

Taksonomija kontekstno odvisnih sistemov

Rok Žontar, Marjan Heričko, Ivan Rozman

Univerza v Mariboru, Fakulteta za elektrotehniko računalništvo in informatiko, Smetanova ulica 17, 2000 Maribor, Slovenija
E-pošta: rok.zontar@uni-mb.si

Povzetek. V delu bomo predstavili rezultate raziskave, ki se osredinja na področje kontekstno odvisnih sistemov. Raziskava temelji na opravljenem pregledu literature, iz katerega smo pridobili nekaj definicij konteksta in kontekstne odvisnosti. Poudariti želimo analizo, ki je bila kot temelj za izdelavo taksonomije kontekstno odvisnih sistemov. V njej vsebovane kategorije vključujejo tako tehnične lastnosti kot tudi nekatere poglede na varnost in zasebnost podatkov. Uporabnost taksonomije je demonstrirana s primerom klasifikacije, ki temelji na osmih kategorijah, predstavljenih v taksonomiji.

Ključne besede: kontekstno odvisni sistemi, senzorji, taksonomija, klasifikacija

Taxonomy of context-aware systems

In this paper we present conclusions, drawn from our research on context-aware systems. Based on reviewing the literature, from which we collected definitions of context and context awareness. We provide the basis for a taxonomy of context-aware systems grouped in eight categories. They include technical aspects of these systems as well as some views of privacy and security. Usefulness of taxonomy is demonstrated with a classification example based on the taxonomy categories.

1 UVOD

Razvoj mobilnih naprav, kot so pametni telefoni in vseprisotnih informacijskih rešitev, ki smo jim priča dandanes, je privedel do popolnoma novega okolja, v katerem uporabniki pričakujejo uporabne storitve in informacije. A količina podatkov in storitev se širi z izjemno hitrostjo. Le-ta je dosegla že tak obseg, da se uporabniki težko sami odločijo, katero storitev izbrati, ali pa za to porabijo veliko časa. To je spodbudilo idejo o uporabi konteksta za zmanjšanje nabora informacij in ponujanje uporabnikom bolj prilagojenih storitev.

Sistemi, ki uporabljajo kontekst, se imenujejo kontekstno odvisni in omogočajo nove razvojne možnosti tako za ponudnike storitev kot tudi za njihove končne uporabnike. Z zajemanjem kontekstnih podatkov in njihovo obdelavo je mogoče prilagajati obnašanje sistema in s tem povečati učinkovitost iskanja informacij in storitev. To pa neposredno izboljšuje uporabniško izkušnjo. Še posebno v kombinaciji z mobilnimi napravami se ti mehanizmi kažejo kot posebej učinkoviti in doživljajo množično uporabo na različnih mobilnih aplikacijskih platformah.

Ta motivacija nas je privedla do raziskave, kjer smo pregledali objave zadnjih let in povzeli rezultate sorodnih del. Opravljen pregled literature je bil podlaga

za izdelavo analize kontekstno odvisnih sistemov. Ta nas je pripeljala do spoznanja, da imajo sistemi kar nekaj podobnosti, a se vendarle razlikujejo v arhitekturi, delovanju in uporabi konteksta. Da bi dosegli višjo stopnjo sistematizacije, smo se odločili za izdelavo taksonomije kontekstno odvisnih sistemov, ki je jedro v tem članku opisane raziskave.

Delo je sestavljeno takole: po uvodnem poglavju predstavljamo definicije kontekstna in kontekstno odvisnega računalništva. Nadaljujemo s tretjim poglavjem, kjer opisujemo osnovne značilnosti kontekstno odvisnih sistemov. Četrto poglavje je jedro raziskave. Na začetku predstavimo taksonomijo kontekstno odvisnih sistemov, ki jo nato podrobno razdelamo po kategorijah. V sklepu predstavljamo klasifikacijo ogrodij in aplikacij glede na predstavljeno taksonomijo. Članek zaključujemo z razpravo, kjer kritično razpravljamo o raziskanem področju in možnosti nadaljnjih raziskav.

