	64 KB ili više

Tabela 12. Minimalni zahtevani parametri mikrokontrolera

Mikrokontrolerom se upravlja programom koji se sastoji od minimalnog izvršnog skupa instrukcija, bez potrebe za postojanjem operativnog sistema.

Šematski prikaz kontekstualnog modula tipa B dat je na Slici 55.



Slika 55. Blok dijagram fizičke arhitekture kontekstualnog modula tipa B

Osnovu ovog modula takođe čini mikrokontroler, koji komunicira sa kontekstualnom platformom korišćenjem 6LoWPAN protokola. Modul omogućava blisku interakciju sa korisnikom posredstvom tastera, RFID identifikacije, kao i svetlosnih i zvučnih alarma. S obzirom na potrebu za postavljanjem na različita ciljna mesta, kao i moguću mobilnost, ovaj modul se oslanja na baterijsko napajanje. Zahtevi vezani za mikrokontroler su navedeni u Tabeli 12.

3.3.1.3 *Prezentacioni moduli*

Iako je prezentacione module moguće realizovati kao namenske računarske sisteme, s obzirom na to da oni ne učestvuju u realizaciji kritičnih funkcija kontekstualne platforme, kao i da ne poseduju specifične sprežne podsisteme, za realizaciju prezentacionih modula mogu da budu korišćeni opštenamenske platforme.

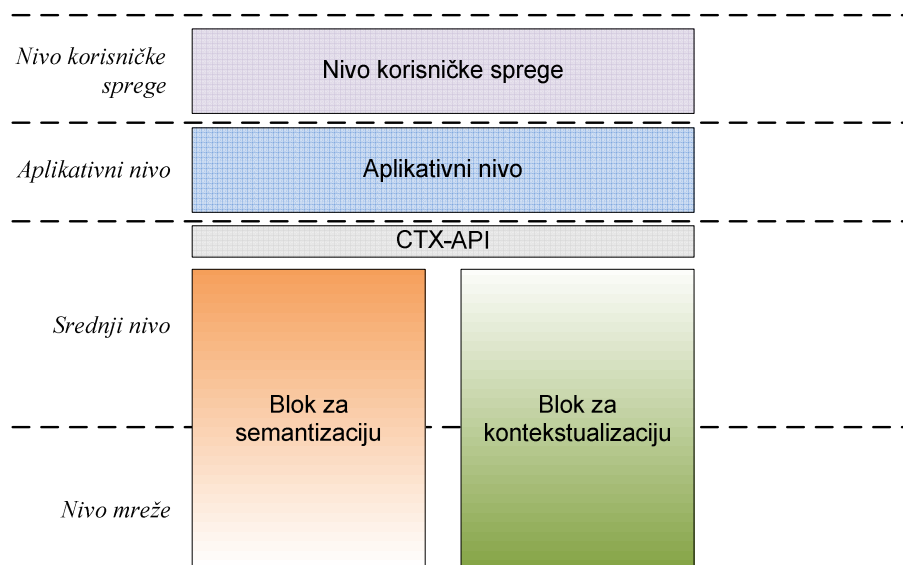
Za realizaciju prezentacionog modula tipa A može se koristiti PC računar priključen na lokalnu mrežu sa monitorom priključenim na HDMI izlaz. Za realizaciju prezentacionog modula tipa B može se koristiti bilo koji tablet novije generacije, koji poseduje ekran osetljiv na dodir i WiFi bežičnu vezu.

3.3.2 Arhitektura programske podrške kontekstualne platforme

U ovom poglavlju biće detaljno razmotrena arhitektura programske podrške kontekstualne platforme. Programska podrška se izvršava na centralnom kontroleru centralne kontekstualne platforme, mikrokontrolerima kontekstualnih modula i kao grafičke aplikacije na prezentacionim modulima. Arhitektura programske podrške za kontekstualnu platformu jedan je od osnovnih fokusa disertacije, tako da će ovoj tematici biti posvećena posebna pažnja.

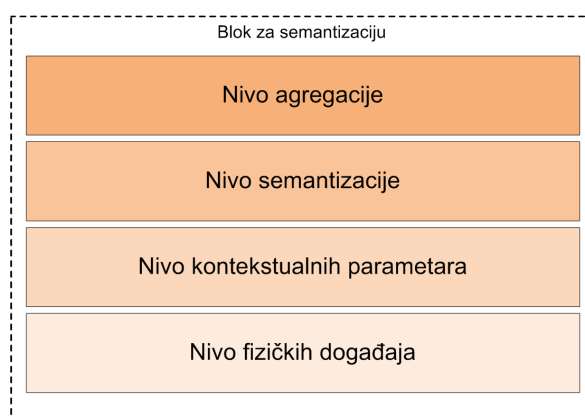
Programska podrška kontekstualne platforme organizovana je nivojski i blokovski. Pri podeli na nivoe poštovana je podela u nivoe korišćena u većini dosadašnjih istraživanja, na nivo mreže, srednji nivo, aplikativni nivo i nivo korisničke sprege, kao što je to detaljno istraženo u poglavlju 1.1.1. Međutim, analizom zahteva i slučajeva korišćenja iz poglavlja 3.1 i 3.2, dolazi se do zaključka da je, osim semantizacije konteksta, odnosno zaključivanja na osnovu kontekstualnih parametara, potrebno omogućiti i upravljanje određenim skupom uređaja, kako bi se realizovala kontekstualna korisnička sprega (što ćemo dalje nazivati *kontekstualizacijom*). Ovo izaziva razlike u realizaciji nivoa mreže i srednjeg nivoa, u zavisnosti od toga da li se obavlja semantizacija konteksta, ili kontekstualizacija okruženja. Iz tog razloga, zaključno sa srednjim nivoom, programska podrška je realizovana u dva bloka: *bloka za semantizaciju* i *bloka za kontekstualizaciju*, iznad kojih se nalaze *aplikativni nivo* i *nivo korisničke sprege*. Naslanjajući se na blokove za semantizaciju i kontekstualizaciju, realizovana je *kontekstualna programska sprega* (CTX API). Ova sprega služi za odvajanje osnovnih blokova kontekstualne aplikacije, odnosno opštenamenskih kontekstualnih funkcija, od konkretnih aplikacija, odnosno korisničkih sprega, i ona pruža usluge aplikacijama, korisničkim spregama i drugim sistemima zasnovano na dostupnim funkcionalnostima kontekstualne platforme (Slika 56).

Model arhitekture platforme



Slika 56. Blokowska i nivojska podela programske podrške na najvišem nivou hijerarhije

Blok za semantizaciju podeljen je na nivoe kao što je to prikazano na Slici 57.

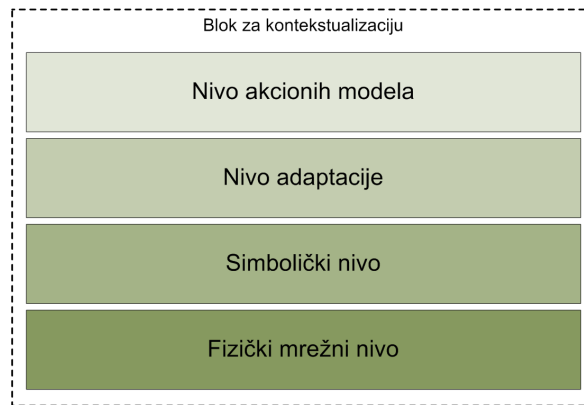


Slika 57. Nivojska organizacija bloka za semantizaciju

Nivo fizičkih događaja realizuje mehanizme evidencije, upravljanja i prikupljanja parametara sa fizičkih ili logičkih senzora, koji su posredstvom senzorskih biblioteka integrisani u kontekstualnu platformu. Nivo kontekstualnih parametara realizuje virtuelne senzore, te omogućava upravljanje događajima i njihovo raspoređivanje. Nivo semantizacije realizuje algoritme za semantizaciju, odnosno mehanizme za parsiranje kontekstualnih polisa, upravljanje kontekstualnim polisama i P-modelima i sl. Nivo agregacije dodatno uopštava zaključke donesene na nivou semantizacije i obezbeđuje dodatne izlaze, poput zauzetosti korisnika i sl. Logički,

nivo fizičkih događaja i nivo kontekstualnih parametara pripadaju nivou mreže, dok nivo semantizacije i nivo agregacije pripadaju srednjem nivou.

Blok za kontekstualizaciju podjeljen je na nivoe kao što je prikazano na Slici 58.



Slika 58. Nivojska organizacija bloka za kontekstualizaciju

Fizički mrežni nivo omogućava pristup rukovaocima svih uređaja koji su dostupni kontekstualnoj platformi (senzorima i aktuatorima), kao i upravljanje radom svih senzorskih i aktuatorskih biblioteka. Simbolički nivo omogućava evidenciju svih instanci senzora i aktuatora, njihovo mapiranje na čitljiva imena na osnovu kojih se mogu adresirati u kontekstualnim skriptovima, kao i skup metoda kojima se obavlja registracija za određene događaje, čekanje tih događaja, te prikupljanje parametara uređaja i izdavanje naredbi uređajima. Nivo adaptacije obezbeđuje kompletan ekstenzioni API za kontekstualne skripte, omogućavajući kontekstualnim skriptovima da koriste funkcije posredstvom kojih se pristupa svim dostupnim funkcionalnostima kontekstualne platforme. Nivo akcionih modela omogućava agregaciju poziva u skriptovne biblioteke, posredstvom kojih se iz kontekstualnih skriptova omogućava izvršavanje više povezanih sekvenci naredbi, odnosno omogućavaju se složene interaktivnosti višeg nivoa apstrakcije. Logički, fizički mrežni nivo i simbolički nivo pripadaju nivou mreže, dok nivo adaptacije i nivo akcionih modela pripadaju srednjem nivou.

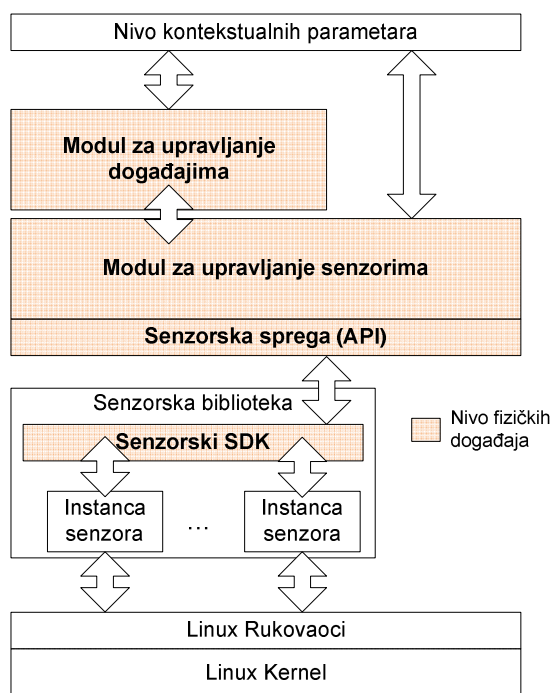
U nastavku poglavlja data je detaljna arhitektura programske podrške za svaki od navedenih nivoa. U sklopu opisa dat je blok dijagram koji prikazuje organizaciju programske podrške po modulima. Zatim, dat je UML dijagram klasa za svaki modul. U cilju bolje ilustracije rada pojedinih modula, dati su UML dijagrami sekvence koji ilustruju važne operacije. Dati UML dijagrami ne predstavljaju kompletnu

Model arhitekture platforme

specifikaciju, već daju detalje do određenog nivoa potrebnog za predstavljanje modela arhitekture programske podrške kontekstualne platforme.

3.3.2.1 Nivo fizičkih događaja

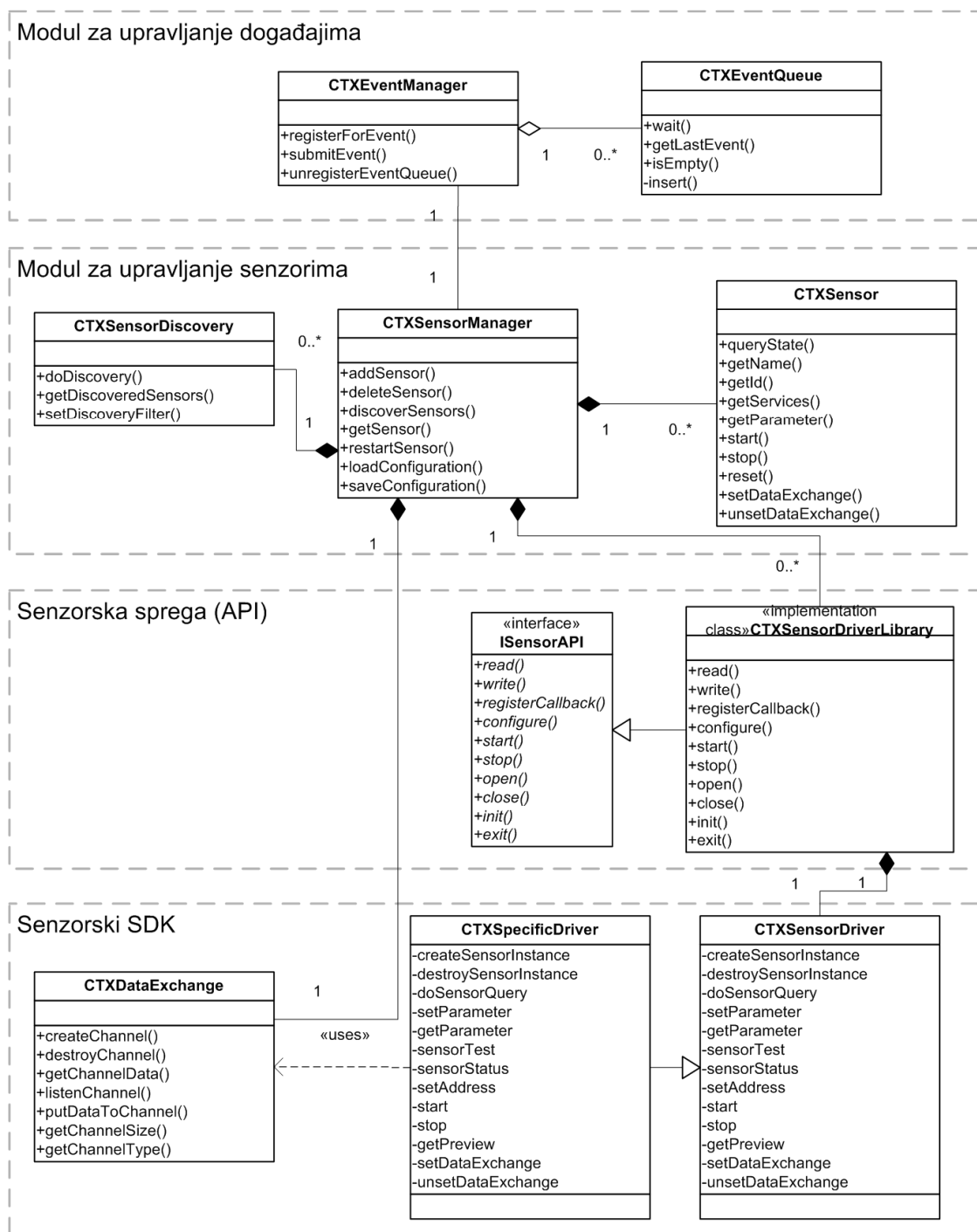
Blok dijagram programske podrške nivoa fizičkih događaja dat je na Slici 59.



Slika 59. Blok dijagram programske podrške nivoa fizičkih događaja

Nivo fizičkih događaja obezbeđuje sledeće funkcionalnosti: (1) spregu za povezivanje sa rukovaocima sensorima, realizovanih u vidu senzorskih biblioteka, kao i SDK za pisanje rukovalaca sensorima; (2) mehanizam učitavanja senzorskih biblioteka, njihove evidencije, inicijalizacije i dodele instancama senzora, realizovan kao modul za upravljanje sensorima; (3) mehanizam registrovanja za događaje koji nastupe na instancama senzora, registrovanje i isporuku događaja, realizovan kao modul za upravljanje događajima; (4) mehanizam dobavljanja i postavljanja parametara rada senzora, realizovan u okviru modula za upravljanje sensorima.

Na Slici 60 dat je UML dijagram klasa koje realizuju module nivoa fizičkih događaja.



Slika 60. UML dijagram klasa za realizaciju nivoa fizičkih događaja

Klasa `CTXSensorManager` kreira spisak svih dostupnih senzora na osnovu ranije snimljene konfiguracije, na osnovu dostupnih senzorskih biblioteka ili na zahtev systemske korisničke sprege. Sve instance senzora u okviru kontekstualne platforme realizovane su klasom `CTXSensor`. `CTXSensorDriverLibrary` i `CTXDiscovery` su deljene klase koje su realizovane u okviru senzorskih biblioteka.

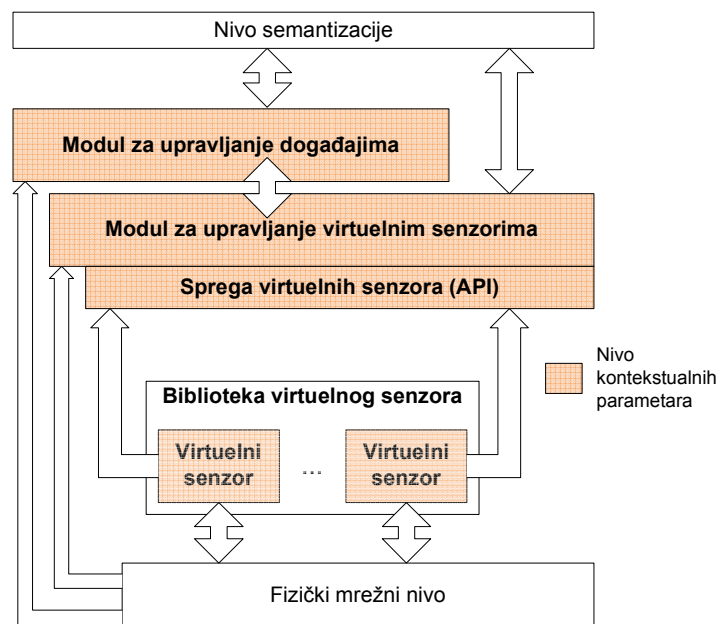
Model arhitekture platforme

CTXSensorDriverLibrary obezbeđuje primitive za upravljanje jednom klasom senzorskih uređaja, dok *CTXDiscovery* obezbeđuje mehanizme za pronalaženje uređaja. Tako, npr. *UPnP discovery* mehanizam može da bude integrisan u kontekstualnu platformu nasleđivanjem klase *CTXDiscovery*. *CTXSensorDriverLibrary* razvija se nezavisno od ostatka programske podrške kontekstualne platforme, uz jedino ograničenje da mora da implementira interfejs *ISensorAPI* da bi integracija sa kontekstualnom platformom bila moguća. Za brži razvoj senzorskih biblioteka, priložena je klasa *CTXSensorDriver*, čijim se nasleđivanjem realizuje konkretan rukovaoc sensorom. Klasa *CTXSpecificDriver* data je kao šablon za kreiranje novog rukovaoca. *CTXSensorDriver* apstrahuje funkcionalnosti kao što su upravljanje instancama senzora, kreiranje i uništavanje instanci, upravljanje razmenom podataka, preuzimanje događaja i njihovo usmeravanje i sl. Klasa *CTXDataExchange* se instancira u okviru kontekstualne platforme, i referenca na tu klasu se prosleđuje rukovaocima sensorima. Ovo omogućava sensorima da obavljaju upis velike količine primljenih podataka u zajednički bafer, tako da se omogući obrada ovih podataka od strane virtuelnih senzora na višim nivoima. Klasa *CTXPhyEventManager* omogućava registraciju modula višeg nivoa za događaje na određenim sensorima, te za prijem tih događaja, odnosno ubacivanje u red čekanja realizovan klasom *CTXEventQueue*. *CTXSensorManager* svakom rukovaocu prosleđuje statičku metodu *submitEvent* klase *CTXPhyEventManager* kao povratnu funkciju, koja se poziva po prijemu novog događaja. Na ovaj način *CTXPhyEventManager* se obaveštava o novom događaju, nakon čega signalizira svim zainteresovanim redovima čekanja. Nivo kontekstualnih parametara, opisan u sledećem poglavlju, povezan je sa nivoom fizičkih događaja posredstvom klase *CTXPhyEventManager*, za prijem novih događaja, klase *CTXSensorManager*, za enumeraciju dostupnih senzora, i klase *CTXDataExchange*, za pristup podacima sa fizičkih senzora u realnom vremenu.

Rukovaoci sensorima mogu komunicirati sa sensorom posredstvom neke od sprega prisutnih na centralnoj kontekstualnoj platformi, ili mogu adresirati odgovarajući kontekstualni modul. U ovom slučaju parametri senzora se proširuju tako da uključe potrebne podatke za identifikaciju kontekstualnog modula (npr. njegovu adresu na mreži).

3.3.2.2 *Nivo kontekstualnih parametara*

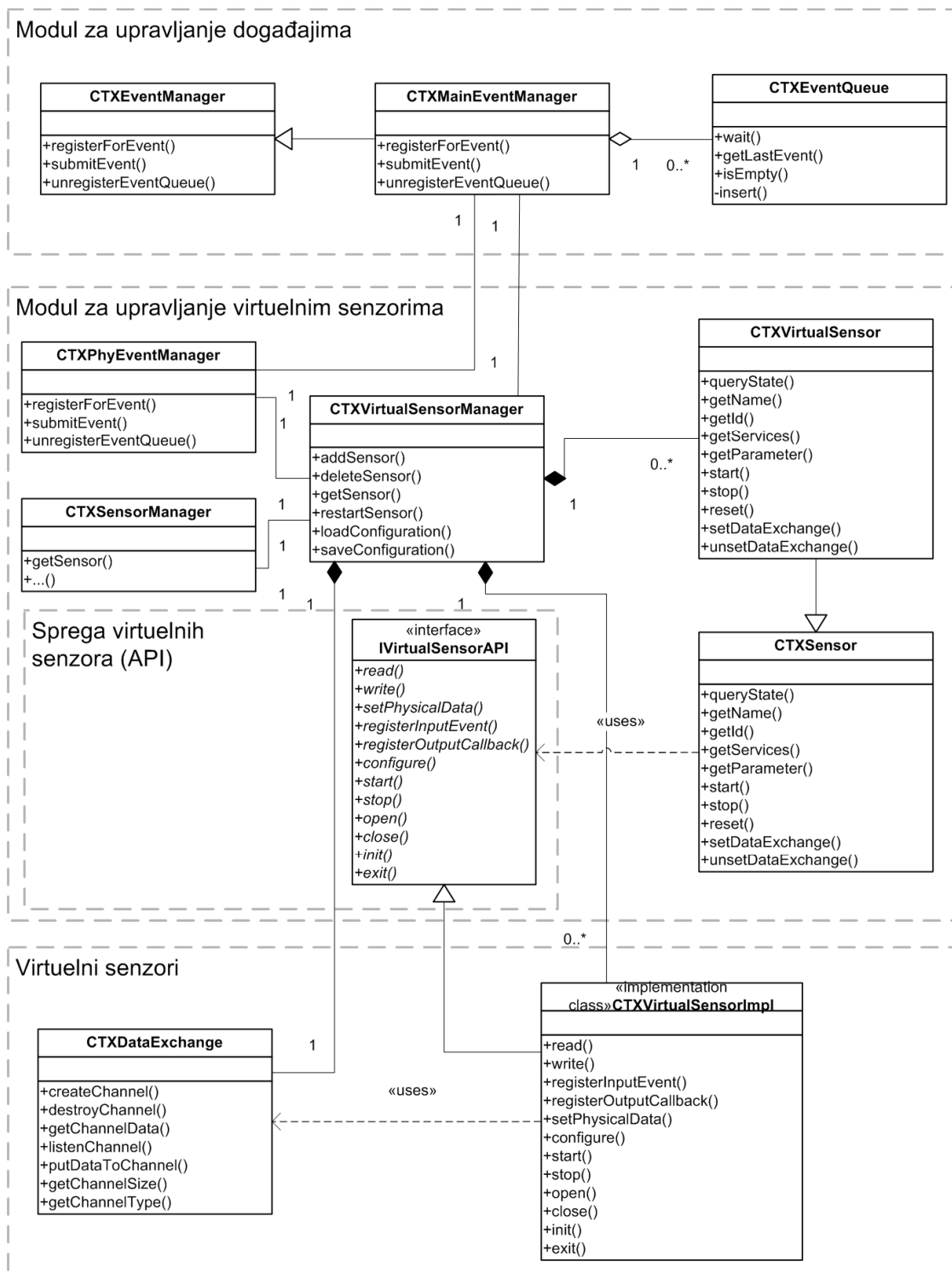
Nivo kontekstualnih parametara obezbeđuje sledeće funkcionalnosti: (1) spregu sa bibliotekama virtuelnih senzora, posredstvom koje virtuelni senzori bivaju obavешteni o fizičkim događajima i mogu da preuzmu fizičke podatke za obradu; (2) mehanizam učitavanja biblioteka virtuelnih senzora, njihove evidencije, inicijalizacije i dodele instancama virtuelnih senzora, realizovan kao modul za upravljanje virtuelnim sensorima; (3) mehanizam registrovanja za događaje koji nastupe na virtuelnim sensorima, detekciju i isporuku događaja, kao i prosleđivanje fizičkih događaja, realizovan kao modul za upravljanje događajima; (4) mehanizam dobavljanja i postavljanja parametara rada virtuelnih senzora, realizovan u okviru modula za upravljanje virtuelnim sensorima. Blok dijagram programske podrške nivoa kontekstualnih parametara dat je na Slici 61.



Slika 61. Blok dijagram programske podrške nivoa kontekstualnih parametara

Na Slici 62 dat je UML dijagram klasa koje realizuju module nivoa kontekstualnih parametara.

Model arhitekture platforme



Slika 62. UML dijagram klasa za realizaciju nivoa kontekstualnih parametara

Klasa `CTXVirtualSensorManager` inicijalizuje sve biblioteke virtuelnih senzora, i kreira odgovarajuće instance virtuelnih senzora korišćenjem klase `CTXVirtualSensor`.

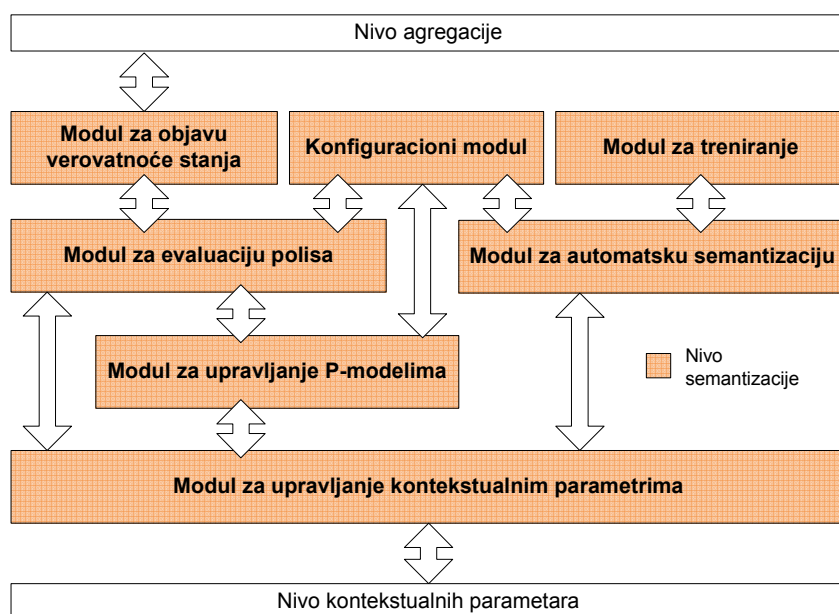
Virtuelni senzori realizovani su u okviru biblioteka kao klase tipa *CTXVirtualSensorImpl* koje implementiraju interfejs *IVirtualSensorAPI*, posredstvom kojeg komuniciraju sa kontekstualnom platformom. *CTXVirtualSensorManager* registrovan je za prijem svih fizičkih događaja korišćenjem *CTXPhyEventManager* klase sa nivoa fizičkih događaja. Ovi fizički događaji se prosleđuju virtuelnim sensorima korišćenjem poziva *registerInputEvent*. Virtuelnim sensorima se takođe prosleđuje instanca klase *CTXDataExchange*, jedinstvena na nivou kontekstualne platforme. Na ovaj način virtuelni senzor, kada god primi novi događaj može da preuzme potrebne podatke za obradu iz odgovarajućeg bafera (ukoliko je to potrebno za njegov rad). Klasa *CTXMainEventManager* omogućava registraciju modula višeg nivoa za događaje na određenim virtuelnim sensorima, kao i prijem tih događaja, odnosno ubacivanje u red čekanja realizovan klasom *CTXEventQueue*. Dodatno, *CTXMainEventManager* prosleđuje i fizičke događaje ukoliko za njih postoji interesovanje, na osnovu veze sa klasom *CTXPhyEventManager*. *CTXVirtualSensorManager* svakom virtuelnom senzoru prosleđuje statičku metodu *submitEvent* klase *CTXMainEventManager* kao povratnu funkciju, koja se poziva po prijemu novog događaja. Na ovaj način *CTXMainEventManager* se obaveštava o novom događaju, nakon čega signalizira svim zainteresovanim redovima čekanja. Nivo semantizacije, opisan u sledećem poglavlju, komunicira sa nivoom kontekstualnih parametara i, posredno, sa nivoom fizičkih događaja, korišćenjem klase *CTXMainEventManager*, posredstvom koje preuzima kontekstualne parametre.

3.3.2.3 Nivo semantizacije

Nivo semantizacije obezbeđuje sledeće funkcionalnosti: (1) prijem kontekstualnih parametara, njihovu obradu i skladištenje, odnosno kreiranje istorije, realizovano u okviru modula za upravljanje kontekstualnim parametrima; (2) izvršavanje i evaluaciju kontekstualnih polisa, u cilju semantizacije konteksta, realizovano u okviru modula za evaluaciju polisa; (3) automatsku semantizaciju, ukoliko je omogućena, realizovanu u okviru modula za automatsku semantizaciju; (4) upravljanje P-modelima i njihovo korišćenje u kontekstualnim polisama, realizovano u okviru modula za upravljanje P-modelima; (5) izračunavanje i objavljivanje verovatnoće stanja okruženja, realizovano u okviru modula za objavljivanje

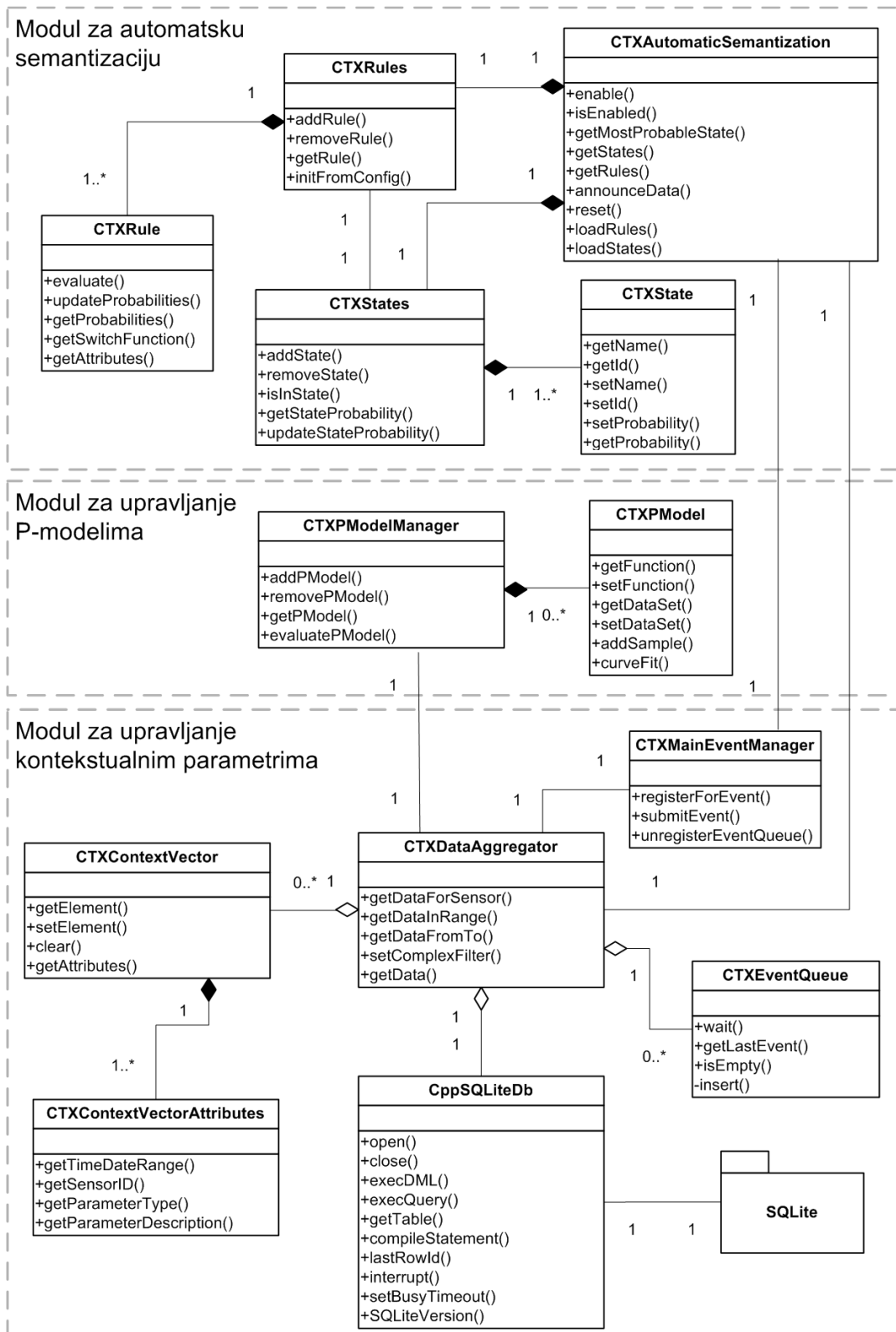
Model arhitekture platforme

verovatnoće stanja; (6) konfigurisanje rada nivoa semantizacije, u smislu izbora kontekstualnih polisa i P-modela, uključivanja/isključivanja automatske semantizacije, inicijalizacije i sl. realizovano u okviru konfiguracionog modula; (7) prilaganje ulaznih podataka za treniranje, u okviru modula za treniranje. Blok dijagram programske podrške nivoa semantizacije dat je na Slici 63.



Slika 63. Blok dijagram programske podrške nivoa semantizacije

Na Slici 64 dat je UML dijagram klasa koje realizuju modul za upravljanje kontekstualnim parametrima, modul za upravljanje P-modelima i modul za automatsku semantizaciju. Središte modula za upravljanje kontekstualnim parametrima je klasa *CTXDataAggregator*. Ova klasa je pretplaćena za događaje koje objavljuje *CTXMainEventManager*. Nakon prijema događaja, *CTXDataAggregator* razvrstava događaje po sensorima i smešta ih u delta liste, po redosledu prijema, koje periodično čuva u *SQLite* bazi podataka posredstvom klase *CppSQLiteDb*. *CTXDataAggregator* omogućava postavljanje filtera pretrage, koji odgovara realizaciji prekidačke funkcije u kontekstualnoj polisi. Nakon upita, *CTXDataAggregator* vraća vektor traženih kontekstualnih parametara, u obliku klase *CTXContextVector*. Modul za upravljanje P-modelima realizovan je klasom *CTXPModelManager*, koja održava spisak P-modela kao instanci klase *CTXPModel*.

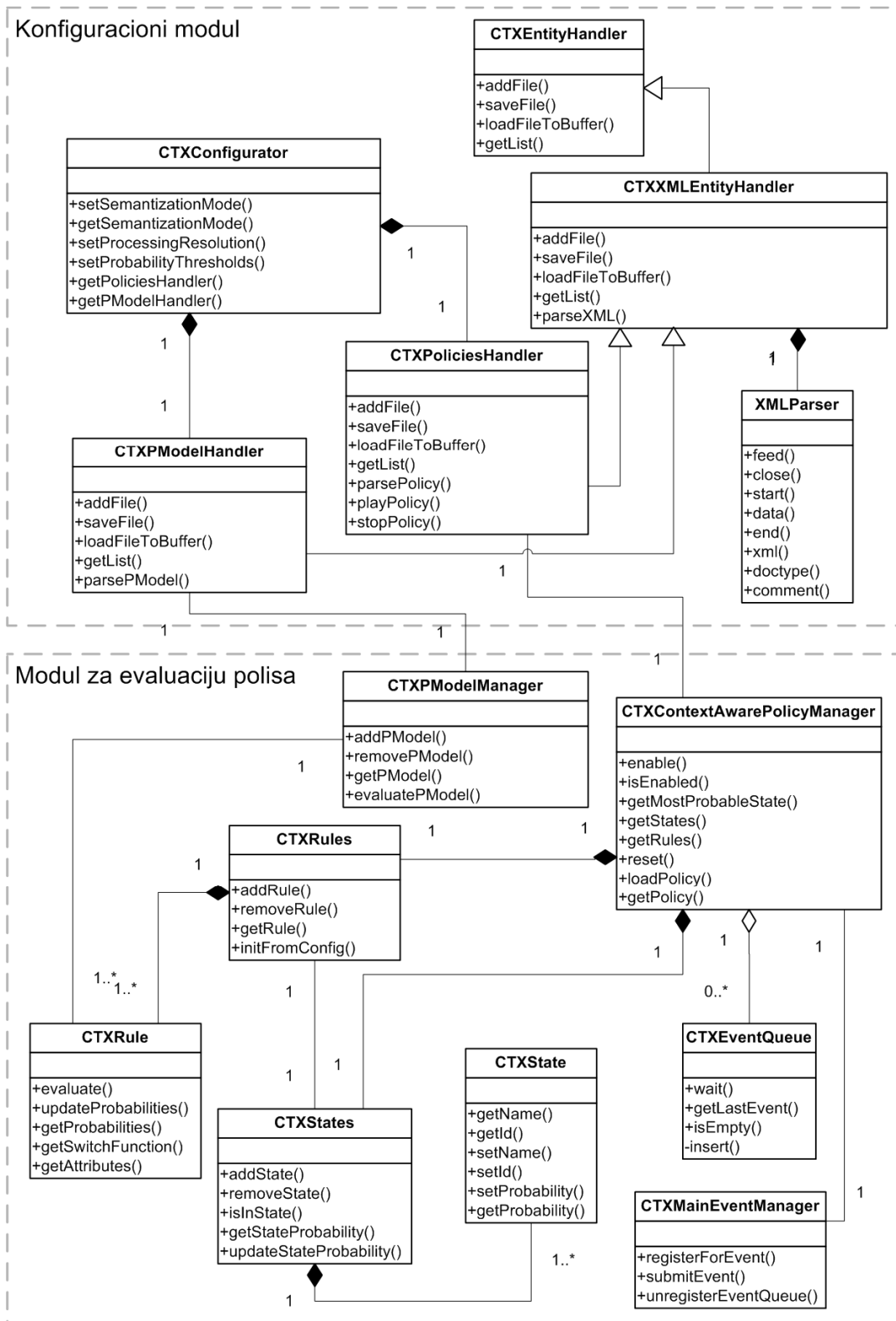


Slika 64. UML dijagram klasa modula za upravljanje kontekstualnim parametrima, modula za upravljanje P-modelima i modula za automatsku semantizaciju

Model arhitekture platforme

CTXPModel instance opisuju P-model, uz mogućnost njegovog zadavanja / proširivanja dodatnim ulaznim semplovima, na osnovu kojih klasa ima mogućnost automatskog izračunavanja polinomske funkcije korišćenjem linearne regresije. Klasa *CTXAutomaticSemantization* realizuje algoritam automatske semantizacije. Trenutni skup pravila i stanja, sa njihovim funkcijama, odnosno verovatnoćama, čuva se u klasama *CTXRules* i *CTXStates*, respektivno. *CTXAutomaticSemantization* omogućava primanje trening unosa korišćenjem metode *announceData*.

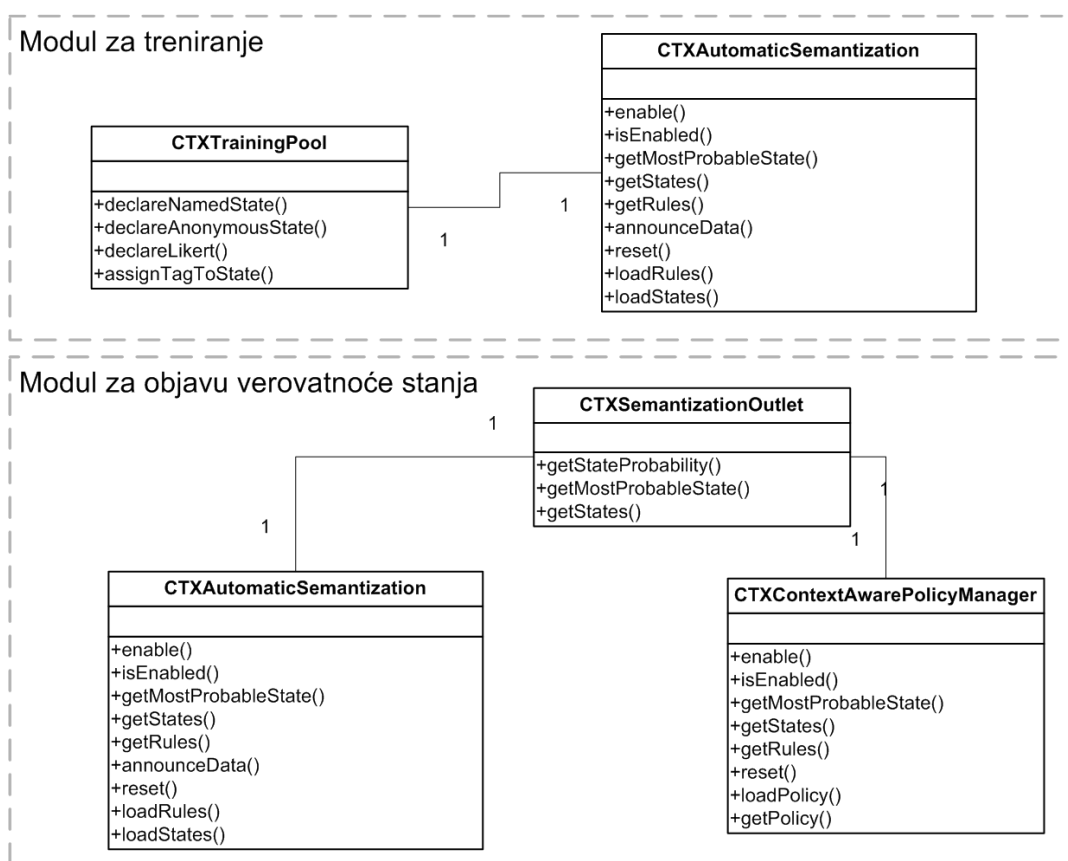
Na Slici 65 dat je UML dijagram klasa koje realizuju modul za evaluaciju polisa i konfiguracioni modul. Klasa *CTXContextAwarePolicyManager* realizuje algoritam za semantizaciju konteksta zasnovanu na polisama. Metoda *loadPolicy* obavlja inicijalizaciju trenutno aktivne kontekstualne polise. Parsiranje polise vođeno je događajima koje objavljuje *CTXMainEventManager*, na koje je *CTXContextAwarePolicyManager* pretplaćen na osnovu sadržaja polise. Nakon parsiranja polise, ista se čuva u internoj reprezentaciji korišćenjem klasa *CTXRules*, *CTXStates*, *CTXRule* i *CTXState* (koje se između ostalog koriste i u okviru modula za automatsku semantizaciju). U ovom slučaju *CTXRule* klasa ima mogućnost dobavljanja verovatnoće na osnovu P-modela, zahvaljujući vezi sa instancom klase *CTXPModelManager*. Klasa *CTXConfigurator* obezbeđuje sve radnje vezane za inicijalizaciju i praćenje stanja rada semantizacionog nivoa. Tako, posredstvom klase *CTXPModelHandler* obavlja se dodavanje i parsiranje novog P-modela, pre nego što se isti prosledi instanci *CTXPModelManager* klase kao *CTXPModel* objekat. *CTXPoliciesHandler* dodaje, uklanja i parsira kontekstualne polise, te je u vezi sa *CTXContextAwarePolicyManager* klasom. Upravljanje P-modelima i kontekstualnim polisama svodi se na operacije sa XML zapisom datoteka, što omogućava apstrahovanje osnovnih operacija u klasu pretka *CTXXMLEntityHandler*, koja za parsiranje XML datoteka koristi Expat *XMLParser*. Konačno, *CTXEntityHandler* enkapsulira osnovne operacije za rad sa datotekama. Dodatno, *CTXConfigurator* omogućava izbor trenutnog režima rada nivoa semantizacije (da li se koristi semantizacija zasnovana na polisama ili automatska semantizacija definiše se pozivom metode *setSemantizationMode*).



Slika 65. UML dijagram klasa modula za evaluaciju polisa i konfiguracionog modula

Model arhitekture platforme

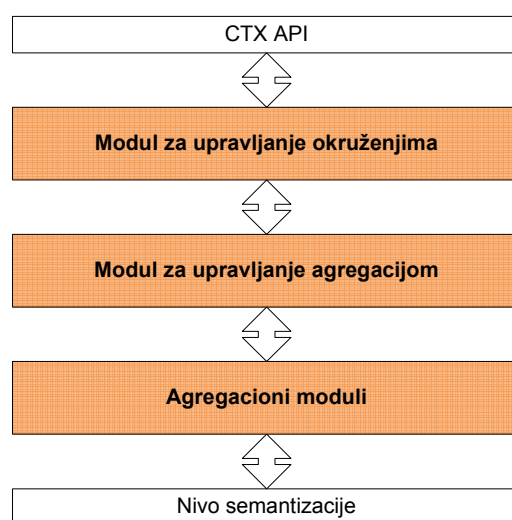
Na Slici 66 dat je UML dijagram klase koje realizuju modul za objavu verovatnoće stanja i modul za treniranje. Modul za objavu verovatnoće stanja omogućava višim nivoima, kao što je nivo agregacije i CTX API, da pristupaju izlaznim informacijama iz postupka semantizacije. Klasa *CTXSemantizationOutlet* omogućava izlistavanje svih stanja, verovatnoća za svako od stanja kao i najverovatnije stanje. *CTXSemantizationOutlet* dolazi do ovih informacija na osnovu veze sa modulom za evaluaciju polisa (klasa *CTXContextAwarePolicyManager*) i modulom za automatsku semantizaciju (klasa *CTXAutomaticSemantization*). Modul za treniranje u osnovi je realizovan kao klasa *CTXTrainingPool* posredstvom koje se obavljaju deklaracije stanja. Ova klasa, dalje, koristi metodu *announceData* klase *CTXAutomaticSemantization* da izvrši postupak treniranja. Nivo agregacije, koji je opisan u sledećem poglavlju, komunicira sa klasama *CTXAutomaticSemantization*, *CTXSemantizationOutlet* i *CTXTrainingPool* za dodatnu semantizaciju.



Slika 66. UML dijagram klase modula za objavu verovatnoće stanja i modula za treniranje

3.3.2.4 Nivo agregacije

Nivo agregacije omogućava sledeće funkcionalnosti: (1) realizaciju dodatnih algoritama semantizacije konteksta, zasnovanih na rezultatima i mehanizmima rada postojećih algoritama na nivou semantizacije, u vidu agregacionih modula; (2) inicijalizaciju, evidenciju i upravljanje radom agregacionih modula, te prikupljanje rezultata njihovog rada, realizovano u okviru modula za upravljanje agregacijom i (3) mogućnost rada kontekstualne platforme za više odvojenih okruženja, pri čemu se evidencija okruženja, inicijalizacija i upravljanje okruženjima obavlja u okviru modula za upravljanje okruženjima. Blok dijagram programske podrške nivoa agregacije dat je na Slici 67.

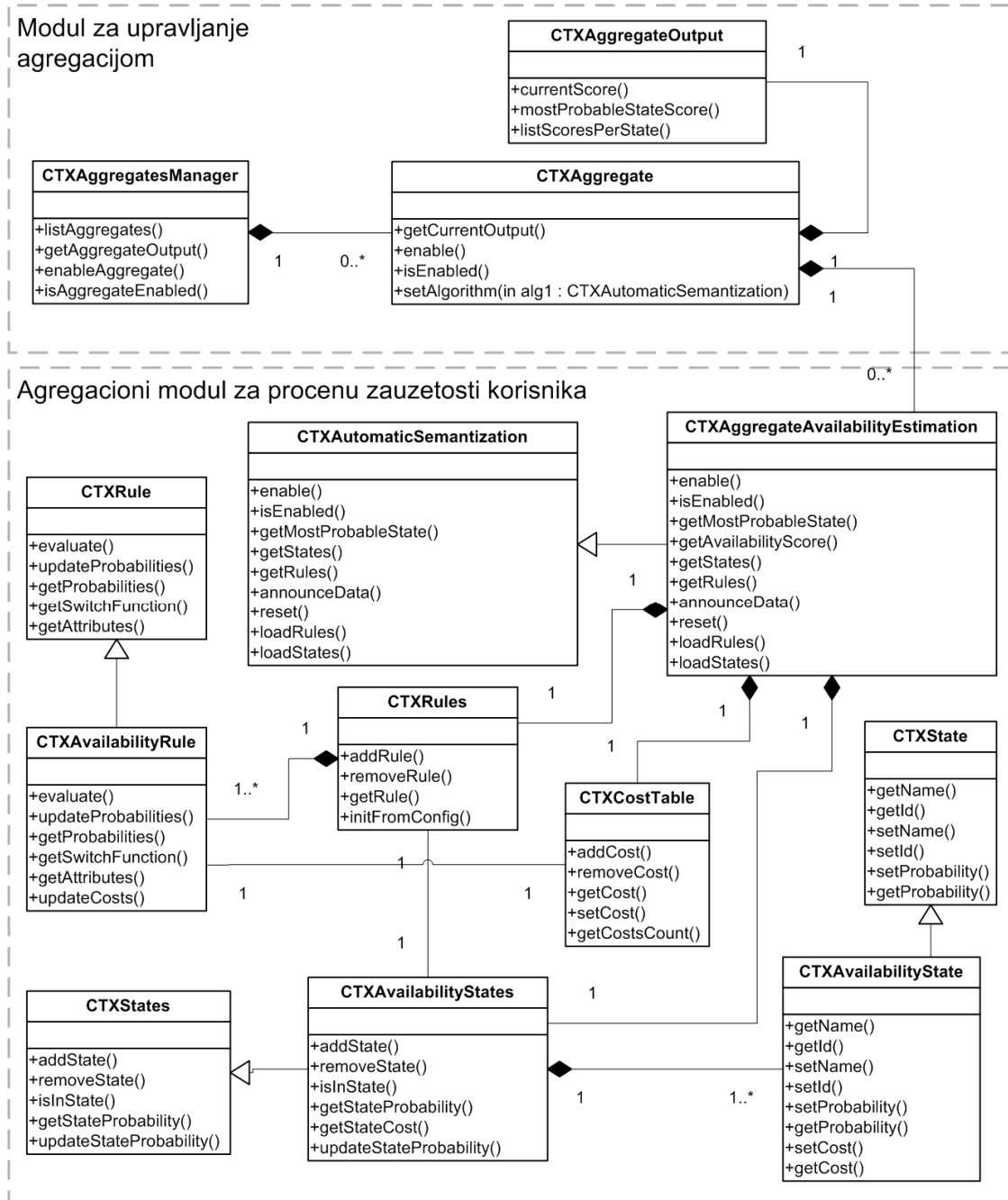


Slika 67. Blok dijagram programske podrške nivoa agregacije

Na Slici 68 dat je UML dijagram klasa koje realizuju agregacioni modul za procenu zauzetosti korisnika i modul za upravljanje agregacijom. Agregacioni moduli realizuju se nasleđivanjem jedne od baznih klasa *CTXAutomaticSemantization* ili *CTXContextAwarePolicyManager*, te proširivanjem njihovih funkcionalnosti. Na ovaj način agregacioni moduli omogućavaju izračunavanje različitih izlaza vezanih za opšte stanje okruženja, odnosno za svako moguće stanje okruženja. Ukoliko je potrebno čuvati dodatne informacije vezano za stanja, odnosno obezbediti dodatna izračunavanja u okviru pravila semantizacije, agregacioni moduli mogu koristiti sopstvene klase koje nasleđuju *CTXRules*, *CTXRule*, *CTXStates* i *CTXState*. Tako, na primer, agregacioni modul za procenu zauzetosti realizuje potrebni algoritam korišćenjem klase *CTXAggregateAvailabilityEstimation*, koja nasleđuje

Model arhitekture platforme

CTXAutomaticEstimation, s tim što omogućava izračunavanje stepena zauzetosti korisnika i njegovo objavljivanje (*getAvailabilityScore*).

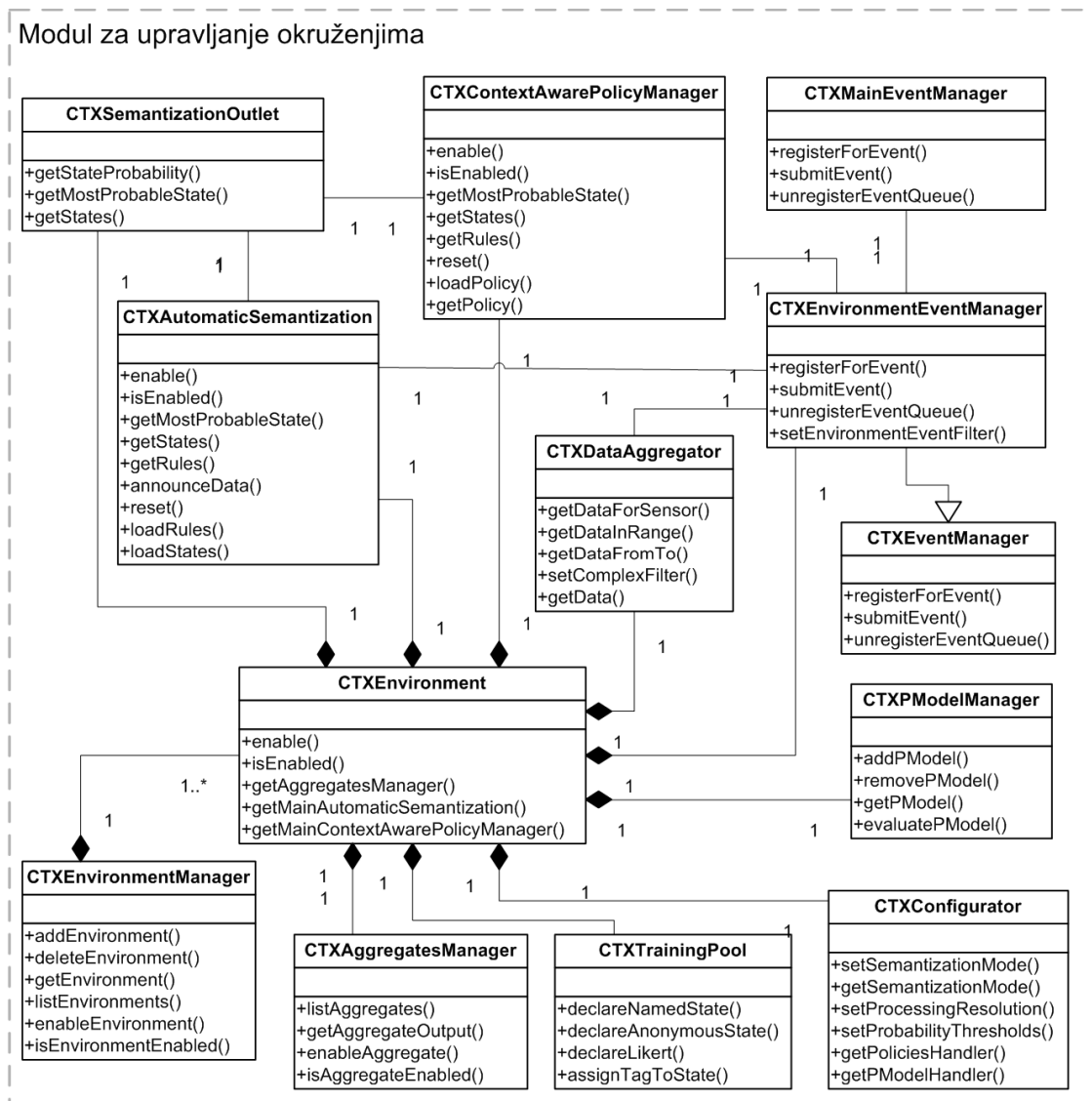


Slika 68. UML dijagram klasa agregacionog modula za procenu zauzetosti korisnika i modula za upravljanje agregacijom

CTXAvailabilityRule proširuje *CTXRule* u smislu da omogućava ažuriranje cene koja se dodeljuje stanju u okviru tog pravila, u odnosu na tabelu cena koja se realizuje

kao *CTXCostTable*. *CTXAvailabilityStates*, odnosno *CTXAvailabilityState* nasleđuju *CTXStates*, odnosno *CTXState* da omoguće dobavljanje cene trenutnog stanja okruženja. Modul za upravljanje agregacijom realizovan je klasom *CTXAggregatesManager*, koja upravlja instancama klase *CTXAggregate*. *CTXAggregate* klasa omogućava objedinjen pogled na raznorodne realizacije agregacionih modula, poput dobavljanja izlaza i omogućavanja/onemogućavanja rada agregacionog modula. *CTXAggregate* klasa pakuje izlaz u *CTXAggregateOutput*, koji se koristi na višim nivoima.

Na Slici 69 dat je UML dijagram klasa koje realizuju modul za upravljanje okruženjima.

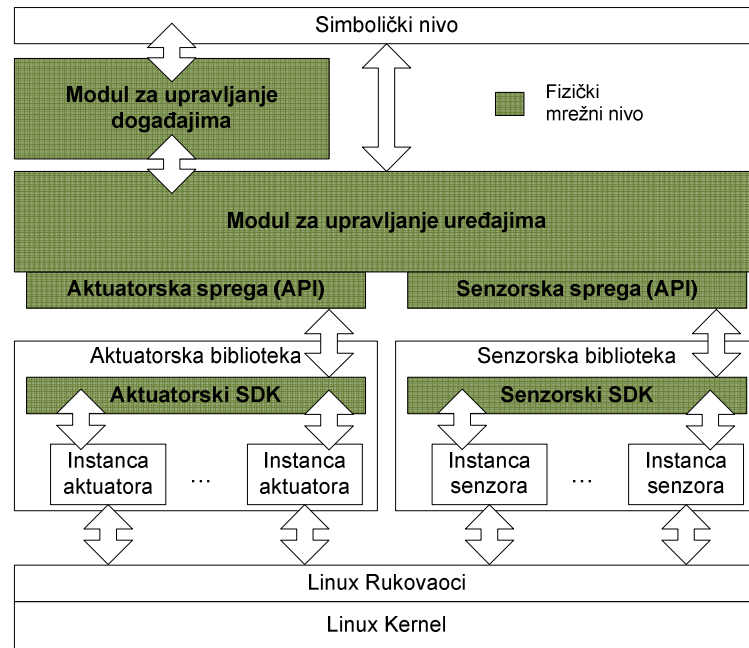


Slika 69. UML dijagram klasa modula za upravljanje okruženjima

Modul za upravljanje okruženjima omogućava kreiranje instanci potrebnih podređenih modula za svako od okruženja ponaosob. Klasa *CTXEnvironmentManager* omogućava dodavanje, uklanjanje i listanje dostupnih okruženja. Za svako od aktivnih okruženja, *CTXEnvironmentManager* instancira po jedan objekat tipa *CTXEnvironment*. Prilikom konstruisanja ovog objekta, kreiraju se instance svih potrebnih podređenih modula. Dupliciranje instanci modula za svako od okruženja obavlja se najvećim delom u okviru nivoa semantizacije i nivoa agregacije. *CTXEnvironmentEventManager* je posebna instanca koja nasleđuje *CTXEventManager* i koja omogućava filtriranje događaja koji pristižu sa senzora koji ne pripadaju traženom okruženju. *CTXEnvironmentEventManager* zamenjuje *CTXMainEventManager* pri komunikaciji sa svim modulima na nivou semantizacije. Operacije na nivou kontekstualnih parametara i nivou fizičkih događaja ostaju identične, sa naznakom da svaka evidentirana instanca senzora u okviru svojih atributa poseduje identifikator koji je vezan za ciljno okruženje.

3.3.2.5 Fizički mrežni nivo

Fizički mrežni nivo bloka za kontekstualizaciju delimično se preklapa sa nivoom fizičkih događaja bloka za semantizaciju, s obzirom da se sa ovog nivoa rukuje uređajima koji su dostupni kontekstualnoj platformi. S druge strane, fizički mrežni nivo dat je posebno kao referenca za one realizacije kontekstualnih platformi koje uključuju rad sa aktuatorima (što u opštem slučaju nije obavezujuća funkcionalnost). Blok za kontekstualizaciju omogućava rad i sa sensorima i sa aktuatorima, s obzirom na potrebu da se kontekstualnim aplikacijama pruži što šira paleta mogućnosti.



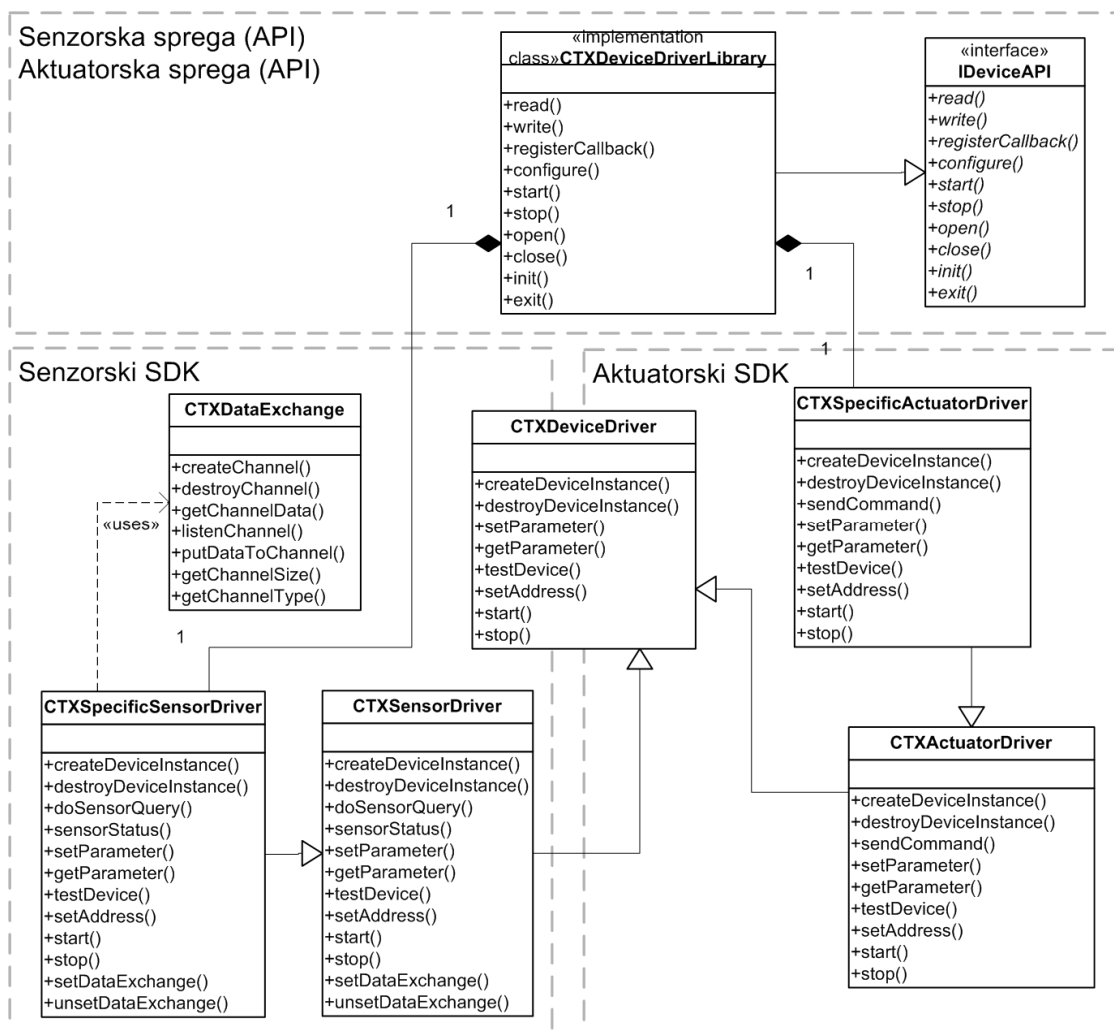
Slika 70. Blok dijagram programske podrške fizičkog mrežnog nivoa

Dakle, fizički mrežni nivo omogućava sve funkcionalnosti navedene za nivo fizičkih događaja u poglavlju 3.3.2.1, sa razlikom što je modul za upravljanje sensorima sastavni deo modula za upravljanje uređajima. Dodatno, on omogućava: (1) spregu za povezivanje sa rukovaocima aktuatorima, realizovanih u vidu aktuatorskih biblioteka, kao i SDK za pisanje rukovalaca aktuatorima; (2) mehanizam učitavanja aktuatorskih biblioteka, njihove evidencije, inicijalizacije i dodele instancama aktuatora, realizovan u okviru modula za upravljanje uređajima; (3) mehanizam dobavljanja i postavljanja parametara rada aktuatora, realizovan u okviru modula za upravljanje uređajima. Blok dijagram programske podrške fizičkog mrežnog nivoa dat je na Slici 70.

Na Slici 71 dat je UML dijagram klasa koje realizuju senzorski i aktuatorski SDK, kao i senzorsku i aktuatorsku spregu (API). Svaki senzor i aktuator realizovan je kao biblioteka *CTXDeviceDriverLibrary*, koja implementira *IDeviceAPI*. Ukoliko biblioteka realizuje rukovalac sensorima, koristi se senzorski SDK. Svaki rukovalac sensorima nasleđuje klasu *CTXSensorDriver*, koja obezbeđuje osnovne funkcionalnosti rukovaoca, poput kreiranja i uništavanja instanci, dodele memorijskih bafera za razmenu podataka, prosleđivanje zahteva odgovarajućim obrađivačima, poput upita stanja senzora i sl. Ukoliko biblioteka realizuje rukovalac aktuatorima, koristi se aktuatorski SDK. Svaki rukovalac aktuatorima nasleđuje klasu

Model arhitekture platforme

CTXActuatorDriver, koja obezbeđuje osnovne funkcionalnosti rukovaoca, uključujući i mehanizme za upućivanje naredbi aktuatorima. *CTXActuatorDriver* i *CTXSensorDriver* nasleđuju klasu *CTXDeviceDriver*, koja obezbeđuje minimalni skup funkcionalnosti zajedničkih za rukovaoce uređajima.



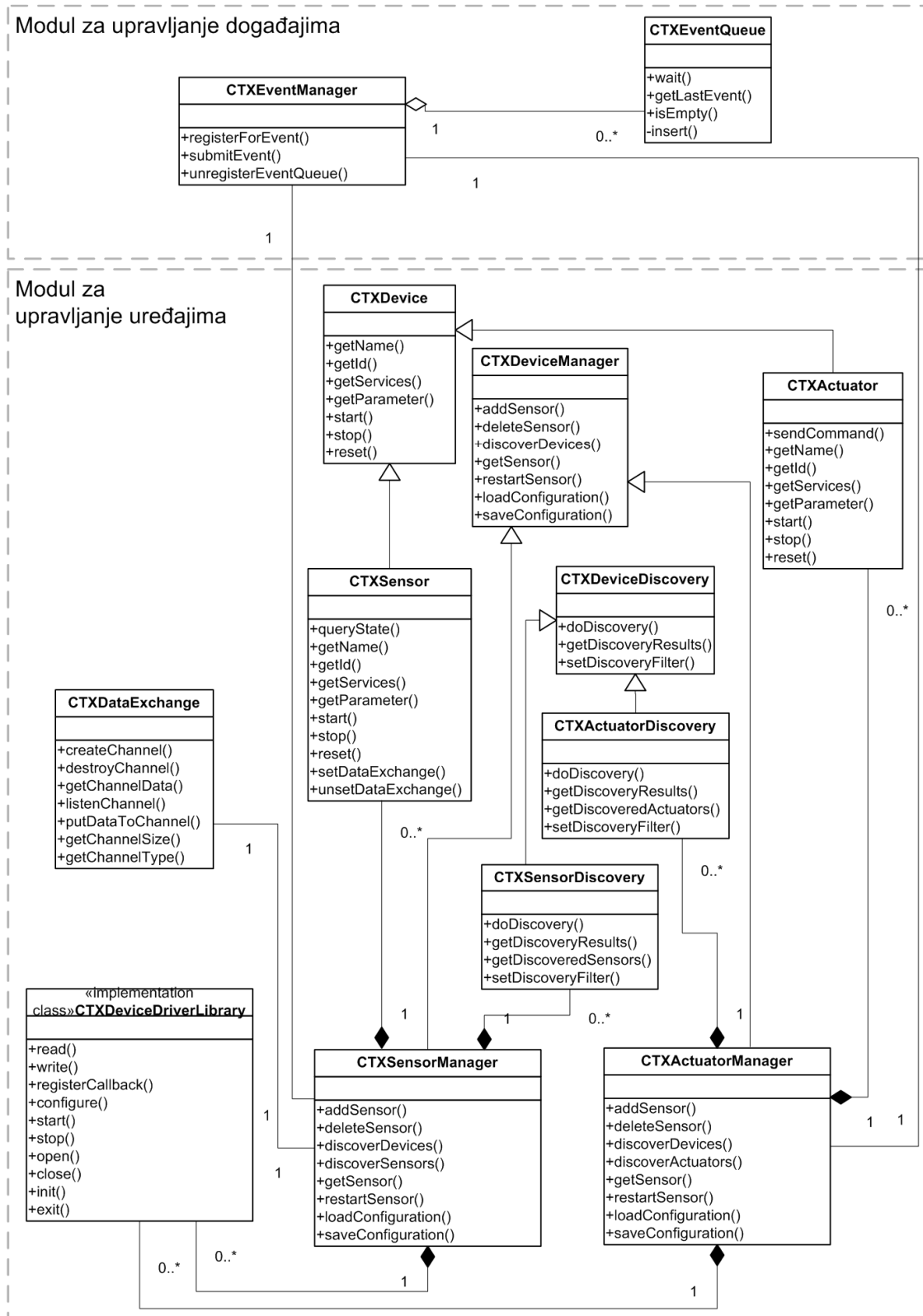
Slika 71. UML dijagram klasa koje realizuju senzorski i aktuatorski SDK i senzorsku i aktuatorsku spregu

Na Slici 72 dat je UML dijagram klasa modula za upravljanje uređajima i modula za upravljanje događajima. Klasa *CTXSensorManager* upravlja svim sensorima prisutnim u okviru kontekstualne platforme, dok klasa *CTXActuatorManager* upravlja aktuatorima. Obe klase nasleđuju bazni *CTXDeviceManager*, koji obezbeđuje mehanizme učitavanja biblioteka uređaja,

kreiranje instanci i vođenje evidencije. Klasa *CTXSensorManager* sve instance senzora predstavlja kao objekte tipa *CTXSensor*, dok klasa *CTXActuatorManager* sve instance aktuatora predstavlja kao objekte tipa *CTXActuator*. Metode pronalaženja senzora, odnosno aktuatora, realizovane su klasama *CTXSensorDiscovery* i *CTXActuatorDiscovery*, respektivno. Suštinska razlika u evidenciji i realizaciji senzora u odnosu na aktuatore, je što se senzori ponašaju kao ulazni uređaji, odnosno omogućavaju upit u trenutno stanje, dok se aktuatorima izdaju komande. Dodatno, senzori podržavaju mehanizam izdavanja veće količine podataka koje mogu da razmenjuju sa višim nivoima ili sa drugim sensorima, realizovan kao instanca klase *CTXDataExchange*. Senzori i aktuatori objavljuju događaje koji su zavisni od sopstvenog stanja. Senzori objavljuju događaje pri svakoj novoj detekciji određenog parametra koji ih aktivira. Aktuatori objavljuju događaje koji su vođeni najčešće samim komandama, odnosno, klasa *CTXActuator* izaziva slanje događaja pri svakom uspešnom izvršavanju određene komande, za potrebe sinhronizacije sa radom nekih drugih aktuatora u kontekstualnoj aplikaciji. Dodatno, aktuatori mogu da objavljuju asinhroni rezultat komande, ukoliko komanda traje neobično dugo, odnosno, ukoliko je moguće promeniti unutrašnje stanje aktuatora na neki drugi način (npr. moguće je aktivirati prekidač za kontrolu svetla korišćenjem kontekstualne platforme, ali se on može aktivirati i ručnim pritiskanjem tastera, pri čemu je neophodno obavestiti kontekstualnu platformu da se takva akcija dogodila). Objavljivanje događaja obavlja se korišćenjem povratne funkcije objekta tipa *CTXEventManager*. *CTXSensorManager* i *CTXActuatorManager* su klase posredstvom kojih simbolički nivo pristupa uslugama fizičkog mrežnog nivoa.

Rukovaoci uređajima mogu komunicirati sa sensorima i aktuatorima posredstvom neke od sprega prisutnih na centralnoj kontekstualnoj platformi, ili mogu adresirati odgovarajući kontekstualni modul. U ovom slučaju parametri senzora se proširuju tako da uključe podatke potrebne za identifikaciju kontekstualnog modula (npr. njegovu adresu na mreži).

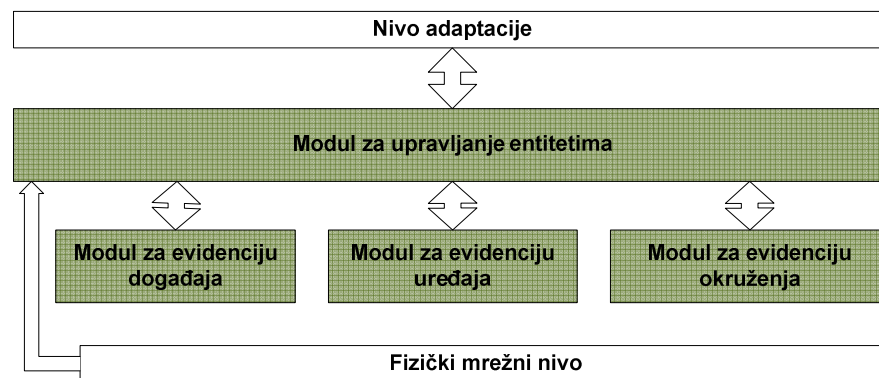
Model arhitekture platforme



Slika 72. UML dijagram klasa modula za upravljanje uređajima i modula za upravljanje događajima

3.3.2.6 Simbolički nivo

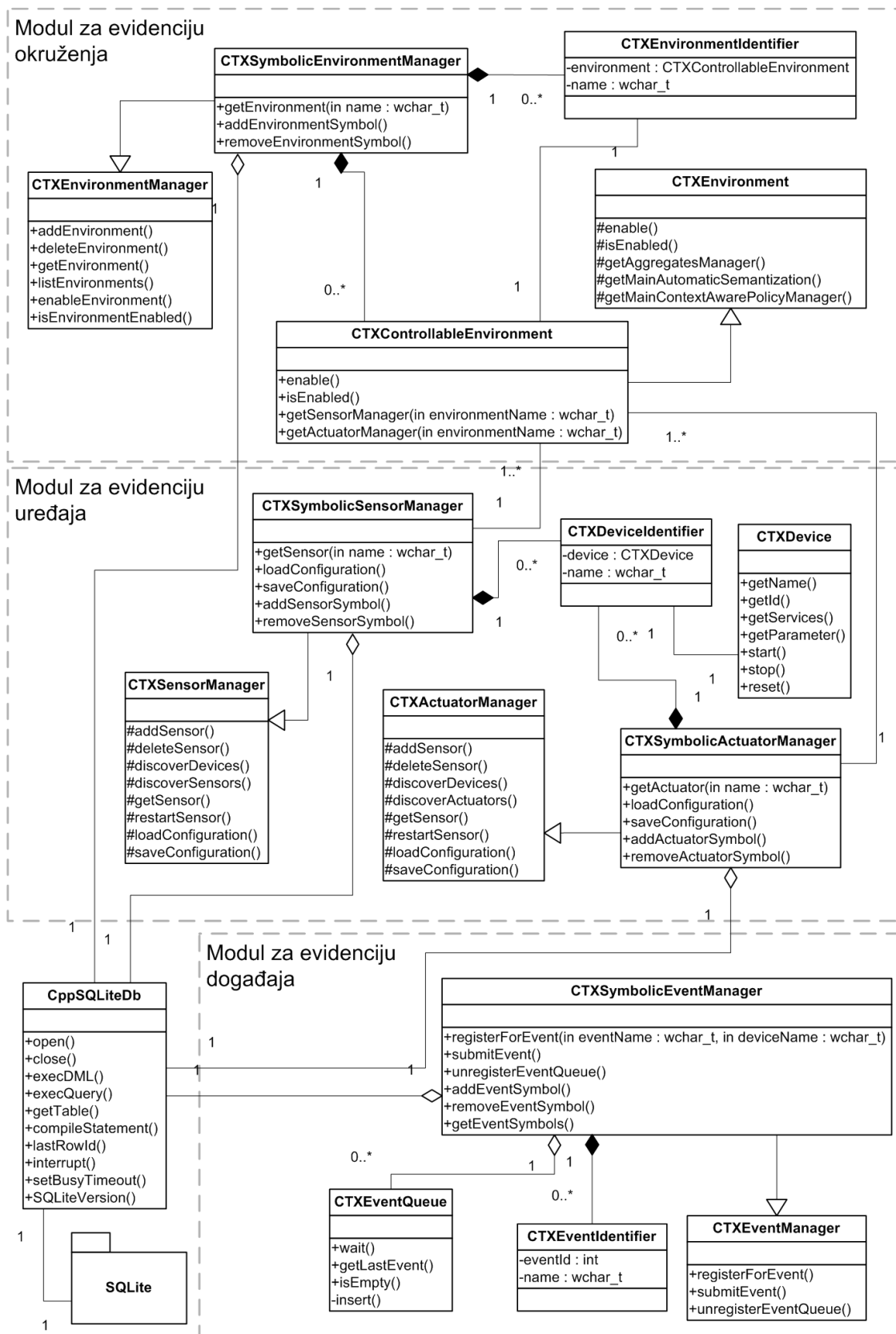
Na simboličkom nivou realizovane su funkcionalnosti koje se odnose na (1) dodeljivanje i evidenciju simboličkih imena uređaja (senzora i aktuatora), događaja i okruženja, realizovano u okviru istoimenih modula; (2) mapiranje simboličkih imena na fizičke identifikatore uređaja i događaja, realizovano u okviru modula za upravljanje entitetima; (3) omogućavanje izdavanja naredbi aktuatorima, odnosno prijavljivanje na događaje sa senzora korišćenjem simboličkih imena, takođe realizovano u okviru modula za upravljanje entitetima. Blok dijagram programske podrške simboličkog nivoa dat je na Slici 73.



Slika 73. Blok dijagram programske podrške simboličkog nivoa

Na Slici 74 dat je UML dijagram klasa modula za evidenciju događaja, modula za evidenciju uređaja i modula za evidenciju okruženja. Klasa *CTXSymbolicEventManager* proširuje osnovni *CTXEventManager*, sa ciljem da omogući korišćenje simboličkih identifikatora pri prijavljivanju i čekanju događaja. Identifikatori se čuvaju u mapi elemenata tipa *CTXEventIdentifier*, dok se za trajno čuvanje mapiranja koristi *SQLite* baza podataka. Klasa *CTXSymbolicSensorManager* proširuje osnovnu klasu *CTXSensorManager* da omogući dobavljanje instance klase *CTXSensor*, koja proširuje *CTXDevice*, na osnovu simboličkog naziva senzora. Klasa *CTXSymbolicActuatorManager* proširuje osnovnu klasu *CTXActuatorManager* da omogući dobavljanje instance klase *CTXActuator*, koja proširuje *CTXDevice*, na osnovu simboličkog naziva aktuatora. Mapiranje reference na odgovarajuću instancu tipa *CTXDevice* ostvareno je korišćenjem *CTXDeviceIdentifier* objekta.

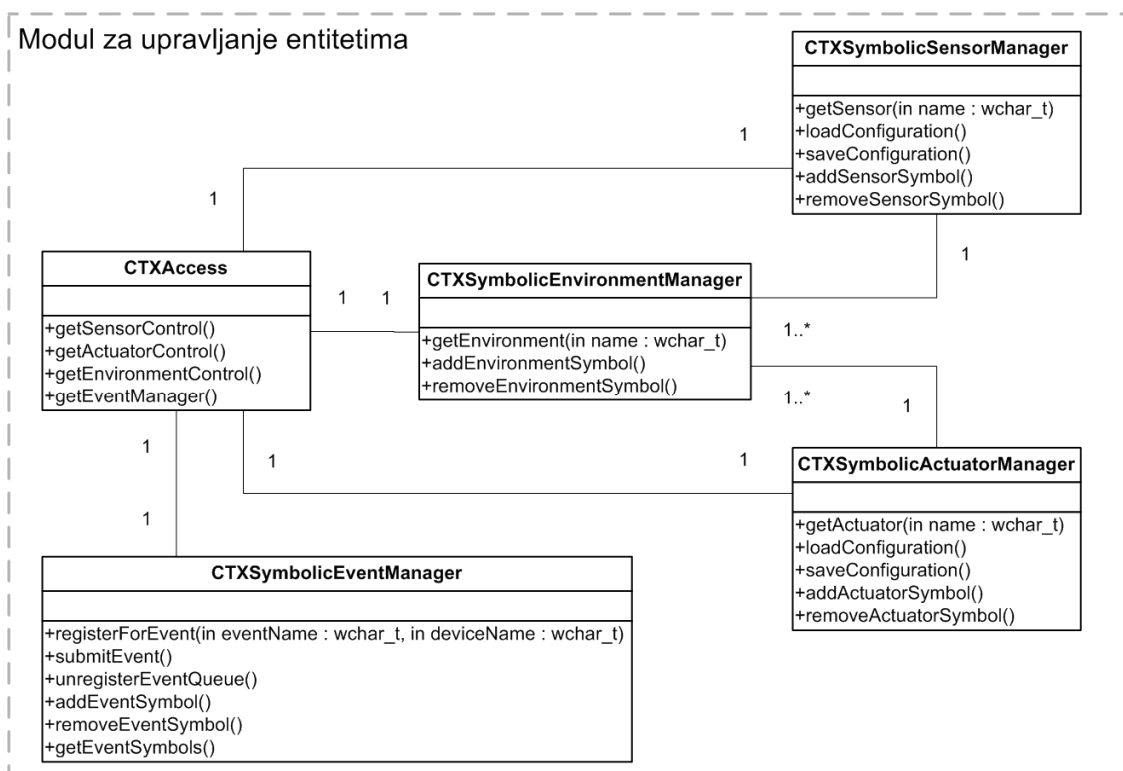
Model arhitekture platforme



Slika 74. UML dijagram klase modula za evidenciju događaja, modula za evidenciju uređaja i modula za evidenciju okruženja

Klasa *CTXSymbolicEnvironmentManager* nasleđuje osnovni *CTXEnvironmentManager*, dodajući mogućnost dobavljanja okruženja, realizovanog klasom *CTXEnvironment*, na osnovu simboličkog naziva okruženja. Klasa *CTXControllableEnvironment*, proširuje *CTXEnvironment*, u cilju ostvarivanja veze sa klasama *CTXSymbolicSensorManager* i *CTXSymbolicActuatorManager*. Mapiranje simboličkih naziva okruženja na reference na odgovarajuće instance tipa *CTXControllableEnvironment*, ostvaruje se korišćenjem klase *CTXEnvironmentIdentifier*.

Na Slici 75 dat je dijagram klasa modula za upravljanje entitetima.



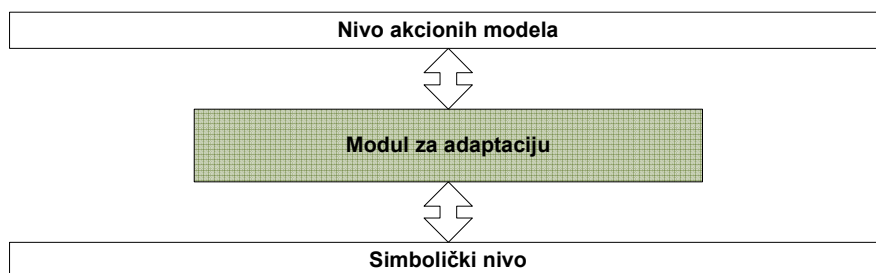
Slika 75. UML dijagram klasa modula za upravljanje entitetima

Pristup svim sensorima i aktuatorima sa nivoa adaptacije, odnosno sa viših nivoa u kojima su realizovane kontekstualne aplikacije, u smislu kontekstualizacije, odnosno povratne kontrole aktuatora i dodatne programske logike nad sensorima, obezbeđen je posredstvom klase *CTXAccess*. Ova klasa omogućava pristup potrebnim blokovima nižeg nivoa hijerarhije, zaduženim za upravljanje događajima (*CTXSymbolicEventManager*), sensorima (*CTXSymbolicSensorManager*), aktuatorima (*CTXSymbolicActuatorManager*) i okruženjima (*CTXSymbolicEnvironmentManager*).

3.3.2.7 Nivo adaptacije

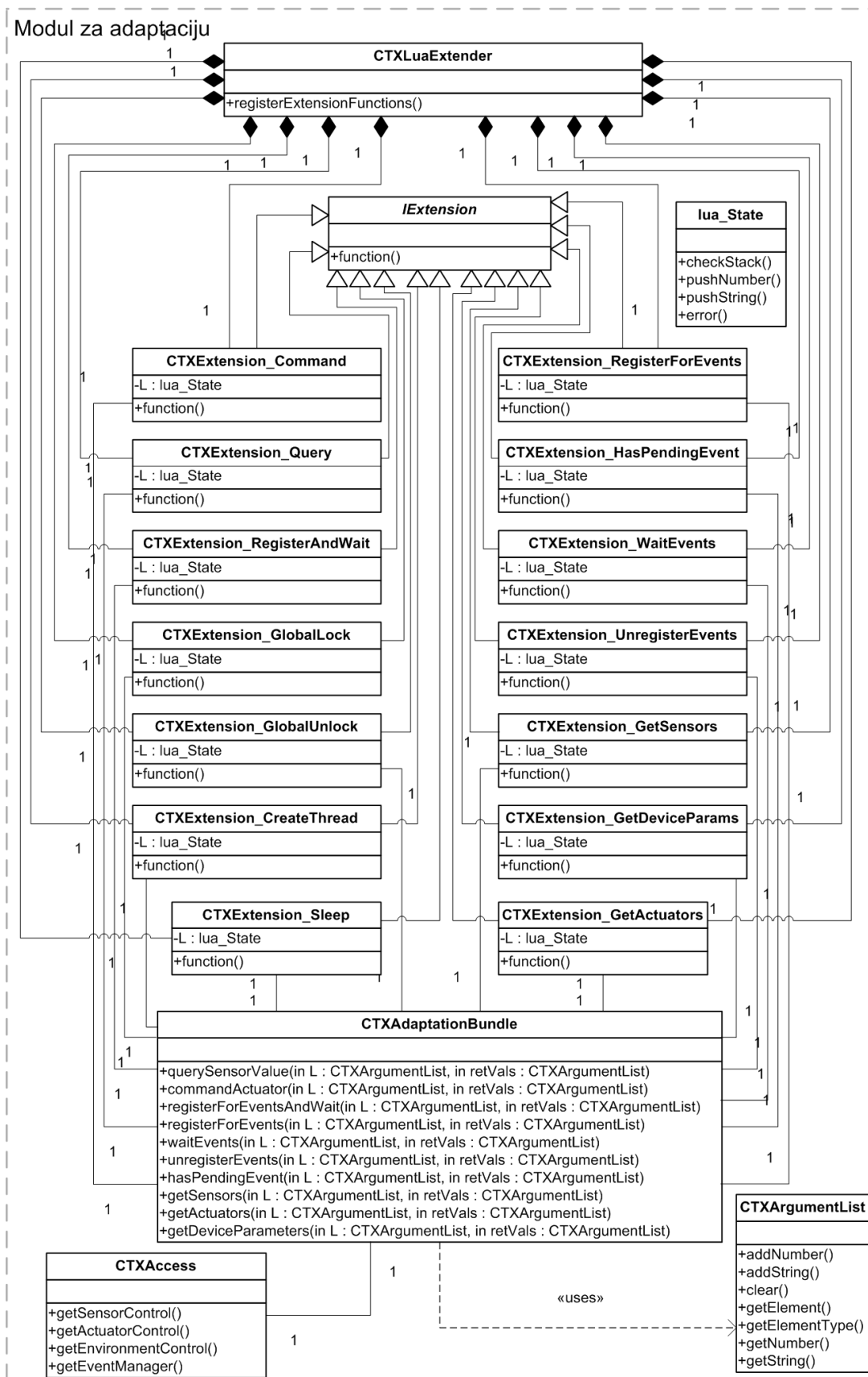
Nivo adaptacije je nivo najniže u hijerarhiji sa kojim se počinje realizacija mehanizma podrške izvršavanju kontekstualnih skriptova. Cilj ovog nivoa je da obezbedi prilagodne mehanizme u cilju proširivanja ciljnog skript interpretera funkcijama na osnovu kojih se omogućava pristup svim potrebnim funkcionalnostima kontekstualne platforme, koje u ovom slučaju obezbeđuje simbolički nivo. Uzevši u obzir rezultate procene upotrebljivosti skript jezika predstavljene u poglavlju 2.4.1.2, ova disertacija podrazumeva Lua jezik kao odabrani skript jezik za realizaciju kontekstualnih platformi, te modeluje arhitekturu kontekstualne platforme u duhu tog jezika, uz napomenu da su predstavljeni principi u potpunosti primenljivi i na realizacije koje koriste neke druge interpretere, odnosno neke druge skript jezike.

Nivo adaptacije realizovan je modulom za adaptaciju, tako da je blok dijagram programske podrške nivoa adaptacije jednostavan i dat na Slici 76.



Slika 76. Blok dijagram programske podrške nivoa adaptacije

Na Slici 77 dat je dijagram klasa modula za adaptaciju. Klasa *CTXLuaExtender* omogućava registrovanje funkcija za proširenje Lua interpretera u sam interpreter, nakon inicijalizacije interpretera na aplikativnom nivou. Interfejs *IExtension* implementira se za svaku funkciju proširenja, koja obezbeđuje mogućnost da se iz Lua skripta ta funkcija poziva i aktivira funkcionalnosti obezbeđene kontekstualnom platformom, kojima se pristupa posredstvom klase *CTXAccess*. Klasa *CTXAdaptationBundle* obezbeđuje API sa pristupom svim funkcionalnostima, na način da omogućava da parametri budu prosleđeni preko liste argumenata tipa *CTXArgumentList*, koja je pogodna za korišćenje u ovom slučaju, jer Lua skript jezik podržava varijabilnu dužinu liste parametara. Objekat tipa *lua_State* prosleđuje se svakoj funkciji proširenja u momentu njenog pozivanja iz skripta. Ovaj objekat sadrži ceo kontekst pozivaoca, uključujući stek sa ulaznim parametrima funkcije.



Slika 77. UML dijagram klasa modula za adaptaciju

Model arhitekture platforme

Proširenja Lua jezika koja su realizovana u cilju omogućavanja kontekstualizacije data su u Tabeli 13.

<i>Naziv Lua funkcije i klase proširenja</i>	<i>Parametri i opis</i>
<code>ctx_command</code> <i>CTXExtension_Command</i>	<ol style="list-style-type: none"> 1. Ime aktuatora 2. Kod komande 3. Parametri <p>Izdavanje komande aktuatoru.</p>
<code>ctx_query</code> <i>CTXExtension_Query</i>	<ol style="list-style-type: none"> 1. Ime senzora 2. Kod stanja 3. Parametri <p>Dobavljanje stanja senzora.</p>
<code>ctx_register_wait</code> <i>CTXExtension_RegisterAndWait</i>	<ol style="list-style-type: none"> 1. Uređeni par (ime senzora, ime događaja), moguće ponavljanje više puta za više događaja <p>Registruje se za navedene događaje i beskonačno čeka na neki od njih sa blokiranjem.</p>
<code>ctx_create_thread</code> <i>CTXExtension_CreateThread</i>	<ol style="list-style-type: none"> 1. Naziv funkcije kojom se realizuje nit. <p>Kreira novu nit i aktivira njeno izvršavanje.</p>
<code>ctx_lock</code> <i>CTXExtension_GlobalLock</i>	<p>Zaključava globalnu kritičnu sekciju. Izaziva grešku ukoliko je kritična sekcija već zaključana.</p>
<code>ctx_unlock</code> <i>CTXExtension_GlobalUnlock</i>	<p>Otključava globalnu kritičnu sekciju. Izaziva grešku ukoliko je kritična sekcija već otključana.</p>
<code>ctx_register_events</code> <i>CTXExtension_RegisterForEvents</i>	<ol style="list-style-type: none"> 1. Uređeni par (ime senzora, ime događaja), moguće ponavljanje više puta za više događaja. <p>Registruje se za navedene događaje i kao povratnu vrednost vraća kod registracije.</p>
<code>ctx_wait_events</code> <i>CTXExtension_WaitEvents</i>	<ol style="list-style-type: none"> 1. Kod registracije 2. Dužina čekanja u milisekundama (0 za beskonačno čekanje) <p>Čeka na događaje koji odgovaraju prethodnoj registraciji, sa vremenskom kontrolom.</p>
<code>ctx_unregister_events</code> <i>CTXExtension_UnregisterEvents</i>	<ol style="list-style-type: none"> 1. Kod registracije <p>Uklanja registraciju za čekanje događaja, nakon što čekanje više nije potrebno.</p>

<p><code>ctx_pending_event</code> <i>CTXExtension_HasPendingEvent</i></p>	<p>1. Uređeni par (ime senzora, ime događaja).</p> <p>Proverava da li ima pristiglih događaja u redu čekanja. Ukoliko ima, funkcija vraća 1, odnosno 0 u suprotnom slučaju.</p>
<p><code>ctx_get_sensors</code> <i>CTXExtension_GetSensors</i></p>	<p>1. Naziv okruženja (opciono).</p> <p>Vraća spisak svih senzora dostupnih kontekstualnoj platformi u tabeli, za navedeno okruženje ili za sva okruženja. Vrsta u tabeli je formata (ime senzora, tip, okruženje, status)</p>
<p><code>ctx_get_actuators</code> <i>CTXExtension_GetActuators</i></p>	<p>1. Naziv okruženja (opciono).</p> <p>Vraća spisak svih aktuatora dostupnih kontekstualnoj platformi u tabeli, za navedeno okruženje ili za sva okruženja. Vrsta u tabeli je formata (ime aktuatora, tip, okruženje)</p>
<p><code>ctx_get_device_params</code> <i>CTXExtension_GetDeviceParams</i></p>	<p>1. Naziv senzora/aktuatora.</p> <p>Vraća spisak svih konfiguracionih parametara i njihovih trenutnih vrednosti.</p>
<p><code>ctx_sleep</code> <i>CTXExtension_Sleep</i></p>	<p>1. Vreme u milisekundama.</p> <p>Odlaganje izvršavanja nastavka skripta za zadati broj milisekundi.</p>

Tabela 13. Kontekstualizaciona proširenja Lua jezika

Ova proširenja Lua jezika odnose se isključivo na zahteve kontekstualizacije, te nisu jedina proširenja koja su realizovana. Klasa *CTXLuaExtender* instancira se i sa aplikativnog nivoa i omogućuje proširivanje dodatnim funkcijama (vidi poglavlje 3.3.2.9). Takođe, s obzirom da se jedan kontekstualni skript izvršava u oblasti jednog okruženja, nije predviđena mogućnost adresiranja okruženja iz funkcija proširenja.

3.3.2.8 Nivo akcionih modela

Nivo akcionih modela je u potpunosti realizovan u skript jeziku (Lua), te ne pripada direktno osnovnoj aplikaciji koja realizuje programsku podršku. Na ovom nivou nalaze se Lua skriptovi koji obezbeđuju omotač (*wrapper*) oko kontekstualnih funkcija kojim je Lua jezik proširen, u cilju omogućavanja konkretnih funkcionalnosti vezanih za određene tipove uređaja. Ovi skriptovi se nazivaju *akcioni modeli*. U opštem slučaju, svaki tip uređaja poseduje sopstveni akcioni model, koji obezbeđuje funkcionalnosti višeg nivoa korišćenjem skupa komandi nižeg nivoa. Na primer,

Model arhitekture platforme

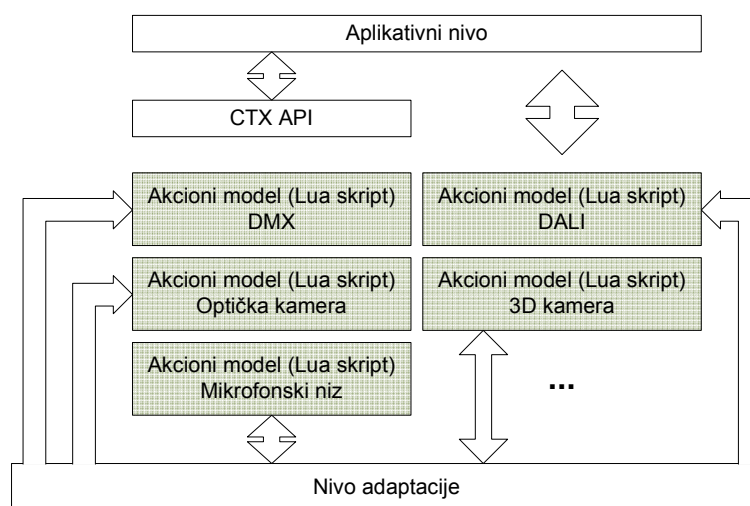
akcioni model koji upravlja DMX rasvetom može da obezbedi Lua funkciju `fade` posredstvom koje se intenzitet DMX svetla menja od trenutne ka zadatoj vrednosti postepeno u vremenu. Ova funkcija u realizaciji koristi isključivo pozive `ctx_command` kojima se omogućuje trenutno postavljanje intenziteta na određenu vrednost. Ilustracija ovog koncepta u obliku izvoda iz Lua skripta koji realizuje odgovarajući akcioni model data je na Slici 78.

```
-----  
-----  
-- function: fadeDMX  
-- short desc:  
--[[  
    Fade DMX light intensity from current to target value  
  ]]  
-- params:  
    -- device: symbolic name of DMX device  
    -- target: target intensity of color  
    -- time: duration of transition to target intensity  
-- return: none  
-----  
-----  
function fadeDMX ( device, target, time )  
    local Ret  
    current = ctx_get_device_param(device)[CTX_DMX_CURRENT_INTENSITY]  
    direction = 1  
  
    howmuch = math.abs(target-current)  
    steplen = time/howmuch  
    if (target<current) then  
        direction = -1  
    end  
  
    for i=0,time,steplen do  
  
        ctx_command(device, CTX_DMX_SET_INTENSITY, current)  
        ctx_sleep(math.floor(steplen+0.5))  
  
        current = current+direction  
    end  
  
    ctx_command(device, CTX_DMX_SET_INTENSITY, target)  
  
end
```

Slika 78. Izvod iz Lua skripta kojim se realizuje jedan akcioni model

Akcioni modeli se izvršavaju sa nivoa aplikacija, pri izvršavanju kontekstualnih skriptova. Međutim, s obzirom na hijerarhiju poziva, oni pripadaju jednom nivou niže. Sa stanovišta kontekstualne aplikacije, akcioni modeli mogu se iskoristiti za realizaciju određenih algoritama zaključivanja koji ne obrađuju velike količine podataka i nemaju

stroge zahteve izvršavanja u realnom vremenu, te na taj način akcioni model preuzima ulogu virtuelnog senzora. Međutim, ovako kreiran akcioni model, koji opisuje senzorsku funkcionalnost višeg nivoa, ne može prijavljivati događaje nivou semantizacije, što dalje zahteva od kontekstualne aplikacije da na sopstveni način realizuje prijem i obradu zaključaka. Poseban značaj akcionih modela je u situacijama u kojima je potrebno brzo obezbediti novu funkcionalnost, kada nije izvodljivo menjati osnovnu programsku podršku kontekstualne platforme, odnosno u slučaju kada nije moguće jednostavno prevođenje koda i kreiranje senzorske biblioteke za određenu platformu. Blok dijagram programske podrške nivoa akcionih modela dat je na Slici 79.



Slika 79. Blok dijagram programske podrške nivoa akcionih modela

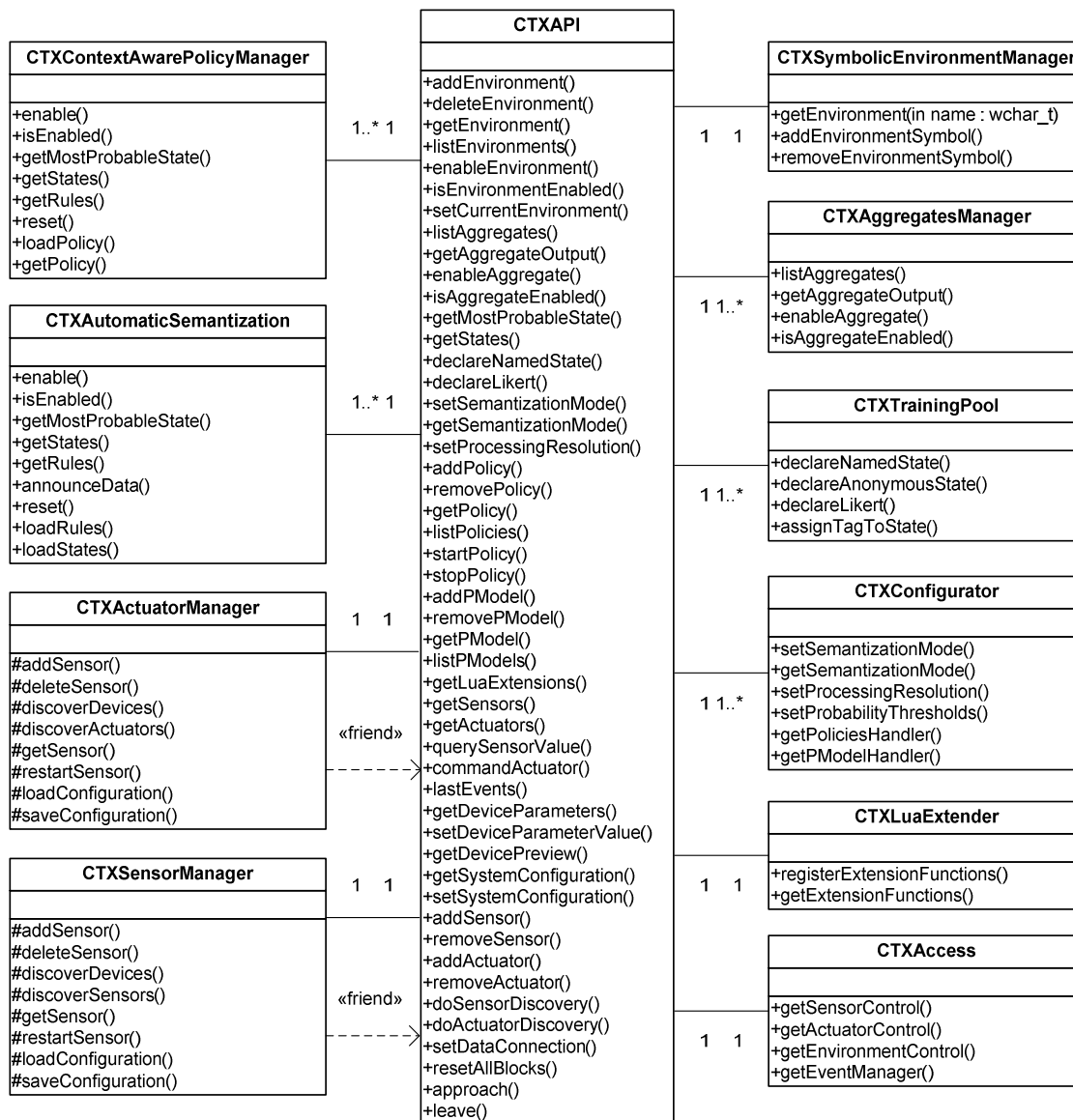
3.3.2.9 Kontekstualna programska sprega (CTX API)

Veliki broj funkcionalnosti koje su specifične za kontekstualnu platformu realizuju se na nivou mreže i na srednjem nivou. Iz tog razloga, između srednjeg nivoa i aplikativnog nivoa obezbeđena je jedinstvena programska sprega za korišćenje na aplikativnom nivou, pod nazivom *kontekstualna programska sprega (CTX API)*. CTX API sadrži spisak svih mogućih metoda koje omogućuju korišćenje kontekstualne platforme. CTX API pristupa kolekciji svih potrebnih klasa posredstvom kojih obezbeđuje potrebne funkcionalnosti.

UML dijagram klasa kontekstualne programske sprege dat je na Slici 80. Klasa CTX API je jedina klasa koja je dostupna aplikativnom nivou i nivou korisničke sprege. S obzirom na osetljivost pojedinih metoda, npr. u smislu slučajeva korišćenja

Model arhitekture platforme

SK3.4 i SK6.4 korisnici CTX API se prijavljuju za korišćenje pozivima *approach* i *leave*, i prilikom prijave moraju da prilože odgovarajući ključ. Na ovaj način je moguće dozvoliti, npr. sistemskoj korisničkoj sprezi da menja osetljiva systemska podešavanja, uključujući, npr. parametre senzora i aktuatora, dok se isto ne dozvoljava kontekstualnim aplikacijama.

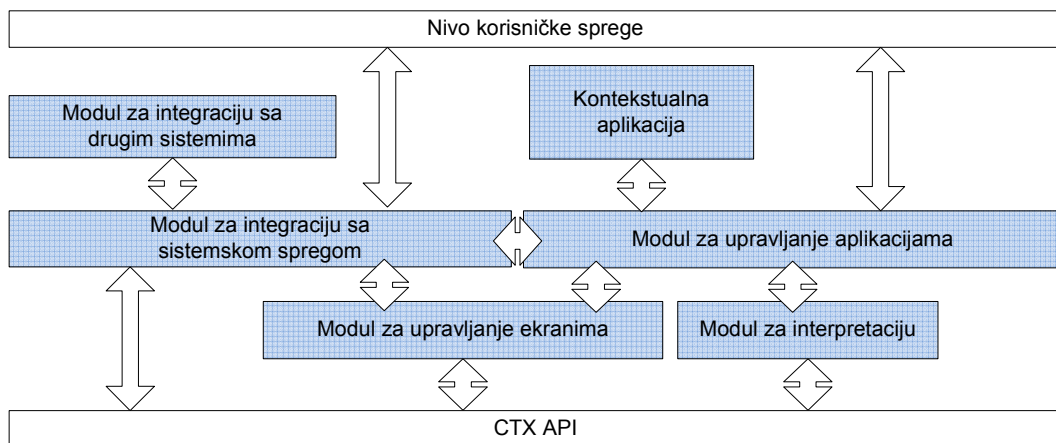


Slika 80. UML dijagram klasa kontekstualne programske sprege

3.3.2.10 Aplikativni nivo

Aplikativni nivo obezbeđuje mehanizme za integraciju ugrađenih kontekstualnih aplikacija, kao i onih aplikacija na drugim uređajima ili sistemima, sa kontekstualnom platformom. U tom smislu aplikativni nivo obezbeđuje: (1) server kontekstualne HTTP

sprege (HTTP API) koji omogućava da zainteresovani uređaji ili sistemi komuniciraju sa kontekstualnom platformom, realizovan u okviru modula za integraciju sa drugim sistemima; (2) integraciju sa sistemskom korisničkom spregom, u smislu realizacije servera na TCP/IP nivou koji obezbeđuje komunikaciju sa web aplikacijom koja realizuje sistemsku korisničku spregu, u okviru modula za integraciju sa sistemskom spregom; (3) evidenciju, dodavanje, uklanjanje, pokretanje i zaustavljanje rada integrisanih kontekstualnih aplikacija, realizovanih u vidu kontekstualnih skriptova, u okviru modula za upravljanje aplikacijama; (4) interpretaciju kontekstualnih skriptova u okviru modula za interpretaciju i (5) komunikaciju sa prezentacionim jedinicama i dodelu prezentacione jedinice, odnosno ekrana priključenog na tu jedinicu, određenim kontekstualnim aplikacijama ili sistemskoj korisničkoj sprezi. Blok dijagram programske podrške aplikativnog nivoa dat je na Slici 81.

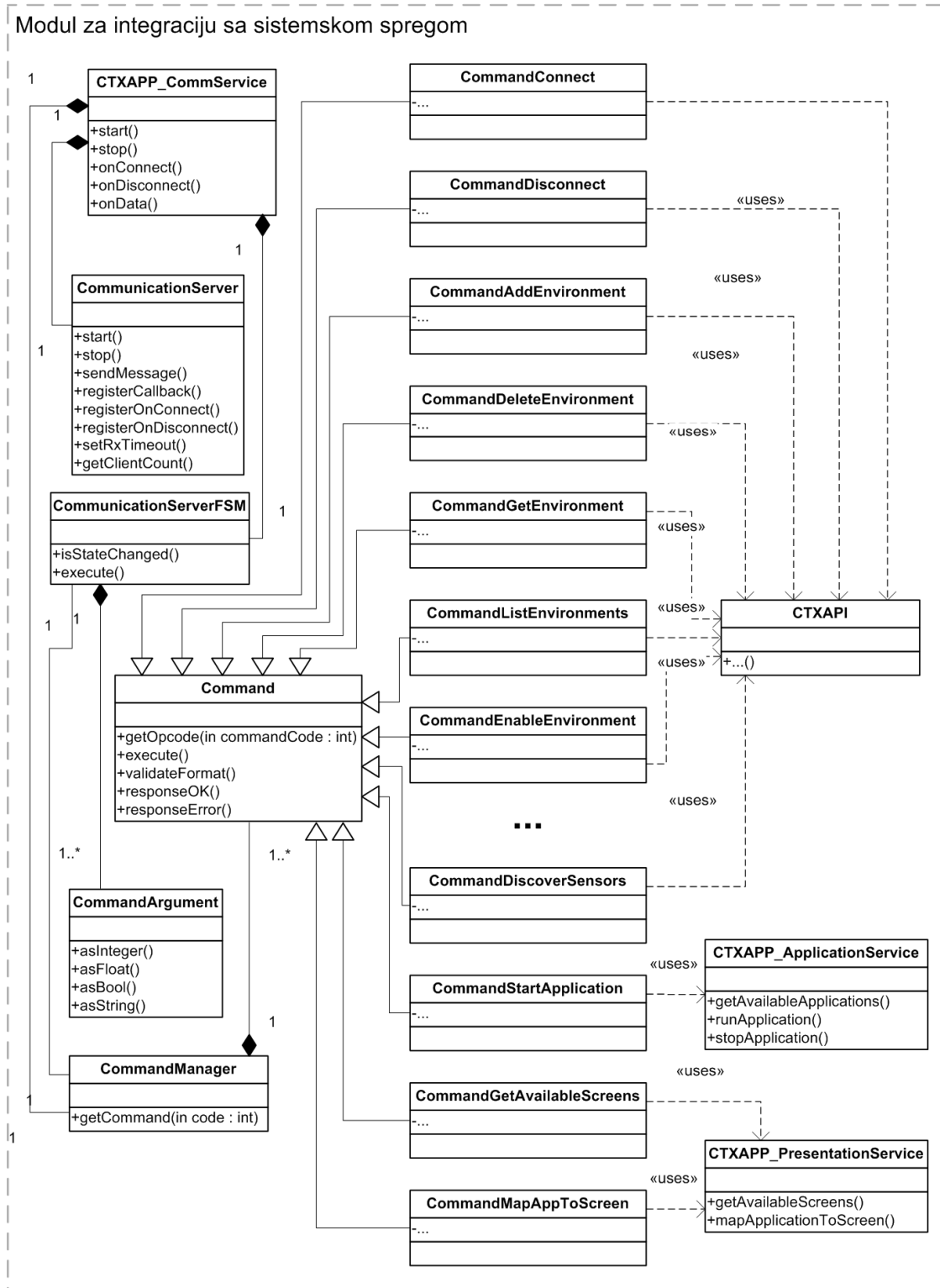


Slika 81. Blok dijagram programske podrške aplikativnog nivoa

Na Slici 82 dat je dijagram klasa modula za integraciju sa sistemskom spregom. Klasa *CTXAPP_CommService* inicijalizuje klasu *CommunicationServer*, koja omogućava TCP/IP komunikaciju sa sistemskom korisničkom spregom ili sa modulima višeg nivoa, koji su realizovani izvan osnovne aplikacije kontekstualne platforme. Klasa *CommunicationServerFSM* realizuje prelaze stanja u toku komunikacione sesije, te aktivira izvršavanje konkretnih komandi korišćenjem klase *CommandManager*. Ova klasa služi kao kontejner svih komandi koje se mogu uputiti kontekstualnoj platformi, sa mehanizmom koji se lako proširuje s obzirom da svaka nova komanda nasleđuje osnovnu klasu *Command*, i registruje se kod klase

Model arhitekture platforme

CommandManager. Klase *CommandConnect* i *CommandDisconnect* realizuju osnovne primitive uspostavljanja i raskidanja sesije, uz prilaganje ključa.



Slika 82. UML dijagram klasa modula za integraciju sa sistemskom spregom

Sve klase naslednice klase *Command* koriste *CTXAPI* za ostvarivanje konkretne funkcionalnosti. U slučaju ostvarivanja funkcionalnosti koju pruža aplikativni nivo, koriste se odgovarajuće klase aplikativnog nivoa, poput *CTXAPP_ApplicationService* i *CTXAPP_PresentationService*.

```

<connect>
  <passkey="c9c6964b63bc506275c7e2933eaa4fd0"/>
</connect>

                                <response>
                                  <error>0</error>
                                  <sessionid>235</sessionid>
                                </response>

<command>
  <passkey="c9c6964b63bc506275c7e2933eaa4fd0"/>
  <sessionid>235</sessionid>
  <code>30</code>
  <environment>Living</environment>
  <type>optical_cameras</type>
</command>

                                <response>
                                  <error>0</error>
                                  <sensors>
                                    <sensor>
                                      <name>Optical camera #5</name>
                                      <address>192.168.0.101</address>
                                      <status>1</status>
                                      <uptime>11:23:22</uptime>
                                    </sensor>
                                  </sensors>
                                </response>

<disconnect>
  <passkey="c9c6964b63bc506275c7e2933eaa4fd0"/>
  <sessionid>235</sessionid>
</disconnect>

                                <response>
                                  <error>0</error>
                                  <sessiondur>00:05:11</sessiondur>
                                </response>

```

Slika 83. Ilustracija sintakse primitiva za uspostavljanje i raskidanje veze, kao i primitive za dobavljanje spiska senzora

Model arhitekture platforme

Konkretan protokol koji se koristi za razmenu podataka sa modulom za integraciju sa sistemskom spregom zasnovan je na XML jeziku, uz poštovanje redosleda operacija povezivanje => komanda => odgovor => raskidanje. Na Slici 83 dat je izgled komandi *connect*, *disconnect* i *get_sensors* u cilju ilustracije korišćenog protokola.

Modul za integraciju sa drugim sistemima realizuje se van okvira osnovne aplikacije kontekstualne platforme. Ovaj modul koristi komunikacione primitive koje obezbeđuje modul za integraciju sa sistemskom spregom, te ostvaruje funkcionalnost HTTP servera koji reaguje na GET i POST zahteve. U opštem slučaju, zainteresovana strana šalje HTTP GET zahtev, i dobija odgovor u XML ili JSON formatu. Ovaj HTTP API mapiran je na komande modula za integraciju sa sistemskom spregom u onoj meri u kojoj to ne ugrožava sigurnost i robustnost kontekstualne platforme, odnosno mapiran je podskup komandi koji ne uključuje komande za konfigurisanje senzora i aktuatora. Na Slici 84 dat je primer jednog zahteva i odgovora za slučaj dobavljanja stepena zauzetosti korisnika (osnovna funkcija *getAggregateOutput*).