2 DEFINICIJA KONTEKSTA IN KONTEKSTNE ODVISNOSTI

Za lažje razumevanje konteksta smo najprej poiskali različne definicije tega termina v literaturi. Prvič izraz obravnavata Schilit in Theimer [1], ki ga definirata kot *lokacijo ali identiteto ljudi in predmetov v bližini ter spremembe med njimi*. Podobne definicije podajo tudi Brown et al. [2] ter Ryan et al. [3]. Čeprav so takšne definicije zelo pogoste, jih je v praksi težko posplošiti. Zato raziskovalci težimo k temu, da bi termin definirali na bolj splošen način. Tega sta se zavedala tudi Dey in Abowd [4], ki sta poiskovala poenotiti pogled na kontekst in kontekstno odvisne aplikacije.

Če gledamo na kontekst v smislu informacijskih storitev in ne toliko mobilnih oz. prenosnih naprav, se

bolj prilega definicija konteksta, kot so jo podali Doukeridis et al. [5]. Njihova definicija identificira dve vrsti konteksta, in sicer uporabniškega in storitvenega. Pri iskanju storitev se preverja ujemanje obeh kontekstov in se tako izključijo vse nerelevantne storitve.

Sami definiciji konteksta in kontekstno odvisnega računalništva sta strogo prepleteni. Tako ni presenetljivo, da sta prvo definicijo kontekstne odvisnosti podala prav Schilit in Theimer [1]. Na začetkih raziskav so definirali aplikacije kot kontekstno odvisne že, če so te seznanjene s kontekstom in se odzivajo v odvisnosti od njegovega stanja.

Treba je poudariti, da v literaturi ne obstaja skupno razumevanje, kaj kontekst dejansko je. Čeprav se večina avtorjev sklicuje na kontekst, ga v svojih prispevkih dojemajo popolnoma drugače. Zato smo se osredotočili na kontekst kot množico dodatnih podatkov o uporabniku, ki mu pomagajo izluščiti pomembne podatke oz. storitve glede na njegove zahteve.

3 KONTEKSTNO ODVISNI SISTEMI

Skozi leta se je razvilo veliko ogrodij in aplikacij, ki jih lahko uvrstimo v skupino kontekstno odvisnih sistemov. Ti se v večini razlikujejo po naboru funkcionalnosti, postavitvi in poimenovanju arhitekturnih slojev ter drugih arhitekturnih konceptov. S ciljem, da izdelamo smiselno taksonomijo, smo najprej raziskali nekaj skupnih značilnosti sistemov, ki so jih omenjali že avtorji pred nami (Baldauf et al. [6] in Lee et al. [7][7]). Na podlagi njihovega dela in predstavljenih meta arhitektur smo razdelali in analizirali kontekstno

odvisne sisteme.

Kontekstno odvisni sistemi imajo pogosto kompleksne arhitekture in veliko podsistemov, ki skrbijo za predstavitev, upravljanje, sklepanje in analizo kontekstnih podatkov. Le z usklajenim sodelovanjem vseh teh podsistemov je mogoče doseči določeno funkcionalnost. Čeprav je bilo predstavljenih veliko različnih sistemov, pa vendar vsi vsebujejo vsaj štiri osnovne korake za upravljanje konteksta [7]. To so:

- pridobivanje konteksta,
- shranjevanje konteksta,
- abstrakcija konteksta in
- uporaba kontekstnih podatkov.

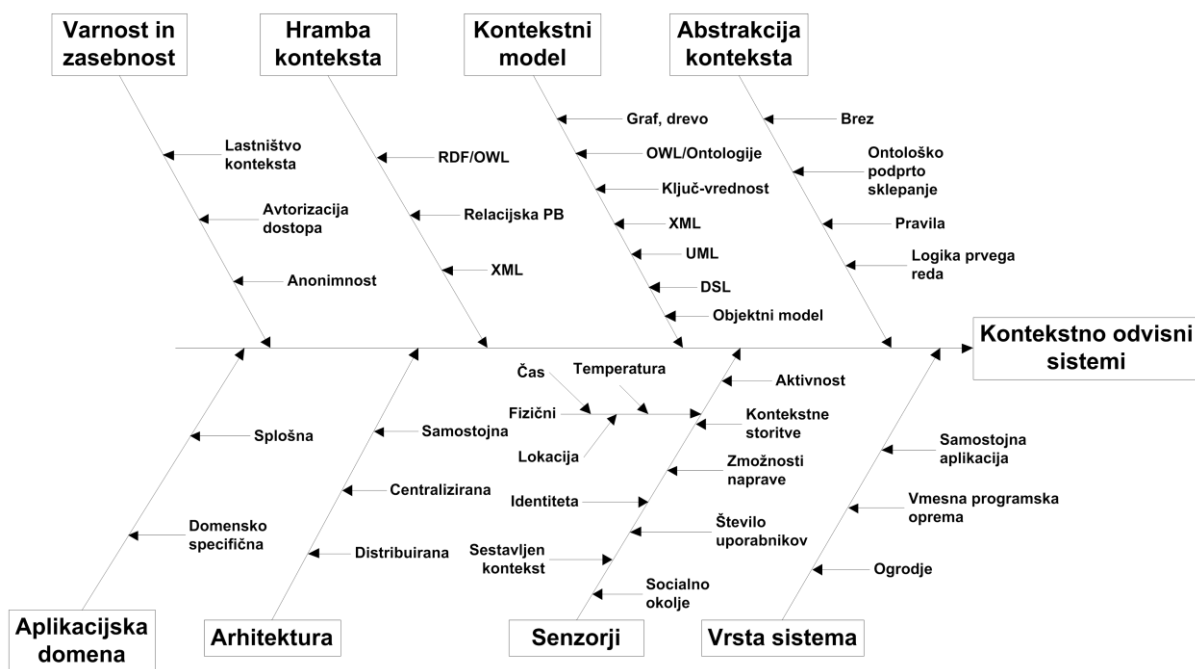
Prvi korak je pridobitev surovih kontekstnih podatkov z uporabo fizičnih ali virtualnih senzorjev. Ko so le-ti zajeti, jih sistemi pogosto shranijo v repozitorij. Za hrambo je treba podatke organizirati v podatkovne ali kontekstne modele. Naslednji korak je opcijski, saj ga uporabljajo le nekateri sistemi in se imenuje abstrakcija konteksta. Zadnji korak v tem procesu je uporaba surovih ali abstrahiranih kontekstnih podatkov v aplikacijah ali storitvah.

4 TAKSONOMIJA

V tem poglavju bomo predstavili in opisali osem kategorij, ki sestavljajo našo taksonomijo. Identificirali bomo vse kriterije in podali razloge, zakaj smo jih uvrstili v taksonomijo.

4.1 Vrsta sistema

Kontekstno odvisni sistemi so lahko, tako kot druga programska oprema, implementirani na različne načine.



Slika 1. Taksonomija kontekstno odvisnih sistemov

Zgodnji sistemi so bili implementirani kot samostojne aplikacije, ki delujejo na določeni napravi ali operacijskem sistemu. Posledično so prilagojeni na specifične senzorje in druge specifične strojne opreme. Vse to pa ima negativne posledice na prenosljivost in množično uporabo. To se novejši sistemi razvijajo v obliki ogrodij in vmesne programske opreme. Medtem ko je možnost ponovne uporabe pri prvih večja, pa se zadnji usmerjajo predvsem v uporabo kontekstna v spletnih storitvah in storitveno-usmerjeni arhitekturi [7]. Kot primere ogrodij lahko naštejemo Context Toolkit [8], Hydrogen [9] in CoBrA [10]. Na drugi strani, lahko najdemo tudi implementacije vmesne programske opreme kot sta Service-Oriented Context-Aware Middleware – SOCAM [11] in Gaia [12].