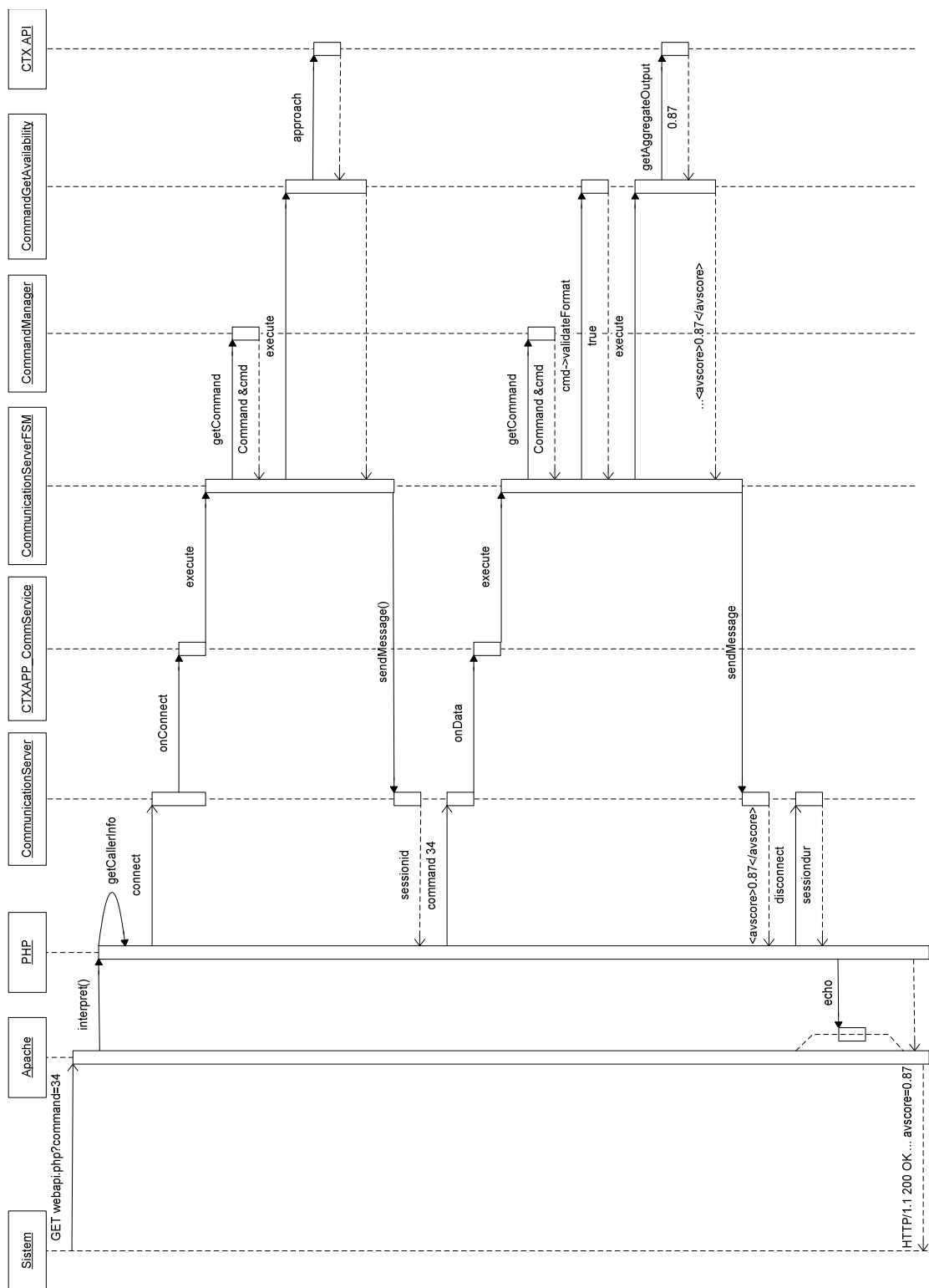
```
GET http://192.168.0.100/webapi.php?command=34&environment=living
HTTP/1.1
Host: localhost
Authorization: Basic k7fnGutONN9/0iA+QGPvA7KTjlc=

HTTP/1.1 200 OK
Server: HTTPd/1.0
Date: Wed, 25 Jul 2012 10:19:07 GMT
Content-Type: text/plain
avscore=0.87
```

Slika 84. Primer jednog HTTP zahteva i odgovora modula za integraciju sa drugim sistemima

Modul za integraciju sa drugim sistemima realizovan je kao PHP skript *webapi.php* koji se izvršava PHP interpreterom, dok se rezultat isporučuje korišćenjem HTTP servera Apache. UML dijagram sekvence dat na Slici 85 ilustruje operaciju uspostavljanja HTTP zahteva, dobavljanja rezultata i isporučivanja natrag pozivaocu, sa prikazom svih entiteta koji učestvuju u tom događaju.

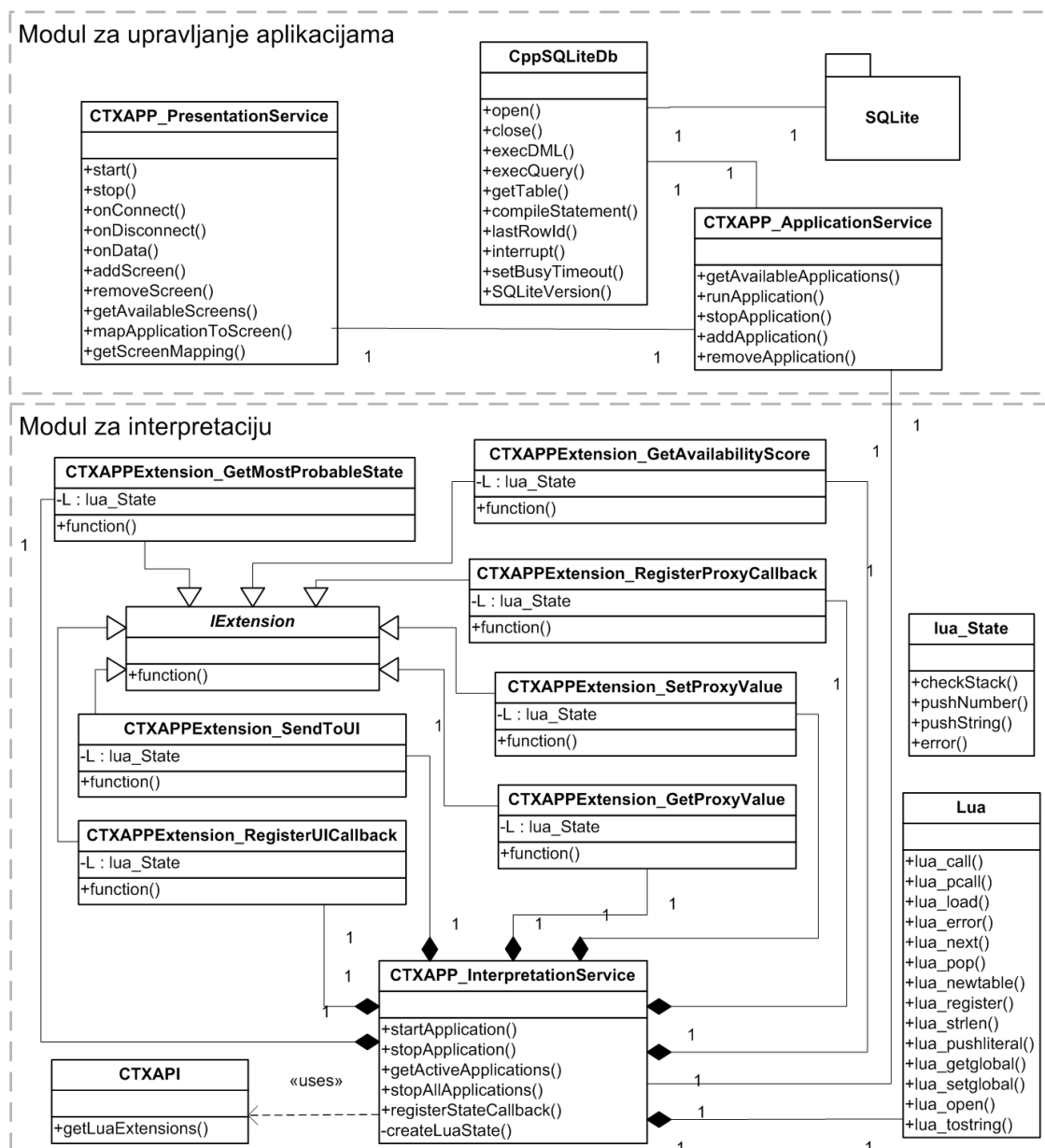
Modul za upravljanje ekranima vodi evidenciju o dostupnim prezentacionim jedinicama, i isporučuje odgovarajući sadržaj za prikaz na prezentacionoj jedinici. Ovaj modul obavlja mapiranje između aktivnih kontekstualnih aplikacija koje koriste grafičku korisničku spregu, i dostupnih prezentacionih jedinica.



Slika 85. UML dijagram sekvence pri uspostavljanju i obradi HTTP API zahteva i isporuci odgovora

Osnovna programska podrška prezentacione jedinice sastoji se od web pregledača, koji izvršava web aplikaciju posluženu od strane modula za upravljanje

obezbeđena su proširenja za razmenu podataka sa grafičkim korisničkim spregama, u vidu popunjavanja i dobavljanja predviđenih polja (*CTXAPPExtension_SetProxyValue*, *CTXAPPExtension_RegisterProxyCallback*, *CTXAPPExtension_GetProxyValue*), odnosno razmene podataka sa UI aplikacijom specifične namene (*CTXAPPExtension_RegisterUICallback*, *CTXAPPExtension_SendToUI*). Takođe, obezbeđena su proširenja koja omogućavaju dobavljanje zaključaka iz procesa semantizacije (*CTXAPPExtension_GetMostProbableState*, *CTXAPPExtension_GetAvailabilityScore* itd).



Slika 87. UML dijagram klasa modula za interpretaciju i modula za upravljanje aplikacijama

Model arhitekture platforme

Klasa *CTXAPP_ApplicationService* omogućava dodavanje, uklanjanje, pokretanje i zaustavljanje aplikacija. Ova klasa koristi informacije koje poseduje *CTXAPP_PresentationService*, da usmeri zahteve Lua UI poziva na ekran koji je dodeljen konkretnoj aplikaciji. Podaci o aplikacijama se čuvaju u SQLite bazi podataka.

U nastavku je dat izvod Lua koda kontekstualnog skripta jedne jednostavne kontekstualne aplikacije razvijene u okviru disertacije, pod imenom *Luminosity*. Aplikacija na osnovu informacije o zauzetosti korisnika određuje boju koja se prikazuje na 4 DMX lampe, postavljene u niz. Dodatno, u zavisnosti od trenutne pozicije korisnika u prostoriji, postavlja se intenzitet osvetljaja na lampama (što je korisnik bliže određenoj lampi, intenzitet je veći, i obratno). Trenutna vrednost nivoa zauzetosti se prikazuje u grafičkoj korisničkoj sprezi. Funkcija *LampFade* podešava vrednost lampe u zavisnosti od trenutnih vrednosti konfiguracije smeštenih u deljenu, globalnu promenljivu *Lamps*. Po jedna instanca ove funkcije pokreće se nakon pokretanja aplikacije u obliku niti, i izvršava se dokle god je aplikacija aktivna. U okviru ove promenljive nalaze se informacije o trenutnom osvetljaju i o stepenu zauzetosti korisnika (Slika 88).

```
function LampFade(dg)

    local oldBright1 = 0

    while (true) do

        -- If skin light intensity for the lamp has changed
        if oldBright1 ~= Lamps[dg].Brightness then
            oldBright1 = Lamps[dg].Brightness
            -- Fade in to new value
            if (Lamps[dg].Brightness > oldBright1) then
                fadeDMX(Lamps[dg].SkinLight, Lamps[dg].Brightness, LAMP_FADE_IN_TIME)
            -- Fade out to new value
            else
                fadeDMX(Lamps[dg].SkinLight, Lamps[dg].Brightness, LAMP_FADE_OUT_TIME)
            end
        end

        if (Lamps[dg].AvailabilityScore > 70) then
            DMX.instruct(Lamps[dg].SkinLight, 0, 255, 0)
        elseif (Lamps[dg].AvailabilityScore > 30) then
            DMX.instruct(Lamps[dg].SkinLight, 255, 255, 0)
        else
            DMX.instruct(Lamps[dg].SkinLight, 255, 0, 0)
        end
    end

    ctx_sleep(20)
end

end
```

Slika 88. Funkcija *LampFade* iz kontekstualne aplikacije *Luminosity*

Funkcija *LampUpdate* izračunava novu vrednost osvetljaja lampe na osnovu prosledene pozicije osobe u prostoriji. Takođe se dobavlja trenutna vrednost stepena zauzetosti korisnika (Slika 89).

```
function LampUpdate ( lamp, x, y )

    local brightness = 0
    local lamp_x = x-Lamps[lamp].Position
    local lamp_y = y

    brightness = ((X_PONDER*lamp_x*lamp_x + lamp_y*lamp_y* Y_PONDER)^(-0.5)) * LAMP_MUL
    availability_score = ctx_get_availability_score()
    ctx_send_to_ui(availability_score)

    if brightness>255 then
        brightness = 255
    elseif brightness<0 then
        brightness = 0
    end

    return brightness, availability_score
end
```

Slika 89. Funkcija *LampUpdate* iz kontekstualne aplikacije *Luminosity*

Funkcija *StateMachine* izvršava se kao posebna nit, koja očekuje događaje sa 3D kamere, dobavlja koordinate trenutne pozicije najviše 4 osobe u prostoriji, te ažurira trenutne parametre promenljive *Lamps* (Slika 90). Na Slici 91 dat je kod koji se izvršava neposredno po pokretanju aplikacije, a koji postavlja boju lampi na početne vrednosti i pokreće sve programske niti.

Aplikativni nivo zahteva od svake kontekstualne aplikacije da deklariše tipove i imena svih uređaja koje će koristiti u svom radu. Ova deklaracija obavlja se posredstvom sistemske korisničke sprege, pri dodavanju nove aplikacije. U slučaju korišćenja aktuatora, proverava se da li je moguće koristiti aktuator kao deljeni resurs, jer je u opštem slučaju moguća potreba za ekskluzivitetom pri korišćenju aktuatora (npr. ukoliko je aktuator multimedijalni plejer koji reprodukuje video materijal, nije pogodno da se koristi od strane dve aplikacije istovremeno; upravljanje lampama iz više aplikacija istovremeno takođe može da izazove oprečne efekte). Iz ovog razloga, klasa *CTXAPP_ApplicationService* neće dozvoliti dodavanje nove aplikacije, ukoliko je već dodata aplikacija koja koristi zahtevane aktuatore, ukoliko ti aktuatori zahtevaju ekskluzivitet korišćenja.

Iako postoje mnogobrojna proširenja za Lua jezik koja omogućavaju operacije niskog nivoa (npr. *LuaSocket* za mrežnu TCP/IP komunikaciju) ove funkcionalnosti

Model arhitekture platforme

nisu omogućene za korišćenje iz kontekstualnih skriptova, zbog zahteva očuvanja robustnosti rada kontekstualne platforme.

```
function StateMachine ()
    local zone
    local lamp
    local i
    local j
    local k
    local max_bright
    local current_as = 100
    local tmpb
    local distances = {
        [1] = {[X] = 0, [Y] = 0},
        [2] = {[X] = 0, [Y] = 0},
        [3] = {[X] = 0, [Y] = 0},
        [4] = {[X] = 0, [Y] = 0}
    }

    while true do

        for i=0,50,40 do

            -- Wait for an event on 3D camera

            dev, event_id = ctx_wait_events(i, evid)

            if (dev ~= nil) then
                -- If presence is detected
                if (event_id % 2 == 1) then
                    -- Get coordinates for all found people
                    distances[1].X, distances[1].Y, distances[2].X, distances[2].Y,
                    distances[3].X, distances[3].Y, distances[4].X, distances[4].Y
                    = ctx_query("3DCam", 2)

                    -- Calculate new brightness and color
                    for k=1,4 do
                        max_bright = MIN_BRIGHT
                        -- Find the maximum possible brightness level depending on all people found
                        for j=1,4 do
                            if distances[j].Y ~= 0 then
                                tmpb, current_as = LampUpdate ( k, distances[j].X, distances[j].Y )
                                if tmpb > max_bright then
                                    max_bright = tmpb
                                end
                            end
                        end
                        -- Assign maximum brightness to the lamp skin light
                        Lamps[k].Brightness = max_bright
                        Lamps[k].AvailabilityScore = current_as
                    end
                    -- If there is no one in front of the Lamps row
                else
                    -- Dim lights
                    Lamps[1].Brightness = MIN_BRIGHT
                    Lamps[2].Brightness = MIN_BRIGHT
                    Lamps[3].Brightness = MIN_BRIGHT
                    Lamps[4].Brightness = MIN_BRIGHT
                end
            else
                print("Timeout!")
            end
        end
    end
end
```

Slika 90. Funkcija *StateMachine* iz kontekstualne aplikacije *Luminosity*

```

-- application entry point

-- Set all lights to initial values
DMX.instruct("Lamp1", 0, 255, 0)
DMX.instruct("Lamp2", 0, 255, 0)
DMX.instruct("Lamp3", 0, 255, 0)
DMX.instruct("Lamp4", 0, 255, 0)

-- Register for 3D camera events
evid = ctx_register_events("3DCam",1,2)

-- Fire up all needed script threads
T1 = create_thread("StateMachine")

T2 = ctx_create_thread("LampFade", 1)
T3 = ctx_create_thread("LampFade", 2)
T4 = ctx_create_thread("LampFade", 3)
T5 = ctx_create_thread("LampFade", 4)

```

Slika 91. Ulazna tačka kontekstualne aplikacije *Luminosity*

3.3.2.11 Nivo korisničke sprege

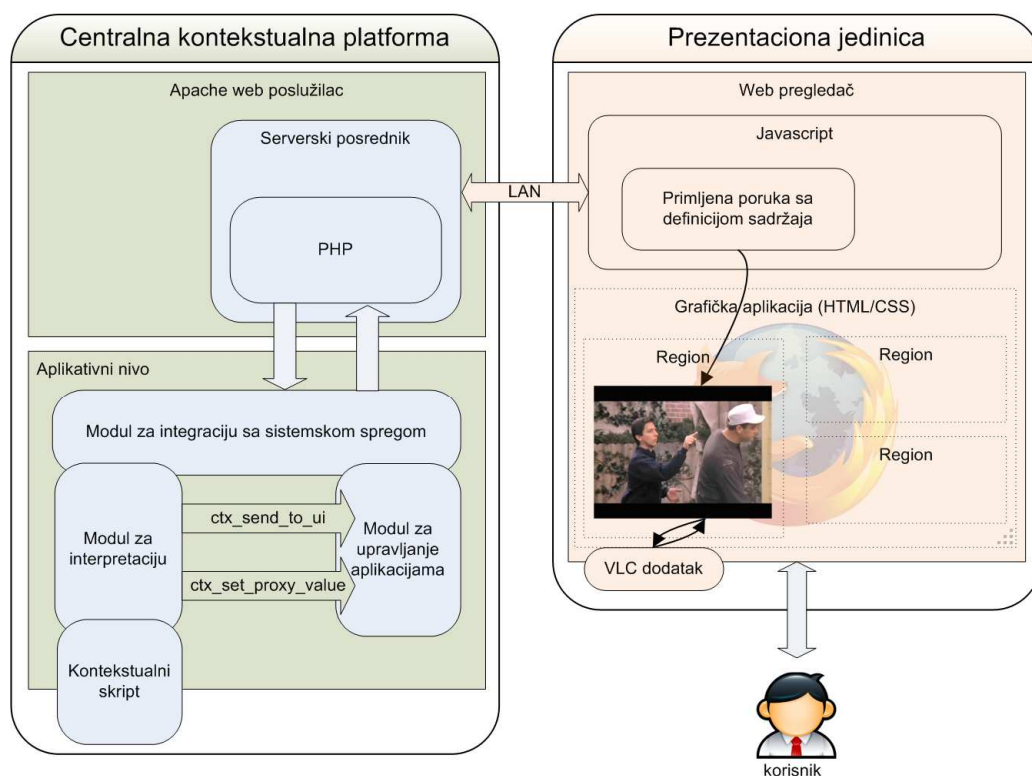
Nivo korisničke sprege omogućava izvršavanje grafičkog dela kontekstualnih aplikacija, kao i grafičkog dela systemske korisničke sprege. U okviru ovog nivoa ne podrazumeva se korišćenje grafičkih aplikacija na mobilnim telefonima i ostalim uređajima, već se podrazumeva da te aplikacije koriste HTTP API koji obezbeđuje aplikativni nivo. Dakle, u okviru nivoa korisničke sprege smeštene su one aplikacije koje se izvršavaju na (1) centralnoj kontekstualnoj platformi, kada se izvršavaju kao posebna Linux aplikacija koja prikazuje sliku na ekranu priključenom korišćenjem lokalnog HDMI izlaza i (2) na prezentacionoj jedinici, kada se aplikacija izvršava u okviru web pregledača, vidljiva na ekranu priključenom na prezentacionu jedinicu korišćenjem HDMI izlaza na prezentacionoj jedinici.

Grafičke aplikacije na centralnoj kontekstualnoj platformi mogu biti realizovane kao (1) desktop aplikacije, korišćenjem GTK ili QT okruženja za razvoj; (2) dinamičke web aplikacije, koje se izvršavaju u web pregledaču, zasnovane na Javascript/HTML/CSS tehnologijama; (3) Flash aplikacije, koje se izvršavaju samostalno ili u okviru web pregledača. Ove aplikacije komuniciraju sa kontekstualnim skriptovima korišćenjem TCP/IP komunikacije (*localhost*) zasnovane na XML koju realizuje modul za upravljanje systemskom korisničkom spregom na aplikativnom nivou, s tim što ovakve aplikacije imaju pristup suženom skupu komandi (ograničenom na dobavljanje *proxy* ili UI naredbe, odnosno slanje korisničke akcije). Svaka ovako realizovana aplikacija poseduje sopstveni *bash* Linux skript koji je pokreće na odgovarajući način. Pri pokretanju aplikacije iz systemske korisničke

Model arhitekture platforme

sprege, najpre se pokreće pripadajući skript koji aktivira grafičku aplikaciju, a tek nakon toga se aktivira kontekstualni skript za upravljanje radom aplikacije.

Grafičke aplikacije realizovane na prezentacionim jedinicama zasnivaju se na periodičnom prozivanju servera korišćenjem HTTP GET zahteva, za dobavljanje novog stanja grafičke sprege. Za slanje informacija ka kontekstualnom skriptu, koriste se HTTP POST zahtevi. Serverski deo ovih aplikacija, koji prihvata HTTP zahteve, koristi dalje TCP/IP komunikaciju da komunicira sa aplikativnim nivoom i realizovan je kao PHP skript na centralnoj kontekstualnoj platformi. Grafički deo je realizovan kao HTML/Javascript/CSS i izvršava se na prezentacionoj jedinici. Ilustracija ovog koncepta data je na Slici 92. Mehanizam je detaljnije opisan u okviru posebnog rada koji se bavi upravljanjem prikazom multimedijalnih prezentacija na prezentacionim jedinicama u okviru web pregledača (Bjelica, 2010d).



Slika 92. Ilustracija ažuriranja korisničke sprege na prezentacionoj jedinici sa centralne kontekstualne platforme

Grafičke aplikacije koje se izvršavaju na centralnoj kontekstualnoj platformi ili na prezentacionoj jedinici, mogu obavljati interakciju sa korisnikom na neki od

sledećih načina: (1) korišćenjem standardne sprege tastatura/miš/ekran osetljiv na dodir, što su događaji koje detektuje grafička aplikacija i o njima obaveštava kontekstualni skript i (2) posredstvom senzora kontekstualne platforme, kada o događaju kontekstualni skript obaveštava grafičku aplikaciju.

Sistemska korisnička sprega koristi TCP/IP komunikaciju sa aplikativnim nivoom, odnosno modulom za integraciju sa sistemskom spregom, uz dozvoljen pristup kompletnom dostupnom API-ju. Sistemska korisnička aplikacija realizovana je kao web aplikacija koju poslužuje server koji se izvršava na centralnoj kontekstualnoj platformi. Ovo omogućava pristup sistemskoj korisničkoj sprezi sa svih uređaja priključenih na lokalnu mrežu, koji poseduju web pregledač. Aplikacija je realizovana korišćenjem okruženja *Google Web Toolkit* (GWT) u Java jeziku i omogućava (1) CRUD operacije nad kontekstualnim aplikacijama, senzorima, aktuatorima, P-modelima, kontekstualnim polisama i korisnicima; (2) aktiviranje/deaktiviranje kontekstualnih polisa, (3) pokretanje i zaustavljanje aplikacija, (4) konfiguraciju kontekstualne platforme i sl. Arhitektura aplikacije koja realizuje sistemsku korisničku spregu je detaljno realizovana u okviru realizacije kontekstualne platforme, međutim, njen opis nadmašuje obim ove disertacije i nije predmet razmatranja.

3.3.3 Neke važne sekvence u okviru kontekstualne platforme

U ovom poglavlju dati su UML dijagrami sekvenci za nekoliko važnih tokova podataka u okviru kontekstualne platforme. Cilj dijagrama je da kompletiraju razumevanje rada kontekstualne platforme, u celini predstavljene nivoima, blokovima, modulima i klasama u poglavlju 3.3.2.

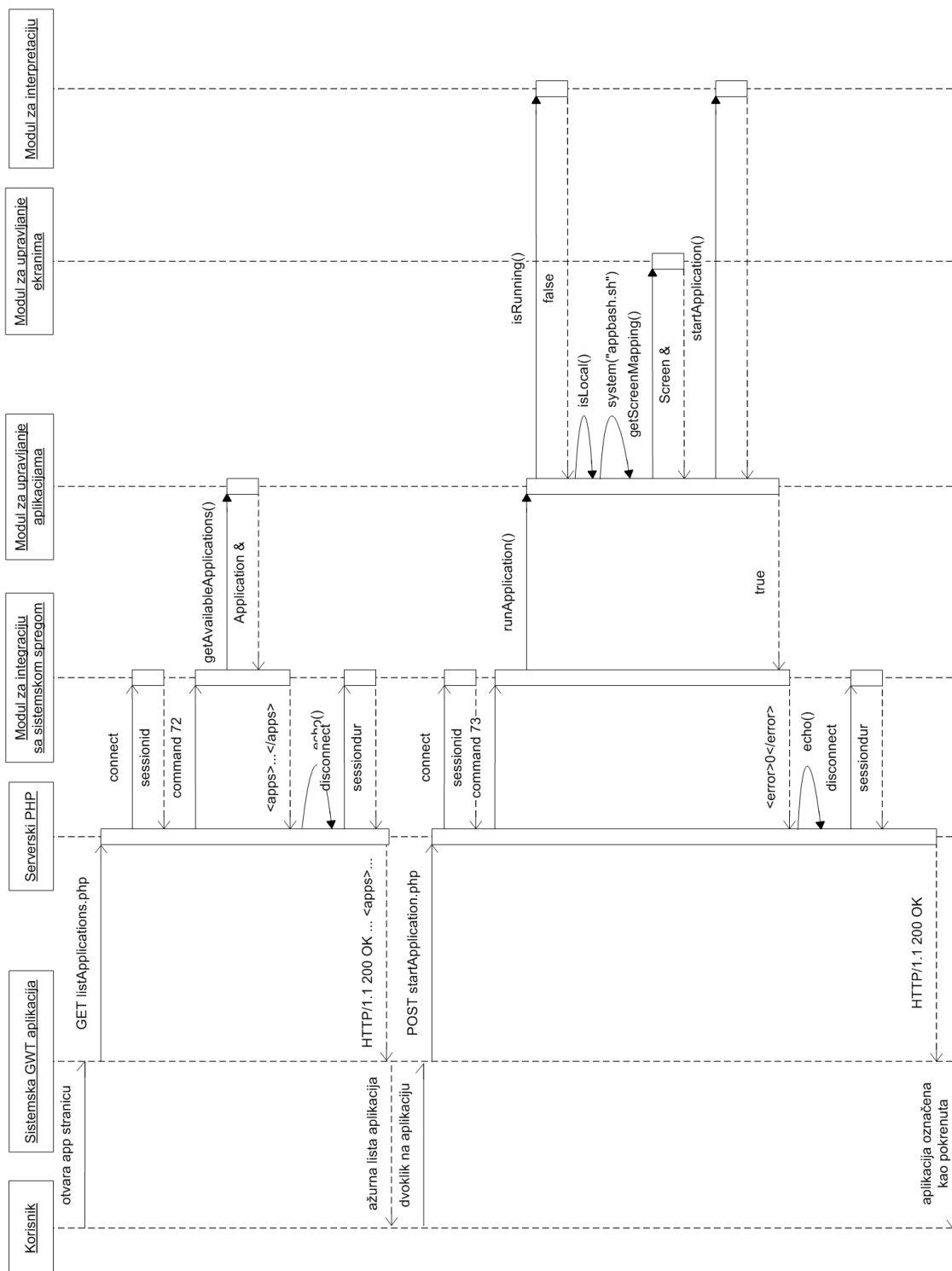
Na Slici 93 dat je UML dijagram sekvence za ilustraciju mehanizma pokretanja kontekstualne aplikacije iz sistemske korisničke sprege.

Na Slici 94 dat je UML dijagram sekvence za ilustraciju prijema novog događaja, izračunavanja nove vrednosti nivoa zauzetosti korisnika korišćenjem automatske semantizacije i njegovu isporuku posredstvom CTX API-ja.

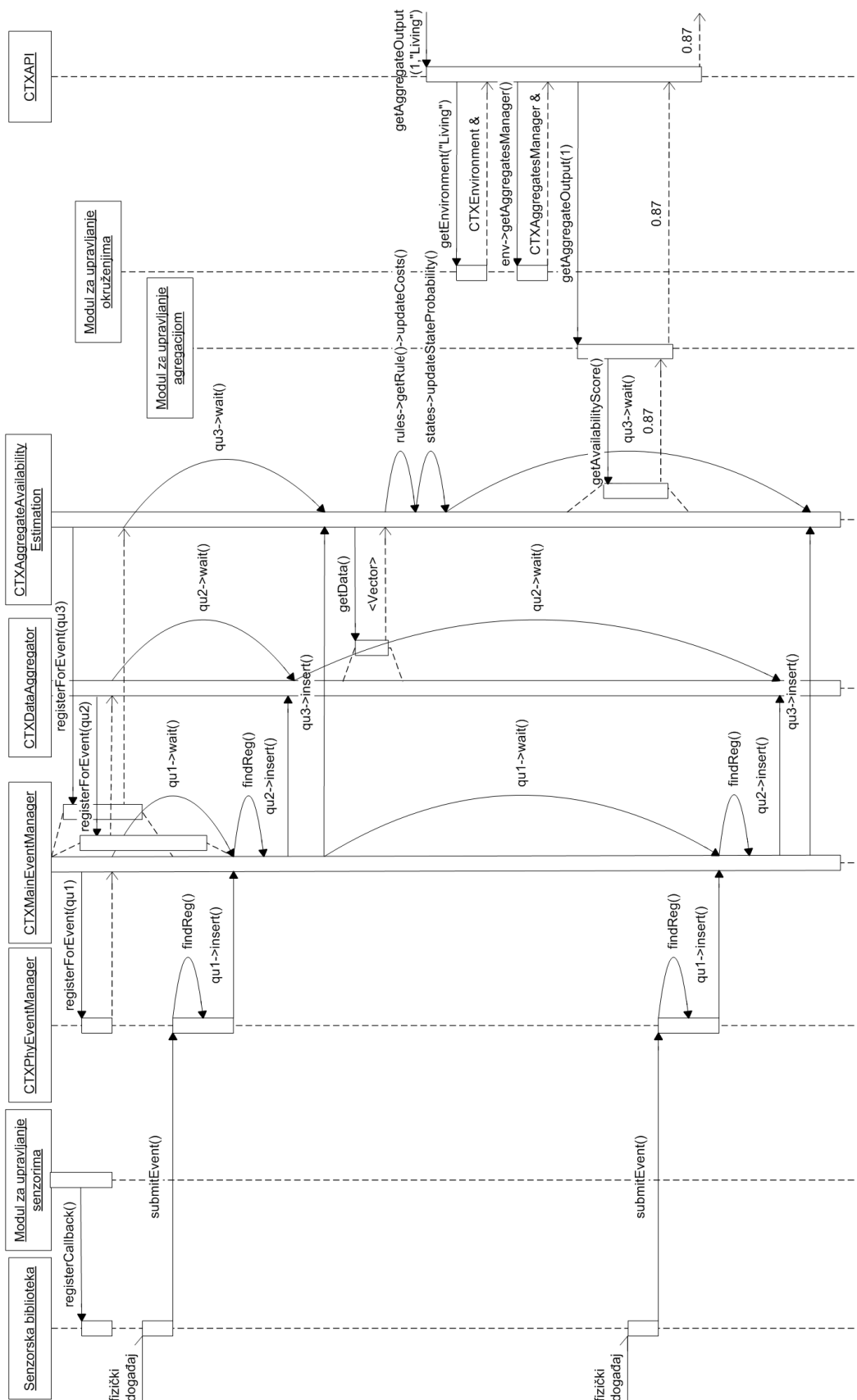
Na Slici 95 i 96 dat je UML dijagram sekvence za ilustraciju postavljanja režima rada aplikacije na TV prijemniku u zavisnosti od verovatnoće gledanja TV programa dobijenog kontekstualnom polisom koja najviše zavisi od detekcije lica korisnika koji gledaju TV program.

Model arhitekture platforme

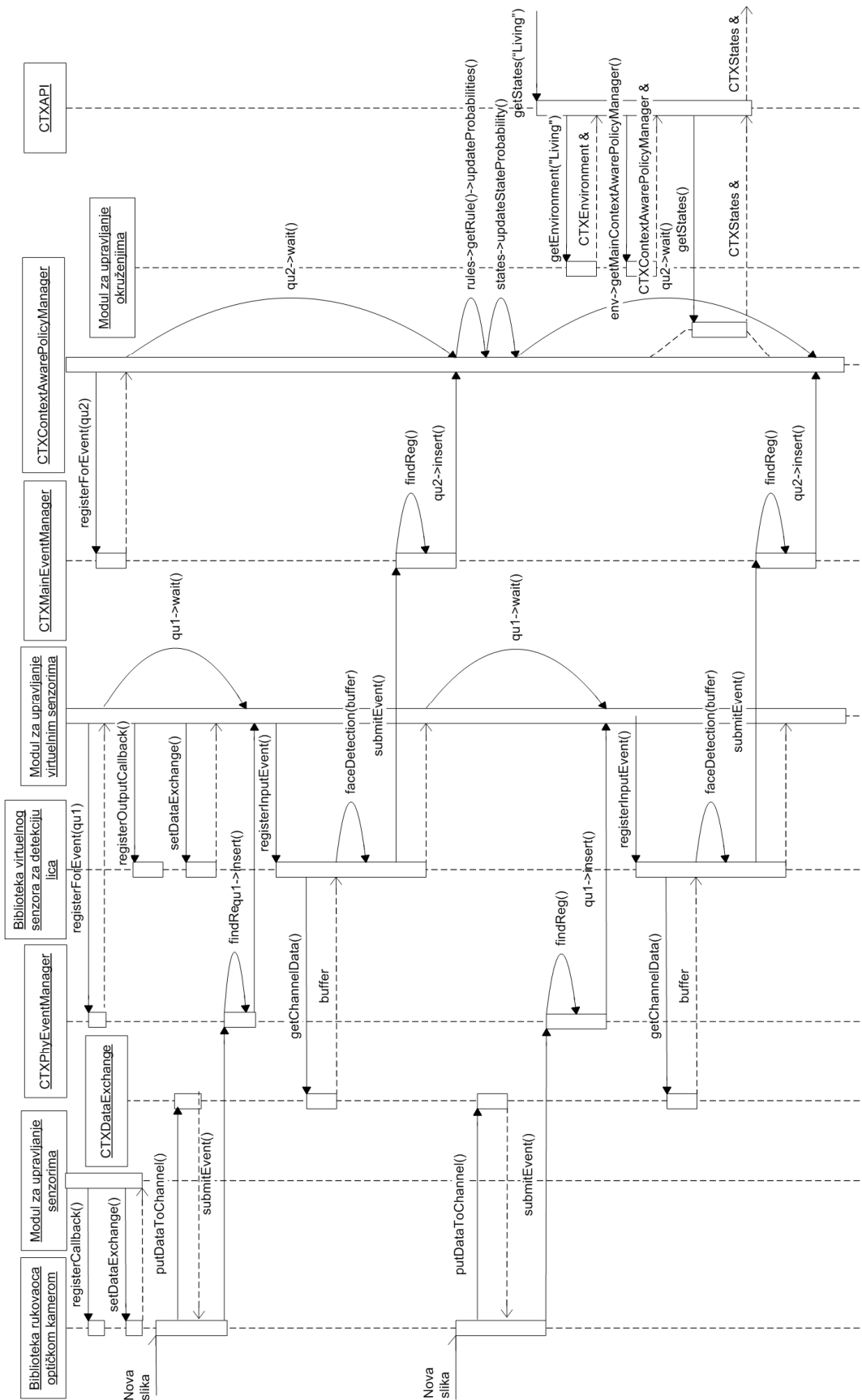
Na Slici 97 dat je UML dijagram sekvence za postupak deklaracije stanja u korisničkoj aplikaciji na mobilnom telefonu.



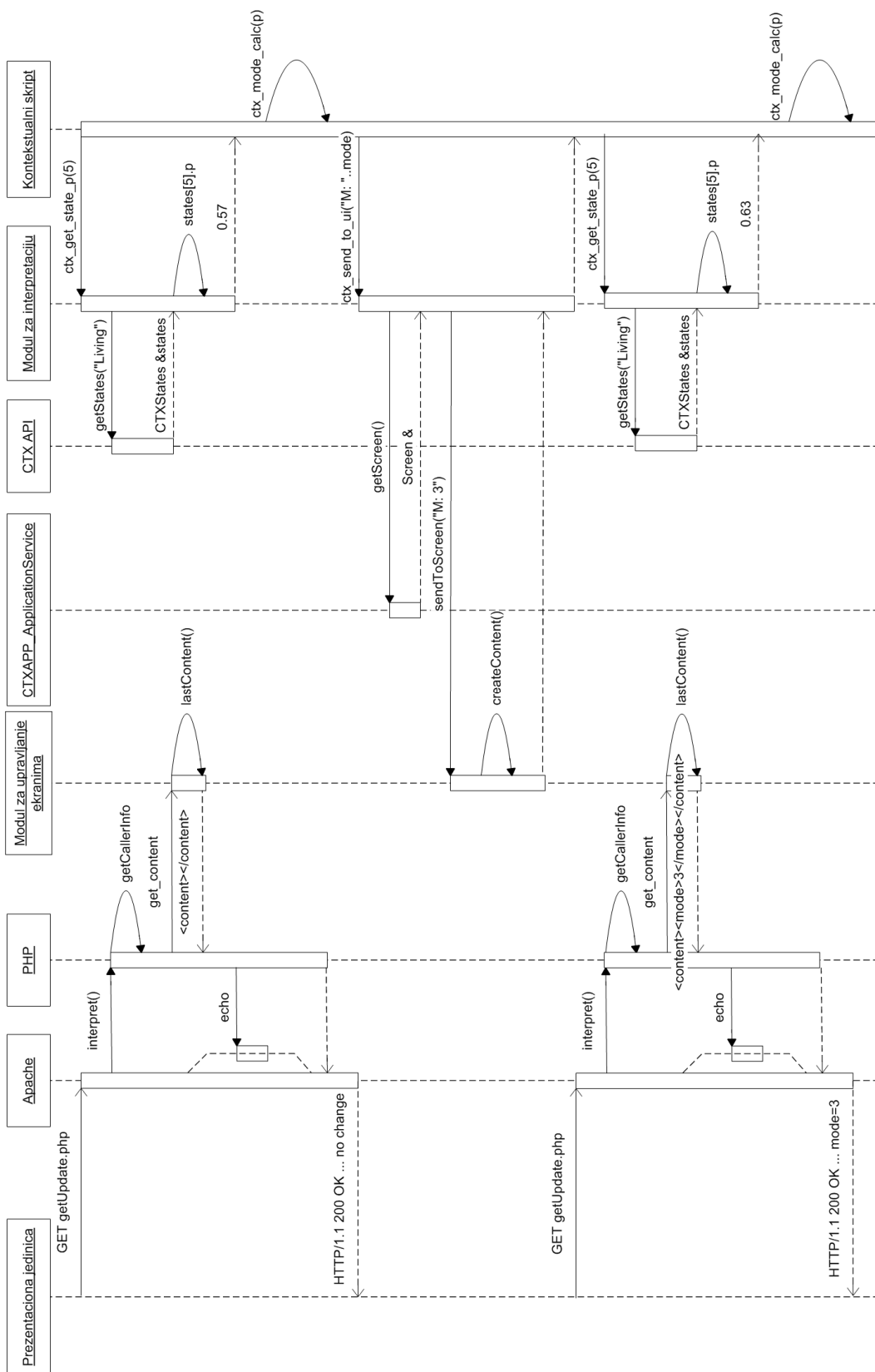
Slika 93. UML dijagram sekvence za ilustraciju mehanizma pokretanja kontekstualne aplikacije iz sistemske korisničke sprege



Slika 94. UML dijagram sekvence za izračunavanje stepena zauzetosti korisnika

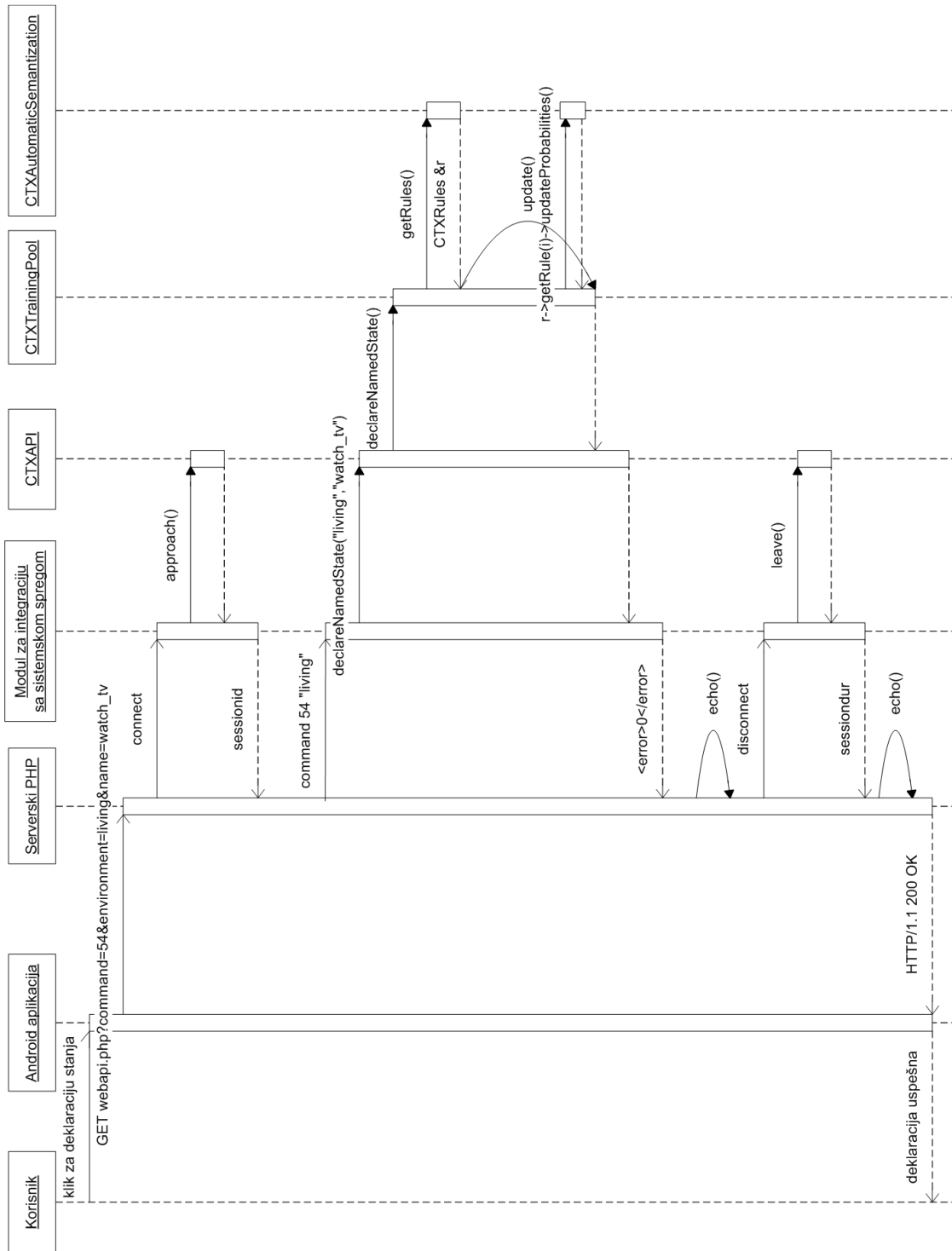


Slika 95. UML dijagram sekvence za izračunavanje nove verovatnoće stanja



Slika 96. UML dijagram sekvence za ažuriranje režima rada grafičke aplikacije na prezentacionoj jedinici u zavisnosti od verovatnoće stanja okruženja

Model arhitekture platforme



Slika 97. UML dijagram sekvence za deklaraciju novog stanja od strane posmatrača korišćenjem Android aplikacije na mobilnom telefonu

3.3.4 Pomoćna programska podrška

Osim programske podrške navedene u poglavlju 3.3.2, za uspešnu realizaciju kontekstualne platforme potrebna je i pomoćna programska podrška, koja neće biti detaljno razmatrana u okviru ove disertacije. Elementi pomoćne programske podrške su:

- *watchdog* proces za praćenje parametara rada procesa koji realizuje programsku podršku kontekstualne platforme. U slučaju da se otkrije veće zauzeće memorije od planiranog, preterano veliko opterećenje procesora i sl. ovaj proces resetuje procese koje prati.
- inicijalizacioni skriptovi, kojima se obezbeđuje odgovarajuća konfiguracija Linux operativnog sistema i obezbeđuju uslovi za pokretanje osnovnih aplikacija.
- skriptovi za ažuriranje verzije programske podrške kontekstualne platforme.
- Aplikacije za mobilne telefone koje omogućuju postupak treniranja (deklaracija stanja, unos cene asinhronog zahteva).

POGLAVLJE 4.

PRIMER REALIZACIJE

U ovom poglavlju dat je opis kontekstualne platforme realizovane u skladu sa predstavljenim modelom arhitekture kontekstualne platforme. Dat je spisak i opis konkretne opreme korišćene za realizaciju platforme, kao i spisak komponenti (senzora, aktuatora, aplikacija) sa kojima je platforma uspešno integrisana i koje su korišćene u njenom radu.

4.1 USER AWARENESS KIT

Naziv razvijene kontekstualne platforme je *User Awareness Kit* (oprema za pažnju prema korisniku) - UAK. UAK je realizovan kao samostalna (*stand alone*) platforma koju je moguće integrisati u ciljno inteligentno okruženje. Ipak, poštovanjem modela arhitekture kontekstualne platforme, omogućena je prenosivost programske podrške na različite ciljne arhitekture. Šta više, zahvaljujući korišćenju prenosivih biblioteka zasnovanih na POSIX standardu, UAK se može prevesti i za Linux i za Windows okruženje, uz minimalne razlike u korišćenom kodu. S tim u vezi, za UAK se može reći da predstavlja programsko rešenje. Ipak, u toku trajanja istraživanja, korišćeni su posebni blokovi fizičke arhitekture razvijene u okviru srodnih istraživanja i integrisani u UAK u cilju što boljeg približavanja predloženom modelu arhitekture. Ovde se prevashodno misli na realizovane kontekstualne module i pojedine

Primer realizacije

senzore/aktuatora koji se u toku istraživanja nisu mogli naći u slobodnoj prodaji (*off the shelf*), što je činjenica koja se može promeniti do trenutka objavljivanja disertacije.

Na temi UAK okruženja izrađen je veći broj naučnih radova. Poređano hronološki, polazna osnova za realizaciju UAK okruženja je sistem za automatizaciju domaćinstva najpre razvijen za samostalni kontroler (Bjelica, 2008), a kasnije ugrađen u set-top boks (STB) i TV prijemnik (Bjelica, 2009a; Bjelica, 2010c; Bjelica, 2010e; Bjelica, 2010f). Kao prototip sa osnovnim funkcijama, UAK je predstavljen na konferenciji pri sajmu potrošačke elektronike CES 2010 u Las Vegasu (Bjelica, 2010b). Nakon toga, u toku istraživanja UAK je napredovao i proširen je funkcijama upravljanja asinhronim zahtevima u smislu otkrivanja nivoa zauzetosti korisnika (Bjelica, 2010g; Bjelica, 2011a) i objavljivanja stepena zauzetosti korišćenjem svetlosnih efekata (Bjelica, 2011c; Bjelica, 2011d). Realizacije aplikacija proširenih UAK platformom predstavljane su u časopisu *IEEE Transactions on Consumer Electronics*, u kome se može pronaći više detalja realizovane platforme i mehanizma integracije (Bjelica, 2011f). Potpuniji prikaz arhitekture platforme i evaluacija algoritama za semantizaciju prihvaćeni su za objavljivanje u časopisu *IEEE Transactions on Systems, Man and Cybernetics* (Bjelica, 2013). Kao potrebni gradivni elementi u istraživanju, najpre su analizirana osnovna polazišta za realizaciju sistema, koja uključuju osnovne teoretske pretpostavke upotrebljivosti i prihvatanja aplikacija od strane korisnika (Bjelica, 2009b; Bjelica, 2009c), zatim su razvijani algoritmi za semantizaciju konteksta (Bjelica, 2010a), i obavljena procena i izbor skript mehanizama i jezika za opis kontekstualnih polisa i kontekstualnih skriptova (Maric, 2010; Bjelica, 2010h; Bjelica, 2011b; Bjelica, 2011g).

4.1.1 Fizička arhitektura kontekstualne platforme UAK

U okviru UAK sistema realizovani su sledeći elementi fizičke arhitekture, koncipirani u poglavlju 3.3.1: (1) centralna kontekstualna platforma; (2) kontekstualni modul tipa A, i (3) prezentacioni modul tipa A.

Centralna kontekstualna platforma je namenski računar (*embedded PC*) sa Linux operativnim sistemom. Za fizičku izvedbu korišćene je namenski računar *Stratus* kompanije *Ionics* (Stratus, 2012). Osim skromnih dimenzija, velikog broja podržanih sprega i zadovoljavajućih performansi, *Stratus* se jednostavno napaja direktnim priključivanjem na strujnu priključnicu. Fizičke karakteristike centralne kontekstualne

platforme date su u Tabeli 14. Izgled centralne kontekstualne platforme dat je na Slici 98.

<i>Naziv parametra</i>	<i>Vrednost</i>
Procesor	1.2 GHz Marvell
Radna memorija	512MB DDR2
Programabilni Flash	512 MB NAND
Mrežne tehnologije	10/100/1000 Base-T gigabitni Ethernet 802.11 b/g WiFi Bluetooth 2.1 ZWave Zigbee (802.15.4), korišćenjem USB dongle CC2531 DK
Napajanje	18W integrisano, 100-240V, 50/60 Hz
Konektori, tasteri, diode	3 programabilna tastera 2 x USB 2.0 tip A RJ45 Ethernet konektor Status LED diode
Sertifikacija	UL, FCC, CE, RoHS
Operativni sistem	Debian 5 Lenny Linux 2.6 kernel Marvel u-boot (v3.4.26)

Tabela 14. Fizičke karakteristike realizovane centralne kontekstualne platforme

Kontekstualni modul tipa A realizovan je u saradnji sa srodnim istraživačkim projektom, a predstavlja odvojenu jedinicu fizičke arhitekture sa mogućnošću upravljanja digitalnim izlazima, PWM, kao i DMX/DALI svetlima, odnosno prijem senzorski ulaza korišćenjem digitalnih i analognih ulaza. Fizičke karakteristike kontekstualnog modula tipa A date su u Tabeli 15.

Ulogu prezentacionog modula tipa A imao je PC računar opšte namene na kome se izvršavao *Mozilla Firefox* web pregledač. Takođe, uspešnu prezentacionu ulogu obavio je i set-top box, kao i TV prijemnik nove generacije sa ugrađenim web pregledačem.

Primer realizacije



Slika 98. Namenski računar kojim je realizovana centralna kontekstualna platforma

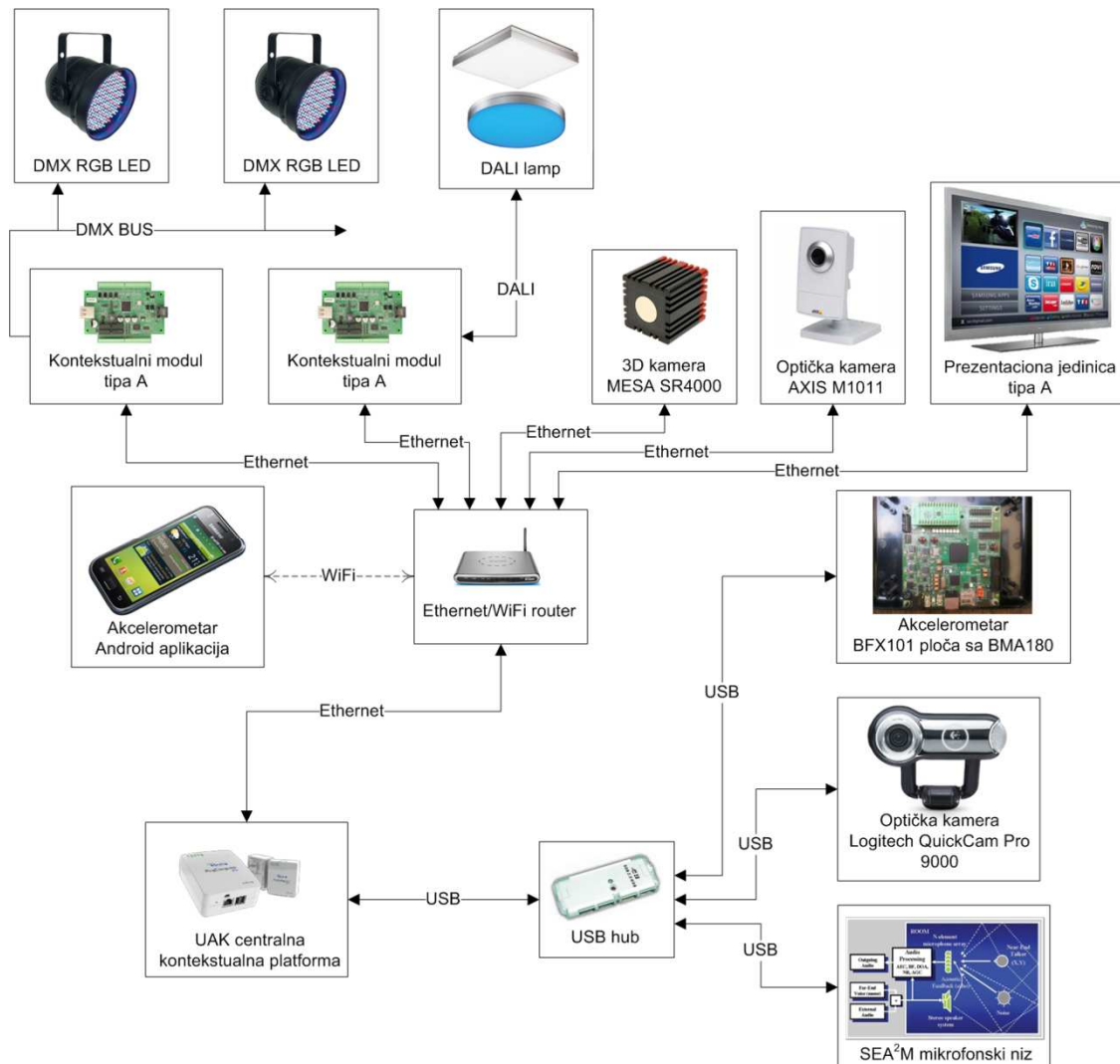
<i>Naziv parametra</i>	<i>Vrednost</i>
Procesor	Microchip PIC32MX
Radna memorija	8 KB
Programabilni Flash	32 KB
Mrežne tehnologije	10/100/1000 Base-T gigabitni Ethernet
Napajanje	9-30V DC
Sprege	4 digitalna ulaza 4 analogna ulaza 4 digitalna izlaza 3 PWM izlaza + PWM GND RS232 izlazne linije DALI konektor DMX bus konektor

Tabela 15. Fizičke karakteristike realizovanog kontekstualnog modula tipa A

U slučaju da je potrebno da se zahtevni algoritmi, poput intenzivne obrade slike i zvuka, realizuju u okviru senzorskih biblioteka u okviru kontekstualne platforme, *Stratus* računar pruža nedovoljno dobre performanse. Za ove slučajeve korišćeni su namenski računarski sistemi boljih karakteristika, poput *Aopen DE67-HA*.

4.1.2 UAK ekosistem

Pod pojmom *UAK ekosistem* podrazumeva se UAK kontekstualna platforma i svi priključeni senzori i aktuatori sa kojima je UAK kontekstualna platforma povezana. Na Slici 99 dat je dijagram UAK ekosistema.



Slika 99. UAK ekosistem

4.1.3 Rukovaoci uređajima

Za podršku sensorima i aktuatorima iz UAK ekosistema, razvijen je veći broj rukovalaca uređajima, u vidu senzorskih i aktuatorskih biblioteka koje realizuju rukovanje sensorima i aktuatorima, kao i u vidu biblioteka virtuelnih senzora, koje obezbeđuju različite algoritme za obradu podataka sa senzora. Spisak realizovanih senzorskih biblioteka dat je u Tabeli 16. Spisak realizovanih aktuatorskih biblioteka dat je u Tabeli 17. Spisak realizovanih biblioteka virtuelnih senzora dat je u Tabeli 18.