4.2 Aplikacijska domena

Z vrsto sistema je tesno povezana tudi aplikacijska domena. Medtem ko lahko za vse kontekstno odvisne sisteme, ki so implementirani kot samostojne aplikacije, z gotovostjo trdimo, da so domensko specifični, pa pri drugih arhitekturah opažamo čedalje večjo težnjo po splošnosti [7]. Zato ni presenetljivo sovpadanje s smernicami razvoja objektno-orientiranih sistemov, kjer se teži k uporabi ogrodij.

4.3 Arhitektura

Arhitekturne stile kontekstno odvisnih sistemov lahko povzamemo v tri kategorije: samostojne, distribuirane in centralizirane. Samostojna arhitektura je najpreprostejša, saj ima aplikacija neposreden dostop do senzorjev in ne omogoča skupne rabe konteksta med napravami. To vrsto arhitekture je mogoče hitro implementirati, a ima ob tem kar nekaj omejitev, saj je omejena le na lastno pridobljene kontekstne podatke. Ta je primerna le za manjše, preproste in domensko specifične aplikacije.

Distribuirani kontekstno odvisni sistemi lahko hranijo kontekstne podatke na več ločenih napravah, zato odpade potreba po dodatnem centralnem strežniku. Vsaka naprava zase je neodvisna, zato izpad posamezne naprave ne vpliva na delovanje sistema. Med slabosti te arhitekture lahko uvrstimo komunikacijske vezi med posameznimi napravami ali agenti. Le-te so pogosto implementirane kot ad hoc rešitve, ki pa jih je težko nadzirati in vzdrževati [7]. Mobilnim napravam namreč primanjkuje virov in računske moči za tako vrsto implementacije. Zato arhitektura distribuiranih sistemov ni primerna za implementacijo računske intenzivnih aplikacij.

V centralizirani arhitekturi so senzorji in naprave povezani s centralnim kontekstnim strežnikom, ki ima na voljo veliko virov in računske moči. Vsi kontekstni podatki so shranjeni tam. Kadar naprava potrebuje dodatne podatke, pošlje poizvedbo na centralni strežnik, to pa ji vrne rezultate povpraševanja. V tej arhitekturi vse povezave vodijo v centralni strežnik, kar omogoča njihovo poenostavitev in posplošitev. Posledica tega sta

lažja implementacija in vzdrževanje. Dodatna prednost je, da lahko v tak sistem na zelo preprost način dodajamo nove naprave in senzorje.

4.4 Senzorji

Senzorji so zelo pomemben del kontekstno odvisnih sistemov, saj skrbijo za zajemanje kontekstnih podatkov. Pogosto je treba zajemati podatke iz več različnih virov, jih združiti in obdelati, da pridobimo ustrezne kontekstne informacije. Podatke lahko zajemamo na več načinov, med njimi so najbolj izpostavljeni fizični in virtualni senzorji ter fuzija obeh zvrsti. V literaturi so najpogosteje omenjeni prav fizični senzorji. Med temi prednjači lokacija [4,5,13,14,15,16,17], kateri sledita čas [4,5,14,17] in temperatura [13]. Druge vrste senzorjev obsegajo še:

- identiteto uporabnika,
- njegove aktivnosti,
- zmožnosti naprave,
- število uporabnikov,
- socialno okolje,
- kontekstne storitve in
- sestavljen kontekst.

Čeprav identitete uporabnika oz. njegovega uporabniškega profila ni mogoče direktno zaznati s senzorji, je le prav, da ga vseeno uvrstimo v to kategorijo. V nemalo raziskavah [5,15,16,17] se je ta koncept izkazal kot ključen za učinkovito delovanje sistema. Uporabnik namreč z nastavitvami profila direktno vpliva na delovanje sistema. Lahko se celo zgodi, da uporabnik prestopi mejo konteksta in postane osrednji del sistema. V [18] sistem predlaga način komunikacije z osebo glede na njene uporabniške nastavitve, lokacijo in aktivnost. Podoben vpliv imata tudi število uporabnikov in socialno okolje, v katerem se uporabnik giblje [14].

Zmožnosti naprave se povezujejo s ponujeno vsebino. V [5] navajajo, da zmožnosti naprave vplivajo na rezultate poizvedb. Tu imajo pomemben vpliv velikost zaslona in podprte vsebine. Ko delamo z multimedijскими vsebinami, prikažemo le tiste, ki so primerne za določen tip naprave.

Kontekstne storitve, kot je bilo že omenjeno, so abstrakcija senzorjev in lahko služijo pridobivanju poljubnih kontekstnih podatkov. Novejši sistemi jih pogosto uporabljajo, saj ponujajo dodatno raven abstrakcije. Sestavljen kontekst, kot je predstavljen v [4], je sinteza več različnih kontekstnih podatkov, ki sami zase ne bi dajali enake stopnje informacij, kot če so povezani skupaj.

4.5 Kontekstni model

Kontekstni modeli so potrebni za shranjevanje in obdelavo kontekstnih podatkov. Izbira in razvoj modela je težavna, saj moremo pokriti čim več možnosti uporabe [19]. Baldauf et al. [6] povzemajo najbolj relevantne pristope k modeliranju konteksta, ki temeljijo na podatkovnih strukturah, uporabljenih za predstavitev

in izmenjavo kontekstnih podatkov. Ločijo šest kategorij:

- modeli ključ-vrednost,
- modeli iz označevalnih shem,
- grafični modeli,
- objektno orientirani modeli,
- modeli na podlagi logike in
- ontološko podprti modeli.

S pregledom literature smo pridobili lasten vpogled v modele kontekstnih ogrodij in aplikacij. Zato smo dopolnili omenjene kategorije, kot je prikazano v taksonomiji. Šesterici smo dodali graf oz. drevo, ki je bilo opisano v [5], in domensko specifičen jezik na katerega smo naleteli v [19]. Modela, ki bi temeljil na logiki, nismo zasledili. Zato in zaradi podobnosti s skupino ontološko podprtih modelov smo se odločili, da ga združimo v eno skupino.

Posebej želimo poudariti ContextUML [13]. To je na jeziku UML¹ temelječa notacija za opis kontekstnih podatkov. V novejši literaturi je opazna čedalje večja težnja k uporabi ontoloških modelov za opis konteksta. Ontologije namreč ponujajo način opisa konceptov in relacij med njimi. Zaradi njihove visoke stopnje izraznosti in zmožnosti uporabe sklepanja jih uvrščamo kot eno pomembnejših tehnologij na tem področju.

4.6 Abstrakcija konteksta

Kontekstne podatke lahko pridobimo neposredno iz senzorjev, a šele njihova abstrakcija privede do novih in bolj uporabnih spoznanj. Kot smo opisali v prejšnjem poglavju, obstaja veliko načinov, kako modelirati kontekst. Razumljivo je, da ima način opisa velik vpliv na zmožnost abstrakcije podatkov v sistemu. Kadar so kontekstni podatki opisani z ontologijo, abstrakcija v večini primerov poteka s semantičnim pristopom in ontološko podprtim sklepanjem.

Nad drugimi kontekstnimi modeli lahko za abstrakcijo uporabimo še pravila ali logiko prvega reda, a je sklepanje v teh primerih močno omejeno.

4.7 Hramba konteksta

Nekateri kontekstno odvisni sistemi hranijo kontekstne podatke v repozitorijih. Ti podatki se lahko uporabijo takoj ali pozneje ter povratno vplivajo na takrat veljavni kontekst. A načini hrambe so različni in močno povezani z izbranim kontekstnim modelom. Najpogosteje se uporabljajo relacijske podatkovne baze. Zelo pogosta je tudi serializacija v XML, ki se uporablja za distribucijo kontekstnih podatkov. S širšo uporabo ontologij se je povečalo tudi hranjenje v zapisih RDF². Kontekstno odvisni sistemi, ki delujejo na storitveni arhitekturi, raje uporabljajo spletne storitve za trajno hrambo kontekstnih podatkov.