<i>Naziv senzorske biblioteke</i>	<i>Opis</i>
<i>BFX_accel.so</i>	Rukovalac akcelerometarskim sensorom BMA180 posredstvom ploče BFX101
<i>optical_cam_usb.so</i>	Rukovalac optičkom kamerom priključenom na USB prolaz
<i>optical_cam_eth_mjpeg.so</i>	Rukovalac optičkom kamerom priključenom na Ethernet, ukoliko podržava MotionJPEG protokol za razmenu slika
<i>sea2m.so</i>	Rukovalac podacima sa mikrofonskog niza SEA ² M
<i>accel_android.so</i>	Rukovalac Android aplikacijom koja vrši monitoring akcelerometra na mobilnom telefonu
<i>3dcam_mesasr4000.so</i>	Rukovalac 3D kamerom MESA SR4000 priključenom na Ethernet

Tabela 16. Realizovane senzorske biblioteke za UAK kontekstualnu platformu

<i>Naziv aktuatorske biblioteke</i>	<i>Opis</i>
<i>ctx_module_dmx.so</i>	Rukovalac DMX magistralom posredstvom kontekstualnog modula
<i>ctx_module_dali.so</i>	Rukovalac lampama priključenim na DALI spregu kontekstualnog modula

Tabela 17. Realizovane aktuatorske biblioteke za UAK kontekstualnu platformu

U okviru šireg istraživanja realizovani su i dodatni rukovaoci, i to: (1) rukovalac PIR sensorom; (2) rukovalac ultrazvučnim sensorom; (3) rukovalac bežičnom

senzorskom mrežom za detekciju osoba na osnovu RSSI; (4) rukovalac kontaktim prekidačima; (5) rukovalac virtuelnog senzora za detekciju bar i QR kodova; (6) rukovalac 3D kamerom *Kinect*. S obzirom da eksperimentalna evaluacija UAK platforme sa ovim rukovaocima nije sprovedena u okviru disertacije, tekst disertacije ne pokriva detalje realizacije ovih rukovalaca. Spisak evaluiranih senzora podudaran je sa sensorima najčešće prisutnim u uređajima potrošačke elektronike (akcelerometri, mikrofoni, kamere), te je skup korišćenih senzora odgovarajući s obzirom na temu ove disertacije.

<i>Naziv biblioteke virtuelnog senzora</i>	<i>Opis</i>
<i>virt_micarray.so</i>	Obezbeđuje zaključke na osnovu mikrofonskog ulaza, kao što su: (1) detekcija govora; (2) ugao trenutnog govornika u odnosu na centar mikrofonskog niza (Papp, 2007)
<i>optical_cam_recognition.so</i>	Realizuje algoritme obrade slike sa optičke kamere, i obezbeđuje sledeće informacije: (1) broj osoba detektovanih kamerom; (2) pozicije osoba na X osi na osnovu težišta objekta osobe; (3) zona dubine detektovane osobe (blizu, srednje daleko, daleko); (4) registrovanje pokreta; (5) detekcija lica.
<i>3d_cam_recognition.so</i>	Realizuje algoritme obrade slike sa 3D kamere, i obezbeđuje sledeće informacije: (1) broj osoba detektovanih kamerom; (2) pozicije osoba X, Y; (3) registrovanje pokreta.
<i>accel_gestures.so</i>	Omogućuje detekciju jednostavnih akcelerometarskih gestova koji su prethodno zadati (npr. krug, trougao, kvadrat, mahanje, brojevi i sl) (Košutić, 2011)

Tabela 18. Realizovane senzorske biblioteke za UAK kontekstualnu platformu

Primer realizacije

U Tabeli 19 dati su primeri tipičnih situacija koje je moguće pouzdano detektovati korišćenjem senzora integrisanih u okviru UAK kontekstualne platforme, na osnovu podataka sa fizičkih i virtuelnih senzora i na osnovu zaključaka kontekstualne polise.

<i>Naziv tehnologije</i>	<i>Situacije koje se pouzdano otkrivaju</i>
<i>Akcelerometar</i>	(1) kretanje korisnika, nasuprot mirovanju; (2) stanje korišćenja telefona, odnosno nekog drugog uređaja, uz dodatnu pomoć mikrofonskog ili nekog drugog senzora; (3) obrasci kretanja, npr. trčanje nasuprot hodanju i sl; (4) kontekstualni korisnički unos korišćenjem gesta.
<i>Optička kamera</i>	(1) broj osoba u prostoriji, na osnovu čega je moguće pomoći u zaključivanju situacije (sastanak, gledanje TV i sl); (2) usmerenost pažnje na osnovu detekcije lica, te je moguće zaključiti npr. aktivnost gledanja TV programa u sobi; (3) na osnovu učestanosti otkrivanja kretanja moguće je pretpostaviti opšte stanje u prostoriji (zabava, sastanak i sl).
<i>3D kamera</i>	(1) broj osoba u prostoriji, na osnovu čega je moguće pomoći u zaključivanju situacije, uz robustnost na nivo osvetljenosti prostorije; (2) obrasci kretanja korisnika na osnovu njihove pozicije u prostoru, na osnovu čega je moguće pretpostaviti opšte stanje u prostoriji. (3) blizina korisnika određenom uređaju/prostoru, odnosno pripadnost zoni, koja definiše određenu aktivnost (npr. gledanje TV programa, pripremanje kafe i sl)
<i>Mikrofonski niz</i>	(1) na osnovu detekcije zvuka daje se doprinos odlučivanju o aktivnosti (npr. mirovanje/spavanje nasuprot diskusiji); (2) na osnovu promene ugla trenutnog govornika moguće je prilično pouzdano odrediti da li je u toku sastanak/razgovor.

Tabela 19. Primeri tipičnih situacija koje je moguće detektovati korišćenjem rukovalaca sensorima, virtuelnih senzora i kontekstualnih polisa

POGLAVLJE 5.

EKSPERIMENTALNI REZULTATI I

DISKUSIJE

U okviru ovog poglavlja izloženi su rezultati rada realizovane kontekstualne platforme i obavljeno poređenje sa ekvivalentnim tehnologijama u oblasti. Eksperimentalno istraživanje za cilj ima nekoliko osnovnih fokusnih tačaka: (1) merenje performanse rešenja, u cilju potvrđivanja funkcionalne ispravnosti i tehničke upotrebljivosti u uređajima potrošačke elektronike; (2) merenje odziva (brzine reagovanja) sistema na različite promene u okruženju, kao ključnog parametra prihvatanja rešenja od strane korisnika, i poređenje odziva sa ekvivalentnim sistemima; (3) merenje tačnosti semantizacije okruženja korišćenjem metoda samoprocene i procene posmatrača, na osnovu tipičnih scenarija korišćenja, i poređenje tačnosti sa nekim vodećim istraživanjima u oblasti i (4) evaluacija potencijala rešenja za razvoj kontekstualnih aplikacija i uporedni prikaz sa ostalim vodećim rešenjima u oblasti.

5.1 MERENJE PERFORMANSI REŠENJA

U ovom poglavlju dati su rezultati merenja tehničkih performansi rešenja u smislu: (1) vremena pokretanja sistema; (2) količine zauzete memorije i ponašanja utroška memorije tokom dužeg vremena korišćenja; (3) zauzeća procesora i ponašanja

Ekperimentalni rezultati i diskusije

zauzeća procesora tokom dužeg vremena korišćenja; (4) broja pokrenutih niti i kretanja broja pokrenutih niti tokom dužeg vremena korišćenja i (5) broja otvorenih datoteka i ponašanja broja otvorenih datoteka tokom dužeg vremena korišćenja. Najpre je data metodologija merenja, zatim su dati rezultati i, konačno, diskusija dobijenih rezultata.

5.1.1 Metodologija merenja performansi rešenja

Za merenje performansi rešenja korišćeni su testovi opterećenja (*stress tests*) razvijeni prevashodno za ovu namenu. Testovi su realizovani u obliku kontekstualne aplikacije (kontekstualnog skripta), dok se za podrazumevani skup funkcionalnosti podrazumevao onaj skup kome je moguće pristupiti iz kontekstualnog skripta, što se u velikoj meri podudara sa funkcionalnošću koju pruža CTX API (vidi poglavlje 3.3.2.9), kao i aplikativni nivo programske podrške (vidi poglavlje 3.3.2.10). Pokrivenost ovog skupa funkcionalnosti razvijenim kontekstualnim skriptom je bila ~78% (25 od 32 poziva). Osnovni tok funkcionalnosti kontekstualnog skripta ogledao se u sledećem: (1) registruj se za sve događaje na svim dostupnim sensorima; (2) obavi kontinualan prijem događaja; (3) na osnovu jednostavne zavisnosti od primljenih događaja aktiviraj svetla na DMX/DALI spregama; (4) periodično obavljaj prijem trenutnog nivoa zauzetosti korisnika i verovatnoće svih dostupnih stanja; (5) periodično šalji ažurirane informacije grafičkoj sprezi. Grafička sprega je izvršavana u okviru web pregledača koji se izvršavao na prezentacionoj jedinici.

Obavljano je merenje vremena pokretanja sistema koje protekne od fizičkog priključivanja *Stratus* računara u strujnu priključnicu, do pojavljivanja ispisa u Android aplikaciji nakon što se ista uspešno povezala na kontekstualnu platformu korišćenjem HTTP sprege. Android aplikacija je realizovana tako da pokušava da obavi konekciju u petlji sa minimalnim čekanjem (100 ms). Ispitivanje je izvršeno merenjem vremena za 20 uzastopnih pokretanja. Takođe, dato je poređenje brzine pokretanja sistema sa ekvivalentnim kontrolerima u smislu funkcionalnosti, koji su po karakteristikama fizičke arhitekture ekvivalentni ili bolji u odnosu na fizičku arhitekturu kontekstualne platforme (*Control4* kontroleri EC100 i HC300 i PRS ExperiencePlayer). U slučaju *Control4* kontrolera, trenutak u kome se podrazumeva završetak postupka pokretanja je onaj u kome je moguće obaviti povezivanje korišćenjem aplikacije *Director*, što je, ispostavilo se, trenutak blizak trenutku

uključivanja signalnih dioda na samom kontroleru (HC300) odnosno poruke *Loading navigator* (EC100). U slučaju PRS kontrolera, trenutak završetka pokretanja nastupa kada postane moguće obaviti povezivanje na kontroler korišćenjem odgovarajućeg TCP/IP API-ja.

Obavljeno je merenje utroška memorije koju zauzimaju svi procesi kontekstualne platforme u okviru operativnog sistema Linux, korišćenjem periodičnog automatskog uvida u datoteku */proc/PID/status* za svaki praćeni proces. Uspešnost testa ogleda se u tome da u toku dužeg perioda korišćenja (praćeno u periodu od preko 12 sati neprekidnog ispitivanja) ne dođe do značajnog povećanja utroška memorije u odnosu na stanje neposredno nakon pokretanja.

Obavljeno je merenje zauzetosti procesora od strane svih procesa kontekstualne platforme u okviru operativnog sistema Linux, korišćenjem periodičnog automatskog pozivanja komande *ps* za svaki praćeni proces. Uspešnost testa ogleda se u tome da u toku dužeg perioda korišćenja (praćeno u periodu od preko 12 sati neprekidnog ispitivanja) ne dođe do značajnih oscilacija ili povećanja zauzeća procesora u odnosu na stanje nakon aktiviranja kontekstualnog skripta.

Obavljeno je merenje broja pokrenutih niti od strane svih procesa kontekstualne platforme u okviru operativnog sistema Linux, korišćenjem periodičnog automatskog uvida u datoteku */proc/PID/status* za svaki praćeni proces. Uspešnost testa ogleda se u tome da u toku dužeg perioda korišćenja (praćeno u periodu od preko 12 sati neprekidnog ispitivanja) ne dođe do značajnih oscilacija ili povećanja broja pokrenutih niti tokom vremena.

Obavljeno je merenje broja otvorenih datoteka od strane svih procesa kontekstualne platforme u okviru operativnog sistema Linux, korišćenjem periodičnog automatskog pozivanja funkcije *lsof* za svaki praćeni proces. Uspešnost testa ogleda se u tome da u toku dužeg perioda korišćenja (praćeno u periodu od preko 12 sati neprekidnog ispitivanja) ne dođe do značajnih oscilacija ili povećanja broja otvorenih datoteka tokom vremena.

Linux skript korišćen za merenje dat je na Slici 100.

Ekperimentalni rezultati i diskusije

```
#!/bin/sh

print2log ()
{
    echo $1
    echo $1 >> $LOG_FILE_NAME
}

# Generate file name
LOG_FILE_NAME="$(ps -p $SCAN_PROC_ID -o comm=)_$(date '+%Y%m%d_%H%M%S').csv"

# look for process call command
print2log "Scanned process ID : $SCAN_PROC_ID"
print2log "Process call command: $(ps -p $SCAN_PROC_ID -o cmd=)"

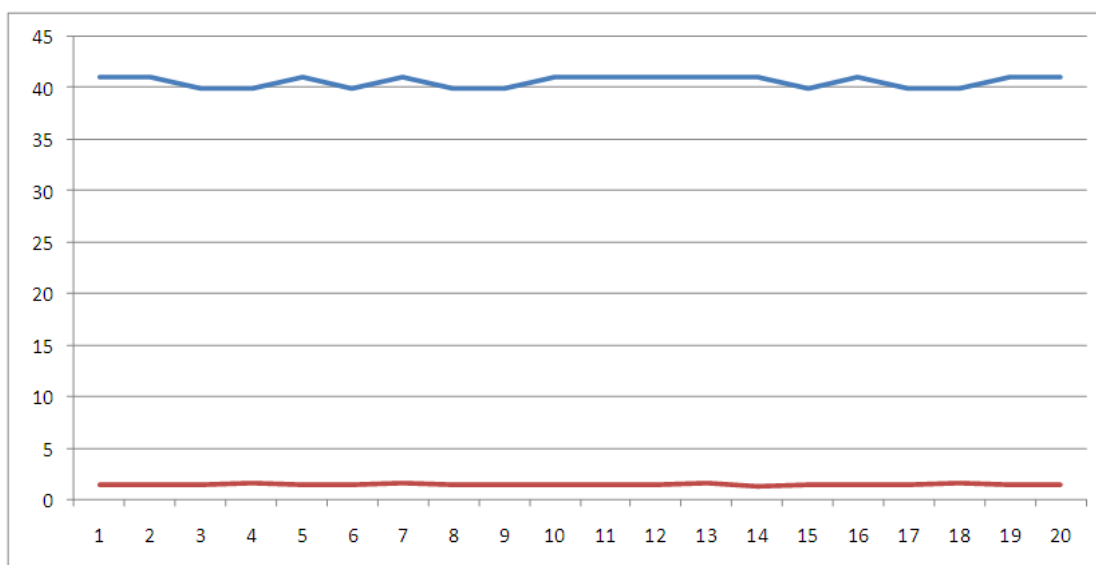
# print csv header
print2log "Time,PID,VmRSS,Thr,Proces opened files,CPU[%]"

while true
do
    ps -p $SCAN_PROC_ID > /dev/null
    if [ $? -ne 0 ]
    then
        echo "Process with ID \"$SCAN_PROC_ID\" does not exist. Exiting..."
        exit 1
    fi
    print2log "\
$(date '+%Y/%m/%d %H:%M:%S'), \
$SCAN_PROC_ID, \
$(cat /proc/$SCAN_PROC_ID/status | grep VmRSS | awk '{print $2}'),\t\
$(cat /proc/$SCAN_PROC_ID/status | grep Threads | awk '{print $2}'),\t\
$(lsof -p $SCAN_PROC_ID | wc -l),\t\
$(ps -p $SCAN_PROC_ID -o pcpu --no-headers),\t\
sleep 5
done
```

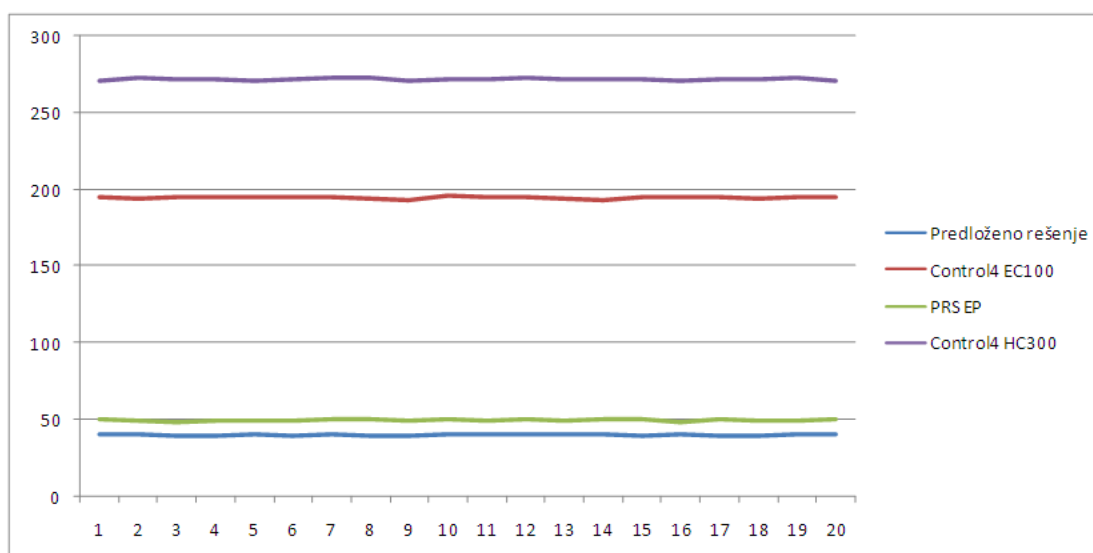
Slika 100. Linux skript korišćen za merenje performansi sistema

5.1.2 Rezultati merenja performansi rešenja

Merenje je otpočelo 14. decembra 2011. godine u 19:10h i trajalo je do 15. decembra 2011. godine u 9:57h, u ukupnom trajanju od 14h 7min. Merenje je otpočelo neposredno nakon uključivanja kontekstualne platforme u rad i pokretanja ispitne aplikacije, sa odmeravanjem na približno svakih 5 sekundi. Rezultati merenja vremena pokretanja sistema dati su na Slici 101. Poređenje vremena pokretanja sa ekvivalentnim kontrolerima dato je na Slici 102. Rezultati merenja količine zauzete memorije dati su na Slici 103. Rezultati merenja broja pokrenutih niti dati su na Slici 104. Rezultati merenja broja otvorenih datoteka dati su na Slici 105. Rezultati merenja zauzetosti procesora dati su na Slici 106.

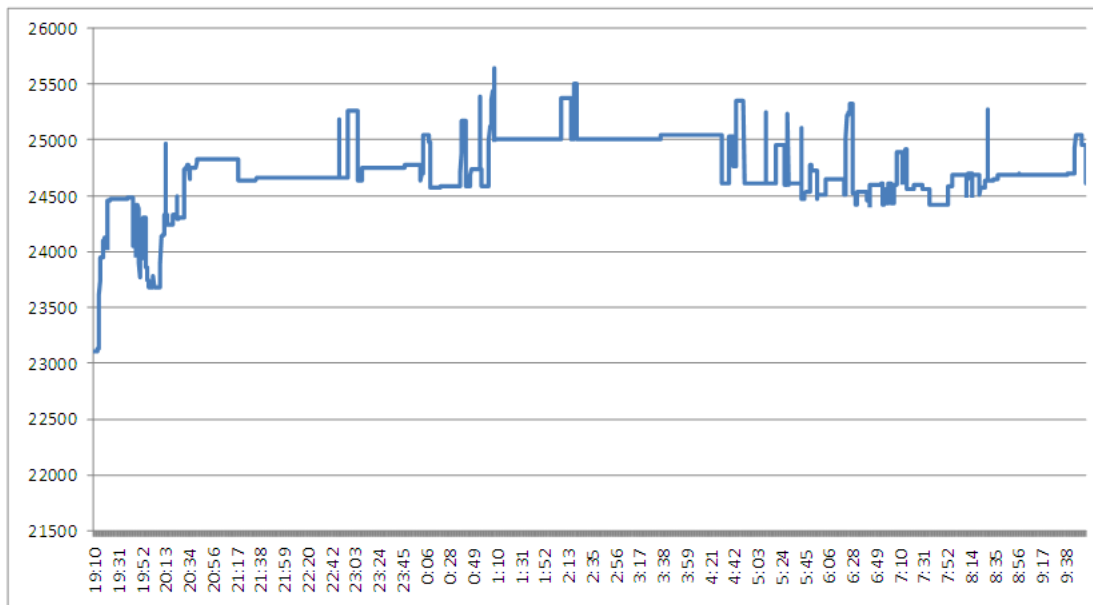


Slika 101. Rezultati merenja vremena pokretanja sistema: na X osi je prikazan redni broj merenja, dok je na Y osi prikazano vreme u sekundama (plava boja – vreme pokretanja celokupnog uređaja i programske podrške; crvena boja – vreme pokretanja aplikacije kontekstualne platforme)

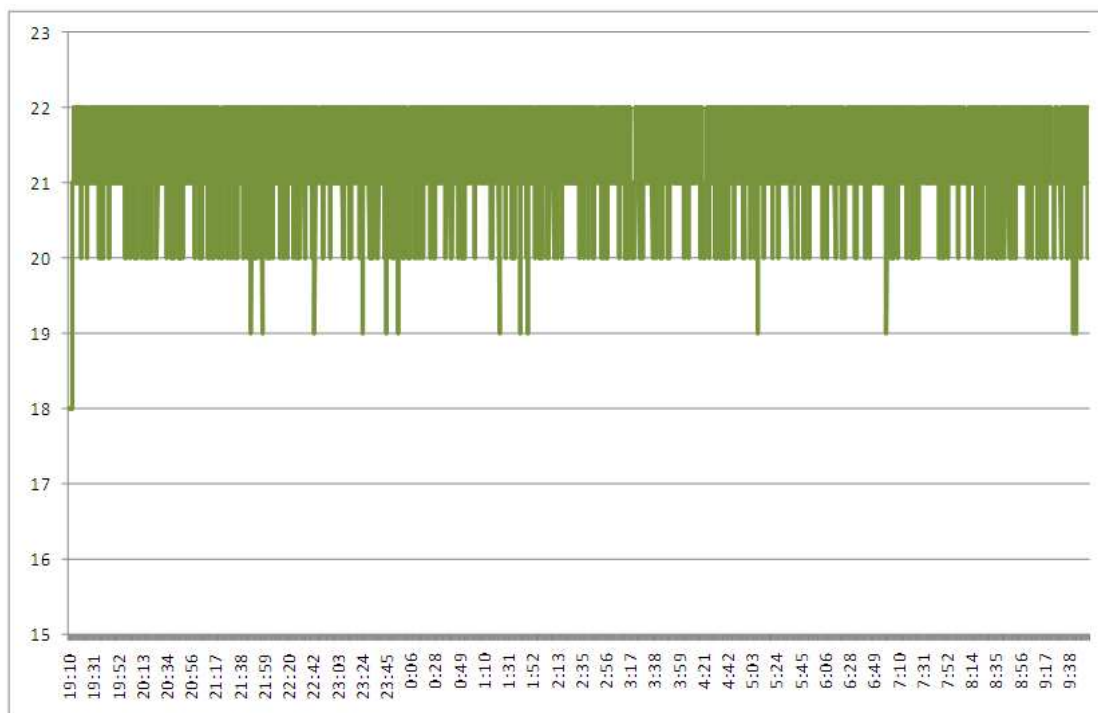


Slika 102. Poređenje vremena pokretanja sistema predložene platforme i platformi trenutno dostupnih na tržištu: na X osi je prikazan redni broj merenja, dok je na Y osi prikazano vreme u sekundama

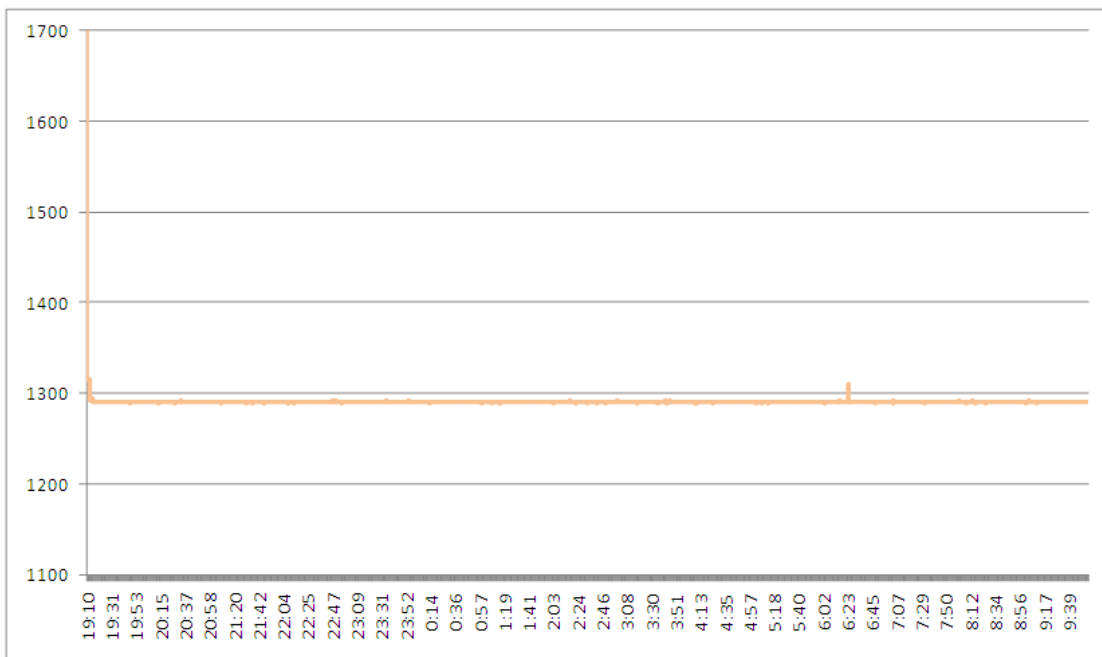
Ekspperimentalni rezultati i diskusije



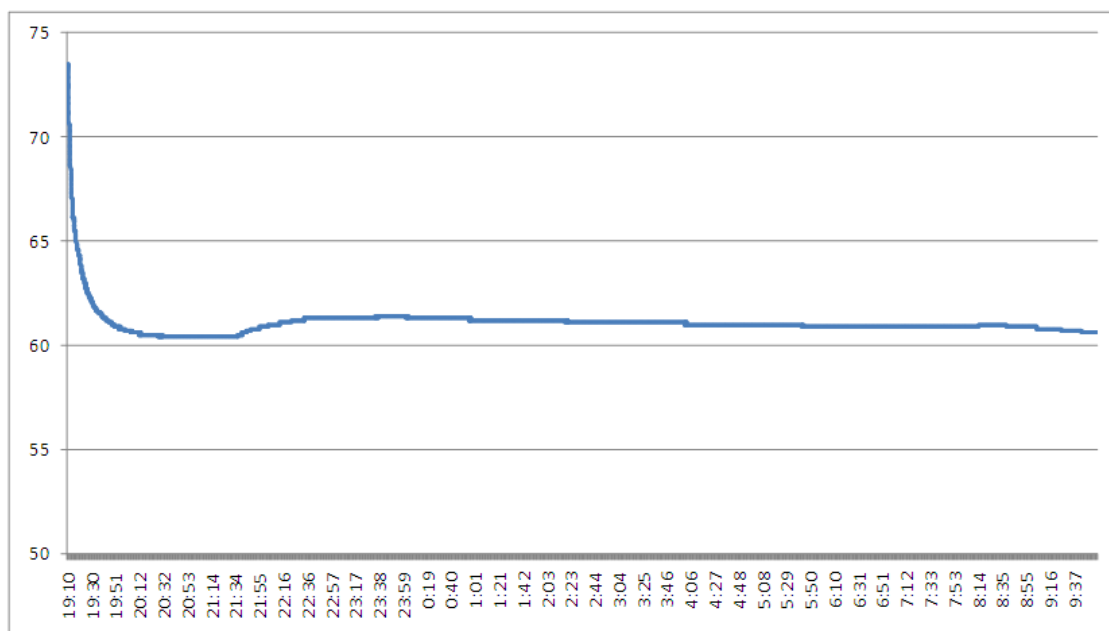
Slika 103. Rezultati merenja količine zauzete memorije: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan utrošak memorije (u Kb)



Slika 104. Rezultati merenja broja pokrenutih niti. na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan broj pokrenutih niti



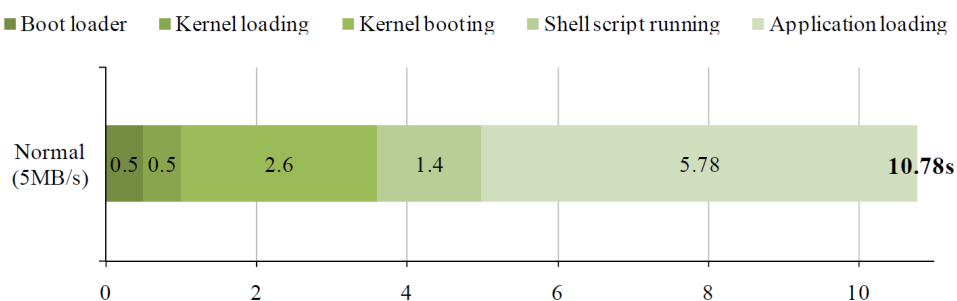
Slika 105. Rezultati merenja broja otvorenih datoteka: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazan broj otvorenih datoteka



Slika 106. Rezultati merenja zauzetosti procesora: na X osi je prikazano trenutno sistemsko vreme, dok je na Y osi prikazano zauzeće procesora (u %)

5.1.3 Diskusija i zaključci

Na osnovu rezultata merenja vremena pokretanja sistema, može se uočiti da je vreme približno 40 sekundi od momenta uključivanja uređaja, što je redom veličine uporedivo, a prema konkretnom merenju bolje u odnosu na kontrolere sa kojima je obavljeno poređenje. Ipak, primetna je velika razlika u vremenu pokretanja različitih kontrolera prilikom ispitivanja, što je i opšti utisak koji se stiče uvidom u izveštaje drugih autora. Ta činjenica se može objasniti nezainteresovanošću proizvođača za ulaganje vremena i novca u optimizaciju vremena pokretanja, u situaciji kada se restart samostalnog kontrolera obavlja izuzetno retko. Ipak, s obzirom da kontekstualna platforma takođe nije optimizovana u ovom pogledu, izmereno vreme pokretanja predstavlja zadovoljavajući rezultat. Dalje, ono što je važno napomenuti je da u strukturi utroška vremena za pokretanje, samo 1-2s vremena zauzima pokretanje aplikacije i procesa kontekstualne platforme, dok ostatak vremena protekne u inicijalizaciji *Stratus* računara i operativnog sistema Linux koji se na njemu izvršava. Iz ovog razloga, vreme pokretanja sistema značajno je brže ukoliko se pokretanje osnovne programske podrške, kao što je učitavanje i podizanje Linux kernela i izvršavanje potrebnih *shell* skriptova, optimizuje. Uzmimo za primer slučaj modernog TV prijemnika, gde je optimizacija vremena pokretanja važan zahtev, sa vremenom pokretanja od približno 11 sekundi, i strukturom raspodele tog vremena datom na Slici 107 (Jo, 2009). Ukoliko se dati sistem proširi programskom podrškom kontekstualne platforme, ukupno vreme pokretanja TV aplikacija, sa trenutnih ~6 sekundi, biće podignuto na 7-8 sekundi. S obzirom da je uticaj na ukupno vreme pokretanja oko 10%, daljom optimizacijom ukupno vreme se može dovesti na raniji, ili na neprimetno duži nivo.



Slika 107. Struktura vremena potrebnog za pokretanje modernog digitalnog TV prijemnika (Jo, 2009)

Ukoliko se programska podrška kontekstualne platforme realizuje kao servis za mobilni telefon (npr. Android servis sa programskom podrškom kontekstualne platforme koja se izvršava u nativnom kodu), njeno pokretanje moguće je paralelizovati, tako da se aktivira tek nakon aktiviranja mobilnog telefona. U tom smislu korisnik ne primećuje povećanje vremena potrebnog za pokretanje programske podrške telefona, osim nešto većeg opterećenja procesora 1-2 sekunde nakon pokretanja.

Na osnovu rezultata merenja utroška resursa može se uočiti stabilno korišćenje resursa od strane kontekstualne platforme tokom dužeg vremena korišćenja. Primetna je, doduše, nešto manja potrošnja memorije na početku ciklusa merenja, nešto veće opterećenje procesora i veći broj pokrenutih niti. Ovo je posledica nešto veće aktivnosti kontekstualne platforme pri učitavanju, odnosno početak punjenja kružnih bafera koji služe, npr. za prikupljanje istorije događaja od strane senzora. Od trenutka kada su kružni baferi popunjeni, količina utrošene memorije postaje konstantna. Takođe, povremeno podizanje nivoa utrošene memorije i njeno vraćanje na konstantnu vrednost posledica je kontinualnog zauzimanja i oslobađanja memorije od strane kontekstualne platforme. S obzirom da je frekvencija odmeravanja pri merenju približno 5 sekundi, trenutna količina zauzete memorije u trenutku odmeravanja je stohastična veličina, koja se kreće oko srednje vrednosti.

Količina utrošene memorije, odnosno *footprint* programske podrške kontekstualne aplikacije, u iznosu od oko 25 MB, svedoči da je moguće izvršavanje programske podrške na novijim uređajima potrošačke elektronike. Potrebno je napomenuti, ipak, da ovaj iznos utroška memorije značajno zavisi od algoritama za obradu slike i kvaliteta slike koji se dobavlja od optičkih i 3D senzora. U slučaju kada se ovi senzori umereno koriste, ili se ne koriste uopšte, količina zauzete memorije opada za u proseku oko 60%.

5.2 MERENJE ODZIVA REŠENJA

U ovom poglavlju dati su rezultati merenja odziva rešenja na različite scenarije odziva. S obzirom na namenu kontekstualne platforme, odziv je izuzetno bitna karakteristika od koga zavisi pravovremena reakcija na stanje okruženja, što je posebno primetno pri realizaciji transparentnih kontekstualnih korisničkih sprega. U

Ekperimentalni rezultati i diskusije

daljem tekstu je najpre opisana metodologija merenja i navedeni odabrani scenariji za merenje odziva, zatim su dati rezultati merenja, poređenje nekih važnih vremena odziva sa ekvivalentnim trenutnim rešenjima i, konačno, diskusija dobijenih rezultata.

5.2.1 Metodologija merenja odziva rešenja

Odziv rešenja se realistično može izmeriti samo ukoliko se celokupan UAK ekosistem posmatra kao crna kutija. Dakle, na ulaz sistema, odnosno u „vidno polje“ senzora dovodi se određena pobuda koja aktivira određenu akciju u kontekstualnoj aplikaciji. Trenutak dovođenja pobude smatra se trenutkom početka merenja vremena, dok se trenutak kada rezultatna akcija postane vidljiva za posmatrača (korisnika), smatra trenutkom završetka merenja vremena.

S obzirom na delikatnost ispitivanja, merenje je obavljeno na osnovu naknadnog uvida u video snimak scene koji je načinjen u toku ispitivanja. Na video snimku se istovremeno mogu videti korisnik i medijum koji predstavlja izlaz (ekran prezentacione jedinice, odnosno svetlosni izvor čija se reakcija očekuje). Analizom video snimka u alatu *Virtual Dub*, koji omogućuje upravljanje reprodukcijom na nivou frejma (tj. rezolucijom od najviše ~40ms), može se jasno izdvojiti prvi frejm na kome je uočena generisana senzorska pobuda, kao i prvi frejm na kome je uočena rezultatna akcija. Razlika u vremenu video zapisa (*timestamp*) ova dva frejma predstavlja vreme odziva rešenja.

Vreme odziva koje je prihvatljivo za korisnika je izuzetno teško precizno kvalifikovati, s obzirom da ono zavisi od korisničkog očekivanja na osnovu prethodnog iskustva i konkretne očekivane funkcionalnosti. Ukoliko je idealna i teoretski zahtevana vrednost realno vreme, onda se očekuju veoma kratka vremena odziva. Različite studije još početkom druge polovine XX veka, odnosno od ranog vremena korišćenja računarskih sistema, bavile su se maksimalnim vremenom odziva do kog je korisnicima prihvatljivo čekanje na osnovu psiholoških i drugih posmatranih metrika koje su povezane sa prirodom čoveka. Na ovaj način došlo se do pravila „2 sekunde“, kao maksimalnog dozvoljenog vremena čekanja koje neće uticati na zadovoljstvo korišćenja sistema (Miller, 1968). Međutim, razvojem uređaja potrošačke elektronike i sve većim brojem korisnika različitih uređaja i servisa, ovo vreme se sve više smanjuje. Npr. najveće dozvoljeno procenjeno vreme čekanja korisnika, koje neće izazvati nelagodnost, pri izmeni programa na TV prijemu je 0.5 sekundi (Kooij,

2006). Sa stanovišta primena kontekstualne platforme za realizaciju kontekstualnih korisničkih sprega, 2 sekunde je preveliko vreme, pa čak i 0.5 sekundi može da bude previše. Za ovakve primene potreban je odziv u realnom vremenu sa stanovišta percepcije korisnika. Na osnovu postojećih istraživanja, prihvatljivo vreme odziva je ispod 200ms (Liukonnen, 2009).

Na osnovu gore navedene analize prihvatljivih vremena odziva, možemo ustanoviti tri bitna vremena odziva za obavljanje ispitivanja: (1) vreme odziva ispod 2 sekunde, kao apsolutni zahtev za sve kontekstualne aplikacije i kao prihvatljivo maksimalno vreme čekanja do promene verovatnoće stanja okruženja na odgovarajuću verovatnoću, ukoliko je stanje statične prirode i ne menja se velikom brzinom (npr. stanje „gleda TV“ ili stanje „sastanak“); (2) vreme odziva ispod 0.5 sekundi, za promene stanja izazvane svesnom korisničkom radnjom, kao što je svesna korisnička aktivacija određene funkcionalnosti; i (3) vreme odziva ispod 0.2 sekundi, za primene u realizaciji kontekstualnih korisničkih sprega u realnom vremenu, poput upravljanja u video igrama i sl. Za ustanovljavanje zadovoljenja ovih kriterijuma koristiće se statistički metod testiranja hipoteze za svaki od ispitnih slučajeva.

Za potrebe merenja odziva sistema realizovana su tri ispitna slučaja u okviru studije ljudskog faktora (*Human Factor Study*), koje ćemo označiti kao Slučaj A, Slučaj B i Slučaj C, u okviru dva okruženja, Okruženja A i Okruženja B. Oba okruženja se sastoje od systemske infrastrukture, i od scene na kojoj se dešavaju radnje koje izazivaju senzorske pobude. U svim ispitnim slučajevima generatori pobude su korisnici (ljudi), koji svojim prisustvom, pokretom ili govorom pobuđuju kontekstualnu platformu i izazivaju određenu akciju kontekstualne aplikacije.

Okruženje A predstavlja simulaciju okruženja u dnevnoj sobi, gde je scena centralizovana oko TV prijemnika koji služi kao prezentaciona jedinica UAK platforme. Najpodesnije mesto za postavljanje senzora u ovom slučaju je upravo TV prijemnik, tako da je 3D kamera postavljena vertikalno iznad TV prijemnika (za praćenje kretanja u celoj sceni), a optička kamera i mikrofonski niz postavljeni na TV prijemnik (Slika 108). U okviru ovog okruženja realizovana su dva ispitna slučaja (Slučaj A i Slučaj B). U okviru Slučaja A mereno je vreme odziva sistema na promenu stanja okruženja, odnosno na promenu verovatnoće stanja „razgovor“ sa niske verovatnoće (ispod 30%) na visoku verovatnoću (iznad 70%). Početak merenja

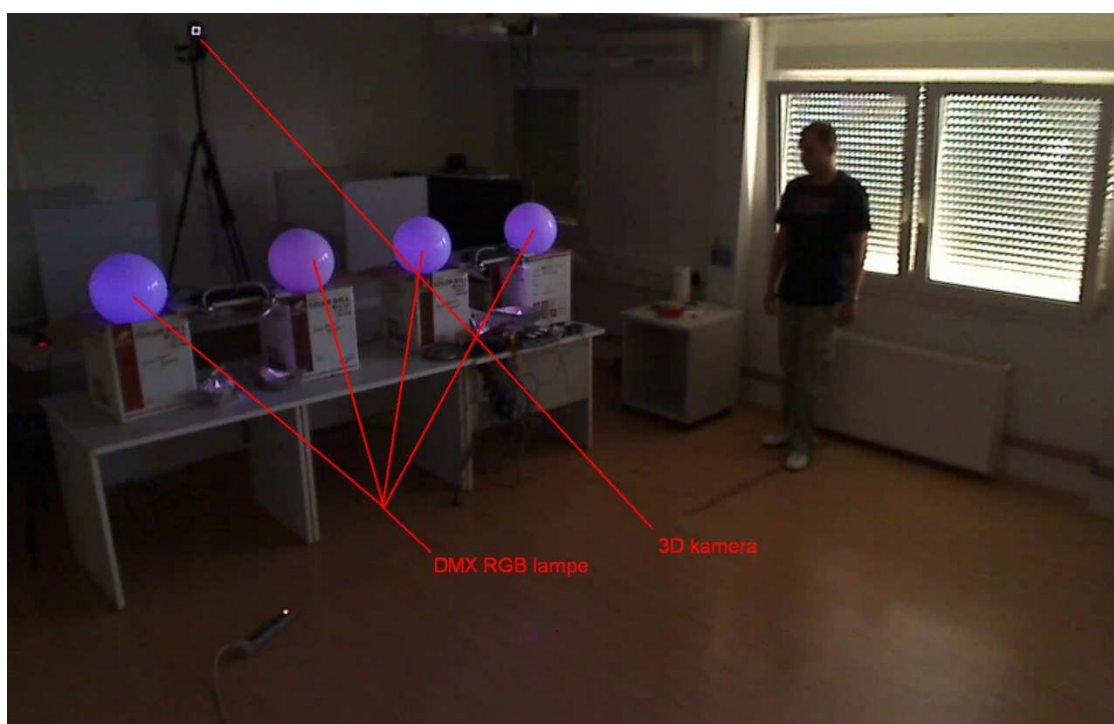
Ekperimentalni rezultati i diskusije

označen je prvim audio odmerkom koji sadrži izgovorenu reč, dok je kraj merenja označen prvim frejmom video snimka u kome se vidi da je verovatnoća stanja „razgovor“, označena vertikalnim indikatorom i brojčanom vrednošću vidljivim u okviru ekrana kontekstualne aplikacije na prezentacionoj jedinici, veća od 70%. Ovaj ispitni slučaj mapiran je na zahtev za vremenom odziva ispod 2 sekunde. Merenje se obavlja za 20 nezavisnih ponovljenih ogleda. U okviru ispitnog Slučaja B dodatno je korišćen uređaj sa akcelerometrom, posredstvom kojeg korisnik izdaje naredbu sistemu. Podizanjem uređaja horizontalno uvis i postavljanjem od sebe, korisnik definiše gest „stop“ i time inicira promenu stanja okruženja na kontekstualnoj platformi. Početak merenja označen je prvim frejmom video snimka, u kome je uređaj postavljen u potpunosti u položaj koji odgovara gestu „stop“. Kraj merenja označen je prvim frejmom video snimka u kome se vidi da je verovatnoća stanja „stop“, označena vertikalnim indikatorom i brojčanom vrednošću vidljivim u okviru ekrana kontekstualne aplikacije na prezentacionoj jedinici, iznad 70%. Ovaj ispitni slučaj odgovara promeni stanja izazvanoj svesnom korisničkom radnjom, te je zahtevano vreme odziva ispod 0.5 sekundi. Merenje se obavlja za 20 nezavisnih ponovljenih ogleda.



Slika 108. Izgled ispitnog Okruženja A: simulacija dnevne sobe sa postavljenim sensorima (optička kamera, mikrofonski niz, 3D kamera) i prezentacionom jedinicom priključenom na TV prijemnik

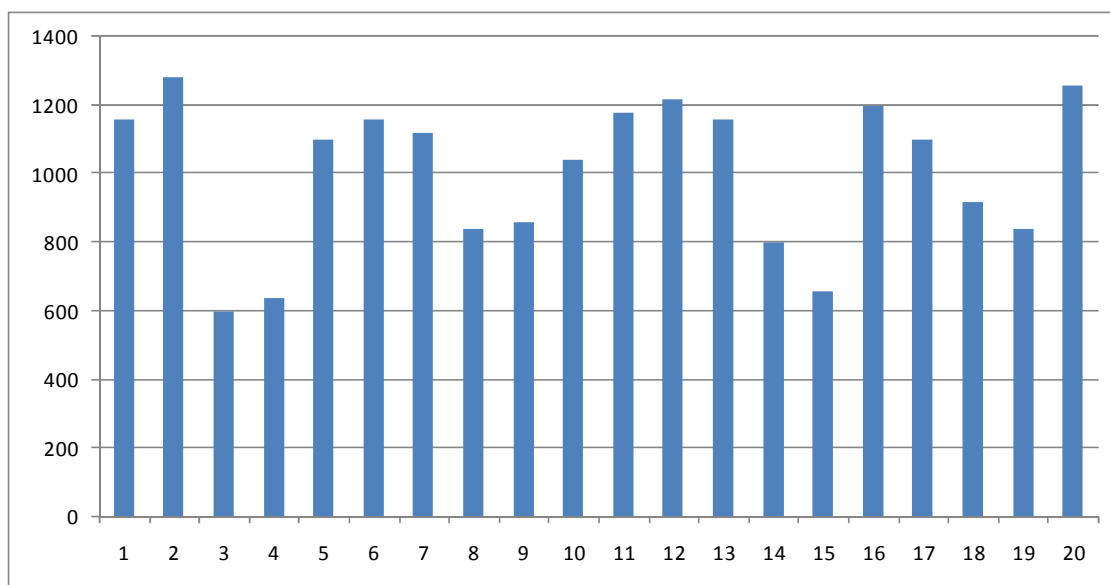
Okruženje B predstavlja scenu u kojoj je realizovana *Luminosity* aplikacija. Scena se sastoji od niza od četiri DMX RGB lampe, čiji intenzitet osvetljaja korespondira trenutnoj poziciji osoba u sceni, dok je jedini korišćeni senzor 3D kamera (Slika 109). U ovom okruženju realizovan je ispitni Slučaj C. U ovom ispitnom slučaju, najpre jedan korisnik, a potom dva korisnika istovremeno obavljaju kretanje po sceni. Početna tačka merenja odgovara prvom frejmu video snimka, u kojem korisnik zaklanja definisanu zonu lampe sa više od 50% tela (na osnovu očitavanja iz programske podrške za obradu slike). Krajnja tačka merenja odgovara prvom frejmu video snimka, u kojem DMX lampa izmeni vrednost osvetljaja sa prethodne vrednosti na novu vrednost. Ispitivanje se obavlja za slučaj kretanja korisnika u situaciji kada se samostalno kreće na sceni, odnosno kada se kreće sa još jednim korisnikom. Merenje se obavlja za više različitih pozicija korisnika u 20 pokušaja.



Slika 109. Izgled ispitnog Okruženja B: Scena za *Luminosity* aplikaciju sa postavljenom 3D kamerom kao jedinim senzorom

5.2.2 Rezultati merenja odziva rešenja

Rezultati merenja odziva u ispitnom Slučaju A dati su na Slici 110. Dat je primer dva frejma u toku merenja u alatu *VirtualDub*, jedan na početku merenja (Slika 111) (tek izgovorena reč, drugi indikator verovatnoće *Conv.* gledano sa desne strane ekrana postavljen na nizak nivo) i na kraju merenja (drugi indikator verovatnoće *Conv.* gledano sa desne strane ekrana prvi put na visokom nivou) (Slika 112).

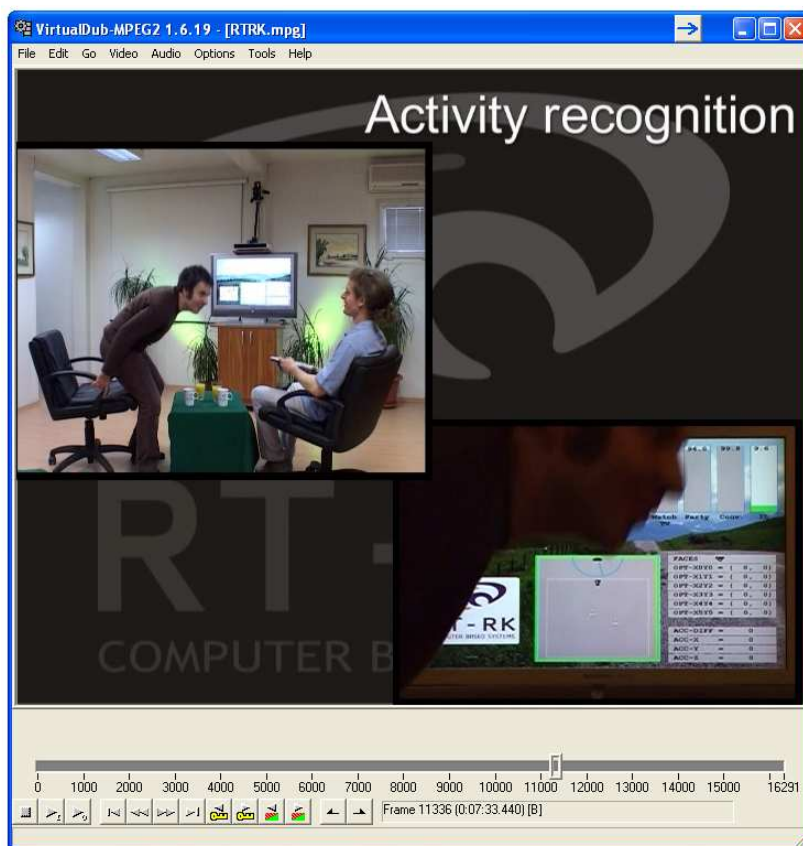


Slika 110. Vreme odziva rešenja za ispitni Slučaj A: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama)



Slika 111. Jedan primer frejma za početak merenja odziva za ispitni Slučaj A

Ekperimentalni rezultati i diskusije



Slika 112. Jedan primer frejma za završetak merenja odziva za ispitni Slučaj A

U Tabeli 20 dati su statistički pokazatelji merenja u ispitnom Slučaju A, dok je na Slici 113 data raspodela dobijenih rezultata.

<i>Srednja vrednost</i>	<i>Standardna devijacija</i>	<i>Varijansa</i>	<i>Medijana</i>
(μ)	(σ)	(σ^2)	(m)
1007	217.77	47422.1	1100

Tabela 20. Statistički pokazatelji merenja za ispitni Slučaj A

Dalje, treba proveriti, da li je uslov da je vreme odziva manje od 2 sekunde za ovaj slučaj ispunjen. Koristićemo statistički z-test za testiranje nulte hipoteze H_0 („Vreme odziva je veće ili jednako 2 sekunde“) odnosno alternativne hipoteze H_1 („Vreme odziva je manje od dve sekunde“) sa pragom značajnosti $\alpha=0.05$. Kada hipoteze izrazimo u milisekundama dobijamo:

$$H_0 : \mu \geq 2000$$

$$H_1 : \mu < 2000$$

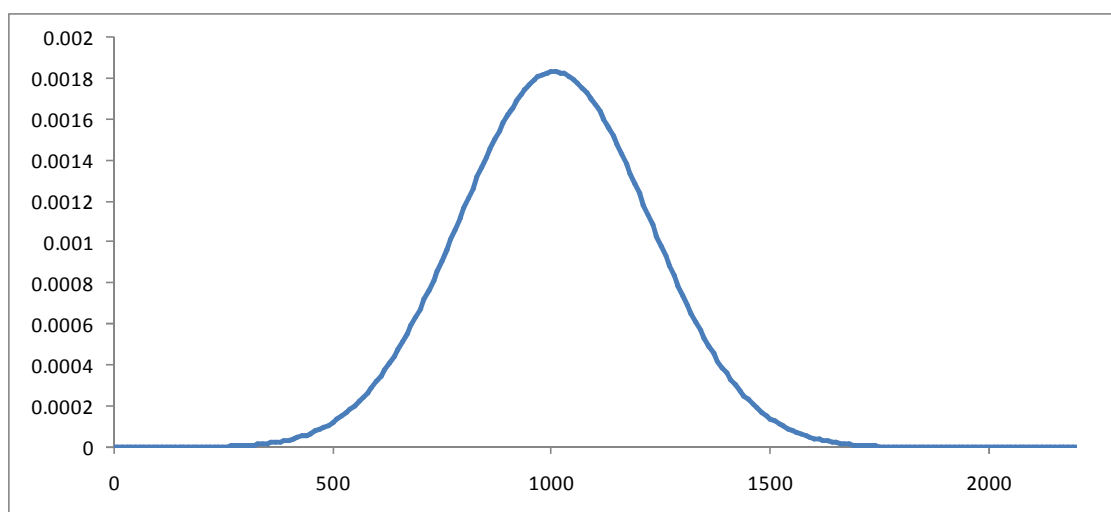
Računamo test statistiku:

$$Z = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{1007 - 2000}{218} \sqrt{20} = -20.37$$

Vrednost Z za koju se hipoteza odbacuje, za zadati prag značajnosti je:

$$R : Z < -1.96$$

S obzirom da je $-20.37 \ll -1.96$, nulta hipoteza se odbacuje u korist alternativne hipoteze, odnosno očekivano vreme odziva je zaista manje od dve sekunde.



Slika 113. Očekivana raspodela odziva rešenja u ispitnom slučaju A

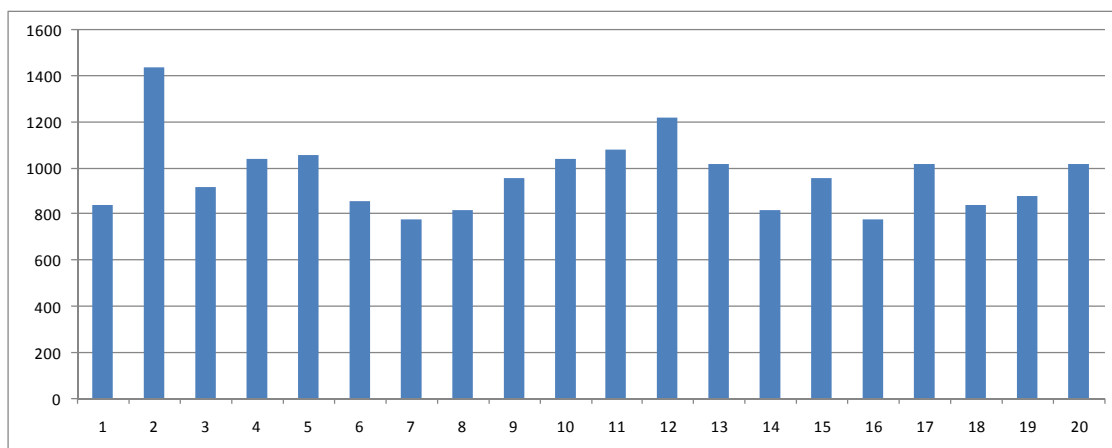
Rezultati merenja odziva za ispitni Slučaj B dati su na Slici 114. Dat je primer dva frejma u toku merenja u alatu *VirtualDub*, jedan na početku merenja (Slika 115) (uređaj postavljen u odgovarajući položaj, indikator verovatnoće *Stop* postavljen na nizak nivo) i na kraju merenja (indikator verovatnoće *Stop* prvi put na visokom nivou) (Slika 116).

U Tabeli 21 dati su statistički pokazatelji merenja u ispitnom Slučaju B, dok je na Slici 117 data raspodela dobijenih rezultata.