4.8 Varnost in zasebnost

Zasebnost podatkov in s tem povezano varovanje je zelo pomemben aspekt kontekstno odvisnih sistemov. Nenehno spremljanje uporabnikov in obdelava njihovih podatkov lahko povzroči strah pri uporabnikih in posledično privede do odklona uporabe aplikacij ali naprav. Zaradi teh dejavnikov je treba v sistemih zagotoviti ustrezne mehanizme za varovanje in omejitev dostopa do teh podatkov. To še posebej velja v sistemih, ki shranjujejo kontekstne podatke v trajni obliki. Tukaj je treba poudariti razlike med klasičnim varovanjem podatkov v informacijskih sistemih in kontekstno odvisnih sistemih. Pri slednjih so pomembni predvsem vidiki do katerih kontekstnih podatkov lahko dostopajo aplikacije. Ali je podatek o lokaciji res potreben? Dovoliti dostop do uporabniškega profila? Kako se beleži zgodovina uporabe? Na vsa ta in številna druga vprašanja je treba odgovoriti, še preden se lotimo razvoja kontekstno odvisnih rešitev.

A na tem področju je opazno očitno pomanjkanje raziskav. Le redki avtorji omenjajo varnostne mehanizme za varovanje zasebnosti uporabnikov. V večini se osredinjajo le na arhitekturo in omogočanje kontekstno odvisnih funkcionalnosti. Langheinrich [20] opozarja na nevarnosti, ki prežijo na uporabnike vseprisotnih mobilnih rešitev. Med drugim poudarja nujnost obveščenosti uporabnika o tem, kateri podatki se zbirajo, in pridobiti njegovo potrditev za to. Hkrati poudarja še pomen anonimnosti in zaključuje z nekaj napotki za zbiranje podatkov:

- zbiranje podatkov samo za določen in točno definiran namen;
- zbiranje izključno podatkov, ki so potrebni za namen (in nič več);
- hraniti podatke le za čas, ki je potreben za namen.

Eden prvih, ki omenjajo zasebnost, je Dey [8]. V Context Toolkitu uvaja preprost mehanizem dostopa za varovanje zasebnosti uporabnikov v obliki lastništva konteksta. CoBrA [10] ponuja ontološko podprt jezik (REI), ki omogoča uporabnikom, da definirajo pravila in s tem kontrolirajo deljenje zasebnih podatkov.

5 KLASIFIKACIJA

Da bi preverili izdelano taksonomijo, smo se odločili izvesti klasifikacijo na primeru. Iz sorodnih del smo izbrali nekaj primerov kontekstno odvisnih sistemov in jih razvrstili glede na kategorije iz klasifikacije. Tabela 1 prikazuje to klasifikacijo.