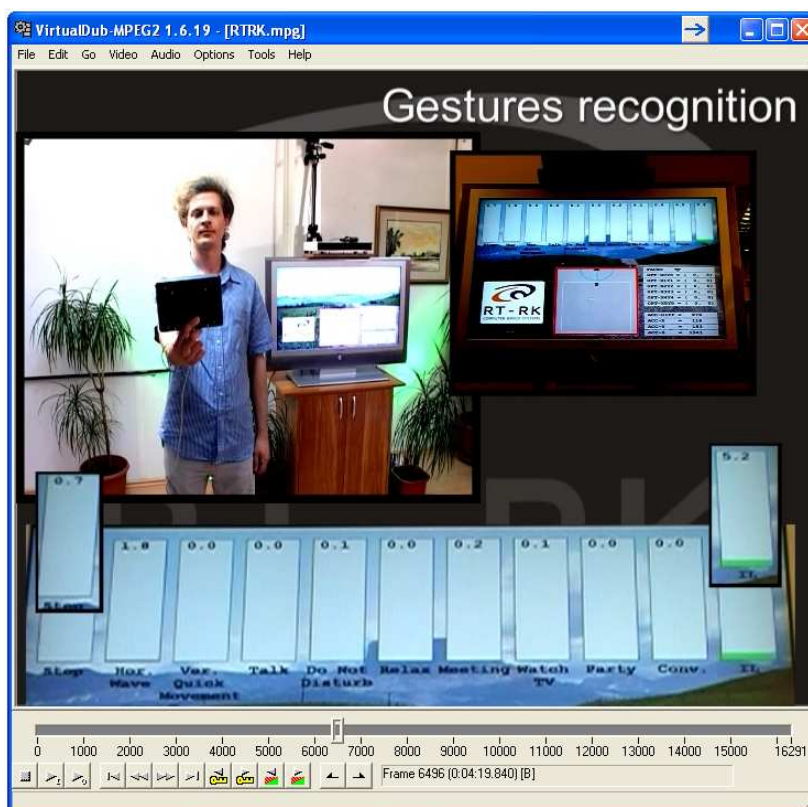
<i>Srednja vrednost</i> (μ)	<i>Standardna devijacija</i> (σ)	<i>Varijansa</i> (σ^2)	<i>Medijana</i> (m)
970	161.38	26042.1	960

Tabela 21. Statistički pokazatelji merenja za ispitni Slučaj B

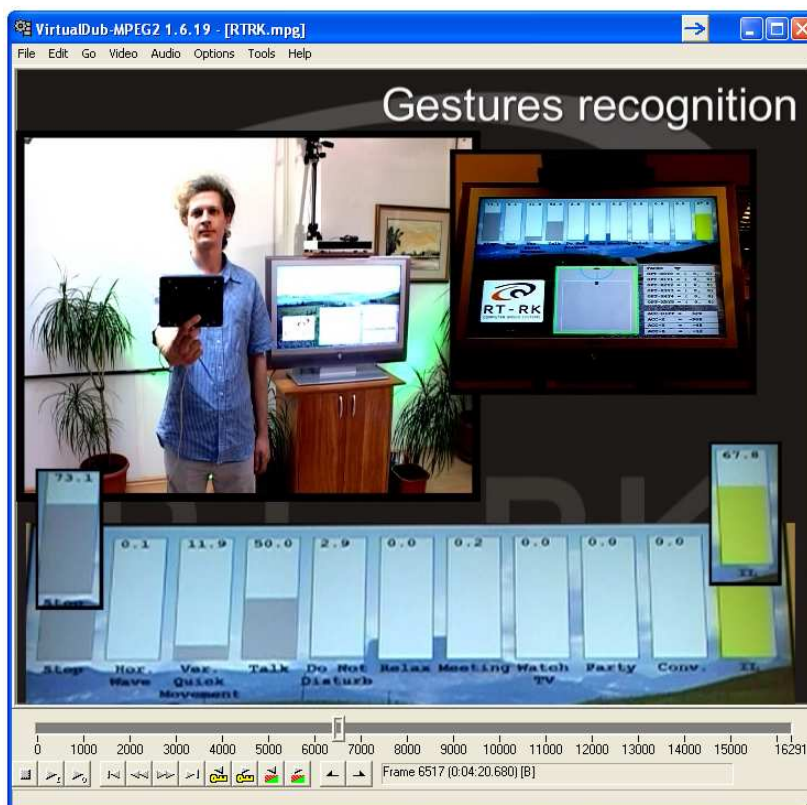
Ekperimentalni rezultati i diskusije



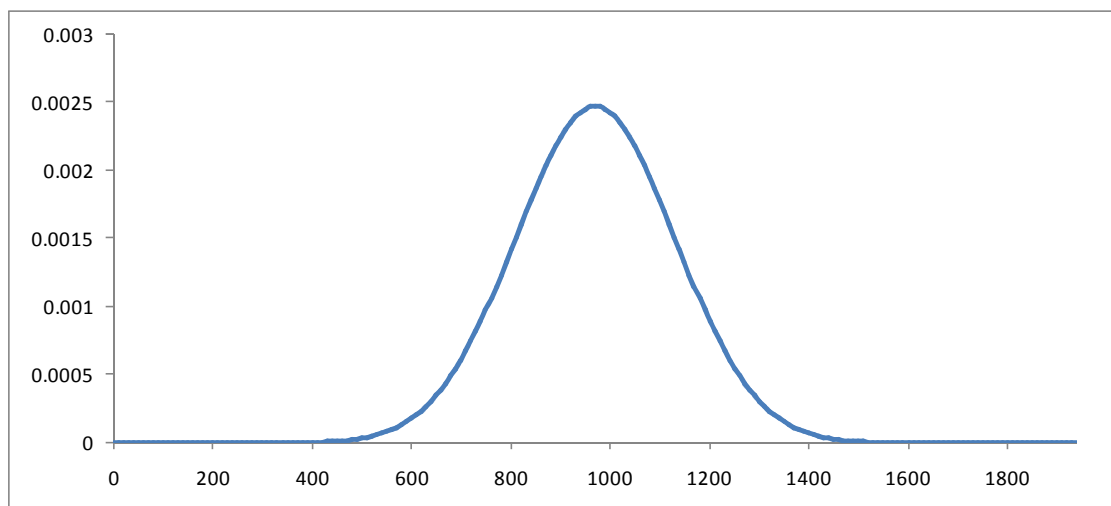
Slika 114. Vreme odziva rešenja za ispitni Slučaj B: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama)



Slika 115. Jedan primer frejma za početak merenja odziva za ispitni Slučaj B



Slika 116. Jedan primer frejma za završetak merenja odziva za ispitni Slučaj B



Slika 117. Očekivana raspodela odziva rešenja u ispitnom slučaju B

Dalje, treba proveriti, da li je uslov da je vreme odziva manje od 0.5 sekundi za ovaj slučaj ispunjen. Koristićemo statistički z-test za testiranje nulte hipoteze H_0 (“Vreme odziva je manje od 0.5 sekundi“) odnosno alternativne hipoteze H_1 („Vreme odziva je veće ili jednako 0.5 sekundi“) sa pragom značajnosti $\alpha=0.05$. Kada hipoteze izrazimo u milisekundama dobijamo:

Ekperimentalni rezultati i diskusije

$$H_0 : \mu < 500$$

$$H_1 : \mu \geq 500$$

Računamo test statistiku:

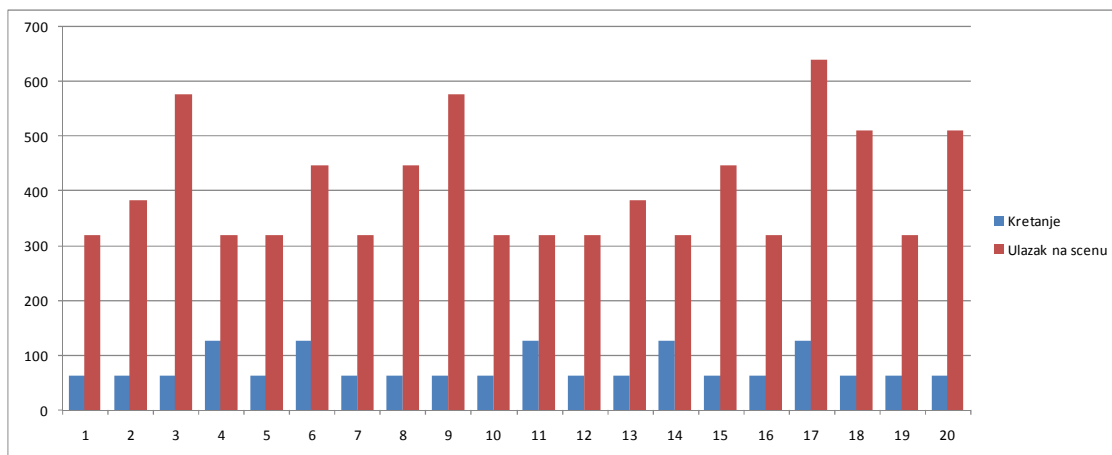
$$Z = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{970 - 500}{161} \sqrt{20} = 13.055$$

Vrednost Z za koju se hipoteza odbacuje, za zadati prag značajnosti je:

$$R : Z > 1.96$$

S obzirom da je $13.055 > 1.96$, nulta hipoteza se odbacuje u korist alternativne hipoteze, odnosno očekivano vreme odziva ne zadovoljava postavljeni kriterijum.

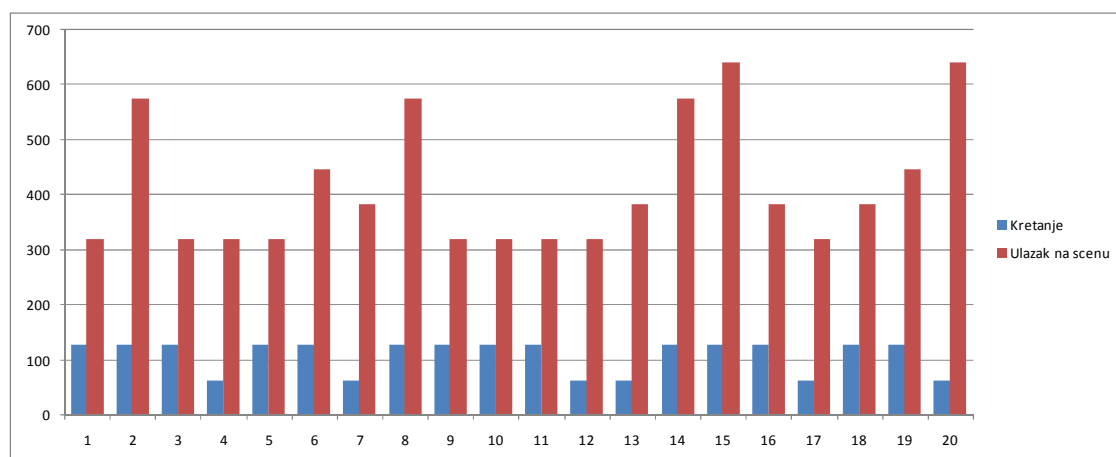
Rezultati merenja odziva za ispitni Slučaj C, sa jednim učesnikom na sceni, dati su na Slici 118. Rezultati merenja odziva za ispitni Slučaj C, sa dva učesnika na sceni, dati su na Slici 119. U toku merenja uočene su bitne razlike u vremenu odziva pri ulasku osobe u scenu po prvi put, u odnosu na vreme odziva kada se osoba pomera unutar scene. Iz tog razloga, na Slikama 118 i 119 su prikazana vremena odziva za oba slučaja. Takođe se mora primetiti da, usled rezolucije snimka od 15 fps, donji prag preciznosti merenja je ~64ms, što bi trebalo da bude dovoljno za procenu zadovoljenja postavljenih kriterijuma.



Slika 118. Vreme odziva rešenja za ispitni Slučaj C sa jednim učesnikom na sceni:
na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama), za slučaj kretanja unutar scene (plavo) i za slučaj ulaska na scenu (crveno)

Da bi se dobio sveobuhvatan pogled na odziv za ovaj slučaj, statički je obrađena situaciju u kojoj se zajedno posmatraju i ulasci na scenu, i kretanje unutar scene, i to na

način da je odnos zastupljenosti uzoraka iz svake od grupa u korelaciji sa prosečnim vremenom provedenim u ulasku/izlasku iz prostorije, u odnosu na vreme boravka u prostoriji (uzet je okvirni odnos 1:10, u odnosu na sprovedeni eksperiment). Podjednako su zastupljeni uzorci kada u eksperimentu učestvuju dva učesnika, u odnosu na uzorke kada u eksperimentu učestvuje jedan učesnik.



Slika 119. Vreme odziva rešenja za ispitni Slučaj C sa dva učesnika na sceni: na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama), za slučaj kretanja unutar scene (plavo) i za slučaj ulaska na scenu (crveno)

U Tabeli 22 dati su statistički pokazatelji merenja u ispitnom Slučaju C.

	<i>Srednja vrednost (μ)</i>	<i>Standardna devijacija (σ)</i>	<i>Varijansa (σ^2)</i>	<i>Medijana (m)</i>
<i>Jedan učesnik, kretanje</i>	80	28.43	808.4	64
<i>Jedan učesnik, ulazak</i>	406.4	106.44	11328.7	352
<i>Dva učesnika, kretanje</i>	108.8	30.1	905.4	128
<i>Dva učesnika, ulazak</i>	416	118.37	14012.6	384
<i>Kumulativno za Slučaj C</i>	123.2	126.1	9429.1	128

Tabela 22. Statistički pokazatelji merenja za ispitni Slučaj C

Ekperimentalni rezultati i diskusije

Na Slici 120 prikazano je šest uzastopnih frejmova iz video sekvence. U prvom frejmu korisnik prvi put zaklanja zonu poslednje lampe (najbliže prozoru) sa više od 50% tela. U sledećim frejmovima korisnik sve više zaklanja zonu. Primetna je promena nivoa osvetljenosti lampe u svakom frejmu, što govori o maksimalnoj brzini odziva za ovaj slučaj.



Slika 120. Šest uzastopnih frejmova iz ispitnog Slučaja C: Na slici se vidi zavisnost osvetljenosti lampe od ulaska korisnika u zonu lampe u svakom frejmu

Sada treba proveriti, da li je uslov da je vreme odziva manje od 0.2 sekundi za ovaj slučaj ispunjeno. Koristićemo statistički z-test za testiranje nulte hipoteze H_0

(“Vreme odziva je veće ili jednako 0.2 sekunde“) odnosno alternativne hipoteze H_1 („Vreme odziva je manje od 0.2 sekunde“) sa pragom značajnosti $\alpha=0.05$, za svaki od posebnih grupa u ovom ispitnom slučaju. Kada hipoteze izrazimo u milisekundama dobijamo:

$$H_0 : \mu \geq 200$$

$$H_1 : \mu < 200$$

Računamo test statistiku za slučaj kretanja jednog učesnika unutar scene (Z_1):

$$Z_1 = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{80 - 200}{28} \sqrt{20} = -19.17$$

Računamo test statistiku za slučaj ulaska učesnika u praznu scenu (Z_2):

$$Z_2 = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{406 - 200}{106} \sqrt{20} = 8.69$$

Računamo test statistiku za slučaj kretanja učesnika unutar scene u kojoj se nalazi još jedan učesnik (Z_3):

$$Z_3 = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{109 - 200}{30} \sqrt{20} = -13.6$$

Računamo test statistiku za slučaj ulaska učesnika u scenu u kojoj se nalazi još jedan učesnik (Z_4):

$$Z_4 = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{416 - 200}{118} \sqrt{20} = 8.2$$

Računamo test statistiku za kumulativni slučaj (Z_5):

$$Z_5 = \frac{\bar{x} - \mu_0}{\sigma} \sqrt{n} = \frac{123 - 200}{126} \sqrt{20} = -2.73$$

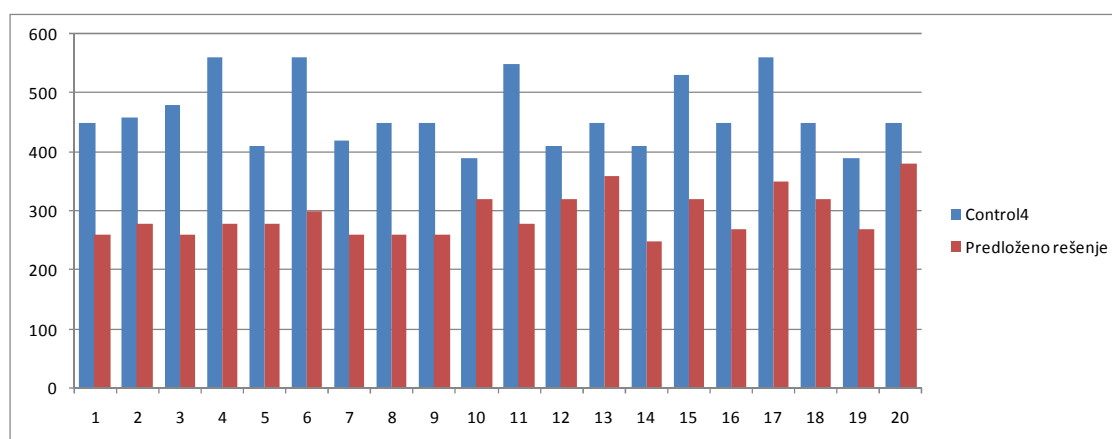
U slučajevima sa statistikama Z_1 , Z_3 i Z_5 , u kojima je srednja vrednost niža od testirane vrednosti, nulta hipoteza se odbacuje u korist alternativne hipoteze, odnosno, vreme odziva je niže od zahtevane granice od 200ms. U slučajevima sa statistikama Z_2 i Z_4 , u kojima je srednja vrednost viša od testirane vrednosti, nulta hipoteza je zadovoljena, odnosno, vreme odziva ne zadovoljava traženi okvir od 200ms.

5.2.3 Poređenje vremena odziva sa postojećim rešenjima

Obavljeno je poređenje vremena odziva predloženog rešenja sa sistemom Control4. Za potrebe poređenja, Control4 Zigbee dimer integrisan je u predloženo rešenje, odnosno dodati su potrebni rukovaoci kako bi bilo moguće kontrolisati svetlo

Ekspperimentalni rezultati i diskusije

korišćenjem *Stratus* namenskog računara i programske podrške kontekstualne platforme. Realizovani rukovalac realizovan je u skladu sa *Control4* specifikacijom, dok su tačne poruke koje se upućuju dimeru replicirane na osnovu prisluškivanja originalnog protokola korišćenjem alata *Wireshark*. Na ovaj način obezbeđeno je poređenje: (1) vremena odziva na naredbu za uključivanje svetla, upućenoj korisničkom akcijom u sistemskoj korisničkoj sprezi kontekstualne platforme, pokrenutoj sa posebnog računara; (2) vremena odziva na naredbu za uključivanje svetla, upućenoj korisničkom akcijom u *Control4 Navigator* aplikaciji na *Control4 7*“ ekranu osetljivom na dodir, povezanom na *Control4 HC300* kontroler. Obavljeno je 20 uzastopnih merenja, i rezultati su dati na Slici 121. Statistički pokazatelji merenja dati su u Tabeli 23.



Slika 121. Poređenje vremena odziva na naredbu za uključivanje svetla predloženog rešenja (crveno) i sistema *Control4* (plavo): na X osi dat je redni broj merenja, dok je na Y osi dato vreme odziva (u milisekundama)

	<i>Srednja vrednost (μ)</i>	<i>Standardna devijacija (σ)</i>	<i>Varijansa (σ^2)</i>	<i>Medijana (m)</i>
<i>Control4</i>	464	57.44	3299	450
<i>Predloženo rešenje</i>	294	37.9	1435.8	280

Tabela 23. Statistički pokazatelji poređenja vremena odziva

5.2.4 Diskusija i zaključci

Na osnovu eksperimentalnih rezultata, u dva od tri ispitna slučaja zadovoljeno je traženo vreme odziva. U ispitnom Slučaju C, vreme odziva je zadovoljavajuće u 3 od 5 podslučajeva, koji uključuju kumulativni slučaj, koji je relevantan sa stanovišta potrošača. Nakon analize strukture utroška vremena pri odzivu, zaključeno je da je veće vreme odziva od dozvoljenog u ispitnom Slučaju C, u slučajevima ulaska učesnika na scenu, rezultat izvršavanja algoritma za obradu slike, koji u ovom slučaju segmentira novi objekat u slici i računa njegovo težište, te objavljuje događaj o tome ostatku sistema koji takođe zahteva određeno vreme za propagaciju. U slučaju kada se učesnici pomeraju unutar scene, vreme odziva je znatno bolje jer se izračunava samo vektor pomeraja nad makroblokom oko tačke prethodnog težišta objekta učesnika u slici. Takođe, nema objavljivanja događaja o prisustvu, te se samo obavlja kontinuirana prozivka nove pozicije od strane kontekstualne aplikacije.

Ispitni Slučaj B nije zadovoljio zahtevano vreme odziva od 0.5 sekundi. Ova situacija se može bolje razumeti ako se uzmu u obzir blokovi uključeni u dedukciju tražene veličine, te propagacija te veličine i njen prikaz u prezentacionoj jedinici. Mehanizam objavljivanja otkrivenog gesta korisnika jednak je semantizaciji i računanju verovatnoće stanja okruženja, odnosno, u postupku otkrivanja gesta učestvuje modul za semantizaciju. Dakle, događaji se agregiraju sa rukovalaca akcelerometrom, preko modula za semantizaciju, odnosno obrade kontekstualnih polisa, do kontekstualnog skripta koji upravlja prezentacionom jedinicom. Upravo u tom dugačkom propagacionom nizu treba tražiti razlog dužeg vremena odziva, koje je, sada razumljivo, uporedivo sa vremenom iz ispitnog Slučaja A. Ukoliko se uzme u obzir izuzetno dobra performansa ispitnog Slučaja C može se izvesti zaključak da je, u slučaju potrebe za izuzetno brзом reakcijom sistema (značajno brže od 1 sekunde), potrebno što veći deo logike rada realizovati u okviru kontekstualnog skripta, umesto korišćenja generalizovanih kontekstualnih polisa. Za većinu realnih aplikacija upravljanje eksplicitnim unosom korisnika direktno iz kontekstualnog skripta nije neobičan zahtev, s obzirom da eksplicitnom unosu ponajmanje doprinose algoritmi semantizacije koje pruža kontekstualna platforma.

Poređenjem vremena odziva sa sistemom *Control4*, pokazano je da predloženo rešenje nudi bolje vreme odziva. S obzirom na to da je ciljni kontrolisani uređaj u

Ekperimentalni rezultati i diskusije

okviru poređenja identičan (*Control4* dimer), vreme odziva koje je dobijeno eksperimentom može se pripisati isključivo osobinama fizičke arhitekture i programske podrške kontekstualne platforme, odnosno kontrolera, i realizovanim mrežnim protokolima između korisničke sprege i platforme. Iz tog razloga može se očekivati bolje vreme odziva i za moguće druge situacije poređenja sa sistemom *Control4*.

5.3 MERENJE TAČNOSTI SEMANTIZACIJE

U ovom poglavlju dati su rezultati merenja tačnosti semantizacije u tipičnom kućnom inteligentnom okruženju. Najpre je data metodologija merenja i opisan scenario za merenje tačnosti semantizacije, zatim su dati rezultati merenja, poređenje tačnosti sa trenutno vodećim istraživanjima i, konačno, diskusija dobijenih rezultata.

5.3.1 Metodologija merenja tačnosti semantizacije

Merenje tačnosti semantizacije je izuzetno kompleksan zadatak, s obzirom na to da realna situacija korišćenja uključuje korisnike i okruženja čije su osobine i mogućnosti najblaže rečeno različite i nepredvidive. Uprkos tome, cilj merenja tačnosti semantizacije je da izoluje slučaj tipičan za jedno inteligentno okruženje i moguću primenu u uređajima potrošačke elektronike, izvede ogled u tom slučaju i uporedi rezultate ogleda sa postojećim, slično izvedenim ogledima u vodećim istraživanjima u oblasti. S obzirom da je naglasak disertacije na modelovanju arhitekture kontekstualne platforme u smislu uređaja potrošačke elektronike, verifikacija i validacija tačnosti algoritama za semantizaciju proverena je u ograničenom obimu. Sveobuhvatno ispitivanje tačnosti algoritama moguće je sprovesti u budućim istraživanjima, dok se eventualno fino podešavanje algoritama, pa čak i njihova zamena novim, ne odražava bitno na predloženu arhitekturu pokrivenu temom ove disertacije.

Ekperimentalno inteligentno okruženje je simulirana dnevna soba. Cilj eksperimenta u ovakvom okruženju je da se potvrdi pouzdanost algoritama za semantizaciju u kontekstu koji je svojstven korisnicima uređaja potrošačke elektronike. Da bi se obavila sveobuhvatna analiza, postavljeni cilj je bio da se proverí što veći deo predloženih algoritama za semantizaciju. Iz tog razloga u evaluaciju su uključeni izlazi iz nivoa agregacije, odnosno procena zauzetosti korisnika. Vrednost zauzetosti korisnika je upravo veličina o kojoj se izvode zaključci u okviru ovog eksperimenta.

Pouzdanost rešenja merena je poređenjem izračunatog stepena zauzetosti korisnika, sa samoprocenom koju su priložili učesnici u sceni, kao i sa procenom datom od strane pet posmatrača. Postavljeno pitanje učesnicima i posmatračima je bilo: „*Kako biste procenili sopstvenu zauzetost (zauzetost posmatranih učesnika) na Likert skali 1-10?*“

S obzirom na cilj rešenja da se ispravno izvršava na različitim uređajima potrošačke elektronike, eksperiment je izveden u tri različite postavke. Postavka 1 pokriva opšti slučaj, u kojem se koriste svi senzori dostupni UAK kontekstualnoj platformi (akcelerometar u mobilnom telefonu, optička kamera, 3D kamera i mikrofonski niz). Postavka 2 pokriva slučaj u kojem su senzori ugrađeni u konzolni uređaj potrošačke elektronike (nalik set-top boksu ili konzoli za video igre), i obuhvata optičku kameru i mikrofonski niz. Postavka 3 pokriva slučaj u kojem se platforma izvršava na mobilnom uređaju, te se kao senzori koriste mikrofonski niz i akcelerometar. Razlike u postavkama su napravljene u smislu procene pouzdanosti pri radu sa različitim dostupnim sensorima, odnosno, nije evaluirana potencijalna razlika u fizičkim arhitekturama uređaja koji mogu da izvršavaju programsku podršku UAK platforme. Iz tog razloga sve tri postavke su evaluirane u slučaju kada se UAK programska podrška izvršava na samostalnom kontroleru. Eksperiment je dodatno pojednostavljen tako što je kontroler programiran da pokrene tri odvojene instance UAK programske podrške (po jednu za svaku postavku), čime je omogućen istovremen rad i izbegnute su razlike u korišćenju scene za svaku od postavki.

Rešenje je instalirano u prostoriju za pauzu i odmor u okviru istraživačkog instituta, koje je preuređeno da simulira okruženje dnevne sobe (Slika 122). Pre obavljanja eksperimenta, prostorija je povremeno posećivana za pripremanje i konzumiranje kafe, gledanje vesti ili kratke razgovore. Posetioci prostorije informisani su o toku eksperimenta, i ohrabreni da koriste prostoriju i sve nove sadržaje koji su u nju postavljeni. Eksperiment je obavljen u dva eksperimentalna dana: prvi dan za treniranje algoritama, a drugi dan za evaluaciju. Prijavljivanje neželjenog asinhronog zahteva obavljeno je sa dva za to namenjena mobilna telefona, na kojima je pokrenuta aplikacija sa mogućnošću definisanja cene asinhronog zahteva, na skali od 0 do 9. Celokupna sesija snimljena je kamerom i potom reprodukovana u formi instrukcijskog snimka i učesnicima u sceni, i posmatračima, zajedno sa grafikonom prijave neželjenih asinhronih zahteva. U toku drugog eksperimentalnog dana zaustavljen je režim

Ekperimentalni rezultati i diskusije

treniranja i obavljena je evaluacija rešenja, uz nastavak snimanja prostorije. Video snimak je naknadno reprodukovao učesnicima i posmatračima, sa zahtevom da unesu subjektivnu procenu nivoa zauzetosti učesnika u prostoriji, kad god procene da je došlo do promene nivoa zauzetosti. Poređenjem dobijenih rezultata sa vrednostima snimljenim u log datoteku od strane odgovarajuće kontekstualne aplikacije, obavlja se procena pouzdanosti rešenja.

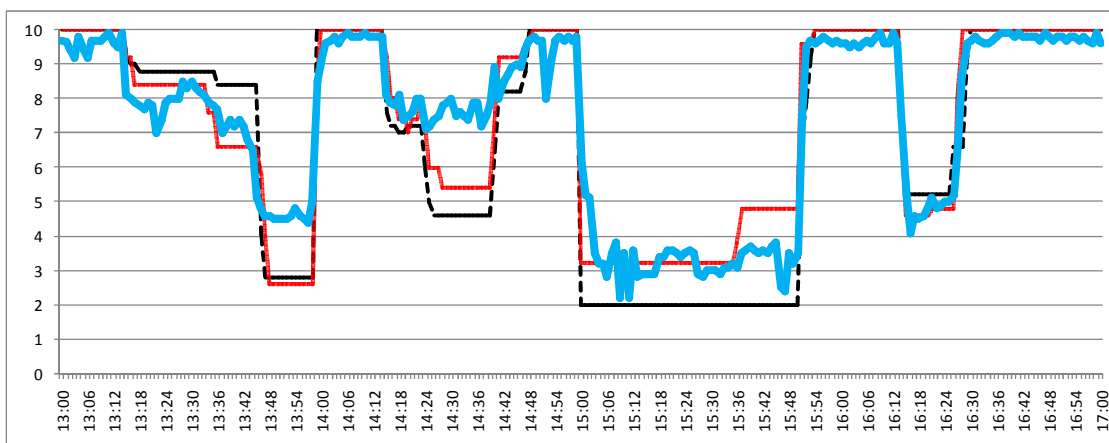


Slika 122. Simulirano okruženje dnevne sobe

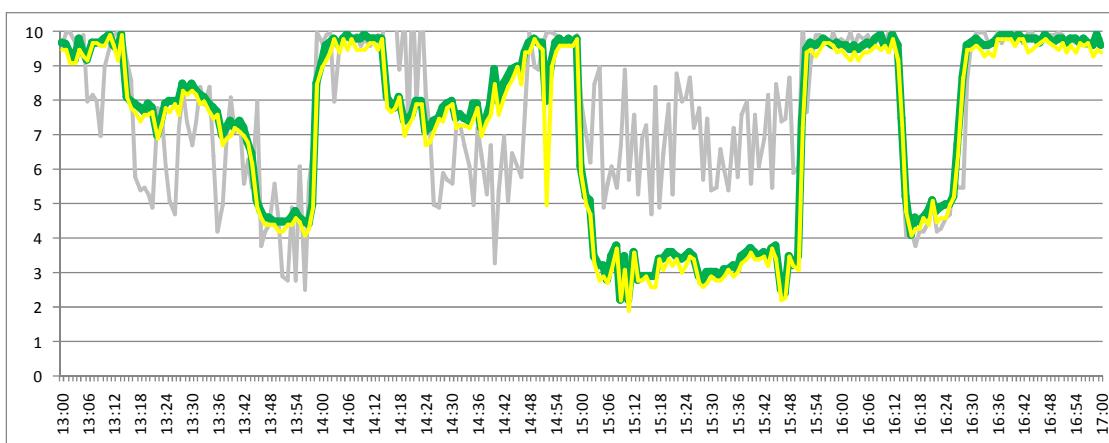
5.3.2 Rezultati merenja tačnosti semantizacije

Na Slici 123 prikazan je izvod iz rezultata eksperimenta u toku jednog eksperimentalnog dana, sa prikazanim nivoima zauzetosti korisnika (izračunatim od strane platforme, prijavljenim samoprocenom učesnika i dobijenim od strane posmatrača). Na Slici 124 prikazana je razlika u proceni zauzetosti korisnika od strane platforme, u zavisnosti od korišćene postavke senzora.

Za dalje izračunavanje rezultata eksperimenta dobijene vrednosti nivoa zauzetosti od strane posmatrača i vrednosti dobijene samoprocenom učesnika su uprosečene, s obzirom na visok stepen korelacije odgovora (sa standardnom devijacijom oko 20%). Poređenjem sa rezultatima izračunatim od strane platforme, dobijene su vrednosti srednje apsolutne greške (*Mean Absolute Error – MAE*) i stepen uspešnosti prepoznavanja nivoa zauzetosti, prikazani u Tabeli 24.



Slika 123. Dobijeni nivoi zauzetosti u toku poslepodneva eksperimentalnog dana u simuliranom okruženju dnevne sobe (10 – minimalna zauzetost, 0 – maksimalna zauzetost). Crvena linija označava prosečan nivo zauzetosti dobijen od strane posmatrača. Isprekidana crna linija označava prosečan nivo zauzetosti dobijen samoprocenom učesnika, dok plava linija označava vrednosti izračunate od strane platforme.



Slika 124. Nivoi zauzetosti izračunati od strane platforme u toku poslepodneva eksperimentalnog dana u simuliranom okruženju dnevne sobe (10 – minimalna zauzetost, 0 – maksimalna zauzetost). Zelena linija označava rezultat zasnovan na Postavci 1. Žuta linija označava rezultat zasnovan na Postavci 2. Siva linija označava rezultat zasnovan na Postavci 3.

<i>Postavka</i>	<i>Srednja apsolutna greška</i>	<i>Uspešnost prepoznavanja</i>
1	0.66 (6.6%)	94%
2	0.71 (7.1%)	93%
3	1.66 (16.6%)	83%

Tabela 24. Srednja apsolutna greška i uspešnost prepoznavanja stepena zauzetosti

Ekperimentalni rezultati i diskusije

Da bi se videlo koliko dobro sistem određuje zauzetost u zavisnosti od određenog stanja okruženja, potrebno je kreirati matricu konfuzije. Za dalje razmatranje podrazumevaće se Postavka 1. U Tabeli 25 dat je spisak stanja okruženja, identifikovanih od strane posmatrača, sa intervalima u kojima su prijavljena.

	Opis stanja okruženja	Interval
S1	Prazna prostorija	$[13^{00}, 13^{15}][14^{00}, 14^{15}]$ $[14^{48}, 14^{52}][14^{54}, 15^{01}]$ $[15^{54}, 16^{14}][16^{29}, 17^{00}]$
S2	Razgovor za stolom uz kafu, TV isključen	$[13^{15}, 13^{46}]$
S3	Ozbiljan razgovor (sa dužim monolozima) nalik sastanku	$[13^{46}, 14^{00}][15^{00}, 15^{37}]$
S4	Neobavezan razgovor uz gledanje TV programa	$[14^{15}, 14^{24}]$
S5	Neobavezan razgovor uz gledanje emisije koja je unapred planirana	$[14^{24}, 14^{40}]$
S6	Razgovor u blizini izlaza iz prostorije (u odlasku)	$[14^{40}, 14^{48}][15^{51}, 15^{54}]$
S7	Brza poseta prostoriji od strane neupućene osobe (npr. čistačica, domar)	$[14^{52}, 14^{54}]$
S8	Razgovor dok je TV prijemnik uključen (ali učesnici razgovora ne gledaju TV program)	$[15^{37}, 15^{51}]$
S9	Učesnici gledaju planiranu TV emisiju u tišini	$[16^{14}, 16^{29}]$

Tabela 25. Spisak stanja okruženja u posmatranom periodu identifikovanih od strane posmatrača

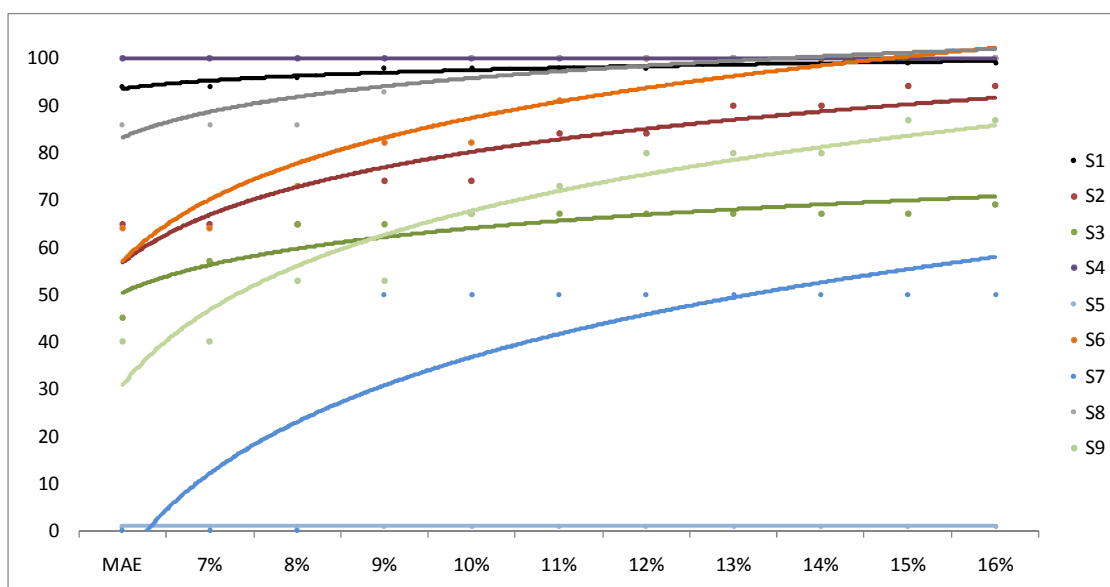
Procena nivoa zauzetosti, data od strane kontekstualne platforme, smatra se *tačnom* ukoliko je dobijeni rezultat u intervalu $[\mu-\varepsilon, \mu+\varepsilon]$, pri čemu je μ srednja vrednost nivoa zauzetosti priloženog od strane posmatrača i samoprocenom, a ε tolerancija u nivou srednje apsolutne greške. U matrici konfuzije dat je stepen slaganja procena od strane platforme, sa procenama od strane posmatrača i učesnika, za svako od stanja okruženja (Tabela 26). Iz matrice se može videti u kojim stanjima je procena najčešće bila ispravna, a u kojima to nije slučaj. Takođe, može se videti aproksimativna raspodela netačnih procena po ostalim stanjima, svaki put kada je izračunati nivom zauzetosti sličan nivou zauzetosti u nekom drugom stanju, u skladu sa zapisima posmatrača i učesnika.

		Procene nivoa zauzetosti od strane platforme										
		S1	S2	S3	S4	S5	S6	S7	S8	S9		
		Prazna prostorija										
		Razgovor za stolom uz kafu, TV isključen										
		Ozbiljan razgovor (sa dužim monolozima) nalik sastanku										
		Neobavezan razgovor uz gledanje TV programa										
		Neobavezan razgovor uz gledanje emisije koja je unapred planirana										
		Razgovor u blizini izlaza iz prostorije (u odlasku)										
		Brza poseta prostoriji od strane neupućene osobe (npr. čistačica, domar)										
		Razgovor dok je TV prijemnik uključen (ali učesnici razgovora ne gledaju TV program)										
		Učesnici gledaju planiranu TV emisiju u tišini										
Nivo zauzetosti dat samoprocenom i izveštajem posmatrača	S1	Prazna prostorija	94					6				
	S2	Razgovor za stolom uz kafu, TV isključen		65		23	1	11				29
	S3	Ozbiljan razgovor (sa dužim monolozima) nalik sastanku		1	45		11	1		37	5	
	S4	Neobavezan razgovor uz gledanje TV programa				100						
	S5	Neobavezan razgovor uz gledanje emisije koja je unapred planirana		26		69	0	4				1
	S6	Razgovor u blizini izlaza iz prostorije (u odlasku)	3	18		6		64	9			
	S7	Brza poseta prostoriji od strane neupućene osobe (npr. čistačica, domar)		25		25		50	0			
	S8	Razgovor dok je TV prijemnik uključen (ali učesnici razgovora ne gledaju TV program)			14					86		
	S9	Učesnici gledaju planiranu TV emisiju u tišini		4		5	46	5			40	

Tabela 26. Matrica konfuzije procene nivoa zauzetosti u zavisnosti od stanja okruženja datog samoprocenom i izveštajem posmatrača

Ekperimentalni rezultati i diskusije

Tolerancija ε direktno određuje broj diskretnih stanja zauzetosti koja se pouzdano mogu odrediti. Što je ε veće, veća je i tačnost procene nivoa zauzetosti za svako od stanja okruženja. Zavisnost vrednosti tolerancije ε i tačnosti procene stepena zauzetosti, kada se uzmu u obzir stanja okruženja, izračunato na osnovu rezultata eksperimenta, data je na Slici 125. Zavisnost vrednosti ε , broja diskretnih stanja zauzetosti koja se mogu odrediti i uspešnosti prepoznavanja stanja zauzetosti, izračunata na osnovu rezultata eksperimenta, data je u Tabeli 27.



Slika 125. Zavisnost vrednosti nivoa tolerancije ε (x osa) i uspešnosti prepoznavanja nivoa zauzetosti korisnika (y osa) za svako od stanja okruženja

<i>Broj diskretnih stanja zauzetosti</i>	<i>Tolerancija ε</i>	<i>Uspešnost prepoznavanja</i>
10	~6%	~61%
7	~7%	~63%
6	~8%	~67%
5	10%	~74%
4	~12%	~79%
3	~17%	~82%
2	25%	~94%

Tabela 27. Zavisnost tolerancije ε i uspešnosti prepoznavanja stanja zauzetosti, od zahtevanog broja diskretnih stanja zauzetosti

5.3.3 Diskusija dobijenih rezultata

Na osnovu ukupne srednje apsolutne greške i stepena uspešnosti prepoznavanja nivoa zauzetosti, datih u Tabeli 24, može se zaključiti kvalitetan i pouzdan rad platforme. U odnosu na korišćene postavke senzora, može se uočiti da odsustvo bitnije razlike u dobijenim rezultatima za Postavku 1 i Postavku 2 svedoči o skromnom doprinosu 3D kamere problemu određivanja nivoa zauzetosti korisnika, dok je mikrofonski niz veoma značajan senzor zahvaljujući kome sve tri postavke omogućuju solidan nivo uspešnosti prepoznavanja nivoa zauzetosti.

Međutim, neophodno je analizirati uspešnost procene nivoa zauzetosti za svako od stanja okruženja, kako bi se procenili granični slučajevi. Na primer, u stanju okruženja S3 („Ozbiljan razgovor sa dužim monolozima nalik sastanku”) uočen je nizak nivo uspešnih procena (45%), uz činjenicu da je platforma često dodeljivala nivo zauzetosti koji odgovara stanju S8 (“Razgovor dok je TV prijemnik uključen, ali učesnici razgovora ne gledaju TV program”), i to u 37% slučajeva. S obzirom na kratko trajanje treninga (1 dan), postoji mogućnost da su u toku trening dana izveštaji o stepenu zauzetosti upućivani većinom dok je TV prijemnik bio uključen, dok je u toku eksperimentalnog dana bilo više dijaloga sa isključenim TV prijemnikom. Takođe, izuzetno loše određivanje nivoa zauzetosti zapaženo je za stanja okruženja S5 (“Neobavezan razgovor uz gledanje emisije koja je unapred planirana”) i S9 (“Učesnici gledaju planiranu TV emisiju u tišini”), s obzirom na to da platforma nije bila u mogućnosti da sazna sadržaj programa i da zaključi da li je gledanje sadržaja planirano ili sporadično. Ovo se može izbeći korišćenjem mehanizama integracije kontekstualne platforme sa drugim sistemima, kada bi dodatni događaji bili obezbeđeni od strane TV prijemnika posredstvom kojih bi se platforma obaveštavala o sadržaju programskog vodiča (*Electronic Program Guide – EPG*). Takođe, moguće je poboljšati uspešnost određivanja nivoa zauzetosti za stanja S5 i S9 produžavanjem vremena odvojenog za trening, kada bi se uspostavila bolja korelacija nivoa zauzetosti korisnika, planiranog gledanja TV programa i npr. zvuka ili pokreta lica. Stanje S7 (“Brza poseta prostoriji od strane neupućene osobe”) je bilo komplikovano za procenu iz razloga što su posmatrači i učesnici, prilikom popunjavanja izveštaja, bili potpuno svesni da osoba koja je u prostoriji nema veze sa eksperimentom, pa su dodeljivali niske nivoe zauzetosti za ovo stanje, kao da je u pitanju prazna prostorija.

Eksperimentalni rezultati i diskusije

Ukoliko se dozvoli veća tolerancija ϵ , uspešnost procene nivoa zauzetosti, gledano po stanjima okruženja (što je najrelevantnija metrika), raste. Broj potrebnih nivoa zauzetosti u praksi je relativno nizak, i često korelira sa brojem režima potrebnih za rad uređaja ili aplikacije. Najčešće je dovoljno 3 do 5 nivoa zauzetosti, odnosno 3 do 5 ekvivalentnih režima rada, na primer *Aktivan*, *Utišan* i *Isključen* za većinu mobilnih telefona, ili *Aktivan*, *Odsutan*, *Zauzet*, *Nevidljiv*, *Isključen* za komunikacione klijent aplikacije. S obzirom na visoku tačnost za 5 diskretnih nivoa (74%) i 3 diskretna nivoa (82%), korišćenje kontekstualne platforme za ove probleme je moguće i eksperiment potvrđuje ispravan rad.

5.3.4 Poređenje rezultata semantizacije sa postojećim rešenjima

Dobijeni rezultati tačnosti procene nivoa zauzetosti od strane predloženog rešenja kontekstualne platforme mogu se uporediti sa prethodnim vodećim istraživanjima u oblasti. U odnosu na jedno od sveobuhvatnijih istraživanja (Fogarty, 2005) predloženo rešenje pokazuje sledeći napredak:

- Procena nivoa zauzetosti, u predloženom rešenju, obavljena je u okruženju domaćinstva (dnevna soba), što je kontekst koji nije obuhvaćen prethodnim istraživanjima. Primena kontekstualnih platformi za procenu zauzetosti korisnika u svakodnevnim okruženjima, poput domaćinstva (uz moguću integraciju sa uređajima za zabavu i komunikaciju) može doprineti prihvatanju uređaja sa takvim funkcijama od strane korisnika;
- Uspešnost prepoznavanja zasnovana na stanjima okruženja (prikazana u Tabeli 27), kada se uporedi sa prethodnim istraživanjima (Fogarty, 2005), pokazuje napredak (74% u odnosu na 48% za pet diskretnih stanja zauzetosti; 94% u odnosu na 82% za dva diskretna stanja zauzetosti);
- Skalabilnost platforme (mogućnost rada sa različitim postavkama senzora), arhitektura platforme i mogućnost interakcije sa okruženjem, doprinosi mogućnosti integracije u različite ciljne uređaje potrošačke elektronike, nasuprot zasnovanosti na PC računaru i prilagođenosti kancelarijskom okruženju u drugim istraživanjima (Fogarty, 2007).

5.4 UPOREDNI PRIKAZ REŠENJA PO POTENCIJALU ZA RAZVOJ KONTEKSTUALNIH APLIKACIJA

U ovom poglavlju dati su rezultati evaluacije predloženog rešenja, kao i ostalih srodnih rešenja u oblasti, po ključnim kriterijumima kontekstualnih platformi sa stanovišta potencijala njihovog korišćenja za razvoj kontekstualnih aplikacija (sa naglaskom na oblast potrošačke elektronike). Kriterijumi evaluacije podeljeni su u sledeće grupe: (1) zahtevi kontekstualne platforme po Dey; (2) opšti zahtevi kontekstualne platforme; (3) integrabilnost i primenljivost za razvoj kontekstualnih aplikacija potrošačke elektronike; (4) podrška za senzore; (5) podrška za aktuatore i scenarije, i (6) cena.

Zahtevi kontekstualne platforme po Dey su važni sa aspekta opštih akademskih kriterijuma koji su postavljeni pred sistem, a koji nisu definisani u okviru ove disertacije, te predstavljaju nepristrasnu osnovu evaluacije i ulazni kriterijum za upotrebu kontekstualne platforme za razvoj aplikacija (ovi zahtevi izneti su u Poglavlju 3.1.1). Opšti zahtevi kontekstualne platforme odnose se na prirodu kontekstualne platforme i osnovne metrike koje određuju nivo njene upotrebljivosti, u smislu analize sprovedene u poglavljima 1.1 i 3.1. Integrabilnost i primenljivost za razvoj kontekstualnih aplikacija potrošačke elektronike je kategorija koja obuhvata minimalni skup osnovnih zahteva navedenih u poglavljima 1.3 i 3.1.2. Podrška za aktuatore, senzore i scenarije su važni da se proceni primenljivost u realnim aplikacijama, koje zahtevaju konkretne mehanizme prikupljanja parametara konteksta, kao i objavljivanje zaključaka upravljanjem različitim aktuatorima. Konačno, cena je presudan faktor u odlučivanju s obzirom na oblast istraživanja (potrošačka elektronika), jer isuviše velika margina cene može da onemogući rešenje da bude dostupno što široj grupi potrošača.

U uporedni prikaz, zajedno sa predloženim rešenjem, uključena su reprezentativna rešenja u oblasti, uključujući potrošačke sisteme dostupne u prodaji, kao i neke akademske platforme u okviru ranijih istraživanja. Reprezentativni sistemi uključeni u evaluaciju, obuhvataju potrošačke sisteme navedene u poglavlju 1.4: Control4, Adhoco, AMX, Crestron, Oblo i PRS. Od akademskih rešenja realizovanih u ranijim istraživanjima, obuhvaćeni su Subtle (Fogarty, 2007), SOCAM (Gu, 2005), CORTEX (Biegel, 2004), i Gaia (Roman, 2002; Gaia Project, 2005).

5.4.1 Metodologija evaluacije potencijala za razvoj kontekstualnih aplikacija

Svaki od kriterijuma evaluacije podeljen je u podkriterijume, i svaki podkriterijum ocenjivan je korišćenjem Likert skale 1-5. Ocene nisu dodeljivane subjektivno, već su mapirane na stepen ispunjenosti zahteva na sledeći način:

- ocena 5 znači da je traženi zahtev u potpunosti ispunjen u okviru evaluiranog rešenja;
- ocena 4 znači da se traženi zahtev može u potpunosti ispuniti, međutim, rešenje se mora koristiti na neadekvatan, neoptimalan ili neočekivan način;
- ocena 3 znači da se traženi zahtev može ispuniti samo dodatnim razvojem u okviru ili zasnovano na evaluiranom rešenju, pri čemu je potrebno uložiti napor od najviše šest inženjer-meseci;
- ocena 2 znači da se traženi zahtev može ispuniti samo značajnim dodatnim razvojem u okviru ili zasnovano na evaluiranom rešenju, pri čemu je potrebno uložiti napor koji premašuje šest inženjer-meseci;
- ocena 1 znači da ispunjenje traženog zahteva u okviru evaluiranog rešenja nije moguće, jer postoje ograničenja u arhitekturi.

Daljom disagregacijom ocena definiše se tzv. *make vs buy* kriterijum, koji označava mogućnost evaluiranog rešenja da bude upotrebljeno u svrhu realizacije kontekstualnih aplikacija za uređaje potrošačke elektronike bez potrebe za izmenama, adaptacijama ili dodatnom implementacijom (ocene 4 i 5), odnosno uz potrebu za implementacijom dodatne logike u okviru aplikacije (ocene 2 i 3). Rešenje koje zasluži ocenu 1 u nekom od kriterijuma, nije odgovarajuće za realizaciju najšireg spektra kontekstualnih aplikacija i može se koristiti samo u ograničenoj meri.

U okviru kriterijuma evaluacije po Dey, definisani su sledeći podkriterijumi: (1) razdvojenost fizičkog nivoa (senzora) od nivoa semantizacije; (2) mogućnost interpretacije konteksta na višem nivou semantike; (3) mogućnost transparentne i distribuirane komunikacije sa sensorima; (4) trajno prikupljanje parametara konteksta, nezavisno od pokretanja aplikacija; (5) mogućnost skladištenja kontekstualnih parametara i uvid u istorijat i (6) mogućnost pretraživanja uređaja i servisa.

U okviru podkriterijuma razdvojenosti fizičkog nivoa (senzora) od nivoa semantizacije, rešenju je dodeljena ocena 5 ukoliko poseduje jasan API na osnovu

kojeg se rešenje može proširiti novim senzorskim rukovaocem, odnosno, ukoliko to proširenje ne zahteva potrebu za dodatnim programiranjem u okviru koda rešenja (idealno, treba da bude moguće dodavanje *plugin*-a sa rukovaocem). Takođe, ocena 5 označava jasnu razdvojenost između fizičkog i srednjeg nivoa, odnosno, mogućnost korišćenja senzora i pristup informacijama sa senzora iz kontekstualne aplikacije bez potrebe za poznavanjem fizičkih detalja implementacije senzora i protokola veze senzora i kontekstualne platforme. Ocena 4 može biti dodeljena zbog jednog od sledećih razloga: (1) proširenje novim senzorskim rukovaocem je jednostavno i moguće, ali postoje ograničenja u smislu mogućih protokola veze i sl, što znači da neće biti moguće proširiti platformu potpuno proizvoljnim sensorom; (2) potrebno je dodatno programiranje u okviru aplikacije kojim se definišu osobine senzora, njihove mogućnosti i sl. a što se inače moglo zaključiti na osnovu prirode senzora ili (3) kontekstualna aplikacija u umerenoj meri mora poznavati tip senzora da bi uspešno pristupila podacima koje on objavljuje. Ocena 3 je dodeljena u slučaju da ne postoji jasan API koji omogućava proširenje rešenja novim rukovaocima, ali sama arhitektura takvo proširenje omogućava uz dodatno programiranje; takođe, ocena 3 se dodeljuje ukoliko se proširenje može obaviti samo uz manju intervenciju u kodu rešenja (npr. „ručno“ dodavanje pojedinih definicija, poziva i sl). Ocena 2 se dodeljuje ukoliko ne postoji potpuna razdvojenost između fizičkog i srednjeg sloja (npr. srednji sloj nije u dovoljnoj meri razvijen i mora se većinski realizovati u okviru aplikacije), odnosno, ukoliko postoje velika ograničenja u pogledu proširivosti. Ocena 1 se dodeljuje ukoliko rešenje ne omogućava razgraničenje fizičkog i srednjeg nivoa, odnosno, ukoliko rešenje ne pruža pomoć aplikaciji pri pristupu senzorskim informacijama, aplikacija mora da poznaje detalje senzora ili protokola, ili ukoliko je rešenje izuzetno ograničeno u smislu fizičkih resursa da ne može da omogući bilo kakvu dodatnu realizaciju koja bi doprinela ispunjenju traženog zahteva.

U okviru podkriterijuma mogućnosti interpretacije konteksta na višem nivou semantike, ocena 5 se dodeljuje u slučaju da srednji nivo omogućava pristup kontekstualnim parametrima sa više različitih nivoa agregacije, i obezbeđuje algoritme za agregaciju po različitim aspektima, kao što su kombinovanje više senzora i podizanje nivoa apstrakcije, pri čemu aplikacija nema potrebu za realizacijom posebnih algoritama u većini slučajeva. Ocena 4 se dodeljuje ukoliko se pojedini

Ekperimentalni rezultati i diskusije

algoritmi ili kombinovanje zaključaka moraju implementirati u okviru aplikacije, ili ukoliko aplikacija nema mogućnost pristupa pojedinim nivoima agregacije (algoritmima) i njihovoj konfiguraciji. Ocena 3 se dodeljuje ukoliko rešenje ne poseduje neke važne algoritme, poput osnovne obrade slike (segmentacija, detekcija pokreta i sl) i zvuka (minimalno detekcija tona nasuprot šumu), ili ukoliko ne omogućava grupisanje senzora i pojednostavljeno izvođenje zajedničkih zaključaka. Ocena 2 se dodeljuje ukoliko, osim pristupa „sirovim“ podacima sa senzora, svi algoritmi ili velika većina algoritama za semantizaciju moraju biti razvijeni u okviru aplikacije. Ocena 1 se dodeljuje ukoliko postoje ozbiljna ograničenja na fizičkom nivou zbog kojih nije moguć čak ni prijem svih kontekstualnih parametara sa potrebnih senzora.

U okviru podkriterijuma mogućnosti transparentne i distribuirane komunikacije sa sensorima, ocena 5 se dodeljuje ukoliko rešenje omogućava izmeštanje senzora/aktuatora na proizvoljne lokacije, njihovo povezivanje sa centralnom platformom, i apstrakciju njihove lokacije i protokola tako da aplikacija nema potrebu za pravljenjem razlike između lokalnih senzora/aktuatora (priključenih na sam uređaj) i senzora/aktuatora koji su udaljeni (npr. udaljeni uređaji na lokalnoj mreži, servisi na internetu). Ocena 4 se dodeljuje ukoliko se pojedini detalji vezani za lokaciju senzora moraju obrađivati u okviru aplikacije, odnosno, ukoliko je apstrakcija lokacije moguća samo nakon posebnih podešavanja sistema i korišćenja odvojenih alata. Ocena 3 se dodeljuje ukoliko rešenje poseduje mogućnost apstrakcije lokacije uređaja zahvaljujući realizovnim protokolima i evidenciji uređaja, ali ne postoji jasan API prema aplikaciji što uslovljava potrebu za dodatnim programiranjem. Ocena 2 se dodeljuje ukoliko se apstrakcija lokacije u potpunosti mora programirati u okviru aplikacije, odnosno, ukoliko rešenje ne omogućava sakrivanje detalja protokola veze udaljenih uređaja. Ocena 1 se dodeljuje ukoliko rešenje ne omogućava korišćenje eksternih sprega, što čini nemogućim dodavanje uređaja koji su prostorno udaljeni (npr. na lokalnoj mreži).

U okviru podkriterijuma trajnosti prikupljanja parametara konteksta, ocena 5 se dodeljuje ukoliko rešenje obavlja akviziciju kontekstualnih parametara kontinualno i nezavisno od stanja rada aplikacije, i ukoliko obezbeđuje pristup ranije primljenim kontekstualnim parametrima, odnosno njihovim prosečnim vrednostima. Ocena 4 se dodeljuje ukoliko pristup ranije primljenim kontekstualnim parametrima nije

omogućen, odnosno ukoliko se oni ne čuvaju u okviru rešenja, već se moraju čuvati u okviru aplikacije. Ocena 3 se dodeljuje ukoliko postoje ograničenja u smislu trajnosti rada rešenja, npr. ukoliko se pri svakom pokretanju aplikacije rešenje mora inicijalizovati ili aktivirati, čime se potencijalno narušava mogućnost korišćenja više aplikacija istovremeno; ova činjenica zahteva programiranje kontrolne aplikacije koja bi obavljala upravljanje tokom izvršavanja rešenja i njegov monitoring. Ocena 2 se dodeljuje ukoliko je kontekst izvršavanja rešenja isključivo vezan za kontekst izvršavanja aplikacije, što zahteva značajne izmene koje bi dovele do ispunjenja zahteva. Ocena 1 se dodeljuje u slučaju da postoje ozbiljna ograničenja u trajnosti pristupa kontekstualnim parametrima, npr. nedovoljno čest pristup (gubljenje određenih parametara), prozivka/pretraga se mora vršiti iz aplikacije i sl.

U okviru podkriterijuma mogućnosti skladištenja kontekstualnih parametara i uvida u istorijat, ocena 5 se dodeljuje u slučaju da se kontekstualni parametri trajno skladište (u zavisnosti od „starosti“ parametra, obavlja se agregacija vrednosti u cilju uštede memorije – npr. čuvaju se samo zaključci, prosečna vrednost ili trend). Ocena 4 se dodeljuje u slučaju da postoje ograničenja u pogledu količine parametara koji se mogu sačuvati, što uslovljava da za parametre koji su stariji od određene vremenske tačke aplikacija mora da obavlja evidenciju. Ocena 3 se dodeljuje u slučaju da se postupak skladištenja mora programirati u okviru aplikacije, ali za to postoji predviđen API koji olakšava taj postupak. Ocena 2 se dodeljuje ukoliko se skladištenje ne obavlja, API ne postoji i taj posao u potpunosti mora biti prepušten aplikaciji. Ocena 1 se dodeljuje ukoliko je trajno skladištenje fizički nemoguće zbog nepostojanja trajne memorije.

U okviru podkriterijuma mogućnosti pretraživanja uređaja i servisa, ocena 5 se dodeljuje ukoliko postoje automatski mehanizmi pretrage zahvaljujući kojima se uređaji pronalaze (dodaju) i uklanjaju bez potrebe za intervencijama iz aplikacije. Ocena 4 se dodeljuje ukoliko se postupak pretrage mora inicirati složenim postupkom iz programskog koda aplikacije, odnosno, ukoliko se za pretragu koriste spoljni alati. Ocena 3 se dodeljuje ukoliko je pretraga omogućena u ograničenom obimu (npr. nije omogućena za tipove uređaja čiji protokoli veze to omogućavaju), pa se pretraga mora dodatno realizovati u okviru aplikacije. Ocena 2 se dodeljuje ukoliko pretraga nije podržana, ali je moguće realizovati u okviru aplikacije zahvaljujući mogućnosti

Ekperimentalni rezultati i diskusije

pristupa spregama na fizičkom nivou i protokolima veze. Ocena 1 se dodeljuje ukoliko rešenje ne podržava uređaje sa protokolima veze na kojima se može sprovesti pretraga, odnosno, ukoliko aplikacija nema pristup fizičkim spregama posredstvom kojih može samostalno da obavi pretragu.

U okviru opštih zahteva kontekstualne platforme, definisani su sledeći podkriterijumi: (1) transparentnost rešenja, odnosno tehnička vidljivost od strane korisnika; (2) dovoljna brzina odziva (u smislu diskusije u poglavlju 5.2.1); (3) mogućnost automatskog prilagođavanja učenjem i mogućnost podešavanja od strane korisnika; (4) mogućnost neposrednog upravljanja rešenjem od strane korisnika.

U okviru podkriterijuma transparentnosti, ocena 5 se dodeljuje ukoliko rešenje omogućava fizičko sakrivanje senzora/aktuatora od strane korisnika, na način da izgled okruženja korisnika bude nepromenjen, odnosno, ukoliko je interakcija korisnika sa rešenjem prirodna i uz minimalno korišćenje tehničkih pristupnih tačaka (npr. aplikacija, panela i sl), već se obavlja korišćenjem postojećih medijuma (npr. mobilni telefon, daljinski upravljač, postojeći prekidači, kontekstualni parametri). Ocena 4 se dodeljuje ukoliko je transparentnost donekle narušena postojanjem pristupnih aplikacija / panela i sl, ali je konfiguracija korisničkog okruženja minimalno izmenjena. Ocena 3 se dodeljuje ukoliko se transparentnost može obezbediti samo uz dodatni razvoj pojedinih komponenti i korisničkih sprega. Ocena 2 se dodeljuje ukoliko samo rešenje ne obezbeđuje transparentnost, već je u tu svrhu potrebno realizovati sve potrebne gradivne blokove. Ocena 1 se dodeljuje u slučaju da je arhitektura sistema takva da nije moguće obezbediti transparentnost u dovoljnoj meri, čak i uz dodatnu realizaciju (npr. rešenje zahteva eksplicitan unos parametara u toku rada korišćenjem PC aplikacije).

U okviru podkriterijuma dovoljne brzine odziva, koriste se metrike navedene u poglavlju 5.2.1. U tom smislu, ocena 5 se dodeljuje ukoliko rešenje zadovoljava sva tri kriterijuma odziva: (1) odziv na promenu stanja ispod 2 sekunde; (2) odziv na aktivaciju ispod 0.5 sekundi i (3) odziv za realno vreme ispod 0.2 sekunde. Ocena 4 se dodeljuje ukoliko se tražena vremena odziva mogu postići ukoliko se mali deo obrade prenese u aplikaciju, odnosno, ukoliko se rešenje koristi na specifičan način. Ocena 3 se dodeljuje ukoliko nije moguće postići traženi odziv za realno vreme, bez dodatnih napora u realizaciji aplikacije (npr. direktno prozivanje senzora i realizacija

optimizovane logike za vremenski kritične slučajeve). Ocena 2 se dodeljuje ukoliko nije moguće postići traženi odziv niti za realno vreme, niti za aktivaciju, bez dodatnih napora u realizaciji aplikacije, poput direktnog prozivanja senzora i optimizovane obrade. Ocena 1 se dodeljuje ukoliko je arhitektura i tehnologija rešenja takva, da ni u kom slučaju aplikacija ne može da nadomesti kašnjenje, te nije moguće ispuniti jedan ili više kriterijuma odziva.

U okviru podkriterijuma mogućnosti automatskog prilagođavanja učenjem i mogućnosti podešavanja od strane korisnika, ocena 5 se dodeljuje ukoliko postoji kontinualan, automatizovan trening platforme (kontinualno učenje) na osnovu kojeg platforma podešava svoj rad prema krajnjem korisniku, odnosno, ukoliko krajnji korisnik ima mogućnost da dodatno, po želji, prilagodi rad platforme bez obzira na parametre definisane u postupku učenja. Ocena 4 se dodeljuje ukoliko faze učenja i prilagođavanja postoje, ali su delimično navođene (npr. korisnik ili administrator deklariraju početak i završetak faza, ili obavljaju manju medijaciju parametara). Ocena 3 se dodeljuje ukoliko postupak učenja nije omogućen u okviru rešenja, ali rešenje poseduje API za podešavanje rada platforme koji omogućava da se postupak učenja realizuje odvojeno, u okviru aplikacije. Ocena 2 se dodeljuje ukoliko postupak učenja i podešavanja moraju da se programiraju u okviru aplikacije, i u tu svrhu ne postoje jednostavni mehanizmi obezbeđeni od strane rešenja. Ocena 1 se dodeljuje ukoliko nije moguće obezbediti učenje i podešavanje, jer rešenje ne omogućava bilo kakvo definisanje parametara rada.

U okviru podkriterijuma mogućnosti neposrednog upravljanja rešenjem od strane korisnika, ocena 5 se dodeljuje ukoliko rešenje omogućava, bez obzira na automatski rad, akviziciju parametara konteksta ili fazu učenja, eksplicitno izdavanje komande određenim uređajima, odnosno izmenu stanja rada platforme. Ocena 4 se dodeljuje ukoliko izdavanje takve komande nije moguće bez prethodnog podešavanja, odnosno ukoliko se komanda izdaje na složen način. Ocena 3 se dodeljuje ukoliko je neposredno upravljanje omogućeno od strane rešenja posredstvom API-ja, ali je u tu svrhu realizovan ograničen broj korisničkih sprega. Ocena 2 se dodeljuje ukoliko korisničke sprege za upravljanje nisu realizovane, ali se mogu realizovati i upravljanje se može obaviti korišćenjem API-ja rešenja. Ocena 1 se dodeljuje ukoliko rešenje ne

Ekperimentalni rezultati i diskusije

poseduje način povezivanja sa korisničkom spregom, odnosno ne poseduje mogućnost za programsko izdavanje komande uređaju ili komande za izmenu režima rada.

U smislu integrabilnosti i primenljivosti za razvoj kontekstualnih aplikacija potrošačke elektronike, definisani su sledeći podkriterijumi: (1) niska potrošnja sistemskih resursa (memorije i procesora); (2) niska potrošnja energije; (3) posedovanje potrebnih licenci i sertifikata; (4) mogućnost portovanja / nivo apstraktnosti; (5) mogućnost integracije sa korisničkom spregom uređaja potrošačke elektronike; (6) mogućnost detekcije stanja korisnika i podešavanja profila; (7) mogućnost povezivanja sa drugim uređajima potrošačke elektronike u okruženju.

U okviru podkriterijuma niske potrošnje sistemskih resursa, ocena 5 se dodeljuje ukoliko je iskorišćenje procesora i memorije takvo, da preostaje preko 80% resursa za razvoj aplikacija, što je definicija minimalnog *overhead*-a dodatnih programskih komponenti (Nejati, 2012). Ocena 4 se dodeljuje ukoliko je iskorišćenje resursa takvo, da postoje uspešno realizovani primeri kontekstualnih aplikacija na istoj platformi koji se bez problema izvršavaju u dužem vremenskom periodu. Ocena 3 se dodeljuje ukoliko rešenje omogućava realizaciju kontekstualne aplikacije na istim sistemskim resursima, ali s obzirom na količinu funkcionalnosti koju rešenje pruža potrebna je pažljiva realizacija nedostajućih elemenata u okviru kontekstualne aplikacije da se ne bi ugrozio integritet aplikacije usled nedostatka resursa. Ocena 2 se dodeljuje po istom osnovu kao i prethodna ocena, ukoliko je stepen rizika veliki jer se većina elemenata realizuje u okviru aplikacije. Ocena 1 se dodeljuje ukoliko rešenje ne poseduje niti jednu realizovanu kontekstualnu aplikaciju, odnosno, bilo koja realizovana kontekstualna aplikacija ne radi ispravno / pravovremeno usled nedostatka sistemskih resursa.

U okviru podkriterijuma niske potrošnje energije, ocena 5 se dodeljuje ukoliko integracija rešenja u uređaj potrošačke elektronike, ne izaziva značajno povećanje potrošnje uređaja (sa marginom od ~10%). Ocena 4 se dodeljuje ukoliko rešenje obezbeđuje potrošnju na nivou prosečnih uređaja potrošačke elektronike sa baterijskim napajanjem. Ocena 3 se dodeljuje ukoliko se adekvatno niska potrošnja može obezbediti samo pažljivim odabiranjem funkcionalnosti koje su uključene/aktivirane u okviru rešenja. Ocena 2 se dodeljuje ukoliko se adekvatno niska potrošnja može obezbediti samo isključivanjem pojedinih blokova funkcionalnosti koje se onda moraju

zameniti posebnom realizacijom u okviru aplikacija. Ocena 1 se dodeljuje ukoliko rešenje značajno utiče na povećanje potrošnje do mere da je to korisniku očigledno, i ako rešenje ne omogućava uključivanje/isključivanje blokova na osnovu čega bi moglo doći do smanjenja potrošnje.

U okviru podkriterijuma posedovanja potrebnih licenci i sertifikata, ocena 5 se dodeljuje ukoliko uređaj, nakon integracije rešenja, nema potrebu za ponovnom evaluacijom sertifikata za niskom potrošnjom energije, bezbednosti, magnetnog zračenja, i sl. Ocena 4 se dodeljuje ukoliko pojedini sertifikati nisu obezbeđeni (kao, npr, za bezbednost pri korišćenju aktuatora). Ocena 3 se dodeljuje ukoliko je neophodna provera programske podrške i fizičkih karakteristika uređaja u smislu ostvarivanja sertifikacionih normi, nakon što se rešenje integriše. Ocena 2 se dodeljuje ukoliko rešenje ne poseduje potrebne sertifikate i ukoliko bi bio potreban značajan dodatni rad za obezbeđivanje svih sertifikacionih zahteva. Ocena 1 se dodeljuje ukoliko rešenje u osnovi svoje arhitekture poseduje značajne prepreke sticanju pojedinih sertifikata, što bi onemogućilo njegovu komercijalnu primenu.

U okviru podkriterijuma mogućnosti portovanja / nivoa apstraktnosti, ocena 5 se dodeljuje ukoliko je moguća primena rešenja (prevođenje) na različite ciljne arhitekture bez izmena, odnosno uz minimalne izmene. Ocena 4 se dodeljuje ukoliko je prevođenje za različite arhitekture moguće, ali uz određene manje izmene. Ocena 3 se dodeljuje ukoliko prenošenje rešenja na ciljnu arhitekturu zahteva ponovnu realizaciju pojedinih blokova. Ocena 2 se dodeljuje ukoliko prenošenje rešenja na ciljnu arhitekturu zahteva ponovnu implementaciju većeg dela programskog koda. Ocena 1 se dodeljuje ukoliko prenošenje nije moguće, ili je izuzetno komplikovano, iz razloga što u svojoj osnovi rešenje koristi usko specifične primitive određene ciljne arhitekture / određenog prevodioca.

U okviru podkriterijuma mogućnosti integracije sa korisničkom spregom uređaja potrošačke elektronike, ocena 5 se dodeljuje ukoliko je korisnička sprega rešenja direktno primenljiva u okviru grafičke sprege ciljne platforme, uz minimalna podešavanja (u smislu funkcionalnosti, rezolucije prikaza, brzine odziva i sl). Ocena 4 se dodeljuje ukoliko grafička sprega može, uz manje izmene, da se primeni na ciljnoj platformi (npr. zasnovana je na otvorenim standardima poput Java, HTML5 i sl). Ocena 3 se dodeljuje ukoliko rešenje ne nudi grafičku spregu koja se direktno može

Ekperimentalni rezultati i diskusije

integrisati, ali omogućava jasne programske sprege za integraciju korisničke sprege ciljnog uređaja sa platformom (na nivou korisničkih akcija). Ocena 2 se dodeljuje ukoliko rešenje ne poseduje određene programske sprege za integraciju korisničke sprege ciljnog uređaja sa platformom, već je potrebna realizacija prilagodnog sloja i grafičkog sloja. Ocena 1 se dodeljuje ukoliko rešenje poseduje sopstvene, usko specifične grafičke sprege i ukoliko ih nije moguće isključiti/prilagoditi ili na neki način povezati grafičku spregu ciljnog uređaja sa rešenjem bez značajnih izmena kompletnog rešenja.

U okviru podkriterijuma mogućnosti detekcije stanja korisnika i podešavanja profila, ocena 5 se dodeljuje ukoliko rešenje poseduje mehanizme i sprege za automatsku detekciju stanja korisnika (kao što je npr. nivo zauzetosti) i objavljivanje tog stanja u cilju izmene režima rada uređaja. Ocena 4 se dodeljuje ukoliko su mehanizmi prisutni, ali nisu uopšteni do nivoa proizvoljne upotrebe nego se koriste u tačno određenim scenarijima. Ocena 3 se dodeljuje ukoliko se algoritmi za detekciju stanja korisnika moraju razvijati u okviru aplikacije, dok se ocena 2 dodeljuje ukoliko je napor potreban za takav razvoj na osnovu dostupnog API-ja rešenja značajan. Ocena 1 se dodeljuje ukoliko rešenje ne poseduje dovoljno senzorskih ulaza da bi se omogućila minimalna preciznost detekcije stanja korisnika, čak i u slučaju da se ta detekcija potpuno realizuje u okviru aplikacije.

U okviru podkriterijuma mogućnosti povezivanja sa drugim uređajima potrošačke elektronike u okruženju, ocena 5 se dodeljuje ukoliko postoje sprege za razmenu informacija između rešenja i uređaja u okruženju, na način da je predviđena brza integracija više različitih klasa uređaja. Ocena 4 se dodeljuje ukoliko sprege za integraciju postoje, ali su jedinstvene i svaki uređaj koji se povezuje mora da implementira klijentsku stranu komunikacionog protokola. Ocena 3 se dodeljuje ukoliko je potrebna realizacija dodatnih programskih blokova da bi se omogućila komunikacija sa drugim uređajima, odnosno ocena 2 ukoliko ta realizacija zahteva značajan napor. Ocena 1 se dodeljuje ukoliko integracija sa drugim uređajima nije predviđena, odnosno rešenje ne poseduje način da razmenjuje informacije sa spoljnim entitetima.

Podrška za senzore se evaluira kroz tri podkriterijuma: (1) podrška za optičke senzore (kamere) i pripadajuće osnovne algoritme obrade slike; (2) podrška za senzore

zvuka (mikrofone) i pripadajuće osnovne algoritme obrade zvuka i (3) podrška za ostale senzore (akcelerometar, toplotni senzor, senzor blizine i sl). U ovim podkriterijumima, ocena 5 se dodeljuje ukoliko su kamere, mikrofoni ili akcelerometar, u najmanjem skupu, podržani i ukoliko postoje osnovni algoritmi detekcije zvuka, pozicije govornika, detekcije pokreta, kretanja i prebrojavanja objekata, kao i detekcije osnovnih gestova. Ocena 4 se dodeljuje ukoliko pojedini algoritmi imaju ograničenja, ili postoje pojedine podgrupe senzora koje nisu podržane. Ocena 3 se dodeljuje ukoliko nije podržan akcelerometarski senzor, odnosno ukoliko nedostaje neki od važnih osnovnih algoritama. Ocena 2 se dodeljuje ukoliko nedostaje podrška za pojedinu grupu senzora, odnosno ukoliko većina algoritama za obradu „sirovih“ podataka sa senzora nije realizovana. Ocena 1 se dodeljuje ukoliko nije moguće integrisati određenu grupu senzora u rešenje, zbog nedostatka potrebnih fizičkih sprega.

Podrška za aktuatore i aktuatorске scenarije se evaluira kroz tri podkriterijuma: (1) podrška za svetlosne efekte; (2) podrška za upravljanje motorima (roletne, vrata i sl); i (3) podrška za ostale aktuatore (brave, HVAC, i sl).

U okviru podkriterijuma podrške za svetlosne efekte, ocena 5 se dodeljuje ukoliko rešenje poseduje podršku za prekidače, dimere i kontrolere svetala u boji (npr. posredstvom DMX ili DALI sprega). Ocena 4 se dodeljuje ukoliko se svetlo može kontrolisati samo posredstvom prekidača i dimera. Ocene 3 i 2 se dodeljuju ukoliko je za podršku kontroli svetla potrebna dodatna implementacija, u manjem ili većem obimu. Ocena 1 se dodeljuje ukoliko rešenje ne omogućava kontrolu svetla zbog fizičkih ograničenja (npr. nedostaju potrebne fizičke sprege).

U okviru podkriterijuma podrške za upravljanje motorizacijom, ocena 5 se dodeljuje ukoliko rešenje omogućava upravljanje vratima i roletnama. Ocena 4 se dodeljuje ukoliko pojedine roletne/vrata nisu upravljiva od strane sistema. Ocene 3 i 2 se dodeljuju ukoliko je za podršku upravljanju vratima i roletnama potrebna dodatna implementacija, u manjem ili većem obimu. Ocena 1 se dodeljuje ukoliko rešenje ne omogućava kontrolu vrata/roletni zbog fizičkih ograničenja (npr. nedostaju potrebne fizičke sprege).

U okviru podkriterijuma podrške za ostale aktuatore, ocena 5 se dodeljuje ukoliko rešenje podržava kontrolu brava i sistema za grejanje i hlađenje. Ocena 4 se

Ekperimentalni rezultati i diskusije

dodeljuje ukoliko postoje problemi sa upravljivošću brava i sistema za grejanje i hlađenje u određenim slučajevima. Ocene 3 i 2 se dodeljuju ukoliko je za podršku upravljanju bravama i sistemima za grejanje i hlađenje potrebna dodatna implementacija, u manjem ili većem obimu. Ocena 1 se dodeljuje ukoliko rešenje ne omogućava kontrolu brava i sistema za grejanje i hlađenje zbog fizičkih ograničenja (npr. nedostaju potrebne fizičke sprege).

Cena se evaluira na sledeći način: ocena 5 se dodeljuje ukoliko je cena u rangu uređaja potrošačke elektronike namenjene širokoj potrošnji; ocena 4 ukoliko je cena dostupna potrošačima, u rangu zbira cena nekoliko uređaja potrošačke elektronike široke potrošnje; ocena 3 ukoliko je cena na granici mogućnosti potrošača; ocena 2 ukoliko je cena nešto iznad granice mogućnosti potrošača (investiciona granica); ocena 1 ukoliko je cena izuzetno visoka (u rangu profesionalne opreme). Evaluacija cena data je na bazi kupovne moći prosečnog stanovnika zemalja zapadne Evrope i Sjedinjenih Američkih Država.

Rezultati po ranije opisanim kriterijumima se predstavljaju i analiziraju na sledeća tri načina: (1) prema kriterijumima; (2) pojedinačno i (3) prema ukupnoj oceni.

Rezultat predstavljen prema kriterijumima obuhvata poređenje rešenja prema nivou ispunjenosti svakog od šest osnovnih kriterijuma. Nivo ispunjenosti kriterijuma prikazan je sa tri veličine: (1) nivo ispunjenosti *buy* kriterijuma, odnosno procenat podkriterijuma koji su ispunjeni sa ocenama 4 i 5, odnosno gde je rešenje primenljivo bez većih izmena; (2) nivo ispunjenosti *make* kriterijuma, odnosno procenat podkriterijuma koji su ispunjeni sa ocenama 2 i 3, odnosno gde su potrebne veće ili manje izmene da bi se omogućila primena i (3) procenat podkriterijuma koji nisu ispunjeni, odnosno kojima je dodeljena ocena 1. Kvalitet rešenja ogleda se u što većem procentu ispunjenosti *buy* kriterijuma, i odsustvu procenta neispunjenih podkriterijuma.

Rezultat predstavljen pojedinačno obuhvata predstavljanje svakog rešenja sopstvenim radar dijagramom sa šest osa, od kojih svaka osa odgovara jednom od šest kriterijuma. Na radar dijagramu se vidi procenat ispunjenosti *buy* i *make* kriterijuma, kao i procenat neispunjenih podkriterijuma. Radar dijagram očigledno pokazuje primenljivost rešenja zahvaljujući upotrebljenim bojama.

Rezultat predstavljen prema ukupnoj oceni predstavlja rangiranje rešenja prema ukupnoj prosečnoj oceni svih kriterijuma, uz uvažavanje težinskih faktora. U ovu svrhu odabrani su sledeći težinski faktori: faktor 2 za zahteve po Dey, kao najvažnije zahteve sa stanovišta oblasti kontekstualnog računarstva; faktor 2 za opšte kontekstualne zahteve koji su takođe izuzetno važni sa aspekta ciljne oblasti; faktor 2 za integrabilnost i primenljivost za razvoj kontekstualnih aplikacija potrošačke elektronike, s obzirom na značaj primene u oblasti potrošačke elektronike; faktor 1 za podršku za senzore, s obzirom da je podrška za pojedinačne senzore tehnički faktor koji je nižeg prioriteta od opšte arhitekture; faktor 0.5 za podršku za aktuatore, s obzirom da je ta podrška od najmanjeg značaja u kontekstualnim platformama koje su prevashodno zasnovane na prikupljanju kontekstualnih parametara posredstvom senzora; faktor 1 za cenu.

5.4.2 Rezultati evaluacije potencijala za razvoj kontekstualnih aplikacija

U ovom poglavlju dati su rezultati evaluacije potencijala za razvoj kontekstualnih aplikacija, u skladu sa metodologijom opisanom u poglavlju 5.4.1. Najpre su dati rezultati evaluacije po kriterijumima, zatim pojedinačno, i na kraju prema ukupnoj oceni.

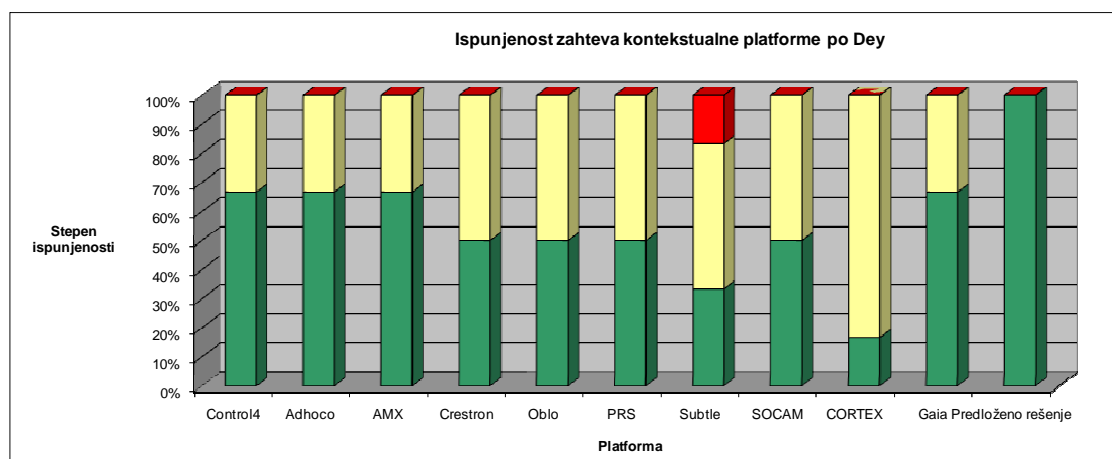
5.4.2.1 Rezultati evaluacije po kriterijumima

U Tabeli 28 dati su brojučani rezultati evaluacije za prvi kriterijum (zahtevi kontekstualne platforme po Dey). Na Slici 126 dat je grafički prikaz stepena ispunjenosti prvog kriterijuma (*make vs buy*).

Ekperimentalni rezultati i diskusije

	<i>Razdvojenost fizičkog nivoa</i>	<i>Interpretacija konteksta</i>	<i>Transparentna i distribuirana komunikacija</i>	<i>Trajnost prikupljanja kontekstualnih parametara</i>	<i>Skladištenje kontekstualnih parametara</i>	<i>Pretraživanje uređaja i servisa</i>
Control4	4	2	5	4	3	4
Adhoco	3	4	4	5	2	4
AMX	4	2	4	4	2	4
Crestron	2	2	4	4	2	4
Oblo	3	3	4	4	2	4
PRS	5	3	5	3	3	4
Subtle	2	4	1	2	4	2
SOCAM	5	2	3	5	4	2
CORTEX	4	3	2	2	2	2
Gaia	4	2	4	4	3	5
Predloženo rešenje	5	4	5	5	4	5

Tabela 28. Ocene evaluacije zahteva kontekstualne platforme po Dey



Slika 126. Grafički prikaz stepena ispunjenosti prvog kriterijuma (*make vs buy*): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo

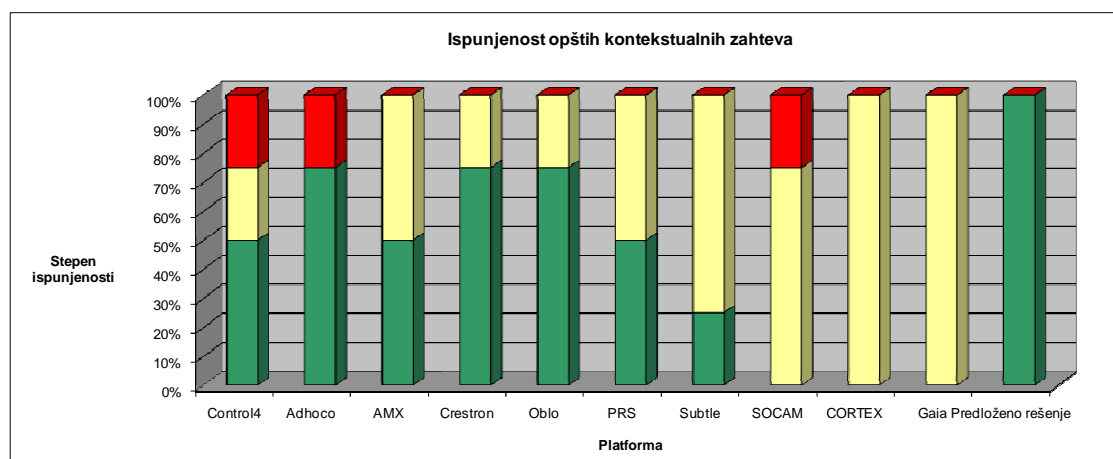
Iz priloženog se može videti da jedino predloženo rešenje u potpunosti ispunjava zahteve kontekstualne platforme po Dey. Ostale evaluirane platforme u delimičnoj meri ispunjavaju zahteve. Nizak skor kontekstualnih platformi iz istraživanja (SOCAM, CORTEX, Subtle) pokazuje da u ovim istraživanjima nisu kompletno sagledani svi potrebni aspekti, već je naglasak bio na algoritmima ili pojedinim delovima arhitekture. Control4, Adhoco i AMX imaju izuzetno visok stepen ispunjenosti ovih zahteva, što pokazuje da se sistemi za automatizaciju domaćinstva koji su dostupni na tržištu mogu često veoma dobro upotrebiti za realizaciju

kontekstualnih aplikacija (iako nedovoljno), što dokazuje opravdanost njihovog uključivanja u razmatranje u poglavlju 1.4.1.

U Tabeli 29 dati su broježani rezultati evaluacije za drugi kriterijum (opšti zahtevi kontekstualne platforme). Na Slici 127 dat je grafički prikaz stepena ispunjenosti drugog kriterijuma (*make vs buy*).

	<i>Transparentnost</i>	<i>Brzina odziva</i>	<i>Prilagodljivost i podešljivost</i>	<i>Neposredno upravljanje</i>
Control4	4	1	3	5
Adhoco	4	1	4	5
AMX	4	3	3	5
Crestron	4	4	2	5
Oblo	4	4	3	5
PRS	4	4	3	3
Subtle	2	4	2	2
SOCAM	3	1	3	2
CORTEX	3	3	2	2
Gaia	3	3	2	2
Predloženo rešenje	4	4	4	5

Tabela 29. Ocene evaluacije opštih zahteva kontekstualne platforme



Slika 127. Grafički prikaz stepena ispunjenosti drugog kriterijuma (*make vs buy*): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo

S obzirom da su u slučaju predloženog rešenja pažljivo sagledani svi aspekti arhitekture kontekstualnih platformi, dobijene su očekivano visoke ocene. Neoptimalna realizacija u pogledu grafičke sprege, jezika korišćenog za realizaciju i sl.

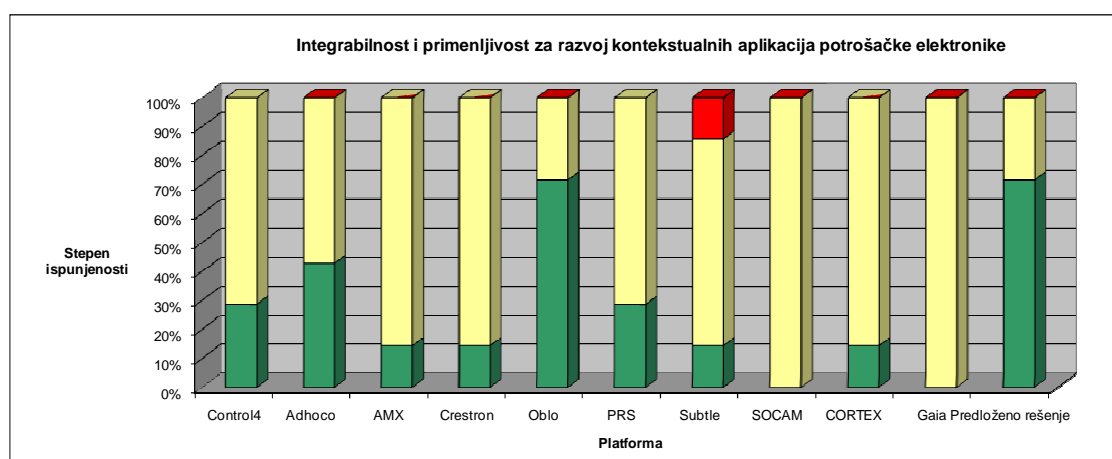
Ekperimentalni rezultati i diskusije

uslovljavaju da Control4, Adhoco i SOCAM nisu u mogućnosti da ispune zahteve realnog vremena. Takođe, u slučaju okruženja CORTEX i Gaia, potreban je znatan napor da bi se došlo do potpunog ispunjenja zahteva.

U Tabeli 30 dati su brožčani rezultati evaluacije za treći kriterijum (integrabilnost i primenljivost za razvoj kontekstualnih aplikacija potrošačke elektronike). Na Slici 128 dat je grafički prikaz stepena ispunjenosti trećeg kriterijuma (*make vs buy*).

	<i>Potrošnja sistemskih resursa</i>	<i>Potrošnja energije</i>	<i>Licence i sertifikati</i>	<i>Portabilnost i apstraktnost</i>	<i>Integracija sa UI</i>	<i>Detekcija stanja korisnika</i>	<i>Komunikacija sa drugim uređajima</i>
Control4	2	3	5	3	3	3	4
Adhoco	3	4	3	3	2	4	4
AMX	2	3	3	3	3	3	4
Crestron	2	3	3	3	3	3	4
Oblo	4	4	2	4	4	3	4
PRS	2	3	4	4	3	3	3
Subtle	3	3	3	2	2	5	1
SOCAM	2	3	2	3	2	2	2
CORTEX	4	3	2	3	2	2	2
Gaia	3	3	2	3	2	2	2
Predloženo rešenje	4	3	4	4	3	5	4

Tabela 30. Ocene evaluacije integrabilnosti i primenljivosti za razvoj kontekstualnih aplikacija potrošačke elektronike



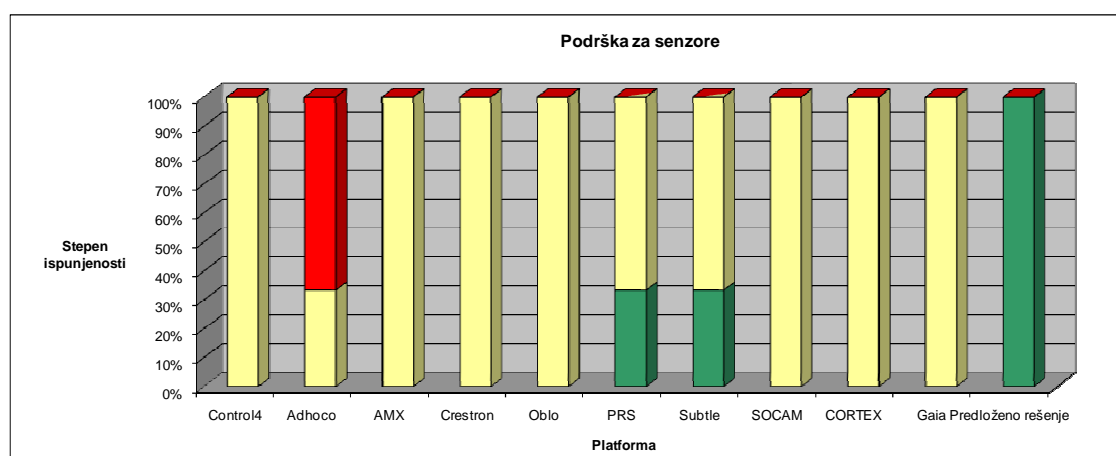
Slika 128. Grafički prikaz stepena ispunjenosti trećeg kriterijuma (*make vs buy*): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo

Rezultati evaluacije trećeg kriterijuma pokazuju da je izuzetno teško u potpunosti zadovoljiti postavljene kriterijume, a zadržati nivo apstraktnosti arhitekture kontekstualne platforme koja se zahteva. Predloženo rešenje, ipak, u ovom pogledu ostvaruje zadovoljavajuće ocene. Kod većine rešenja potreban je znatan napor u cilju ispunjenja kriterijuma integrabilnosti; iako označeno kao moguće uz implementaciju, u praksi se pokazuje da je napor potreban za implementaciju izuzetan, često na granici izvodljivosti.

U Tabeli 31 dati su brožčani rezultati evaluacije za četvrti kriterijum (podrška za senzore). Na Slici 129 dat je grafički prikaz stepena ispunjenosti četvrtog kriterijuma (*make vs buy*).

	<i>Podrška za optičke senzore</i>	<i>Podrška za senzore zvuka</i>	<i>Podrška za ostale senzore (akcelerometar i sl)</i>
Control4	3	2	3
Adhoco	1	1	3
AMX	3	2	3
Crestron	2	3	3
Oblo	2	2	2
PRS	5	2	3
Subtle	2	5	3
SOCAM	2	2	2
CORTEX	2	2	2
Gaia	2	2	2
Predloženo rešenje	5	5	4

Tabela 31. Ocene evaluacije podrške za senzore



Slika 129. Grafički prikaz stepena ispunjenosti četvrtog kriterijuma (*make vs buy*): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo

Ekperimentalni rezultati i diskusije

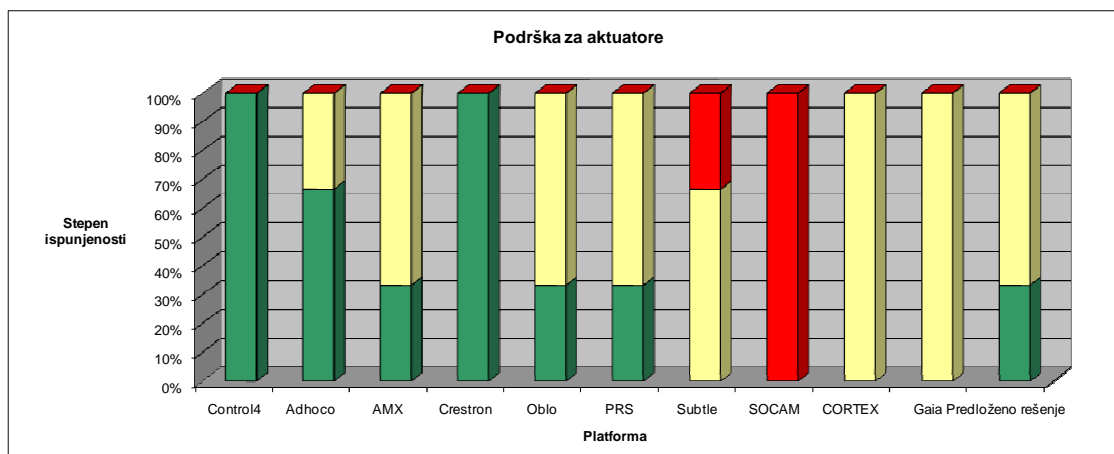
Rezultati evaluacije četvrtog kriterijuma pokazuju da je predloženo rešenje najkompletnije u pogledu podrške za različite senzore, koja je siromašna u većini istraživanja sistema za automatizaciju domaćinstva, ali i kontekstualnih platformi u celini. Ipak, većina rešenja omogućava dodatno programiranje u cilju omogućavanja podrške za više senzora, što je očekivano s obzirom da je većina rešenja zasnovana na opštenamenskim fizičkim arhitekturama, uz posedovanje uobičajenih sprega poput USB, Ethernet i RS232.

U Tabeli 32 dati su brožani rezultati evaluacije za peti kriterijum (podrška za aktuator). Na Slici 130 dat je grafički prikaz stepena ispunjenosti petog kriterijuma (*make vs buy*).

	<i>Podrška za svetlosne efekte</i>	<i>Podrška za upravljanje motorizacijom (vrata, roletne)</i>	<i>Podrška za ostale aktuator (brave, HVAC i sl)</i>
Control4	4	5	5
Adhoco	4	4	2
AMX	4	3	3
Crestron	5	4	5
Oblo	4	2	2
PRS	5	2	2
Subtle	1	2	2
SOCAM	1	1	1
CORTEX	2	2	2
Gaia	2	2	2
Predloženo rešenje	5	2	2

Tabela 32. Ocene evaluacije podrške za aktuator

Podrška za aktuator nije tipična u kontekstualnim platformama; ipak, ona može biti značajna za brojne nove kontekstualne aplikacije. Prirodno, podrška za aktuator je veoma visoka u sistemima za automatizaciju domaćinstva. Predloženo rešenje poseduje visok nivo kontrole svetlosnih efekata, i ipak pruža dobar kompromis između uobičajenih funkcionalnosti kontekstualne platforme i funkcionalnosti opštenamenske platforme za automatizaciju domaćinstva, čime se omogućava realizacija šireg spektra kontekstualnih aplikacija.

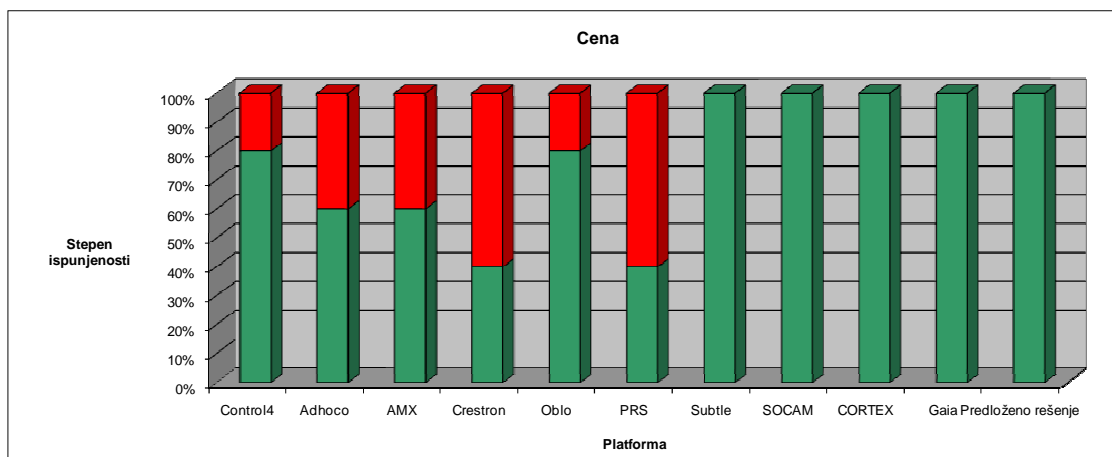


Slika 130. Grafički prikaz stepena ispunjenosti petog kriterijuma (*make vs buy*): zelena boja – rešenje primenjivo bez većih izmena; žuta boja – rešenje primenjivo uz izmene; crvena boja – rešenje nije primenjivo

U Tabeli 33 dati su rezultati procene cene (šesti kriterijum). Na Slici 131 dat je grafički prikaz stepena ispunjenosti šestog kriterijuma. Rezultati po ovom kriterijumu su indikativni, s obzirom da je promena cena konstantna, kao i da platforme koje su realizovane u okviru istraživanja (uključujući realizovanu platformu) nemaju definisanu cenu. Za ove slučajeve pretpostavljena je minimalna cena.

	<i>Ocena za procenjenu cenu</i>
Control4	4
Adhoco	3
AMX	3
Crestron	2
Oblo	4
PRS	2
Subtle	5
SOCAM	5
CORTEX	5
Gaia	5
Predloženo rešenje	5

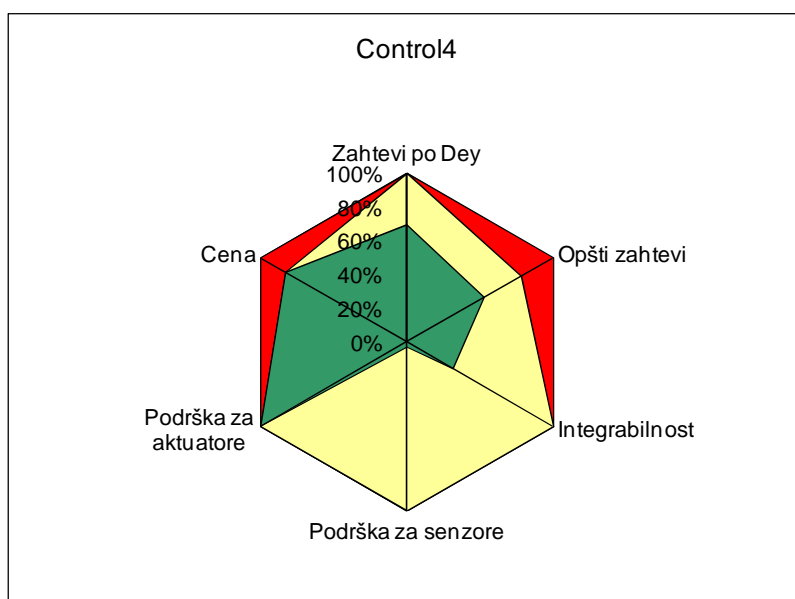
Tabela 33. Ocene evaluacije cene rešenja



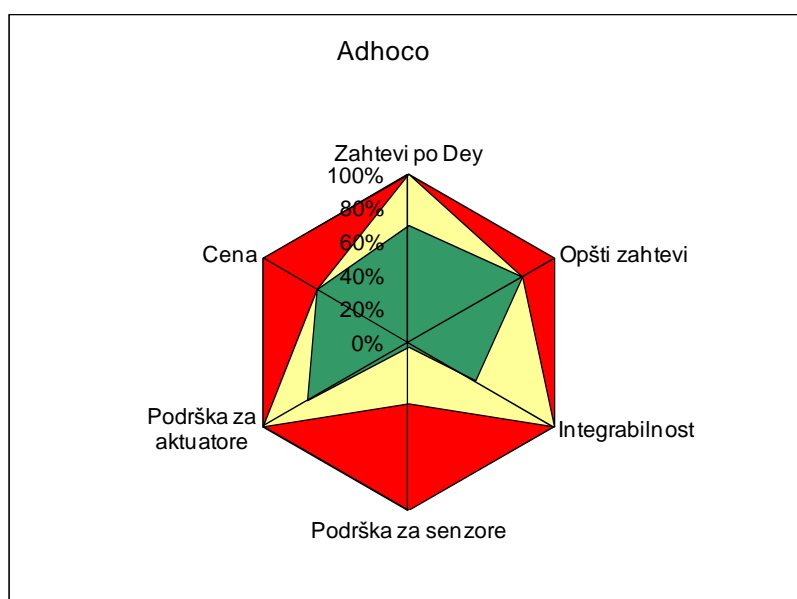
Slika 131. Grafički prikaz stepena ispunjenosti šestog kriterijuma (cena): zelenom bojom prikazan je nivo efikasnosti cene – što je nivo veći, cena je niža

5.4.2.1 Rezultati pojedinačne evaluacije

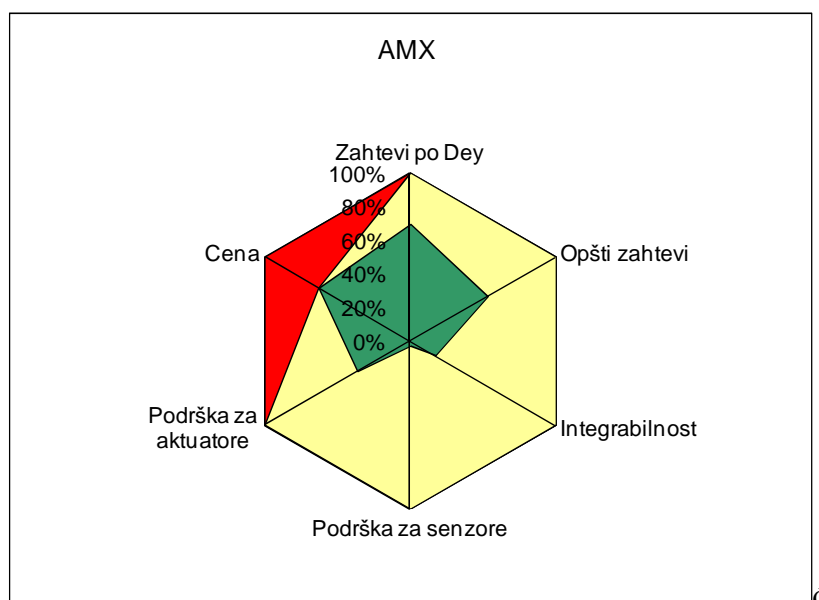
U ovom poglavlju dati su rezultati pojedinačne evaluacije rešenja, predstavljeni radar grafikonom. Sa ovih grafikona lako se može uočiti opšti stepen ispunjenosti svih zahteva od strane svakog od rešenja – što je veća ispunjenost zelenom bojom, ispunjenost zahteva je veća, dok postojanje crvene boje označava nemogućnost ispunjavanja pojedinih zahteva. Grafikoni za svako od rešenja, uključujući i predloženo rešenje, dati su na Slikama 132 – 142.



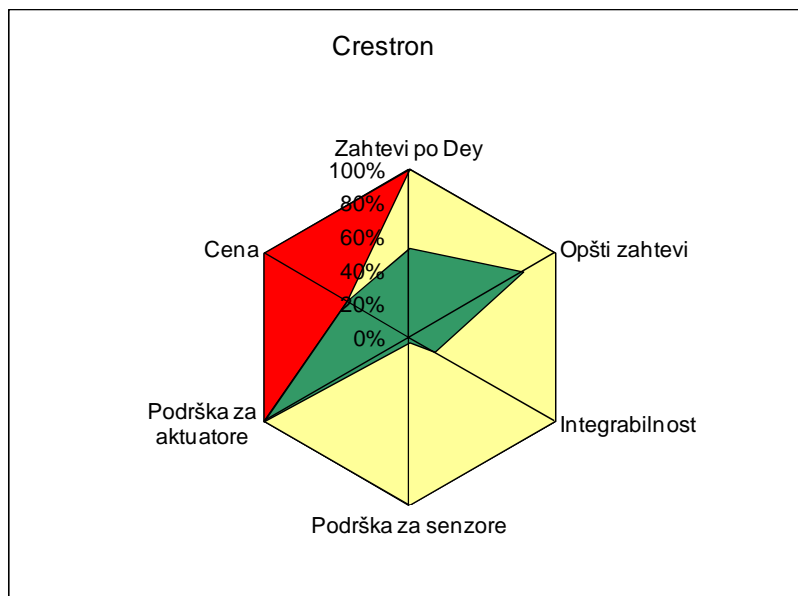
Slika 132. Grafički prikaz pojedinačne evaluacije za okruženje Control4



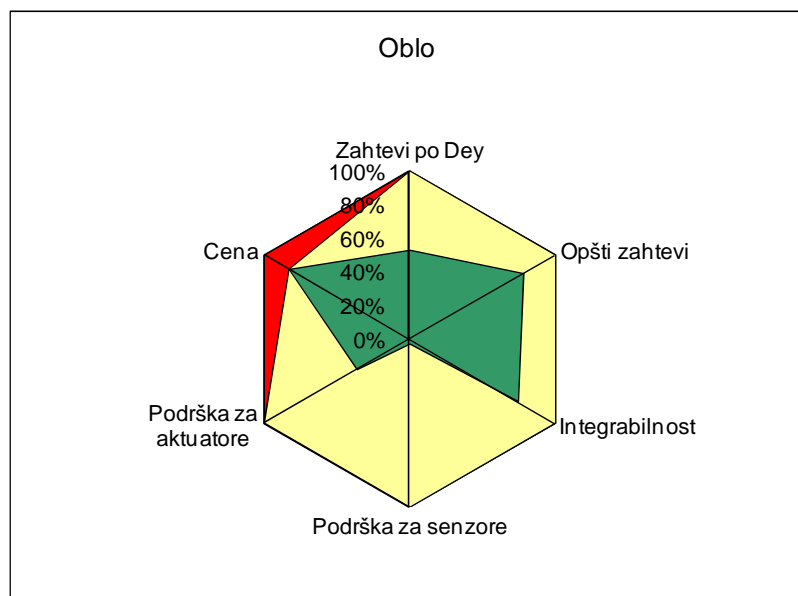
Slika 133. Grafički prikaz pojedinačne evaluacije za okruženje Adhoco



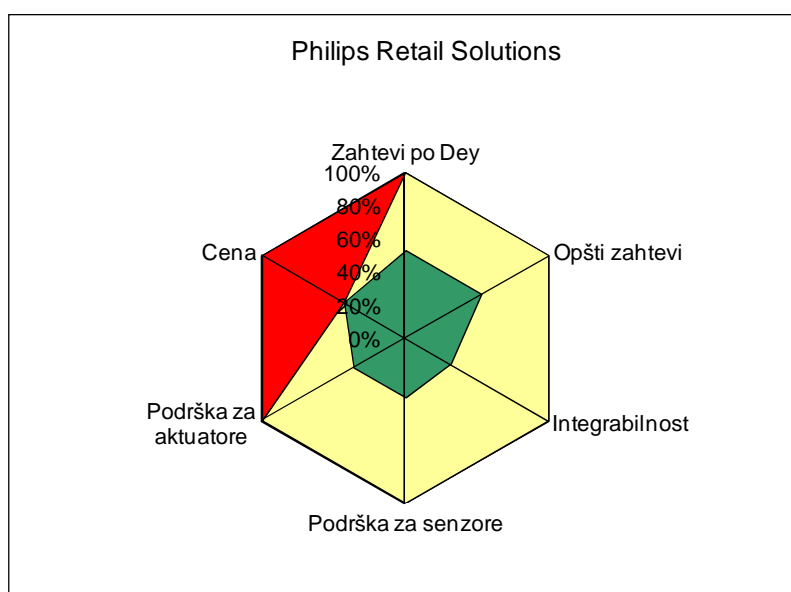
Slika 134. Grafički prikaz pojedinačne evaluacije za okruženje AMX



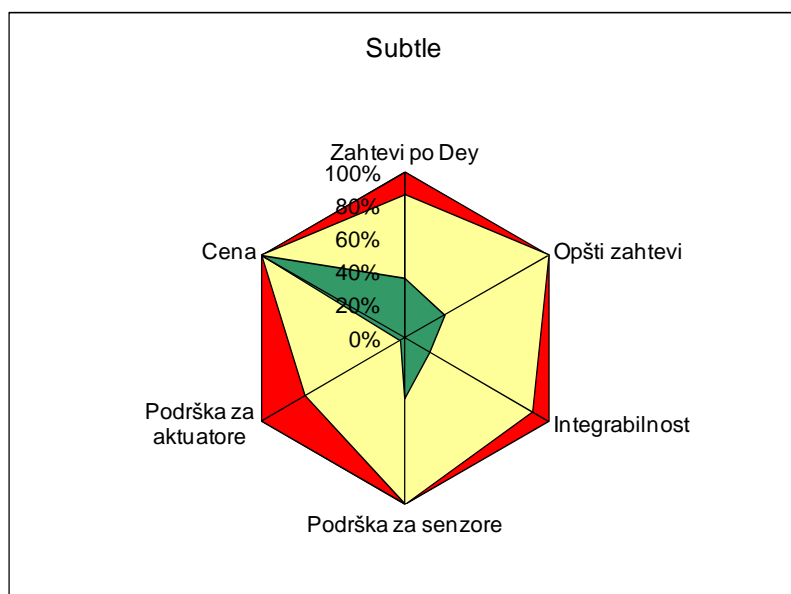
Slika 135. Grafički prikaz pojedinačne evaluacije za okruženje Crestron



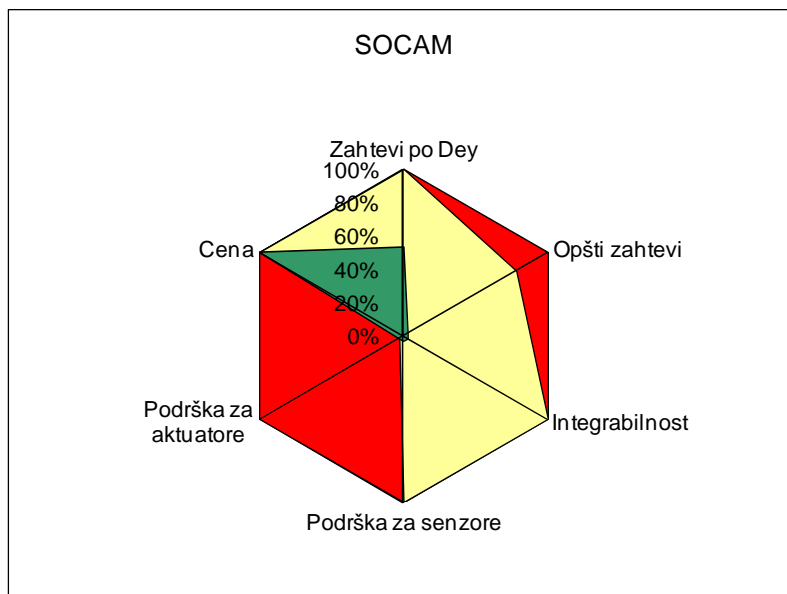
Slika 136. Grafički prikaz pojedinačne evaluacije za okruženje Oblo



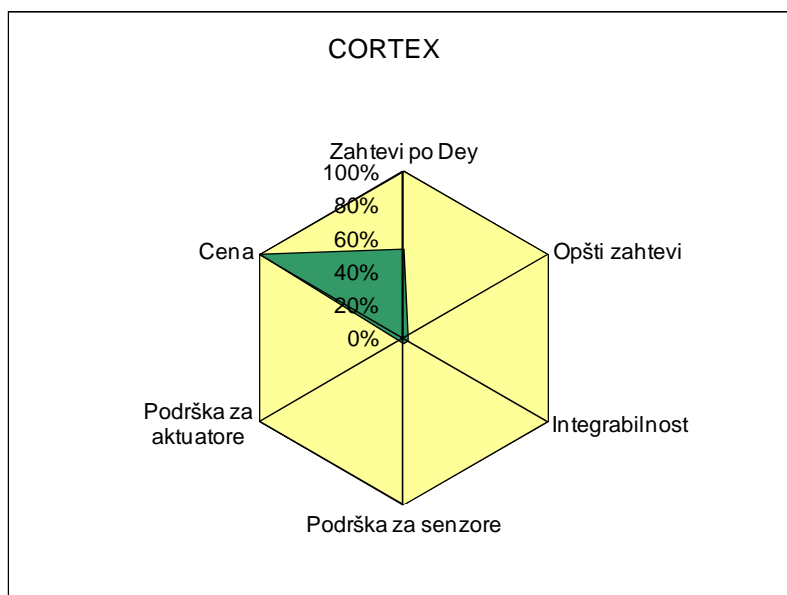
Slika 137. Grafički prikaz pojedinačne evaluacije za okruženje Philips Retail Solutions



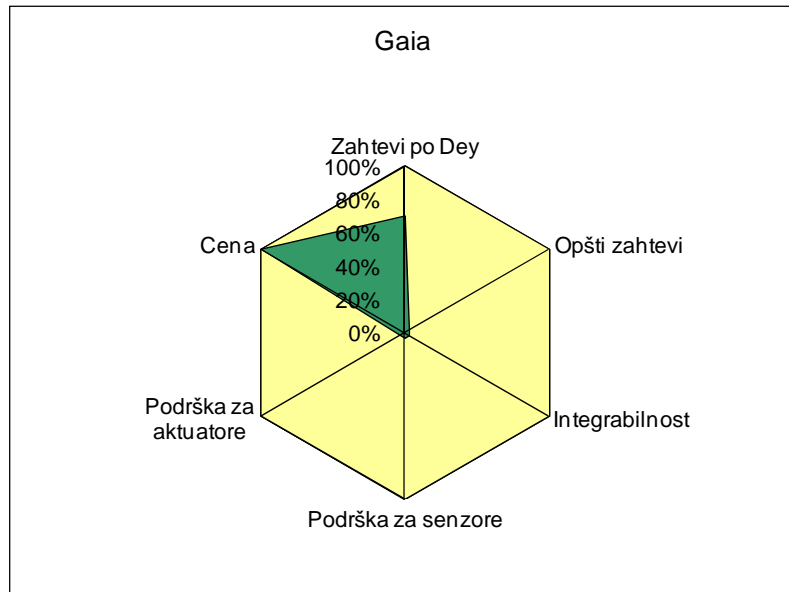
Slika 138. Grafički prikaz pojedinačne evaluacije za okruženje Subtle



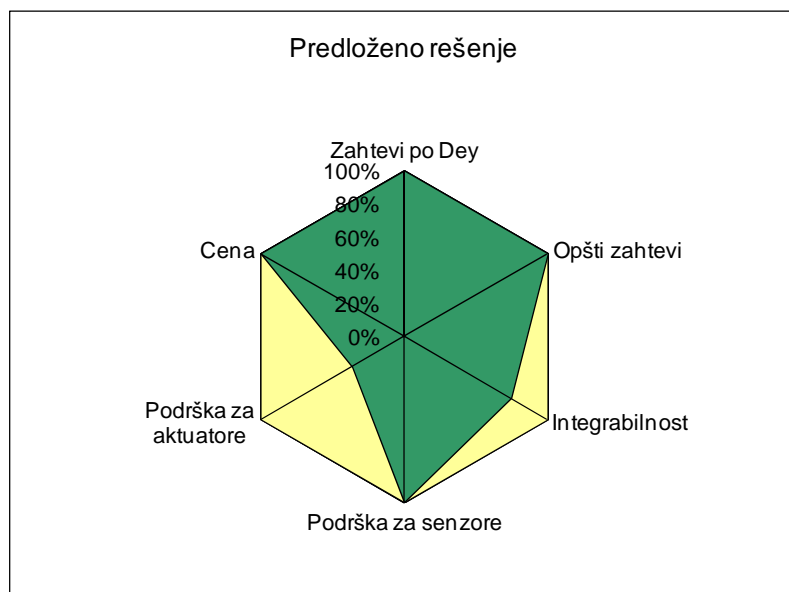
Slika 139. Grafički prikaz pojedinačne evaluacije za okruženje SOCAM



Slika 140. Grafički prikaz pojedinačne evaluacije za okruženje CORTEX



Slika 141. Grafički prikaz pojedinačne evaluacije za okruženje Gaia



Slika 142. Grafički prikaz pojedinačne evaluacije za predloženo rešenje

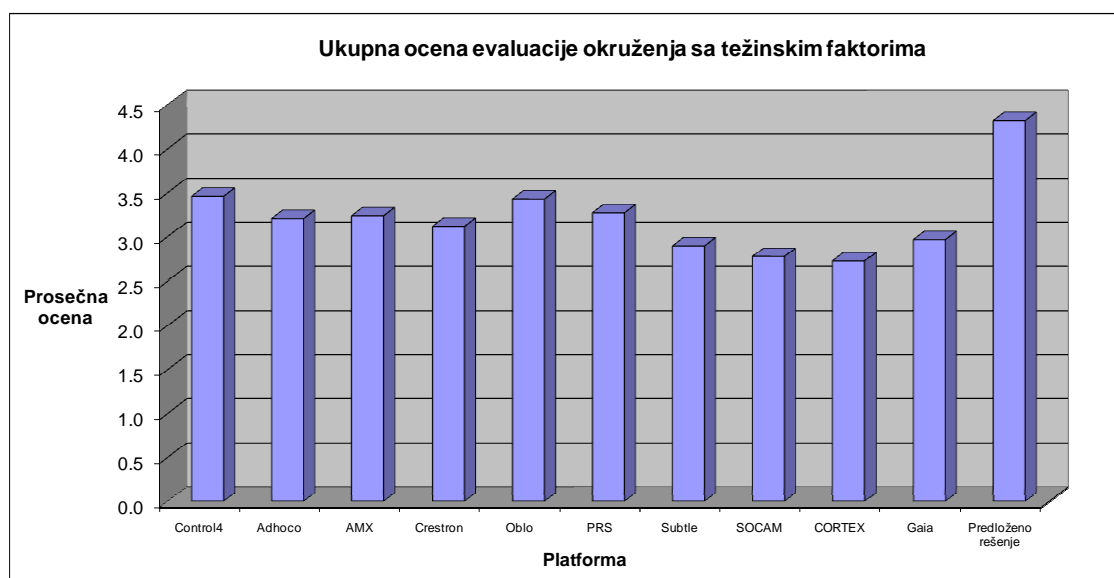
Ukoliko se posmatraju isključivo funkcionalni zahtevi, uz zanemarivanje cene, sa priloženih grafikona uočava se da okruženja AMX, CRESTRON, Oblo, PRS, CORTEX i Gaia, zajedno sa predloženim rešenjem, ne poseduju nijedan kriterijum koji se ne može zadovoljiti. Među ovim rešenjima, predloženo rešenje poseduje

Ekperimentalni rezultati i diskusije

najveći stepen zadovoljenja postavljenih kriterijuma. Rešenja predložena u prethodnim istraživanjima, poput okruženja CORTEX i Gaia ne poseduju dovoljno specifičnih tehničkih rešenja da bi se uspešno i brzo mogla primeniti u uređajima potrošačke elektronike. Ipak, ova okruženja, zajedno sa predloženim rešenjem, mogu predstavljati osnovu za dalja istraživanja u oblasti disertacije.

5.4.2.2 Rezultati evaluacije prema ukupnoj oceni

U ovom poglavlju dati su rezultati evaluacije prema ukupnoj oceni, izvedenoj na osnovu svih evaluiranih kriterijuma, uz primenu težinskih faktora definisanih u poglavlju 5.4.1. Ovi rezultati služe da pokažu da predloženo rešenje u proseku nadmašuje dosadašnja rešenja (Slika 143). S obzirom da prosečna ocena najčešće nije valjan pokazatelj uspešnosti određene implementacije, uzevši u obzir mogućnost neispunjenja određenih ključnih zahteva, dati su rezultati evaluacije po kriterijumima i pojedinačne evaluacije u prethodnim poglavljima (5.4.2.1 i 5.4.2.1).



Slika 143. Rezultati evaluacije rešenja prema ukupnoj oceni

POGLAVLJE 6.

PRIMERI PRIMENE REŠENJA

U ovom poglavlju dati su primeri nekoliko realizovanih primena kontekstualne platforme. Primeri obuhvataju kontekstualne korisničke sprege realizovane kao integrisane kontekstualne aplikacije, i kontekstualne aplikacije, realizovane u okviru platforme, ili kao posebna rešenja koja komuniciraju sa kontekstualnom platformom korišćenjem HTTP API-ja.

6.1 KONTEKSTUALNE KORISNIČKE SPREGE

U ovom poglavlju predstavljeno je nekoliko primera kontekstualnih korisničkih sprega realizovanih na osnovu predložene kontekstualne platforme.

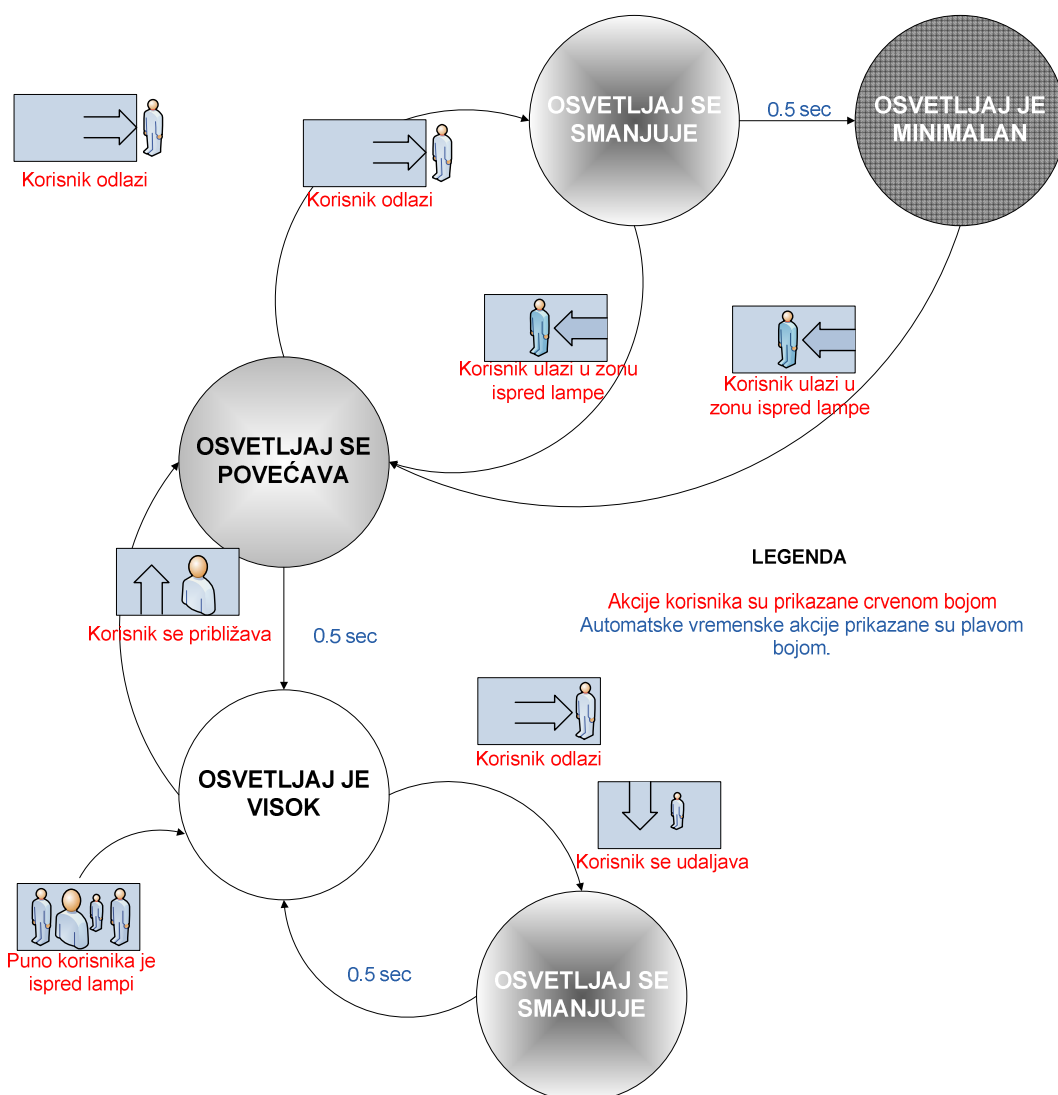
6.1.1 Sprega za privlačenje pažnje korisnika

Sprege za privlačenje pažnje realizovane su kao kontekstualne aplikacije *Luminosity* i *LightBeat*, koje koriste kontekstualnu platformu i DMX svetla u boji za kreiranje impresivnih interaktivnih efekata u cilju privlačenja pažnje korisnika.

Aplikacija *Luminosity* kao jedini senzor koristi 3D kameru, koja snima scenu veličine 3 x 4m, u okviru koje se prati kretanje korisnika. Sa jedne strane scene postavljen je niz od više DMX lampi u boji. Intenzitet osvetljenosti svake lampe korespondira udaljenosti korisnika od lampe u tom trenutku. Ukoliko je korisnik bliže lampi, intenzitet osvetljenosti je veći, a ukoliko se korisnik udaljava od lampe,

Primeri primene rešenja

intenzitet njene osvetljenosti se smanjuje. Ukoliko se ova postavka koristi, npr. za isticanje određenih zona od interesa pored kojih često prolaze ljudi (npr. izlog prodavnice, reklamni štand i sl), prilikom prolaska kreira se svetlosni „talas“ koji ih prati dok se kreću, sa ciljem privlačenja pažnje. Ukoliko korisnik zastane, korisnička sprega angažuje njegovu pažnju, s obzirom da je mesto u nizu lampi najosvetljenije baš tu gde korisnik stoji. Prilikom prilaska nizu lampi, osvetljenje je sve jače dok se, konačno, nakon što korisnik priđe potpuno blizu, aktivira usmereno svetlo koje prezentuje predmet od interesa, ili se aktivira određena multimedijalna prezentacija, čime je svrha sprege u potpunosti ispunjena jer ovim korisnik praktično pristaje na uslugu koja se nudi. Ilustracija prelaza stanja u algoritmu rada aplikacije *Luminosity* data je na Slici 144.



Slika 144. Ilustracija prelaza stanja u algoritmu rada aplikacije *Luminosity*

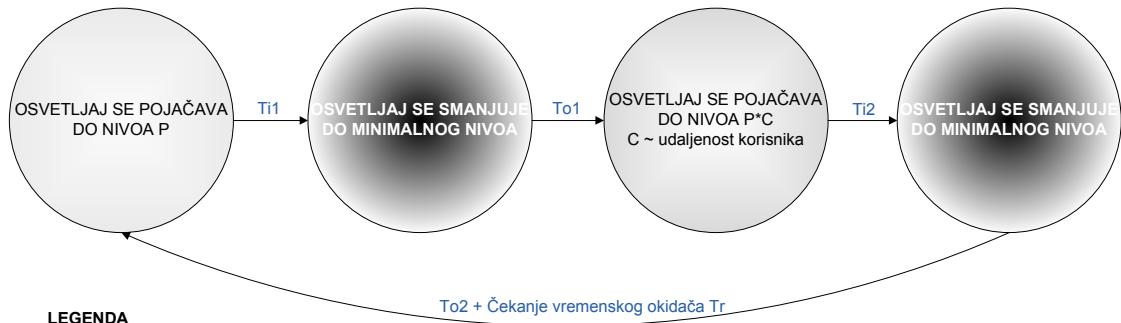
Prikaz rada aplikacije *Luminosity* u laboratorijskim uslovima dat je na Slici 145.



Slika 145. Prikaz rada aplikacije *Luminosity* u laboratorijskim uslovima

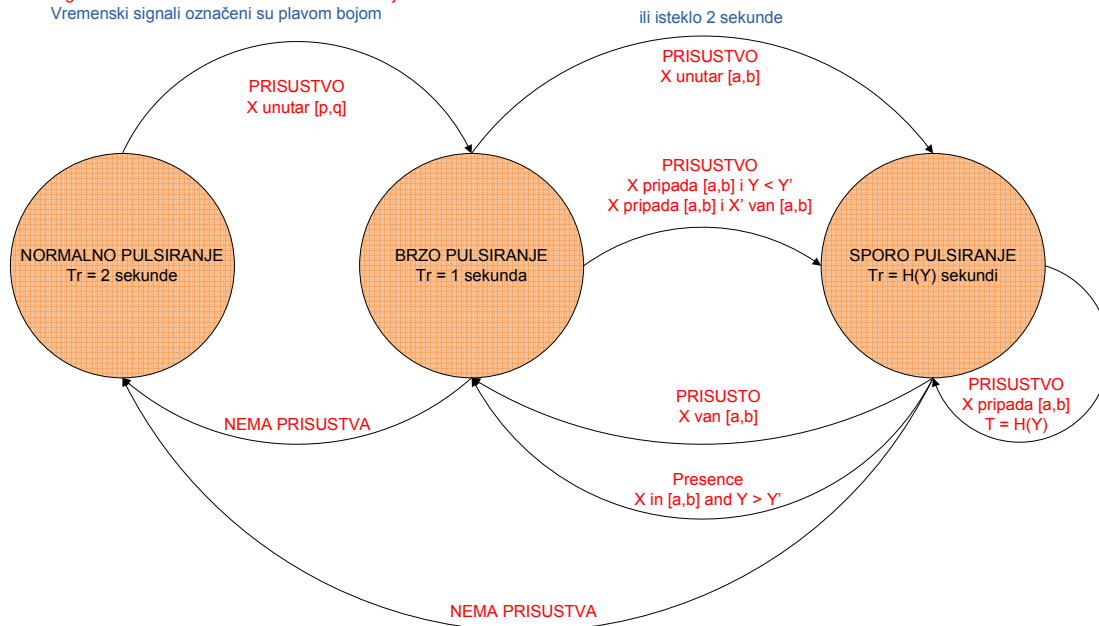
Aplikacija *LightBeat* koristi istovetnu postavku kao *Luminosity*, s tim što je ponašanje lampi u odnosu na korisnika nešto drugačije. U situaciji u kojoj na sceni nema korisnika, lampe postepeno podižu i spuštaju nivo osvetljaja (pulsiraju) u cilju imitacije otkucaja srca, srednjom brzinom. Nakon što korisnik uđe u zonu jedne od lampi, tempo pulsiranja te lampe se ubrzava u cilju privlačenja pažnje. Nakon što se korisnik zaustavi ispred lampe, tempo se postepeno usporava, i biva sve sporiji što je korisnik bliže. U slučaju da korisnik poželi da se udalji, i postepeno se počne udaljavati od lampe, tempo pulsiranja se naglo ubrzava. Cilj ovakvog ponašanja je da angažuje pažnju korisnika u smeru izazivanja emocija, s obzirom da lampe oponašaju živo biće sa otkucajima srca. U paraleli sa pulsiranjem, čuje se sinhronizovan zvuk srčanog ritma. Maksimalan nivo osvetljenosti lampe pri otkucaju ekvivalentan je udaljenosti korisnika, kao u slučaju aplikacije *Luminosity*. Ilustracija prelaza stanja u algoritmu rada aplikacije *LightBeat* data je na Slici 146.

Primeri primene rešenja



LEGENDA

X – Horizontalna koordinata detektovane osobe
 Y – Udaljenost detektovane osobe od kamere
 X', Y' - Poslednje koordinate detektovane osobe
 a,b – X koordinate koje definišu zonu ispred lampe
 p,q – X koordinate koje definišu širu zonu, ispred i u blizini lampe, a,b unutar [p,q]
 P – Predefinisani maksimalni nivo osvetljenosti
 C – Umnožak nivoa osvetljenosti [0,1]
 H(Y) – Brzina pulsiranja u zavisnosti od udaljenosti Y
 Tr – Vektor vremenskih okidača
 Ti1, To1, Ti2, To2 – Konfigurabilni parametri pulsiranja
 Signali sa 3D kamere označeni su crvenom bojom
 Vremenski signali označeni su plavom bojom



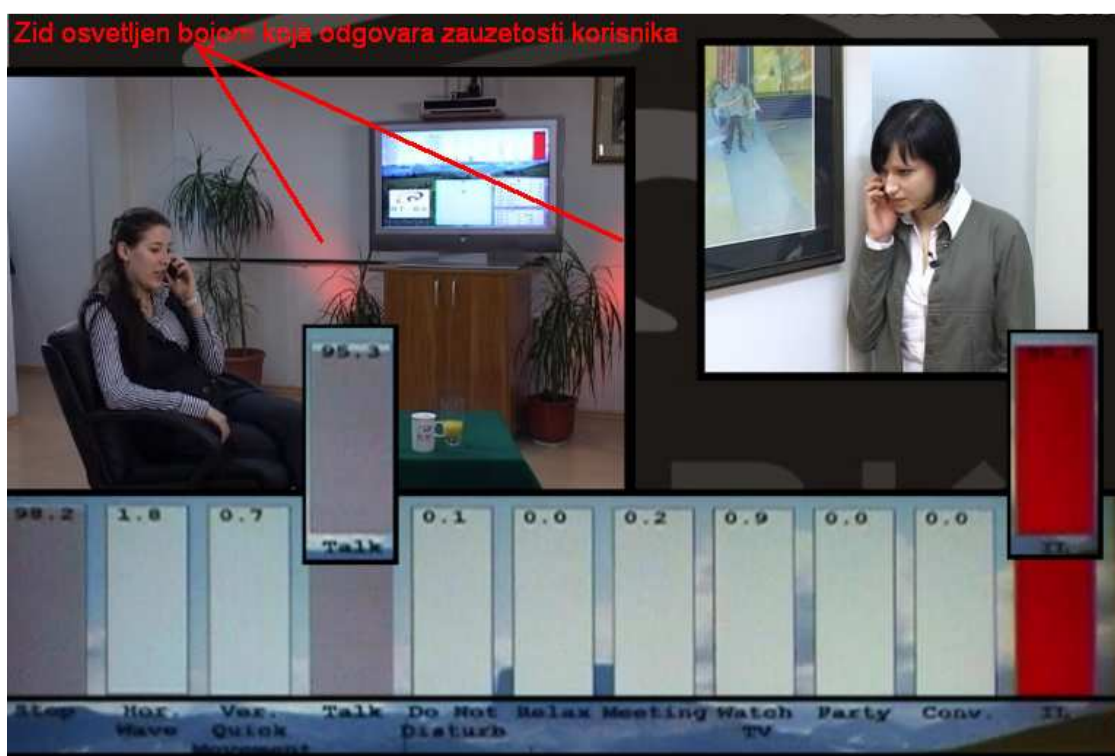
Slika 146. Ilustracija prelaza stanja u algoritmu rada aplikacije *LightBeat*

6.1.2 Sprega za objavljivanje zauzetosti korisnika

Izlaz algoritma za detekciju zauzetosti korisnika, koji je omogućen u okviru kontekstualne platforme, može se iskoristiti za postavljanje boje svetla na pogodnim mestima, vidljivim od strane korisnika, u cilju smanjenja broja neželjenih socijalnih asinhronih zahteva. Na primer, svetlo u boji može biti postavljeno na ulazu u prostoriju ili u hodnik ispred, tako da bude vidljivo sa spoljne strane. Ukoliko je boja svetla crvena, to znači da je zauzetost korisnika visoka i da zahtevi u tom trenutku nisu poželjni. Ukoliko je boja svetla žuta, to znači da je zauzetost korisnika umerena i da su

dozvoljeni samo hitni zahtevi. Ukoliko je boja svetla zelena, to znači da je zauzetost korisnika niska i da su svi zahtevi dozvoljeni. S obzirom na analizu navedenu u poglavlju 1.6.2, očekivan efekat je da boje znatno bolje utiču na odlučivanje korisnika koji mogu da upute neželjeni socijalni zahtev, zahvaljujući kognitivnim i perceptivnim faktorima, ali i objektivnim faktorima, kao što je činjenica da su svetla postavljena u neposrednoj blizini osobe koja je predmet zahteva.

Na Slici 147 prikazan je izgled situacije u laboratorijskim uslovima, u kojoj kontekstualna platforma zaključuje visoku zauzetost osobe koja razgovara telefonom i objavljuje tu informaciju posredstvom kontekstualne korisničke sprege, korišćenjem svetla u crvenoj boji koje obasjava zid kancelarije.



Slika 147. Izgled kontekstualne korisničke sprege za objavljivanje zauzetosti korisnika

6.2 KONTEKSTUALNE APLIKACIJE

U ovom poglavlju predstavljeni su primeri korišćenja informacija koje pruža kontekstualna aplikacija za rad drugih aplikacija.

6.2.1 Komunikacioni klijent

U okviru ranijih istraživanja u okviru instituta RT-RK i Fakulteta tehničkih nauka, razvijena je komunikaciona aplikacija za TV prijemnik (*SkypeTV*) koja omogućava razmenu tekstualnih poruka i uspostavljanje poziva sa TV prijemnika, posredstvom telefona povezanog *Bluetooth* vezom (Lakobrija, 2007). To rešenje je prošireno kontekstualnim funkcionalnostima koje nudi predložena kontekstualna platforma (Bjelica, 2011f), u cilju odabiranja optimalnog režima rada komunikacione aplikacije, koji se zaključuje na osnovu nivoa zauzetosti korisnika koji objavljuje kontekstualna platforma.

Podržani režimi u kontekstualnoj aplikaciji su *Aktivan*, *Pasivan* i *Isključen*. U režimu *Aktivan*, aplikacija funkcioniše sa najvišim nivoom interakcije sa korisnikom: dolazni pozivi se objavljuju zvučnim signalom i porukom koja se postavlja na sredinu TV ekrana. U režimu *Pasivan*, zvučni signal je utišan a dolazni pozivi se objavljuju korišćenjem informacionog okvira u uglu ekrana. U režimu *Utišan*, ne postoji objavljivanje dolaznih poziva, već su uvid u propuštene pozive i mogućnost iniciranja povratnog poziva omogućeni isključivo na korisnički zahtev, pritiskom odgovarajućeg tastera na daljinskom upravljaču. Izbor trenutno aktivnog režima direktno zavisi od procenjenog nivoa zauzetosti korisnika. Režim *Aktivan* postavlja se u slučaju da je nivo zauzetosti korisnika ispod 30%. Režim *Pasivan* postavlja se u slučaju da je nivo zauzetosti korisnika između 30% i 70%. Režim *Isključen* postavlja se u slučaju da je nivo zauzetosti korisnika preko 70%. Da bi se izbegla naizmenična brza promena režima rada aplikacije u slučaju da je nivo zauzetosti korisnika na graničnim vrednostima, koristi se histerezijski pomeraj od 5%.

Na slikama koje slede prikazane su tipične situacije korišćenja komunikacione aplikacije u laboratorijskim uslovima u simuliranom okruženju dnevne sobe. Na Slici 148 dat je izgled komunikacione aplikacije i prikaz situacije za režim rada *Aktivan*. Može se primetiti da osobe u prostoriji imaju neformalan razgovor i ne posmatraju TV program, te je njihov stepen zauzetosti nizak. Na Slici 149 dat je izgled komunikacione aplikacije i prikaz situacije za režim rada *Pasivan*. U ovom slučaju osobe u prostoriji razgovaraju i gledaju TV program, i zaključeni nivo zauzetosti je umeren. Na Slici 150 dat je izgled komunikacione aplikacije i prikaz situacije za režim rada *Isključen*. U ovom slučaju osobe gledaju TV program u tišini te je zaključeni nivo zauzetosti visok.

Na Slici 151 korisnik aktivira spisak propuštenih obaveštenja, što je jedini način za interakciju sa komunikacionom aplikacijom u režimu *Isključen*.



Slika 148. Izgled komunikacione aplikacije i situacija u režimu *Aktivan*



Slika 149. Izgled komunikacione aplikacije i situacija u režimu *Pasivan*



Slika 150. Izgled komunikacione aplikacije i situacija u režimu *Isključen*



Slika 151. Izgled komunikacione aplikacije i situacija u režimu *Isključen*, sa korisnikom koji na zahtev izaziva prikaz spiska propuštenih obaveštenja

6.3 OSTALE KOMERCIJALNE PRIMENE

Na osnovu celokupnog predloženog modela ili njegovih delova realizovano je nekoliko komercijalnih kontekstualnih platformi u vidu samostalnih kontrolera, koje su od skora dostupne širem krugu korisnika, a već izvesno vreme u probnom radu. U toku tog perioda prijavljeno je nekoliko merljivih parametara koje su navedene u nastavku:

- Povećan protok ljudi u inteligentim okruženjima u kojima je instalirana kontekstualna platforma za 12-37%;
- Doprinos kontekstualne platforme iznosu prodaje proizvoda za 3%;
- U slučaju posebne instalacije u zoni za odmaranje, broj korisnika povećan za 27%, utrošeno vreme boravka;
- Četiri godine razvoja sistema gotovo bez bilo kakvih promena u modelu arhitekture, uz dodavanje brojnih proširenja.

POGLAVLJE 7.

ZAKLJUČAK

Omogućavanje da uređaji potrošačke elektronike budu svesni sopstvenog fizičkog okruženja je izuzetno težak zadatak. Njegova složenost evidentna je u velikom broju istraživanja iz oblasti kontekstualnog računarstva, automatizacije domaćinstva, različitih kontekstualnih aplikacija za uređaje potrošačke elektronike i sl. U okviru disertacije dat je jedinstven pogled na problematiku realizacije sistema, uređaja ili aplikacije sa kontekstualnim funkcijama. Data je analiza i predlog algoritama, detalji fizičke arhitekture i iscrpan predlog modela programske podrške za realizaciju kontekstualnih platformi. Predloženi model realizovan je u okviru praktičnog sistema, i u okviru eksperimentalne evaluacije pokazano je da realizovani sistem funkcioniše u skladu sa postavljenim kriterijumima, uporedivo ili bolje od dosadašnjih rešenja u oblasti.

U okviru disertacije predložen je model arhitekture kontekstualne platforme koji je primenljiv na oblast potrošačke elektronike. Primenom ovog modela, u celini ili u delovima, daje se dobra podrška realizaciji kvalitetnih, skalabilnih i proširivih kontekstualnih aplikacija. Rešenje kontekstualne platforme na osnovu predloženog modela omogućava ponovnu iskoristivost u ponovljenim upotrebama, u različitim uređajima potrošačke elektronike. Zahvaljujući predloženim mehanizmima, problemi specifičnosti ciljnog okruženja, korišćenih senzora i aktuatora ili korišćenih protokola

Zaključak

veze, rešavaju se izmenom i prilaganjem odgovarajućih programskih biblioteka bez potrebe za daljim izmenama u programskoj podršci.

U okviru disertacije predloženi su algoritmi za semantizaciju konteksta, kao i za detekciju zauzetosti korisnika na osnovu procenjenog stanja okruženja i zadatih unosa cene asinhronih zahteva u zavisnosti od stanja okruženja. Osnovni doprinos ovih algoritama je u njihovoj jednostavnosti i efikasnosti. Predložen je algoritam minimalne kompleksnosti koji omogućava dobre performanse u smislu izvršavanja na uređajima potrošačke elektronike, interakcije sa skript jezicima te moguće tačnosti postupka semantizacije. Predloženi algoritmi donose novi pogled na problem semantizacije, uvodeći koncepte P-modela koji modeluju važne aktivnosti korisnika i omogućavaju izračunavanje međuzavisnosti tih aktivnosti. Algoritmi donose bitan osvrt na problematiku semantizacije za uređaje potrošačke elektronike, koja zahteva jednostavnost, robustnost i predvidljivost. Iz tog razloga, umesto dalje generalizacije, algoritmi omogućavaju prilaganje konkretnih, eksperimentalno utvrđenih P-modela za važne aktivnosti korisnika uređaja potrošačke elektronike, poput gledanja TV programa, obavljanja telefonskog poziva i sl.

Disertacija se posebno bavi praktičnim aspektima poput upotrebe postojećih, komercijalnih senzora i aktuatora za korišćenje u okviru algoritama, uz davanje pregleda tehnologija aktuelnih u trenutku pisanja disertacije. Predloženi mehanizmi integracije sa sensorima i aktuatorima upravo su uslovljeni praktičnim ograničenjima koja donosi oprema trenutno dostupna na tržištu. U tom smislu, rešenja predložena u okviru disertacije imaju za cilj brzu i jednostavnu primenu u realizaciji konkretnih komercijalnih projekata, koji uključuju kontekstualne funkcije uređaja potrošačke elektronike.

Konačno, disertacija ukazuje na problem preopterećenosti korisnika brojnim funkcijama uređaja potrošačke elektronike koji su im danas na raspolaganju, uključujući asinhronu zahteva koje ti uređaji ispostavljaju bez obzira na korisnički kontekst. Takođe, korisnici *nolens volens* prihvataju koncepte uređaja, i zamenjuju ranije, životne koncepte, kojima se čovek anatomski i psihološki prilagođavao hiljadama godina, novim, tehnološkim konceptima čija je evolucija počela u drugoj polovini dvadesetog veka. Mišljenje autora je da ovakvo stanje ima svoju granicu, te da je pitanje vremena kada će je korisnici identifikovati i početi da odbijaju dalje

korišćenje široke palete uređaja i tehnologija. Novi izazovi, poput pažljivog upravljanja resursima korisnika, kao što su njegovo vreme i pažnja, te pružanje tehnoloških usluga u potpunosti prilagođenih kontekstu korisnika, biće presudni za uspeh budućih uređaja potrošačke elektronike. Ovaj zahtev postaće ključan već u narednih nekoliko godina, i od tada će postati nezaobilazan u dizajnu novih uređaja.

Rezultati dobijeni u okviru rada na disertaciji predstavljaju dobru osnovu za buduća istraživanja u oblasti kontekstualnih platformi za uređaje potrošačke elektronike. Najpre, od interesa je sprovođenje studije ljudskog faktora na značajnom uzorku stvarne korisničke populacije. Na osnovu tih podataka došlo bi se do zaključaka u kojim oblastima života se koncept uspešno primenjuje, a u kojima ne, te bi se omogućilo dalje unapređivanje metoda i tehnika semantizacije i kontekstualizacije okruženja. Ova studija se može sprovesti samo u saradnji sa industrijom, odnosno sa proizvođačima uređaja potrošačke elektronike. Na primer, s obzirom na sve veću rasprostranjenost Android OS, studija ljudskog faktora za korisnike mobilnih uređaja može da uključi realizaciju Android aplikacije sa kontekstualnom platformom, koja će omogućavati korisnicima da odgovaraju npr. na upite o zauzetosti i svojim aktivnostima korišćenjem grafičke korisničke sprege, dok se programska podrška kontekstualne platforme izvršava u nativnom kodu. Ovim bi se omogućilo da korisnici uključeni u studiju dobrovoljno preuzmu i koriste aplikaciju u definisanom vremenskom periodu. Studija namenjena proceni realizacije za inteligentna okruženja najpre može da upotrebi platformu u okviru okruženja sa velikim protokom i izmenljivošću ljudi, poput studentskih prostorija, muzeja, kulturnih spomenika, reklamnih štandova i sl.

Takođe, od značaja je i ispitivanje performansi predloženog rešenja na različitim uređajima potrošačke elektronike, kao i poređenje tako dobijenih rezultata. Posebni aspekti od značaja su uticaj virtuelizacije na rad kontekstualne platforme, odnosno, npr. zavisnost performanse od okruženja korišćenog za realizaciju. Jedno od ispitivanja može da uključi procenu performansi kontekstualne platforme, kao što je, npr. brzina odziva, kada se ona realizuje u jeziku Java.

Model arhitekture kontekstualne platforme može se dodatno validirati implementacijom pojedinih delova ili u celini za rešavanje različitih novih klasa problema, realizacijom novih kontekstualnih aplikacija ili kontekstualnih korisničkih

Zaključak

sprega. Ovim se omogućava procena upotrebljivosti kontekstualne platforme na osnovu studija slučaja, kada se platforma posmatra kao crna kutija koja podržava rad ciljnih aplikacija. Jedan od zanimljivih smerova za istraživanja je primena kontekstualne platforme za medijaciju socijalnih asinhronih zahteva korišćenjem svetlosnih indikatora. Potrebno je sprovesti sveobuhvatnu studiju koja bi procenila efikasnost takve medijacije u različitim inteligentnim okruženjima.

LITERATURA

- Abe, M., Morinishi, Y., Maeda, A., Aoki, M., & Inagaki, H. (2009). A Life Log Collector Integrated with a Remote-Controller for Enabling User Centric Services. *IEEE Transactions on Consumer Electronics*, 55(1), 295-302.
- Abi Research (2012). Home Automation, Security, and Monitoring. *Research Report*, 1-60, <http://www.abiresearch.com/research/1003235>.
- Abid, Z., Chabridon, S., & Conan, D. (2009). A Framework for Quality of Context Management. *Lecture Notes in Computer Science*, 5786/2009, 120-131.
- Achilleos, A., Yang, K., Georgalas, N. (2010). Context modelling and a context-aware framework for pervasive service creation: A model-driven approach. *Pervasive and Mobile Computing*, 6(2), 281-296.
- Acrossair (2012). Nearest Tube. http://www.acrossair.com/apps_nearesttube.htm.
- Adamczyk, P. D., Iqbal, S. T., & Bailey, B. P. (2005). A Method, System and Tools for Intelligent Interruption Management. *Proceedings of the 4th International Workshop on Task Models and Diagrams*, 123-126.
- Adamczyk, P. D., & Bailey, B. P. (2004). If Not Now, When?: The Effects of Interruption at Different Moments Within Task Execution. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 6(1), 271-278.
- Adams, M., Hofstede, A. H. M. Ter, Aalst, W. M. P. Van Der, & Edmond, D. (2007). Dynamic, extensible and context-aware exception handling for workflows. *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS*, 1-18.

LITERATURA

- AdSense (2012). <https://www.google.rs/adsense/>.
- Agarwal, S., Joshi, A., Finin, T., Yesha, Y., & Ganous, T. (2007). A pervasive computing system for the operating room of the future. *Mobile Networks and Applications*, 12(2–3), 215–228.
- Agre, P. E. (2001). Welcome to the Always-On World. *IEEE Spectrum*, 38(1), 10-13.
- Al-Masri, E., & Mahmoud, Q. H. (2009). SmartCon: a context-aware service discovery and selection mechanism using Artificial Neural Networks. *International Journal of Intelligent Systems Technologies and Applications*, 6(1-2), 144-156.
- Alahuhta, P., Löthman, H., Helaakoski, H., Koskela, A., & Rönning, J. (2007). Experiences in developing mobile applications using the apricot agent platform. *Personal and Ubiquitous Computing*, 11(1), 1–10.
- Alex, H., Kumar, M., & Shirazi, M. (2005). MidFusion: An adaptive middleware for information fusion in sensor network applications. *Information Fusion*, 9(3), 332–343.
- Altmann, E. M., & Trafton, J. G. (2007). Timecourse of recovery from task interruption: Data and a model. *Psychonomic Bulletin & Review*, 14(6), 1079-1084.
- Altosaar, M., Vertegaal, R., Sohn, C., & Cheng, D. (2006). AuraOrb: Social Notification Appliance. *Proceedings of the CHI '06 extended abstracts on Human factors in computing systems*, 381-386.
- Ambulant Player (2012). Ambulant Open Source. <http://www.ambulantplayer.org>.
- Anagnostopoulos, C. B., Ntarladimas, Y., & Hadjiefthymiades, S. (2007). Situational computing: An innovative architecture with imprecise reasoning. *Journal of Systems and Software*, 80(12), 1993–2014.
- Appelbaum, S. H., Marchionni, A., & Fernandez, A. (2008). The multi-tasking paradox: perceptions, problems and strategies. *Management Decision*, 46(9), 1313-1325.
- Australian Government Research and Statistics (2009). Get the Picture, Free-to-air TV, Audiences (viewing patterns). <http://www.screenaustralia.gov.au/gtp/wftvviewtimeofday.html>, pristupano u novembru 2009.
- Avrahami, D., Gergle, D., Hudson, S. E., & Kiesler, S. (2007). "Improving the match between callers and receivers: A study on the effect of contextual information on cell phone interruptions. *Behaviour & Information Technology*, 26(3), 247-259.

- Baek, S. H., Lee, H. J., Lim, S. Y., & Huh, J. D. (2005). Managing mechanism for service compatibility and interaction issues in context-aware ubiquitous home. *IEEE Transactions on Consumer Electronics*, 51(2), 524–528.
- Bailey, B. P., Adamczyk, P. D., Chang, T. Y., & Chilson, N. A. (2006). A framework for specifying and monitoring user tasks. *Computers in Human Behavior*, 22(4), 709-732.
- Bailey, B. P., & Iqbal, S. T. (2008). Understanding Changes in Mental Workload during Execution of Goal-Directed Tasks and Its Application for Interruption Management. *ACM Transactions on Computer-Human Interaction*, 14(4), 1-28.
- Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2001). The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface. *Proceedings of Interact*, 1, 593-601.
- Balarin, F. (1997). Hardware-Software Co-Design of Embedded Systems: The Polis Approach. *Springer*, 1-297.
- Balasubramaniam, S., & Indulska, J. (2004). Vertical handover supporting pervasive computing in future wireless networks. *Computer Communications*, 27(8), 708–719.
- Balzarotti, D., Costa, P., & Picco, G. P. (2007). The LightTS tuple space framework and its customization for context-aware applications. *Web Intelligence and Agent Systems*, 5(2), 215-231.
- Bao, L., & Intille, S. (2004). Activity Recognition from User-Annotated Acceleration Data. *Lecture Notes in Computer Science*, 3001-2004, 1-17.
- Bardram, J. E., & Hansen, T. R. (2004). The AWARE architecture: supporting context-mediated social awareness in mobile cooperation. *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 6(3), 192-201.
- Beach, A., Gartrell, M., Han, R., & Mishra, S. (2010). CAwbWeb: Towards a Standardized Programming Framework to Enable a Context-Aware Web. *University of Colorado, Technical Report*, 1-9.
- Begole, J., Matsakis, N. E., & Tang, J. C. (2004). Lilsys: Sensing Unavailability. *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 6(3), 511-514.
- Beigl, M., Zimmer, T., & Decker, C. (2002). A Location Model for Communicating and Processing of Context. *Personal and Ubiquitous Computing*, 6(5-6), 341-357.

LITERATURA

- Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2003a). Context-aware middleware for resource management in the wireless internet. *IEEE Transactions on Software Engineering*, 29(12), 1086–1099. 2003.
- Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2003b). Dynamic binding in mobile applications: A middleware approach. *IEEE Internet Computing*, 7(2), 34–42.
- Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2006). A mobile computing middleware for location- and context-aware internet data services. *ACM Transactions on Internet Technology*, 6(4).
- Bellotti, F., Berta, R., De Gloria, A., & Margarone, M. (2005). Implementing tour guides for travelers. *Human Factors and Ergonomics in Manufacturing*, 15(4), 461–476.
- Benedetto, M. D., & Nardis, L. D. (2006). Tuning UWB signals by pulse shaping: Towards context-aware wireless networks. *Signal Processing*, 86(9), 2172–2184.
- Bernstein, A., Vorburger, P., & Egger, P. (2007). A scenario-based approach for direct interruptability prediction on wearable devices. *International Journal of Pervasive Computing and Communications*, 3(4), 426-438.
- Bessai, K., Claudepierre, B., Saidani, O., & Nurcan, S. (2008). Context-aware Business Process Evaluation and Redesign. *9th Workshop on Business Process Modeling, Development and Support*, 1-10.
- Bhattacharyya, M., Kumar, A., & Bayomi, M. (2010). System Framework and Protocols for Ubiquitous Computing Based Monitoring of an Oil Platform. *Designing Solutions-Based Ubiquitous and Pervasive Computing: New Issues and Trends*, IGI Global, 138-157.
- Biegel, G., & Cahill, V. (2004). A framework for developing mobile, context-aware applications. *Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication*, 361–365.
- Bigdelou, A., Schwarz, L., Benz, T., & Navab, N. (2012). A flexible platform for developing context-aware 3D gesture-based interfaces. *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, 335-336.
- Bihler, P., Imhoff, P., & Cremers, A. B. (2011). SmartGuide – A Smartphone Museum Guide with Ultrasound Control. *Procedia Computer Science*, 5/2011, 586-592.

- Bjelica, M. (2008). Realizacija sistema za automatizaciju domaćinstva sa grafičkom korisničkom spregom na TV prijemniku. *Diplomski – master rad, Fakultet tehničkih nauka*, 1-110.
- Bjelica, M. Z., Teslić, N., Jovanović, Z. & Marčeta, Z. (2009a). Koncept i realizacija upravljanja domaćinstvom korišćenjem postojećih uređaja kućne elektronike. *Zbornik radova konferencije TELFOR*, 823-826.
- Bjelica, M. Z. & Teslic, N. (2009b). A Concept of System Usability Assessment: System Attentiveness as the Measure of Quality. *IEEE Eastern European Regional Conference on the Engineering of Computer Based Systems*, 144-145.
- Bjelica, M. Z. & Teslic, N. (2009c). A Concept of Usability Assessment for User-Centered Multimedia Applications. *Proceedings of International Multiconference on Computer Science and Information Technologies (IMCSIT)*, 443-450.
- Bjelica, M. Z., & Teslic, N. (2010a). Characterizing Application Attentiveness to its Users: A Method and Possible Use Cases. *International Journal of Computer Science and Applications*, 7(3), 60-83.
- Bjelica, M. Z., & Teslic, N. (2010b). Multi-purpose user awareness kit for consumer electronic devices. *Digest of Technical Papers International Conference on Consumer Electronics*, 239-240.
- Bjelica, M. Z., Papp, I., Teslic, N. & Coulon, J. –M. (2010c). Set-top box-based home controller. *Proceedings of 14th International Symposium on Consumer Electronics (ISCE)*, 1-4.
- Bjelica, M. Z., Mrazovac, B., Stefanovic, D. & Teslic, N. (2010d). Sistem za raspodeljenu reprodukciju multimedijalnih prezentacija korišćenjem web pregledača. *Zbornik radova konferencije TELFOR 2010*, 990-993.
- Bjelica, M. Z. & Teslic, N. (2010e). A Concept and Implementation of the Embeddable Home Controller. *Proceedings of 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 173-177.
- Bjelica, M. Z., Teslić, N., Jovanović, Z. & Marčeta, Z. (2010f). Home Appliances as Home Controllers: Concepts and Set-Top Box Implementation. *TELFOR Journal*, 2(2), 107-112.

LITERATURA

- Bjelica, M. Z., Pilipović, M., Kaštelan, I. & Mrazovac, B. (2010g). Sistem za procenu zauzetosti korisnika u kućnom i kancelarijskom okruženju. *Zbornik radova konferencije TELFOR 2010*, 819-822.
- Bjelica, M. Z., Mrazovac, B. & Teslic, N. (2010h). Towards ubiquitous distributed multimedia: UPnP-based control and scripting. *Proceedings of the 15th European Conference on Circuits and Systems for Communications*, 39-42.
- Bjelica, M. Z., Mrazovac, B., Papp, I. & Teslic, N. (2011a). User Availability Estimation Prototype for Home and Office Use. *IEEE International Conference on Consumer Electronics, Las Vegas, USA*, 259-260.
- Bjelica, M. Z., Zivkovic, A., Stefanovic, D., Narancic, D., & Vojnovic, V. (2011b). Home automation system behavior definition by the use of script languages. *Telecommunication forum (TELFOR)*, 920-923.
- Bjelica, M. Z., Mrazovac, B., Papp, I., & Teslic, N. (2011c). Busy flag just got better: Application of lighting effects in mediating social interruptions. *34th International Convention on Information and Communication Technology, Electronics and Microelectronics*, 975-980.
- Bjelica, M. Z., & Teslic, N. (2011d). Context-Aware Lighting as an Immersive User Interface for Mediating Social Interactions. *10th WSEAS International Conference on Applied Computer and Applied Computational Science*, 254-259.
- Bjelica, M. Z., Mrazovac, B., Duffy, R., & Brophy, L. (2011e). Scalable Multisensory System for a Customizable User Wake Up Experience. *IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 303-307.
- Bjelica, M. Z., Mrazovac, B., Papp, I. & Teslic, N. (2011f). Set-Top Box-Based Communication Client with the Automatic Operation Profile Selection. *IEEE Transactions on Consumer Electronics*, 57(3), 1433-1441.
- Bjelica, M. Z., Mrazovac, B. & Teslic, N. (2011g). Evaluation of the Available Scripting Languages for Home Automation Networks: Real World Case Study. *Proceedings of International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 611-614.
- Bjelica, M. Z., Mrazovac, B., Teslic, N., Papp, I., & Stefanovic, D. (2012). Cloud-Enabled Home Automation Gateway with the Support for UPnP Over IPv4/IPv6

and 6LoWPAN. *IEEE International Conference on Consumer Electronics, Las Vegas, USA*, 524-525.

- Bjelica, M. Z., Mrazovac, B., Papp, I. & Teslic, N. (2013). Context-Aware Platform with User Availability Estimation and Light-based Announcements. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 1-12 (rad prihvaćen za objavljivanje).
- Black, D., Clemmensen, N. J., & Skov, M. B. (2009). Supporting the supermarket shopping experience through a context-aware shopping trolley. *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group*, 1-4.
- Blake, M., Kahan, D., & Nowlan, M. (2007). Context-aware agents for user-oriented web services discovery and execution. *Distributed and Parallel Databases*, 21(1), 39–58.
- Bolchini, C., Schreiber, F. & Tanca, L. (2007). A methodology for a very small database design. *Information Systems*, 32(1), 61–82.
- Borderless (2008). The Borderless Communicator – IP communication and mobile computing – Borderless Blog – Journal of Cyber Kinetic IP news – January 20, 2008 – What percentage of Skype users are online all the time? <http://idisk.mac.com/hhbv-Public/Blogs/88743277.html>.
- Bottaro, A., & Gerodolle, A. (2008). Home SOA - facing protocol heterogeneity in pervasive applications. *Proceedings of the 5th international conference on Pervasive services*, 1-8.
- Bottazzi, D., Corradi, A., & Montanari, R. (2006). Context-aware middleware solutions for anytime and anywhere emergency assistance to elderly people. *Communications Magazine, IEEE*, 44(4), 82–90.
- Boukerche, A., Pazzi, R. W. N., & Araujo, R. V. (2006). Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4), 586–599.
- Bouvin, N. O., Christensen, B., Frank, K. G., & Hansen, A. (2003). HyCon: A framework for context-aware mobile hypermedia. *New Review of Hypermedia and Multimedia*, 9, 59–88.

LITERATURA

- Brilingaite, A., & Jensen, C. S. (2007). Enabling routes of road network constrained movements as mobile service context. *GeoInformatica*, 11(1), 55–102.
- Brunato, M., & Battiti, R. (2005). Statistical learning theory for location fingerprinting in wireless LANs. *Computer Networks*, 47(6), 825–845.
- Bucchiarone, A., Marconi, A., Pistore, M., & Sirbu, A. (2011). A Context-Aware Framework for Business Processes Evolution. *IEEE International Enterprise Distributed Object Computing Conference Workshops*, 146-154.
- Burstein, F., Zaslavsky, A., & Arora, N. (2005). Context-aware mobile agents for decision-making support in healthcare emergency applications. *International Workshop on Context Modeling and Decision Support*, 1-16.
- Butter, T., Aleksy, M., Bostan, P., & Schader, M. (2007). Context-aware User Interface Framework for Mobile Applications. *27th International Conference on Distributed Computing Systems Workshops*, 39-44.
- Butterfield, E. C. (1964). The interruption of tasks: methodical, factual and theoretical issues. *Psychological Bulletin*. 62(5), 309-322.
- Byun, H. E., & Cheverst, K. (2004). Utilizing context history to provide dynamic adaptations. *Applied Artificial Intelligence*, 18(6), 533–548.
- C4Central. (2011). *New lifestyle applications available via the Control4 4Store*. <http://c4central.com/2011/05/12/new-lifestyle-applications-available-via-the-control4-4store/>.
- Cabri, G., Leonardi, L., Mamei, M., & Zambonelli, F. (2003). Location-dependent services for mobile users. *IEEE Transactions on Systems*, 33(6), 667–681.
- Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3), 289–308.
- Canalys (2012). Smart phones overtake client PCs in 2011 - Vendors shipped 488 million smart phones in 2011, compared to 415 million client PCs. <http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011>.
- Cano, J. -C., Manzoni, P., & Toh, C. -K. (2006). UbiqMuseum: A Bluetooth and Java Based Context-Aware System for Ubiquitous Computing. *Wireless Personal Communications*. 38(2), 187-202.

- Capra, L., Emmerich, W., & Mascolo, C. (2003). CARISMA: Context-aware reflective middleware system for mobile applications. *IEEE Transactions on Software Engineering*, 29(10), 929–945.
- CarFinder (2011). Find Your Car with AR: Augmented Car Finder. <http://itunes.apple.com/rs/app/id370836023>.
- Carrol, A., & Heiser, G. (2010). An analysis of power consumption in a smartphone. *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, 1-14.
- Carton, A. M., & Aiello, J. R. (2009). Control and Anticipation of Social Interruptions: Reduced Stress and Improved Task Performance, *Journal of Applied Social Psychology*, 39(1), 169-185.
- Cashmore, P. (2009). Augmented Reality Twitter is the Coolest Thing Ever. *Mashable Social Media*, <http://mashable.com/2009/08/04/augmented-reality-twitter/>.
- Ceri, S., Daniel, F., Facca, M. F., & Matera, M. (2007). Model-driven engineering of active context-awareness. *World Wide Web*, 10(4), 387–413.
- CES (2008). Consumer Electronics Innovation: Improving Time-to-Market through Better Alignment with Semiconductor Suppliers. *Panel Discussion, Consumer Electronics Show, Las Vegas, USA*.
- Chan, A. T. S., & Chuang, S. N. (2003). MobiPADS: A reflective middleware for context-aware mobile computing. *IEEE Transactions on Software Engineering*, 29(12), 1072–1085.
- Chan, A. T. S., Chuang, S. N., Cao, J., & Leong, H. V. (2004). An event-driven middleware for mobile context awareness. *The Computer Journal*, 47(3), 278–288.
- Chan, A. B., Liang, Z.-S.J., Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. *IEEE Conference on Computer Vision and Pattern Recognition 2008*, 1-7.
- Chang, K. -H., Chen, M. Y., & Canny, J. (2007). Tracking free-weight exercises. *Proceedings of the 9th international conference on Ubiquitous computing*, 1-19.
- Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, Cambridge University Press, 18, 197–207.

LITERATURA

- Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., & Chakraborty, D. (2004). Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Computing*, 8(6), 69–79.
- Chen, H., & Mohapatra, P. (2005). A context-aware HTML/XML document transmission process for mobile wireless clients. *World Wide Web*, 8(4), 439–461.
- Chen, C. -M., & Li Y. -L. (2010). Personalised context-aware ubiquitous learning system for supporting effective English vocabulary learning. *Interactive Learning Environments*, 18(4), 341-364.
- Cheverst, K., Byun, H. E., Fitton, D., Sas, C., Kray, C., & Villar, N. (2005). Exploring Issues of User Model Transparency and Proactive Behaviour in an Office Environment Control System. *User Modeling and User-Adapted Interaction*, 15(3-4), 235-273.
- Choi, J. H., Shin, D. K., & Shin, D. I. (2005). Research and implementation of the context-aware middleware for controlling home appliances. *IEEE Transactions on Consumer Electronics*, 51(1), 301–306.
- Control4 (2012). Control4 Home Automation and Smart Home Control. <http://www.control4.com/>.
- CryDev (2010). CryEngine Lua Utilities, CryWiki Crysis community forum, http://www.crydev.net/wiki/index.php/CryEngine_Lua_Uilities.
- Czerwinski, M., Horvitz, E., Wilhite, S. (2004). A diary study of task switching and interruptions. *Proceedings of the SIGCHI conference on Human factors in computing systems*.6, 175-182.
- Cube Smart (2002). Social Interruption and the Loss of Productivity. *Cube Smart Inc*, 1-8.
- Curino, C., Giani, M., Giorgetta, M., Giusti, A., Murphy, A. L., & Picco, G. P. (2005). Mobile data collection in sensor networks: The TinyLime middleware. *Pervasive and Mobile Computing*, 1(4), 46–469.
- Danninger, M., & Stiefelhagen, R. (2008). A context-aware virtual secretary in a smart office environment. *Proceedings of the 16th ACM international conference on Multimedia*, 1-10.

- David, L., Endler, M., Barbosa, S. D. J., & Filho, J. V. (2011). Middleware Support for Context-Aware Mobile Applications with Adaptive Multimodal User Interfaces. *Fourth International Conference on Ubi-Media Computing*, 1-6.
- Davidow, W. H. (2012). Our Tools Are Using Us. *IEEE Spectrum*, 49(8), 45-48 (INT), August 2012.
- De Guzman, E. S., Sharmin, M., & Bailey, B. P. (2007). Should I Call Now? Understanding What Context is Considered When Deciding Whether to Initiate Remote Communication via Mobile Devices. *Proceedings of Graphics Interface Conference*, 143-150.
- Dey, A. K. (2000). Providing Architectural Support for Building Context-Aware Applications. *Doctoral thesis, Georgia Institute of Technology, USA*, 1-188.
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4-7.
- Dey, A. K., Abowd, G. D., & Salber, D. (2001). Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, 16, 97-166.
- Dey, A. K., & Newberger, A. (2009). Support for context-aware intelligibility and control. *Proceedings of the 27th international conference on Human factors in computing systems*.
- Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., & Herlocker, J. L. (2005). TaskTracer: A desktop environment to support multi-tasking knowledge workers. *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*, 75-82.
- Dumas, B., Lalanne, D., & Oviatt, S. (2009). Multimodal Interfaces: A Survey of Principles, Models and Frameworks. *Lecture Notes in Computer Science*, 5440/2009, 3-26.
- EC Directive 2002/95/EC (2003). Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment. *Official Journal of the European Union*, 37, 19-23.
- EC Regulation 1275 (2008). Commission Regulation (EC) No 1275/2008 of 17 December 2008 implementing Directive 2005/32/EC of the European Parliament

LITERATURA

- and of the Council with regard to ecodesign requirements for standby and off mode electric power consumption of electrical and electronic household and office equipment. *Official Journal of the European Union*, 339, 45-52.
- Elliot, A., Maier, M., Moller, A., Friedman, R., & Meinhardt, J. (2007). Color and Psychological Functioning: The Effect of Red on Performance Attainment. *Journal of Experimental Psychology*, 136(1), 154-168.
- Emmerich, P. (2009). Beginning Lua with World of Warcraft Add-ons, *Apress*.
- EN 55014-2. (2009). Electromagnetic compatibility - Requirements for household appliances, electric tools and similar apparatus - Part 2: Immunity - Product family standard. *International Electrotechnical Commission (IEC)*.
- Executive Order 13221 (2001). Energy Efficient Standby Power Devices. *White House, USA*.
- Fahy, P., & Clarke, S. (2004). CASS – a middleware for mobile context-aware applications. *Workshop on Context Awareness, MobiSys 2004*.
- Favela, J., Tentori, M., Castro, L. A., Gonzalez, V. M., Moran, E. V., & Martínez-García, A. I. (2007). Activity recognition for context-aware hospital applications: Issues and opportunities for the deployment of pervasive networks. *Mobile Networks and Applications*, 12(2–3), 155–171.
- Feng, Y., -H., Teng, T. -H., & Tan, A. -H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1), 455-463.
- Fleuret, F., Berclaz, J., Lengagne, R., Fua, P. (2008). Multicamera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 267-282.
- Fogarty, J., & Hudson, S. E. (2007). Toolkit support for developing and deploying sensor-based statistical models of human situations. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 135-144.
- Fogarty, J., Hudson, S. E., Atkinson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., & Yang, J. (2005). Predicting Human Interruptibility with Sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1), 119-146.
- Fogarty, J., Lai, J., & Christensen, J. (2004). Presence versus availability: The design and evaluation of a context-aware communication client. *International Journal of Human-Computer Studies*, 61(3), 299–317.

- Franklin, D. (2001). The Intelligent Classroom: Competent Assistance in the Physical World. *PhD Thesis, Northwestern University, Evanston, Illinois, USA*, 1-199.
- Friday, A., Wu, M., Finney, J., Schmid, S., Cheverst, K., & Davies, N. (2003). Network layer access control for context-aware IPv6 applications. *Wireless Networks*, 9(4), 299–309.
- Fu, L. (2011). Design of QR code-based Mall shopping guide system. *International Conference on Information Science and Technology*, 450-453.
- Gaia Project (2005). <http://gaia.cs.uiuc.edu/>.
- Gales, M., & Young, S. (2008). The application of hidden Markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3), 195-304.
- Gandon, F. L., & Sadeh, N. M. (2004). Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3), 241–260.
- Garrett, R. K., & Danzinger, J. N. (2008). IM = Interruption Management? Instant Messaging and Disruption in the Workplace. *Journal of Computer-Mediated Communication*. 13(1), 23-42.
- Gellersen, H. W., Schmidt, A., & Beigl, M. (2002). Multi-sensor context-awareness in mobile devices and smart artefacts. *Mobile Networks and Applications*, 7(3), 341–351.
- Giaffreda, F., de Carvalho, R., Melia, T., Giaffreda, F., & de Carvalho, T. (2007). Media Network enablers for enhanced mobility management. *BT Technology Journal*, 25(2), 95–105.
- Gievska, S., Lindeman, R., & Sibert, J. (2005). "Examining the Qualitative Gains of Mediating Human Interruptions during HCI. *Proceedings of 11th International Conference on Human-Computer Interaction*, 605-614, 2005.
- Ghanam, Y., Shouman, M., & Greenberg, S. (2009). Object-Specific Interfaces in Smart Homes. *University of Calgary, Technical report*, 1-9.
- Gonzalez-Castano, F. J., Garcia-Reinoso, J., Gil-Castineira, F., Costa-Montenegro, E., & Pousada-Carballo, J. M. (2005). Bluetooth-assisted context-awareness in educational data networks. *Computers Education*, 45(1), 105–121.
- Google Search (2009). <http://googleblog.blogspot.com/2009/12/personalized-search-for-everyone.html>.

LITERATURA

- GPL FAQ (2012). Frequently Asked Questions about the GNU Licenses. <http://www.gnu.org/licenses/gpl-faq.html#MereAggregation>.
- Gu, T., Pung, H. K., & Zhang, D. Q. (2004). Toward an OSGi-based infrastructure for context-aware applications. *IEEE Pervasive Computing*, 3(4), 66–74.
- Gu, T., Pung, H. K., Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1), 1-18.
- Guang-yao J., Xiao-yi L., Myoung-Soon P. (2006). An indoor localization mechanism using active RFID tag. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 1-4.
- Guha, S., Daswani, N., & Ravi, J. (2006). An Experimental Study of the Skype Peer-to-Peer VoIP System. *Proceedings of 5th International Workshop on Peer-to-Peer Systems (IPTPS), Santa Barbara, California, USA*, 1-6.
- Guillemin, A. (2003). Using Genetic Algorithms to Take Into Account User Wishes in an Advanced Building Control System. *PhD thesis, Ecole Polytechnique Federale de Lausanne*, 1-185.
- Guinness World Records (2011). Kinect Confirmed As Fastest-Selling Consumer Electronics Device, http://community.guinnessworldrecords.com/_Kinect-Confirmed-As-Fastest-Selling-Consumer-Electronics-Device/blog/3376939/7691.html.
- Halstead, M. H. (1977). Elements of Software Science. *Elsevier, North-Holland*, New York.
- Hameed, S., Ferris, T. Jayaraman, S., & Sarter, N. (2009). Using Informative Peripheral Visual and Tactile Cues to Support Task and Interruption Management. *Human Factors*, 51(2), 126-135.
- Harman, W. S. (2006). Interruptions in the Goal Striving Process. *Doctoral Dissertation, University of Washington*, 1-160.
- Harold, E. R., & Means, W. S. (2004). XML in a Nutshell Third Edition. *O'Reilly Media*.
- Hatala, M., & Wakkary, R. (2005). Ontology-based user modeling in an augmented audio reality system for museums. *User Modeling and User-Adapted Interaction*, 15(3–4), 339–380.

- Hazzan, O., & Dubinsky, Y. (2008). Agile Software Engineering, *Undergraduate Topics in Computer Science, Springer*, 1-303.
- Hill, R., Al-Muhtadi, J., Campbell, R., Kapadia, A., Naldurg, P., & Ranganathan, A. (2004). A middleware architecture for securing ubiquitous computing cyber infrastructures. *IEEE Distributed Systems Online*, 5(9).
- Ho, J., & Intille, S. (2005). Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. *Proceedings of the SIGCHI conference on Human factors in computing system*, 909-918.
- Hodgetts, H. M., & Jones, D. M. (2007). Reminders, Alerts and Pop-ups: The Cost of Computer-Initiated Interruptions. *Lecture Notes in Computer Science*, 4550, 818-826.
- Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., & Altmann, J. (2002). Context-awareness on mobile devices – the hydrogen approach. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 292–302.
- Hollrback (2011). Hollrback by Hollrback. <http://itunes.apple.com/us/app/hollrback/id411757911?mt=8>.
- Holte, M. B., Moeslund, T. B., Fihl, P. (2010). View-invariant gesture recognition using 3D optical flow and harmonic motion context. *Computer Vision and Image Understanding*, 114(12), 1353-1361.
- Hong, D., Dickson, K. W., Chiu, V. Y., Shen, S. C. C., & Kafeza, E. (2007). Ubiquitous enterprise service adaptations based on contextual user behavior. *Information Systems Frontiers*, 9(4), 343–358.
- Hong, J., Suh, E., & Kim, S. -J. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36, 8509-8522.
- Hong, J. -H., Yang, S. -I., & Cho, S. -B. (2010). ConaMSN: A context-aware messenger using dynamic Bayesian networks with wearable sensors. *Expert Systems with Applications*, 37(6), 4680-4686.
- Horn, M. S., & Jacob, R. J. K. (2007). Tangible programming in the classroom with tern. *CHI '07: extended abstracts on Human factors in computing systems*, 1-6.
- Horvitz, E., Koch, P., & Apacible, J. (2004). BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption. *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 6(3), 507-510.

LITERATURA

- Hoschka, P. (1998). An Introduction to the Synchronized Multimedia Integration Language. *IEEE Multimedia*, 5(4), 84-88.
- Hu, J., & Feijs, L. (2006). IPML: Extending SMIL for Distributed Multimedia Presentations. *Lecture Notes in Computer Science*, 4270, 60-70.
- Huang, L. -Y., & Hsieh, Y. -J. (2011). Consumer electronics acceptance based on innovation attributes and switching costs: The case of e-book readers. *Electronic Commerce Research and Applications*, online version.
- Huang, P. -C., Chien, C. -Y., Kuo, Y. -H., & Wu, M. -Z. (2009). An integration platform for developing context-aware applications based on context reasoning and service automation techniques. *International Conference on Advanced Communication Technology*, 1819-1823.
- Hussmann, S., Ringbeck, T., & Hagebeucker, B. (2008). A Performance Review of 3D TOF Vision Systems in Comparison to Stereo Vision Systems. *Stereo Vision, Intech*, book chapter, 103-120.
- IDC (2011). Media Tablet and eReader Markets Beat Second Quarter Targets, Forecast Increased for 2011, According to IDC. *IDC – Press Release*, <http://www.idc.com/getdoc.jsp?containerId=prUS23034011>.
- Ierusalimschy, R., Figueiredo, L. H. de, & Filho, W. C. (1996). LUA – an extensible extension language, *Software – Practice & Experience*, 26(6), 635-652.
- Indulska, J., & Sutton, P. (2003). Location management in pervasive systems. *CRPITS'03: Proceedings of the Australasian Information Security Workshop*, 143–151.
- Intelligent Profile (2011). Intelligent Profile Lite – Tomáš Novák. <https://play.google.com/store/apps/details?id=eu.novintech.app.intelligentprofile>.
- Intille, S. S. (2002). Designing a home of the future. *IEEE Pervasive Computing*, 1(2), 76–82.
- Iqbal, S. T., & Bailey, B. P. (2007). Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 697-706.
- Jo, H., Kim, H., Roh, H. -G. & Lee, J. (2009). Improving the Startup Time of Digital TV. *IEEE Transactions on Consumer Electronics*, 55(2), 721-727.

- JP Morgan (2011). Big jump to 99m tablet sales worldwide forecast for 2012. *TabTimes*, <http://tabtimes.com/news/ittech-stats-research/2011/12/16/big-jump-99m-tablet-sales-worldwide-forecast-2012>.
- Julien, C., & Roman, G. C. (2006). EgoSpaces: Facilitating rapid development of context-aware mobile applications. *IEEE Transactions on Software Engineering*, 32(5), 281–298
- Jung, J. Y., & Lee, J. W. (2007). ZigBee Device Design and Implementation for Context-Aware U-Healthcare System. *Second International Conference on Systems and Networks Communications*, 2007.
- Jutla, D. N., Bodorik, P., & Zhang, Y. (2006). PeCAN: An architecture for users' privacy-aware electronic commerce contexts on the semantic web. *Information Systems*, 31(4–5), 295–320.
- Kahl, G., Spassova, L., Schöning, J., Gehring, S., & Krüger, A. (2011). IRL SmartCart - a user-adaptive context-aware interface for shopping assistance. *Proceedings of the 16th international conference on Intelligent user interfaces, Palo Alto, CA, USA*.
- Kao, T. H., & Yuan, S. M. (2004). Designing an XML-based context-aware transformation framework for mobile execution environments using CC/and XSLT. *Computer Standards Interfaces*, 26(5), 377–399.
- Kasteren, T. van, Noulas, A., Englebienne, G., Kröse, B. (2008). Accurate activity recognition in a home setting. *Proceedings of the 10th international conference on Ubiquitous computing*, 1-9.
- Kellerer, W., Wagner, M., Balke, W. T., & Schulzrinne, H. (2004). Preference-based session management for IP-based mobile multimedia signaling. *European Transactions on Telecommunications*, 15(4), 415–427.
- Khedr, M., & Karmouch, A. (2004). Negotiating context information in context-aware systems. *IEEE Intelligent Systems*, 19(6), 21–29.
- Khedr, M., Karmouch, A. (2005). ACAI: agent-based context-aware infrastructure for spontaneous applications. *Journal of Network and Computer Applications*, 28(1), 19-44.
- Kirsh, D. (2000). A Few Thoughts on Cognitive Overload. *Intellectica*, 1(30), 19-51.
- Knez, I. (1997). Changes in Females' and Males' Positive and Negative Moods As a Result of Variations in CCT, CRI and Illuminance Levels. *Right Light*, 4, 149-154.

LITERATURA

- Knight, J. F., Bristow, H. W., Anastopoulou, S., Baber, C., Schwirtz, A., & Arvanitis, T. N. (2007). Uses of accelerometer data collected from a wearable system. *Personal and Ubiquitous Computing*, 11(2), 117–132.
- Kooij, R., Ahmed, K., & Brunnstrom, K. (2006). Perceived Quality of Channel Zapping. *Proceedings of the Fifth IASTED International Conference on Communication Systems and Networks*, 155-158.
- Korhonen, J., Ojala, T., Ristola, A., Kesti, M., Kilpelänaho, V., Koskinen, M., et al. (2007). Mobile fair diary: Hybrid interface for taking, browsing and sharing context-aware notes. *Personal and Ubiquitous Computing*, 11(7), 577–589.
- Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., & Malm, E. J. (2003a). Managing context information in mobile devices. *IEEE Pervasive Computing*, 2(3), 42–51.
- Korpipää, P., & Mäntyjärvi, J. (2003b) An ontology for mobile device sensor-based context awareness. *Lecture Notes in Computer Science*, 2680/2003, 451–458.
- Košutić, N. (2011). Realizacija biblioteke za prepoznavanje korisničkih gestova korišćenjem akcelerometra zasnovano na Skrivenim Modelima Markova. *Završni (bachelor) rad, Fakultet tehničkih nauka, Univerzitet u Novom Sadu*, 1-33.
- KPMG (2008). The Consumer Electronics Boom. *KPMG's Information, Communication & Entertainment (ICE) study*, 1-28.
- Kuan-Ta, C., Chun-Ying, H., Huang, P., & Chin-Laung, L. (2006). Quantifying Skype user satisfaction, *Proceedings of the 2006 conference on Application, technologies, architectures and protocols for computer communications*, 399-410.
- Kusznir, J. & Cook, D. J. (2010). Designing Lightweight Software Architectures for Smart Environments. *Sixth International Conference on Intelligent Environments*, 220-224.
- Kwon, O. B. (2003). “I know what you need to buy”: context-aware multimedia-based recommendation system. *Expert Systems with Applications*, 25(3), 387-400.
- Kwon, O. B. (2004). Modeling and generating context-aware agent-based applications with amended colored petri nets. *Expert Systems with Applications*, 27(4), 609–621.
- Kwon, O. B., Choi, S. C., & Park, G. R. (2005a). NAMA: A context-aware multi agent based web service approach to proactive need identification for personalized reminder systems. *Expert Systems with Applications*, 29(1), 17–32.

- Kwon, O. B., Yoo, K. D., & Suh, E. H. (2005b). UbiDSS: A proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. *Expert Systems with Applications*, 28(1), 149–161.
- Ladd, A. M., Bekris, K. E., Rudys, A., Kavraki, L. E., & Wallach, D. S. (2005). Robotics based location sensing using wireless ethernet. *Wireless Networks*, 11(1–2), 189–204.
- Lakobrija, R., Smiljković, N., Papp, I., Kukolj, D. & Savić, M. (2007). Jedno rešenje proširenja TV uređaja Skype aplikacijom. *Zbornik radova konferencije TELFOR*, 549-552.
- Lieberman, H., & Chu, A. (2007). An interface for mutual disambiguation of recognition errors in a multimodal navigational assistant. *Multimedia Systems*, 12(4–5), 393–402.
- Lin, P., Tsai, C. Y., & Lai, Y. C. (2005). A SIP-based mobility management platform for WLAN location-aware broadcasting and multicasting application. *Wireless Communications and Mobile Computing*, 5, 647–663.
- Linden, G., Smith, B. & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76-80.
- Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V. (2009). uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6), 657-675.
- Liu, S., & Cheng, L. (2011). A context-aware reflective middleware framework for distributed real-time and embedded systems. *Journal of Systems and Software*, 84(2), 205-218.
- Liu, W. (2008). Focusing on Desirability: The Effect of Decision Interruption and Suspension of Preferences. *Journal of Consumer Research*, 35, 640-652.
- Liukonnen, T. (2009). Human Reaction Times as a Response to Delays in Control Systems – Notes in Vehicular Context, *Department of Information Processing Science, University of Oulu*, 1-13.
- Lopes, F., Delicato, F. C., Batista, T., & Pires, P. F. (2009). Context-based heterogeneous middleware integration. *Proceedings of the 2009 Workshop on Middleware for Ubiquitous and Pervasive Systems*, 1-7.
- Lua Download (2012). Lua: download. <http://www.lua.org/download.html>.

LITERATURA

- Magableh, B., & Barrett, S. (2011). Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2003b). Dynamic binding in mobile applications: A middleware approach. *IEEE Internet Computing*, 7(2), 34–42.
- Maric, S., Bjelica, M. Z., Mrazovac, B. & Teslic, N. (2010). Realizacija upravljanja raspodeljenim multimedijalnim sistemom korišćenjem skriptova. *Zbornik radova konferencije ETRAN*, 1-4.
- Malandrino, D., Mazzoni, F., Riboni, D., Bettini, C., Michele, C. & Scarano, V. (2010). MIMOSA: context-aware adaptation for ubiquitous web access. *Personal and Ubiquitous Computing*, 14(4), 301-320.
- Mark, G., Gudith, D., & Klocke, U. (2008). The Cost of Interrupted Work: More Speed and Stress. *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 107-110.
- Markus, L. & Keil, M. (1994). If we build it, they will come: Designing information systems that people want to use, *MIT Sloan Management Review*, 35(4), 11-25.
- Matthews, D., Chalmers, D., & Wakeman, I. (2011). MediateSpace: decentralised contextual mediation using tuple spaces. *Proceedings of the Third International Workshop on Middleware for Pervasive Mobile and Embedded Computing*, 1-6.
- McCrickard, D. S., Chewar, C. M., Somervell, J. P., & Ndiwalana, A. (2003). “A Model for Notification Systems Evaluation – Assessing User Goals for Multitasking Activity. *ACM Transactions on Computer-Human Interaction*, 10(4), 312-338.
- McFarlane, D. C. (2002). The Scope and Importance of Human Interruption in Human-Computer Interaction Design. *Human-Computer Interaction*, 17(1), 1-61.
- McKeever, S., Ye, J., Coyle, L., Dobson, S. (2009). A Context Quality Model to Support Transparent Reasoning with Uncertain Context. *1st International Workshop on Quality of Context (QuaCon), Stuttgart, Germany*, 1-13.
- McNair, J., & Zhu, F. (2004). Vertical handoffs in fourth-generation multinet network environments. *IEEE Wireless Communications*, 11(3), 8–15.
- Meier, A., & Lebot, B. (1999). One Watt Initiative : a Global Effort to Reduce Leaking Electricity. *Lawrence Berkeley National Laboratory*, 1-7.
- Meng, D., & Poslad, S. (2009). A Reflective Goal-Based System for Context-Aware Adaptation. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 12(3), 105-110.

- Milewski, A. E., & Smith, T. M. (2000). Providing Presence Cues to Telephone Users. *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 89-96.
- Miller, R. B. (1968). Response time in man-computer conversational transactions. *Proceedings of the Spring Joint Computer Conference, AFIPS Press*, 33, 267-277.
- Miori, V., Tarrini, L., Manca, M., & Tolomei, G. (2006). An open standard solution for domotic interoperability. *IEEE Transactions on Consumer Electronics*, 52(1), 97-103.
- MisterHouse (2008). MisterHouse – It Knows Kung-Fu. <http://misterhouse.sourceforge.net/>.
- Monaci, G., Gritti, T., Vignoli, F., Walmink, W., & Hendriks, M. (2011). Flower Power. *Proceedings of the 19th ACM International conference on Multimedia*, New York, 909-912.
- Mrazovac, B., Bjelica, M. Z., Kukolj, D., Todorovic, B. M., Vukosavljev, S. (2012). Human Presence Detection Using Radio Irregularity in Wireless Networks. *SENSORNETS 2012, Rome, Italy*, 920-923.
- Mrazovac, B., Bjelica, M. Z., Simic, Dj., Tikvic, S., & Papp, I. (2011). Gesture Based Hardware Interface for RF Lighting Control. *9th IEEE International Symposium on Intelligent Systems and Informatics*, 309-314.
- Mrazovac, B., Bjelica, M. Z., Teslic, N., & Papp, I. (2011). Towards Ubiquitous Smart Outlets for Safety and Energetic Efficiency of Home Electric Appliances. *IEEE International Conference on Consumer Electronics – Berlin*, 324-328.
- Munarriz, R. A. (2012). BlackBerry Loses to Apple Again. *Aol Original, Daily Finance*, <http://www.dailyfinance.com/2012/02/09/blackberry-loses-to-apple-again/>
- Murphy, A. L., & Picco, G. P. (2004). Using coordination middleware for location-aware computing: A lime case study. *Lecture Notes in Computer Science*, 263–278.
- Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D. L., Morley, D., Pfeffer, A., Pollack, M., & Tambe, M. (2007). An intelligent personal assistant for task and time management. *AI Magazine*, 28(2), 47-61.
- Mylyn (2012). <http://www.eclipse.org/mylyn/>.

LITERATURA

- Neisse, R., Wegdam, M., & Sinderen, M. van. (2008). Trustworthiness and Quality of Context Information. *The 9th International Conference for Young Computer Scientists*, 1925-1931.
- Nejati, S., Di Alesio, S., Sabetzadeh, M., & Briand, L. (2012). Modeling and Analysis of CPU Usage in Safety-Critical Embedded Systems to Support Stress Testing, *ACM/IEEE 15th International Conference on Model Driven Engineering Languages & Systems*.
- Nokia Situations (2011). <http://betalabs.nokia.com/apps/nokia-situations>.
- Norrie, M. C., Signer, B., Grossniklaus, M., Belotti, R., Decurtins, C., & Weibel, N. (2007). Context-aware platform for mobile data management. *Wireless Networks*, 13(6), 855–870.
- O’Grady, M. J., O’Hare, G. M. P., & Sas, C. (2005). Mobile agents for mobile tourists: A user evaluation of Gulliver’s genie. *Interacting with Computers*, 17(4), 343–366.
- O’Hare, G. M. P., & O’Grady, M. J. (2003). Gulliver’s genie: A multi-agent system for ubiquitous and intelligent content delivery. *Computer Communications*, 26(11), 1177–1187.
- OpenRemote (2012). Open Remote – Home of the Digital Home. <http://openremote.org>.
- Orland, K. (2011). Microsoft Announces Windows Kinect SDK for Spring Release. http://www.gamasutra.com/view/news/33136/Microsoft_Announces_Windows_Kinect_SDK_For_Spring_Release.php.
- Ouden, E. D. (2006). Development of a Design Analysis Model for Consumer Complaints: Revealing a New Class of Quality Failures. *Eindhoven University of Technology, Ph.D. dissertation*, 1-177.
- Papakos, P., Capra, L., & Rosenblum, D. S. (2010). VOLARE: context-aware adaptive cloud service discovery for mobile systems. *Proceedings of the 9th International Workshop on Adaptive and Reflective Middleware*, 32-28.
- Papp, I., Saric, Z., Jovicic, S., Teslic, N. (2007). Adaptive microphone array for unknown desired speaker’s transfer function. *Journal of the Acoustical Society of America*, 122(2), 1-6.
- Parent’s Choice (2011). Kinect for Xbox 360. http://www.parents-choice.org/product.cfm?product_id=28940.

- Plaza, E., Arcos J. L. (2001). Context Aware Agents for Personal Information Services. *Lecture Notes in Computer Science*, 2182/2001, 44-55.
- ProfileRules (2011). Profile Rules – Instituto Eldorado. <https://play.google.com/store/apps/details?id=org.eldorado.android.profilemanager>.
- Rademacher, G. (2012). Railroad Diagram Generator v1.21.383 April 8, 2012. <http://railroad.my28msec.com/rr/ui>.
- Ranganathan, A., Al-Muhtadi, J., & Campbell, R. H. (2004). Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3(2), 62–70.
- Ranganathan, A., Campbell, R. H., Ravi, A., & Mahajan, A. (2002). Conchat: A context-aware chat program. *IEEE Pervasive Computing*, 1(3), 51–57.
- Rehman, K., Stajano, F., & Coulouris, G. (2007). An architecture for interactive context-aware applications. *IEEE Pervasive Computing*, 6(1), 73–80.
- Renaud, K., Ramsay, J., Hair, M. (2008). Think Before You Link: Controlling Ubiquitous Availability. *Interactions*. 15(3), 65-68.
- Research and Markets (2010). How Innovation and Pricing Will Shape Consumer Electronics' Future Growth, <http://www.euromonitor.com/how-innovation-and-pricing-will-shape-consumer-electronics-future-growth/report>.
- Riekkki, J., Huhtinen, J., Ala-Siuru, P., Alahuhta, P., Kaartinen, J., & Röning, J. (2003). Genie of the net, an agent platform for managing services on behalf of the user. *Computer Communications*, 26(11), 1188–1198.
- Riva, O., & Toivonen, S. (2007). The DYNAMOS approach to support context-aware service provisioning in mobile environments. *Journal of Systems and Software*, 80(12), 1956–1972.
- Rocha, R. C. A., & Endler, M. (2006). Middleware: Context management in heterogeneous, evolving ubiquitous environments. *Distributed Systems Online*, IEEE, 7(4), 1.
- Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., & Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE Pervasive Computing*, 1(4), 74–83.
- Russel, S. J., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach (3rd edition). *Prentice Hall*.

LITERATURA

- Ryzko, D., & Henryk R. (2010). Distributed Default Logic for Context-Aware Computing in Multi-Agent Systems. *Lecture Notes in Computer Science*, 6086/2012, 476-483.
- Sajal, K. D., Nirmalya, R., & Abhishek, R. (2006). Context-aware resource management in multi-inhabitant smart homes: A framework based on Nash H-learning. *Pervasive and Mobile Computing*, 2(4), 372–404.
- Samaan, N., & Karmouch, A. (2005). A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. *IEEE Transactions on Mobile Computing*, 4(6), 537–551.
- Satoh, I. (2007). A Spatial Communication Model for Ubiquitous Computing Services. *Fourth European Conference on Universal Multiservice Networks*, 32-44.
- Satoh, I. (2010). Mobile Agent-based Context-aware Services. *Journal of Universal Computer Science*, 16(15), 1929-1952.
- Schilit, B. N., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. *First Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, USA, 85-90.
- Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Laerhoven, K. V., & Velde, Walter V. de (1999). Advanced Interaction in Context. *Lecture Notes in Computer Science*, 1707-1999, 89-101.
- Schmidt, A., Takaluoma, A., & Mantyjarvi, J. (2000). Context-aware telephony over WAP. *Personal Technologies*, 4(4), 225–229.
- Schroeder, S. (2009). Browse the Reality on Your Android Phone With Layar. *Mashable Social Media*, <http://mashable.com/2009/08/18/layar/>.
- Schulzrinne, H., Wu, X., Sidiroglou, S., & Berger, S. (2003). Ubiquitous computing in home networks. *IEEE Communications Magazine*, 41(11), 128–135.
- Schwaber, K. (2009). Agile Project Management with Scrum. *Best Practices, Microsoft Professional*, 1-192.
- Selburn, J. (2011). Consumer Platforms Q3 2011 Market Tracker. *IHS iSuppli*, http://www.isuppli.com/Abstract/P13853_20110809220629.pdf.
- Shaer, O., & Jacob, R. J. K. (2009). A specification paradigm for the design and implementation of tangible user interfaces. *ACM Transactions on Computer-Human Interaction*, 16(4), 1-39.

- Sheng, Q. Z., Pohlenz, S., Yu, J., Wong, H. S., Ngu, A. H. H., & Maamar, Z. (2009). ContextServ: A platform for rapid and flexible development of context-aware Web services. *Proceedings of the 31st International Conference on Software Engineering*, 1-4.
- Shin, C., Kim, H., Kang, C., Jang, Y., Choi, A., & Woo, W. (2010). Unified Context-Aware Augmented Reality Application Framework for User-Driven Tour Guides. *International Symposium on Ubiquitous Virtual Reality*, 52-55.
- Signer, B., Grossniklaus, M., & Norrie, M. C. (2007). Interactive paper as a mobile client for a multi-channel web information system. *World Wide Web*, 10(4), 529–556.
- Skype (2012). Free Skype internet calls and cheap calls to phones online – Skype. <http://www.skype.com/intl/en/home>.
- Soldatos, J., Dimakis, N., Stamatis, K., & Polymenakos, L. (2007). A breadboard architecture for pervasive context-aware services in smart spaces: Middleware components and prototype applications. *Personal and Ubiquitous Computing*, 11(3), 193–212.
- Sridevi, S., Sayantani, B., Amutha, K. P., Mohan, C. M., & Pitchiah, R. (2010). Context Aware Health Monitoring System. *Lecture Notes in Computer Science*, 6165/2010, 249-257.
- Stober, S., & Nürnberger, A. (2010). MusicGalaxy - An Adaptive User-Interface for Exploratory Music Retrieval. *Proceedings of 7th Sound and Music Computing Conference (SMC'10)*, 382-389.
- Stober, S., & Nürnberger, A. (2012). Adaptive music retrieval—a state of the art. *Multimedia Tools and Applications*, 1-28.
- Strasser, H. (2011). Digital Convergence. <http://www.digitalconvergence.eu/>.
- Strassner, J., Meer, S. van der, O'Sullivan, D., & Dobson, S. (2009). The Use of Context-Aware Policies and Ontologies to Facilitate Business-Aware Network Management. *Journal of Network and Systems Management*, 17(3), 255-284.
- Stratus (2012). Ionics Plug. Create your own cloud! Stratus. <http://www.ionicsplug.com/stratus.html>.
- Suh, J., & Woo, C. –W. (2011). Design and Development of a Social Intelligence Based Context-Aware Middleware Using BlackBoard. *IEEE International*

LITERATURA

- Conference on Tools with Artificial Intelligence (ICTAI), Boca Raton, USA, 908-910.*
- Sung, M., Gips, J., Eagle, N., Madan, A., Caneel, R., Devaul, R., et al. (2005). Mobile-IT Education (MIT. EDU): m-Learning applications for classroom settings. *Journal of Computer Assisted Learning*, 21(3), 229–237.
- TAT (2009). TAT Augmented ID. <http://www.tat.se/blog/tat-augmented-id/>.
- Thawani, A., Gopalan, S., & Sridhar, V. (2005). Context-Aware Personalized Ad Insertion in an Interactive TV Environment. *TV'04: the 4th Workshop on Personalization in Future TV - Methods, Technologies, Applications for Personalized TV*, 1-7.
- Tiny XML (2012). Tiny XML download. <http://sourceforge.net/projects/tinyxml/>.
- Tsunoda, T., & Hoshino, M. (2008). Automatic metadata expansion and indirect collaborative filtering for TV program recommendation system. *Multimedia Tools and Applications*, 36(1-2), 37-54.
- Tullio, J., Dey, A. K., Chalecki, J., & Fogarty, J. (2007). How it works: a field study of non-technical users interacting with an intelligent system. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1-10.
- Tyan, J., & Mahmoud, Q. H. (2005). A comprehensive service discovery solution for mobile ad hoc networks. *Mobile Networks and Applications*, 10(4), 423–434.
- Udugama, A., Kuladinithi, K., Gorg, C., Pittmann, F., & Tionardi, L. (2007). NetCAPE: Enabling seamless IMS service delivery across heterogeneous mobile networks. *Communications Magazine, IEEE*, 45(7), 84–91.
- UK Office for National Statistics (2003). UK 2000 Time Use Survey, TV Viewing. http://www.statistics.gov.uk/timeuse/summary_results/activities_age_gender.asp, pristupano u novembru 2009.
- Ullmer, B. A. (2002). Tangible interfaces for manipulating aggregates of digital information. *Massachusetts Institute of Technology, Ph.D. thesis*, 1-268.
- Umapathy, V. (2011). An adaptive user interface for open educational content. *Massachusetts Institute of Technology, Thesis (M. Eng.)*, 1-67.
- Valtonen, M., Vainio, A. -M., & Vanhala, J. (2009). Proactive and adaptive fuzzy profile control for mobile phones. *IEEE International Conference on Pervasive Computing and Communications*, 1-3.

- Vastenburger, M. H., Keyson, D. V., & Ridder, H. de (2009). Considerate Home Notification Systems: A User Study of Acceptability of Notifications in a Living Room Laboratory. *International Journal of Human-Computer Studies*, 67(9), 814-826.
- Vergnes, D., Giroux, S., Chamberland, D. -T. (2005). Interactive Assistant for Activities of Daily Living. *From Smart Homes to Smart Care: International Conference on Smart Homes and Health Telematics*, 15(1), 229-236.
- Vertegaal, R. (2003). Attentive User Interfaces. *Communications of the ACM*, 46(3), 31-33.
- Vilas, A. F., Redondo, R. P. D., Arias, J. J. P., Cabrer, M. R., Solla, A. G., & Duque, J. G. (2010). Context-aware personalization services for a residential gateway based on the OSGi platform. *Expert Systems with Applications*. 37(9), 6538-6546.
- Vlist, B. J. J. van der, Gerrit, N., Jun, H. & Feijs, L. M. G. (2010). Semantic connections: Exploring and manipulating connections in smart spaces. *IEEE Symposium on Computers and Communications*, 1-4.
- Weiser, M. (1991). The Computer for the 21st century. *Scientific American*, 265(9), 66-75.
- Wikitude (2012). Wikitude lab takes AR to the web. <http://www.wikitude.com/>.
- Wilson, D., Doyle, J., Weakliam, J., Bertolotto, M., & Lynch, D. (2007). Personalised maps in multimodal mobile GIS. *International Journal of Web Engineering and Technology*, 3(2), 196-216.
- Wood, A., Stankovic, J., Virone, G., Selavo, L., Zhimin, H., Qihua, C., Thao, D., Yafeng, W., Lei, F., & Stoleru, R. (2008). Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE Network*. 22(4), 26-33.
- Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., Yi, M. (2009). Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 210-227.
- Wu, C. L., Liao, C. F., & Fu, L. C. (2007a). Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology. *IEEE Transactions on Systems, Man and Cybernetics*, 37(2), 193-205.

LITERATURA

- Wu, Z., Wu, Q., Cheng, H., Pan, G., Zhao, M., & Sun, J. (2007b). ScudWare: A semantic and adaptive middleware platform for smart vehicle space. *Intelligent Transportation Systems, IEEE*, 8(1), 121–132.
- Xu, Y., Spasojevic, M., Gao, J., & Jacob, M. (2008). Designing a vision-based mobile interface for in-store shopping. *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, 1-10.
- Xu, J. Y., Sun, Y., Wang, Z., Kaiser, W. J., & Pottie, G. J. (2011). Context guided and personalized activity classification system. *Proceedings of the 2nd Conference on Wireless Health*, 1-10.
- Yau, S. S., & Karim, F. (2003). An energy-efficient object discovery protocol for context-sensitive middleware for ubiquitous computing. *IEEE Transaction on Parallel and Distributed Systems*, 14(11), 1074–1084.
- Yaghmaie, M., & Bahreininejad, A. (2011). A context-aware adaptive learning system using agents. *Expert Systems with Applications*, 38(4), 3280-3286.
- Yu, Z., Zhou, Z., Yu, Z., Zhang, D., & Chin, C. (2006). An OSGI-based infrastructure for context-aware multimedia services. *Communications Magazine, IEEE*, 44(10), 136–142.
- Yu-Ren Chen (2010). Developing a Smart Home the System the Using RF4CE-compliant ZigBee Wireless Sensor Networks. *Master thesis, National Taipei University of Education, Department of Computer Science*.
- Yung-Bok, K., Marie, K., & Yong J. L. (2008). COSMOS: a middleware platform for sensor networks and a u-healthcare service. *Proceedings of the 2008 ACM symposium on Applied computing*.
- Zhang, Z., Wang, N., Chen, Y., Lu, Y., Shi, Y., Wang, R., & Lu, X. (2011). SREngine: An OSGi-based Context-aware inference engine for smart room. *International Conference on Pervasive Computing and Applications*, 267-271.
- Zheng, D., Yan, H., & Wang, J. (2011). Research of the Middleware Based Quality Management for Context-Aware Pervasive Applications. *International Conference on Computer and Management*, 1-4.
- Zhu, F., Mutka, M. W., & Ni, L. M. (2005). Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, 4(4), 81–90.

- Zigelbaum, J., Horn, M. S., Shaer, O., & Jacob, R. J. K. (2007). The tangible video editor: collaborative video editing with active tokens. *Proceedings of the 1st international conference on Tangible and embedded interaction*, 1-5.
- Zimmer, T. (2006). QoC: Quality of context - improving the performance of context-aware applications. *In Advances in Pervasive Computing. Adjunct Proceedings of Pervasive 2006*, 1-6.

LITERATURA

BIOGRAFIJA

Milan Bjelica rođen je 1985. godine u Zrenjaninu, Republika Srbija. Master tezu „Realizacija sistema za automatizaciju domaćinstva sa grafičkom korisničkom spregom na TV prijemniku“ odbranio je 2008. godine na Fakultetu tehničkih nauka Univerziteta u Novom Sadu.

Od 2009. godine zaposlen je u istraživačko-razvojnem institutu RT-RK, gde učestvuje u razvoju inovativnih kontrolera za inteligentna okruženja u okviru grupacije i-Novi. U okviru istraživanja u sklopu doktorske disertacije, 2010. godine radi na razvoju programske podrške sistema za automatizaciju domaćinstva kompanije *Control4*, SAD. Aktivno učestvuje u razvoju rešenja koja uključuju korisničke aspekte, uključujući kontekstualne algoritme i korisničke sprege. Studije ljudskog faktora za kontekstualne aplikacije, koje su delom uključene u ovu disertaciju, počev od 2012. godine realizuje u saradnji sa univerzitetom *Sheffield Hallam, Sheffield*, Velika Britanija. Rezultati njegovog rada integrisani su u rešenja automatizacije inteligentnih okruženja i automatizacije domaćinstva kompanija RT-RK, *Control4* i *Philips*.

Na Fakultetu tehničkih nauka od 2009. godine radi na doktorskoj disertaciji u oblasti kontekstualnog računarstva, i zaposlen je kao asistent na predmetima *Programska podrška u televiziji i obradi slike, Arhitektura procesora signala i Programiranje namenskih računarskih struktura 1*. Predvodi tim inženjera i studenata za realizaciju korisničkih sprega i igara za inteligentna okruženja koja uključuju integraciju različitih uređaja potrošačke elektronike, poput TV prijemnika, pametnih telefona i računara. Autor je ili koautor osam radova u časopisima (pet kao prvi autor),

BIOGRAFIJA

preko 40 radova na međunarodnim i nacionalnim konferencijama, tri tehnička rešenja i jednog patenta. Aktivno učestvuje u projektima Ministarstva prosvete i nauke Republike Srbije (ciklus istraživanja 2011-2014).


Govori engleski. Oženjen je Nadom, i tata malog Vasilija.

PRILOG: SENZORSKE TEHNOLOGIJE

Kao prilog u nastavku je data analiza trenutno dostupnih fizičkih senzorskih tehnologija, koje se mogu integrirati u kontekstualnu platformu a bile su dostupne su na tržištu u trenutku pisanja disertacije (te su stoga moguće za korišćenje u uređajima potrošačke elektronike). Dat je spisak sledećih grupa senzorskih tehnologija, razvrstano po primeni/osobinama:


- Senzori dometa
- Senzori dodira i blizine
- Senzori setekcije pokreta
- Senzori za detekciju objekata
- Senzori za identifikaciju
- Senzori za detektovanje meteoroloških prilika
- Senzori za detektovanje nivoa svetlosti
- Senzori zvuka
- Senzori pokreta / orijentacije
- Senzori za detekciju težine
- Senzori za detekciju kontakta elemenata

Osnovni faktori koje treba uzeti u obzir pri odlučivanju o upotrebi određenog senzora u uređaju potrošačke elektronike sa kontekstualnim funkcijama su (1) mrtvi uglovi, odnosno, granične zone detekcije; (2) interferencija sa uređajima u blizini; (3) cena (izražena u poenima, za međusobno poređenje cene različitih senzora); (4) domet; (5) mogućnosti u smislu traženog slučaja korišćenja.

3D Time of Flight (TOF) kamera	
Tip:	Senzor dometa
Opis funkcionisanja:	Merenje faze odbijenog svetlosnog talasa, koji se emituje na frekvenciji skoro infracrvenog spektra (<i>near infrared – NIR</i>).
Opseg:	2m – 5m (teoretski 2m – 15m)
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	6/10
Sprege sa okolinom:	USB 2.0, Ethernet (za gotove kamere)
Cena:	~3000
Prednosti:	<ul style="list-style-type: none"> • jedan senzor za širok spektar aplikacija • pouzdana tehnologija • moguća detekcija obeležja korisnika
Nedostaci:	<ul style="list-style-type: none"> • cena • nemogućnost rada pri direktnoj sunčevoj svetlosti
Proizvođači/modeli:	Panasonic Electric Works / <i>D-Imager</i> SoftKinetic / <i>DepthSense</i> Fotonic PMDTechnologies / <i>PMDVision</i> MESA Imaging / <i>SwissRanger</i>
Slike:	

3D skener struktuiranog svetla	
Tip:	Senzor dometa
Opis funkcionisanja:	Projekcija svetlosnih šablona (<i>pattern</i>) i snimanje distorzija.
Opseg:	1m – 4m (~6 m ²)
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	7/10
Sprege sa okolinom:	USB
Cena:	150
Prednosti:	<ul style="list-style-type: none"> • jedan senzor za širok spektar aplikacija • pouzdana tehnologija • moguća detekcija obeležja korisnika • cena
Nedostaci:	<ul style="list-style-type: none"> • nešto sporiji odziv • nemogućnost rada pri direktnoj sunčevoj svetlosti


Proizvođači/modeli:	Microsoft / <i>Kinect</i>
Slike:	

Stereo kamera	
Tip:	Senzor dometa
Opis funkcionisanja:	Na osnovu posmatranja scene od strane više kamera računa se 3D pozicija tačke snimljene iz različitih uglova.
Opseg:	4m x 6m
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	8/10
Sprege sa okolinom:	IEEE-1394a (<i>Firewire</i>), USB
Cena:	2000
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • stabilnost tehnologije (postoji duže vreme)
Nedostaci:	<ul style="list-style-type: none"> • nedovoljna prisutnost u uređajima potrošačke elektronike
Proizvođači/modeli:	Point Grey / <i>Bumblebee</i> Vivitar Loreo
Slike	

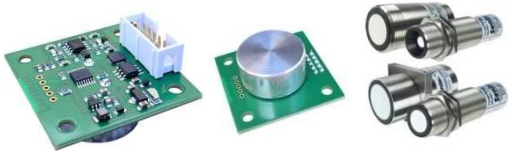
Infracrveni senzor (IR)	
Tip:	Senzor dometa / Senzor blizine
Opis funkcionisanja:	Emitovanje infracrvenog zraka. Kada objekat preseče zrak, senzor se aktivira.
Opseg:	1-2 m ³
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✗ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne izlazne linije
Cena:	10 (samo senzor)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • ne zahteva fizički kontakt sa korisnikom • trajnost

Prilog: Senzorske tehnologije

Nedostaci:	<ul style="list-style-type: none"> • osetljivost na eksterne izvore infracrvenog zračenja, poput grejnih tela ili sunčeve svetlosti • osetljivost na prljavštinu i prašinu • ne postoji uređaj potrošačke elektronike sa komercijalnom spregom
Proizvođači:	Sharp World Acroname Robotics Avagotech
Slike:	

Mikrotalasni senzor (radar)	
Tip:	Senzor dometa
Opis funkcionisanja:	Emitovanje mikrotalasa, prikupljanje reflektovanih talasa i merenje faznog pomeraja.
Opseg:	do 8-12m (ugao širenja 60°)
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Analogne izlazne linije
Cena:	20 (samo senzor), 200 (uređaj)
Prednosti:	<ul style="list-style-type: none"> • visoka preciznost i pouzdanost • moguće meriti brzinu i smer kretanja objekta • otporno na promene temperature, isparenja, prašinu i dim • prihvatljiva cena
Nedostaci:	<ul style="list-style-type: none"> • osetljiv na refleksiju talasa iz okruženja
Proizvođači:	Innosent Wadeco SiversIMA
Slike:	


Ultrazvučni senzor (sonar)	
Tip:	Senzor dometa
Opseg:	do 10 m ³
Opis funkcionisanja:	Merenje energije odbijenog signala od 40 kHz. Izmene u primljenoj energiji označavaju prisustvo objekta.

Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne ili analogne izlazne linije
Cena:	20 (samo senzor), 200 (uređaj)
Prednosti:	<ul style="list-style-type: none"> • preciznost i pouzdanost • prihvatljiva cena • jednostavan za korišćenje i implementaciju
Nedostaci:	<ul style="list-style-type: none"> • osetljiv na promene temperature, pritiska, višestruki eho, vlažnost • staromodna tehnologija
Proizvođači:	Sick UK Devantech / <i>Robot Electronics</i> Polaroid Murata SensComp Omron Industrial Automation Senix
Slike:	

Tasteri	
Tip:	Senzor dodira
Opis funkcionisanja:	Fizički pritisak tastera od strane korisnika obezbeđuje električni kontakt koji signalizira pritisak.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije
Cena:	<1 (za taster), >10 (za tastature)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • niska cena • jednostavno korišćenje
Nedostaci:	<ul style="list-style-type: none"> • potrebna dodatna fizička arhitektura i ugradnja
Proizvođači:	Grayhill Storm Interface C&K Components


Prilog: Senzorske tehnologije



Senzor zasnovan na Holovom efektu	
Tip:	Senzor dodira
Opis funkcionisanja:	Holov senzor daje električni signal proporcionalan promeni jačine magnetnog polja. Funkcioniše u sklopu nekog pomičnog elementa (dugme, pedala, fizički slajder i sl).
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✗ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije.
Cena:	<1-5
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • niska cena • jednostavno korišćenje • trajnost • detekcija nivoa pomeraja dugmeta/pedale
Nedostaci:	<ul style="list-style-type: none"> • zahteva fizički kontakt sa korisnikom • sistem zahteva trajni magnet kao izvor magnetnog polja • osetljivost na okruženja u otvorenom prostoru • potrebna dodatna fizička arhitektura
Proizvođači:	Micronas Honeywell Cherrycorp
Slike:	


Kapacitivni senzor	
Tip:	Senzor blizine
Opis funkcionisanja:	Kada se objekat nađe u zoni detekcije, električna kapacitivnost senzora se menja. Ova promena se konvertuje u električni signal koji se može meriti.
Opseg:	15 cm
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10


Sprege sa okolinom:	Digitalne linije
Cena:	100 (gotov uređaj), 2-3 (rešenje zasnovano na integrisanom kolu)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • ne zahteva fizički kontakt sa korisnikom • trajnost
Nedostaci:	<ul style="list-style-type: none"> • skuplji za upotrebu u većoj količini (npr. za realizaciju tastera) • potrebna kalibracija jer performanse zavise od okruženja, korisnika i sl.
Proizvođači/Modeli:	Omron Atmel / <i>Q-prox</i>
Slike:	

Termalni senzor	
Tip:	Senzor za detekciju objekata
Opseg:	Oko 500° C
Opis funkcionisanja:	Koristi se tehnologija termalne infracrvene slike za identifikaciju korisnika u zoni detekcije, smer kretanja, brojanje objekata, gustinu saobraćaja i sl.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	5/10
Sprege sa okolinom:	Serijska, Ethernet, Firewire, USB
Cena:	~1500
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • srednja cena • jednostavnost korišćenja
Nedostaci:	<ul style="list-style-type: none"> • skupa rešenja za spoljnu upotrebu • potrebno podešavanje na mestu instalacije
Proizvođači/Modeli:	Trafsys Electrophysics / <i>TVS-50SC</i> SenSource
Slike:	

Optička kamera sa ugrađenim algoritmima	
Tip:	Senzor za detekciju objekata / Senzor detekcije pokreta
Opis funkcionisanja:	Fotografisanje posmatrane scene i primena algoritama obrade slike u cilju otkrivanja pokreta, segmentacije/prebrojavanja objekata i sl.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	USB, Ethernet, WiFi
Cena:	20-1000
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • jednostavno korišćenje • postoji bežično rešenje
Nedostaci:	<ul style="list-style-type: none"> • relativno visoka cena • ograničena performansa u uslovima slabe osvetljenosti i pri čestim promenama osvetljenosti • osetljivost na vremenske uslove za korišćenje u otvorenom prostoru
Proizvođači/Modeli:	Veliki broj proizvođača Logitech Axis / <i>M1010</i> Dilax / <i>Optocount</i> Linksys / <i>PVC2300</i> sa audio funkcijom i PoE
Slike:	


Pasivni infracrveni senzor (PIR)	
Tip:	Senzor detekcije pokreta
Opis funkcionisanja:	Merenje infracrvene svetlosti koju zrače objekti u vidnom polju.
Opseg:	10 m x 10 m
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije
Cena:	1-5 (samo senzor), >20 (uređaj)
Prednosti:	<ul style="list-style-type: none"> • trajnost • jednostavnost • niska cena

Nedostaci:	<ul style="list-style-type: none"> potrebni posebni algoritmi ili samostalan uređaj koji razlikuje npr. ljude od kućnih ljubimaca
Proizvođači/Modeli:	Kube / TR257 PerkinElmer Nicera Paradox
Slike:	


Kombinovani senzor za detekciju pokreta (PIR/radar)	
Tip:	Senzor za detekciju pokreta
Opis funkcionisanja:	Kombinovano korišćenje pasivne infracrvene tehnologije i mikrotalasne tehnologije, što doprinosi smanjenju broja lažnih alarma.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Analogne linije
Cena:	50
Prednosti:	<ul style="list-style-type: none"> srednje visoka cena bolje performanse od samostalnih senzora napredne funkcije, poput izbegavanja alarma za kućne ljubimce
Nedostaci:	<ul style="list-style-type: none"> nešto viša cena od pojedinačnih senzora
Proizvođači:	Guardall Paradox
Slike:	

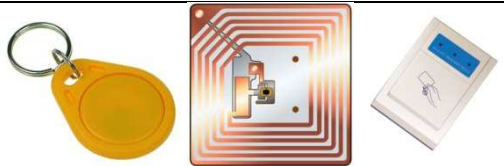
Čitač bar/QR kodova	
Tip:	Senzor za identifikaciju
Opis funkcionisanja:	Očitavanje identifikacionog broja objekat (EAN notifikacija) skeniranjem šablona odštampanog na objektu (koji se sastoji od vertikalnih linija različite debljine ili šablona kvadratića - QR) korišćenjem svetlosnog izvora, sočiva i senzora osveljaja. Takođe postoje realizacije koje koriste kameru i algoritme obrade slike.

Prilog: Senzorske tehnologije

Opseg	0-50 cm
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✗ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Serijska sprega, USB
Cena:	>150
Prednosti:	<ul style="list-style-type: none"> • Jednostavnost • Niska cena
Nedostaci:	<ul style="list-style-type: none"> • zahteva direktnu optičku vidljivost za rad • bar kod mora biti odštampan na svim objektima od interesa i ne može se dinamički menjati • ograničena daljina za uspešno očitavanje • nizak nivo sigurnosti identifikacije (u poređenju sa RFID rešenjima) • zahteva kontroler u neposrednoj blizini
Proizvođači:	Honeywell Security Nonus
Slike:	


SMART kartice	
Tip:	Senzor za identifikaciju
Opis funkcionisanja:	Integrirano kolo smešteno je u plastični okvir najčešće veličine kreditne kartice. Identifikacija se obavlja umetanjem kartice u čitač ili prisanjanjem kartice na terminal, te čitanjem podataka iz memorije u okviru integrisanog kola. Mogu biti zasnovane na RFID.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Čitač kartica spregnut standardnim spregama (npr. USB)
Cena:	Niska cena.
Prednosti:	<ul style="list-style-type: none"> • Jednostavnost • Niska cena
Nedostaci:	<ul style="list-style-type: none"> • zavisi od prljavštine, prašine, vremenskih uslova i uopšte uslova čuvanja kartice, s obzirom na to da čitljivost opada vremenom • niska brzina očitavanja (oko 4 sekunde) • potreban je čitač kartica za identifikaciju

Proizvođači/Modeli:	Huayuan / <i>EM4200</i> Atmel
Slike:	

RFID	
Tip:	Senzor za identifikaciju
Opis funkcionisanja:	Identifikacija korišćenjem radio frekvencije. Sastoji se od dve komponente: transpondera, koji se nalazi ugrađen u objekat koji treba identifikovati (aktivni tag), i čitača, koji emituje energiju, pobuđuje transponder i očitava emitovani kod korišćenjem radio talasa.
Opseg:	do 1 cm (kHz), do 1 m (MHz), do 5 m (UHF)
Prostor upotrebe:	✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne sprege
Cena:	10 (aktivni tag), >500 (čitač)
Prednosti:	<ul style="list-style-type: none"> • Jednostavnost • visoka brzina očitavanja (0.5 sekundi) • nije potreban fizički kontakt • popularna sprega • ne zahteva direktnu optičku vidljivost • konfigurabilnost
Nedostaci:	<ul style="list-style-type: none"> • visoka cena za pojedinačne tagove
Proizvođači/Modeli:	Huayuan Atmel
Slike:	

Biometrijski senzori	
Tip:	Senzor za identifikaciju
Opis funkcionisanja:	Identifikuju korisnika na osnovu njegovih jedinstvenih fizičkih parametara, poput otiska prsta, glasa ili identifikacije mrežnjače oka. Koriste se mehanizmi optičkog skeniranja / snimanja kamerom / snimanja zvuka i analize slike/zvuka odgovarajućim algoritmima.

Prilog: Senzorske tehnologije

Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	5/10
Sprege sa okolinom:	Digitalne sprege
Cena:	Izuzetno visoka (osim za čitače otiska prsta)
Prednosti:	<ul style="list-style-type: none"> • izuzetno visoka pouzdanost identifikacije ljudi
Nedostaci:	<ul style="list-style-type: none"> • niska brzina očitavanja (~5 sekundi) • izuzetno kompleksni algoritmi za identifikaciju • osetljivost na prljavštinu i prašinu kod optičkih metoda, poput identifikacije otiska prsta • problem zaštite privatnosti, što može uzrokovati odbijanje korišćenja ove tehnologije od strane potrošača
Proizvođači/Modeli:	Atmel (čitač otiska prsta)
Slike:	

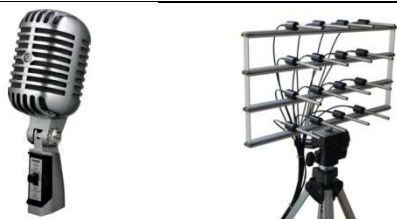
Senzori za temperaturu, vlažnost, vazdušni pritisak, vetar i sunčevo zračenje	
Tip:	Senzor za detektovanje meteoroloških prilika
Opis funkcionisanja:	Merenje određene komponente meteorološke prilike korišćenjem različitih tehnika.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne i analogne linije
Cena:	10 (samostalni senzori), >200 (stanice za merenje meteoroloških prilika)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • niska cena • jednostavno korišćenje
Nedostaci:	<ul style="list-style-type: none"> • kompleksnija integracija sa okruženjem
Proizvođači/Modeli:	NewMountain / <i>NM150</i> meteorološka stanica Rainwise Sparkfun / <i>SHT15</i> senzor vlažnosti i temperature Hygrasgard




Senzor osvetljenosti	
Tip:	Senzor za detektovanje nivoa svetlosti
Opis funkcionisanja:	Merenje nivoa svetlosti korišćenjem CCD senzora.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne i analogne linije
Cena:	<10 (samostalni senzori)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • niska cena • jednostavno korišćenje
Nedostaci:	<ul style="list-style-type: none"> • za korišćenje u otvorenom prostoru potrebna je odgovarajuća montaža i pakovanje (zaštita od vlage, prašine i sl). • preciznost zavisi od mesta postavljanja
Proizvođači/Modeli:	Douglas Lighting Controls / WPS-5527K Kodenshi Jenoptik
Slike:	


Senzori zvuka (mikrofoni)	
Tip:	Senzor zvuka
Opis funkcionisanja:	Konvertuje pritisak zvuka na membranu mikrofona u električne signale.
Opseg:	Omnidirekcionni mikrofoni (zvučni signal iz celokupnog radijusa se prikuplja) i direkcionni (prikuplja zvuk iz smera postavljanja mikrofona).
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru

Prilog: Senzorske tehnologije

Zrelost tehnologije:	6/10
Sprege sa okolinom:	Analogne linije, potrebna A/D konverzija
Cena:	Veliki rang u zavisnosti od kvaliteta (1 - >1000)
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • ne zahteva fizički kontakt sa korisnikom • trajnost • kvalitetni audio algoritmi mogu da potisnu eho, šum i sl. i podignu upotrebljivost na visok nivo • mikrofonski nizovi omogućavaju otkrivanje pozicije govornika
Nedostaci:	<ul style="list-style-type: none"> • potrebni algoritmi obrade zvuka za napredne primene • osetljivost na položaj govornika • osetljivost na glasnoću govora • osetljivost na izvore šuma i reverberaciju prostora
Proizvođači/Modeli:	Mnogo proizvođača Sennheiser Honeywell Shure
Slike:	


RGB senzor	
Tip:	Senzori za identifikaciju
Opis funkcionisanja:	Detektuje vidljivi svetlosni spektar u vidu nivoa crvene, plave i zelene komponente.
Opseg:	talasne dužine 380 nm – 750 nm ili više (do ~1000 nm)
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✗ u otvorenom prostoru
Zrelost tehnologije:	5/10
Sprege sa okolinom:	Analogne linije, potrebna A/D konverzija
Cena:	50
Prednosti:	<ul style="list-style-type: none"> • pouzdanost • trajnost • ne zahteva kontakt sa korisnikom • mogućnost identifikacije na osnovu boje
Nedostaci:	<ul style="list-style-type: none"> • osetljivost na eksterne izvore svetlosti

Proizvođači/Modeli:	TAOS (ams) Hamamatsu Keyence
Slike:	

Akcelerometar (Žiroskop)	
Tip:	Senzor pokreta / orijentacije
Opis funkcionisanja:	Pokret mehaničkog tela unutar ispitne komore meri se piezoelektričnim, piezorezistivnim ili kapacitivnim komponentama i konvertuje u električni signal, koji odgovara slobodnom ubrzanju. Pojedina rešenja koriste Holove senzore.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije (samostalan senzor), komercijalne sprege, poput WiFi/Ethernet ukoliko se komunicira sa aplikacijom na mobilnom telefonu koja ima pristup akcelerometru
Cena:	~3
Prednosti:	<ul style="list-style-type: none"> • omogućava kreiranje korisničkih sprega koje podržavaju gestove • omogućava detekciju kretanja, mirovanja i orijentacije • prisutan u uređajima potrošačke elektronike, poput pametnih telefona • jednostavnost i pouzdanost
Nedostaci:	<ul style="list-style-type: none"> • za uspešno merenje mora biti u sastavu merenog objekta • za naprednu detekciju gestova potrebno je koristiti više akcelerometara i složene algoritme
Proizvođači/Modeli:	Bosch / <i>BMA150</i>
Slike:	

Senzor opterećenja	
Tip:	Senzor za detekciju težine
Opis funkcionisanja:	Na osnovu merača istežanja opruge, koja pri različitim istežanjima ima različitu električnu otpornost, dobija se električna veličina koja korespondira mehaničkom opterećenju primenjenom na senzor.

Prilog: Senzorske tehnologije

Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije (samostalan senzor)
Cena:	~10
Prednosti:	<ul style="list-style-type: none"> • omogućava pouzdanu detekciju prisustva u određenoj zoni, uključujući diskriminaciju po težini • nevidljiv za korisnika ako se pravilno postavi
Nedostaci:	<ul style="list-style-type: none"> • komplikovana instalacija i ožičenje • zavisi od karakteristika poda
Proizvođači/Modeli:	Sparkfun / <i>SEN-10245</i>
Slike:	

Kontaktni senzori	
Tip:	Senzor za detekciju kontakta elemenata
Opis funkcionisanja:	Na osnovu uspostavljenog kontakta dve kontaktne jedinice (ploče), uspostavlja se protok struje i omogućava signal na izlazu.
Prostor upotrebe:	<ul style="list-style-type: none"> ✓ u zatvorenom prostoru ✓ u otvorenom prostoru
Zrelost tehnologije:	10/10
Sprege sa okolinom:	Digitalne linije (samostalan senzor)
Cena:	~10
Prednosti:	<ul style="list-style-type: none"> • izuzetno jednostavan za korišćenje • pouzdana detekcija
Nedostaci:	<ul style="list-style-type: none"> • potrebno postavljanje na mesta kontakta
Proizvođači/Modeli:	Keyence / <i>GT2</i>
Slike:	