¹ <http://www.uml.org/>

² <http://www.w3.org/RDF/>

Tabela 1: Klasifikacija kontekstno odvisnih sistemov na podlagi naše taksonomije

Kontekstno odvisni sistem	Vrsta sistema	Aplikacijska domena	Arhitektura	Senzorji	Kontekstni model	Abstrakcija konteksta	Hramba konteksta	Varnost in zasebnost
AMS[1]	Samostojna aplikacija	Domensko specifična	Samostojna	Lokacija	/	/	/	/
Context Toolkit [8]	Ogrodje	Splošna	Distribuirana	Lokacija, identiteta, čas, aktivnost	Ključ-vrednost	Pravila	/	Avtorizacija
Hydrogen [9]	Ogrodje	Splošna	Distribuirana	Lokacija, identiteta, čas	Objektni model	/	/	/
Gaia [12]	Ogrodje	Splošna	Distribuirana	Lokacija	Ključ-vrednost	Logika prvega reda	XML	/
CoBrA [10]	Ogrodje	Splošna	Centralizirana	Lokacija, aktivnost	Ontologije	Ontološko podprto sklepanje	RDF/OWL	Avtorizacija
SOCAM [11]	Vmesna programska oprema	Splošna	Centralizirana	Kontekstne storitve	Ontologije	/	RDF/OWL	/
Enhanced CoCA [21]	Ogrodje	Splošna	Distribuirana	Lokacija, aktivnost	Ontologije	Logika prvega reda	Relacijske PB	/
Context Management Framework [22]	Ogrodje	Splošna	Centralizirana	Lokacija, čas, temperatura, aktivnost, zmožnosti naprave	Ontologije	Ontološko podprto sklepanje	XML	/

6 SKLEP

V raziskavi smo se ukvarjali s kontekstom in kontekstno odvisnimi sistemi v računalništvu. Cilj je bil pridobiti pregled nad omenjenim raziskovalnim področjem. Zanimalo nas je, kaj sploh kontekst je in kako ga definirajo drugi raziskovalci. Potem smo si zastavili vprašanje, kako zaznati kontekst in kako ga lahko uporabimo za izboljšanje delovanja sistema. Da bi dobili odgovore na ta in podobna vprašanja, smo se lotili pregleda sorodne literature. Pri tem so bile arhitekture sistemov in uporabljene tehnologije v ospredju naše pozornosti.

Da bi bolje razumeli kontekst, smo najprej poiskali njegove definicije in definicije kontekstno odvisnih sistemov. Ugotovili smo, da imajo avtorji skupno razumevanje konteksta, a ga je težko izraziti. Najbolj razširjena in uveljavljena je definicija, ki sta jo podala Dey in Abowd [4]. Nato smo pogledali, kako so zgrajeni sistemi, ki temeljijo na kontekstu. Na podlagi pridobljenega znanja smo ugotovili kar nekaj sovpadanja med posameznimi sistemi. Zato smo izdelali taksonomijo področja, ki omogoča sistematičen pristop k analizi kontekstno odvisnih sistemov. Izbrali smo osem kategorij, s pomočjo katerih lahko natančno klasificiramo rešitve. Za lažjo predstavitev smo izdelali diagram ribje kosti, ki ponazarja izdelano taksonomijo.

Za konec lahko ugotovimo, da je področje kontekstno odvisnih sistemov še dokaj neraziskano. Možnosti za nadaljnje raziskave lahko najdemo v vsaki od navedenih kategorij. Mogoče je še največ dela

odprtega pri zajemanju konteksta. Trenutni sistemi sicer promovirajo različne vrste senzorjev, a na koncu implementirajo le redke. Potrebni so inteligentni mehanizmi, kako zajeti tudi druge uporabniške kontekste. Prav tako so odprta področja še na hrambi in obdelavi konteksta. Nazadnje je treba omeniti še varnost in zasebnost. Žal smo morali ugotoviti, da se le malo raziskav ukvarja s tem področjem.

LITERATURA

- [1] B. Schilit and M. Theimer, "Disseminating Active Map Information to Mobile Hosts," IEEE Network, vol. 8, no. 5, pp. 22-32, 1994.
- [2] P. J. Brown, J. D. Bovey, and X. Chen, "Context-Aware Applications: From the Laboratory to the Marketplace," IEEE Personal Communications, vol. 4, no. 5, pp. 58-64, 1997.
- [3] N. Ryan, J. Pascoe, and D. Morse, "Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant," in In Computer Applications in Archaeology, 1997.
- [4] Anind K. Dey and Gregory D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," in HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, 1999.
- [5] Christos Doukeridis, Nikos Loutas, and Michalis Vazirgiannis, "A System Architecture for Context-Aware Service Discovery," Electronic Notes in Theoretical Computer Science, vol. 146, no. 1, pp. 101-116, 2006.
- [6] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg, "A survey on context-aware systems," International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, no. 4, pp. 263-277, 2007.
- [7] Sankeun Lee, Juno Chang, and Sang-goo Lee, "Survey and Trend Analysis of Context-Aware Systems," Information-An International Interdisciplinary Journal, vol. 14, no. 2, pp. 527-548, 2011.

- [8] Anind K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001.
- [9] T. Hofer et al., "Context-awareness on mobile devices - the hydrogen approach," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003, pp. 6-9.
- [10] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 3, no. 18, pp. 197-207, 2003.
- [11] T. Gu, H.K. Pung, and D.Q. Zhang, "A middleware for building context-aware mobile services," in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, Milan, 2004.
- [12] Manuel Román, Christopher Hess, Renato Cerqueira, Roy H. Campbell, and Klara Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces," *IEEE Pervasive Computing*, vol. 1, pp. 74-83, 2002.
- [13] Quan Z. Sheng and Boualem Benatallah, "ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development," in *Proceedings of the International Conference on Mobile Business (ICMB '05)*, Washington, DC, 2005, pp. 206-212.
- [14] Albrecht Schmidt, "Implicit Human Computer Interaction Through Context," in *2nd Workshop on Human Computer Interaction with Mobile Devices*, Edinburgh, 1999.
- [15] Stephan Reiff-Marganiec and Hong Qing Yu, "Automated Context-aware Service Selection for Collaborative Systems," in *CAiSE '09 Proceedings of the 21st International Conference on Advanced Information Systems Engineering*, Amsterdam, 2009, p. 261.
- [16] Keita Fujii and Tatsuya Suda, "Semantics-based context-aware dynamic service composition," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, Maj 2009.
- [17] Nasser Ghadiri, Mohammad Ali Nematbakhsh, Ahmad Baraani-Dastjerdi, and Nasser Ghasem-Aghaee, "A Context-Aware Service Discovery Framework Based on Human Needs Model," *Lecture Notes in Computer Science*, pp. 404-409, 2007.
- [18] Jiehan Zhou and Jukka Riekkii, "Context-Aware Pervasive Service Composition," in *2010 International Conference on Intelligent Systems, Modelling and Simulation*, Liverpool, 2010, pp. 437-442.
- [19] Karen Henriksen and Jadwiga Indulska, "Developing context-aware pervasive computing applications: Models and approach," *Pervasive and Mobile Computing*, vol. 2, no. 1, pp. 37-64, 2006.
- [20] Marc Langheinrich, "Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems," in *UbiComp '01 Proceedings of the 3rd international conference on Ubiquitous Computing*, London, 2001, pp. 273-291.
- [21] Dejene Ejigu, Marian Scuturici, and Lionel Brunie, "CoCA: A Collaborative Context-Aware Service Platform for Pervasive Computing," in *Fourth International Conference on Information Technology, INT'07*, Las Vega, NV, 2007, pp. 297-302.
- [22] Panu Korpipaa, Jani Mantjarvi, Juha Kela, Heikki Keranen, and Esko-Juhani Malm, "Managing context information in mobile devices," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 42-51, July-September 2003.

Rok Žontar je diplomiral leta 2009 na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Zaposlen je kot raziskovalec po nacionalni shemi mladih raziskovalcev v isti ustanovi. Njegova raziskovalna zanimanja vključujejo kontekstno odvisne sisteme, modelno usmerjen razvoj, razvoj mobilnih aplikacij in druge.

Marjan Heričko je diplomiral leta 1989, magistriral leta 1993 in doktoriral leta 1998 na Fakulteti za elektrotehniko, računalništvo in informatiko. Zaposlen je kot redni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru.

Ivan Rozman je diplomiral leta 1977, magistriral leta 1980 in prejel doktorat znanosti leta 1983 na takratni Visoki tehniški šoli v Mariboru. Zaposlen je kot redni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko. Prav tako je predstojnik Inštituta za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko.