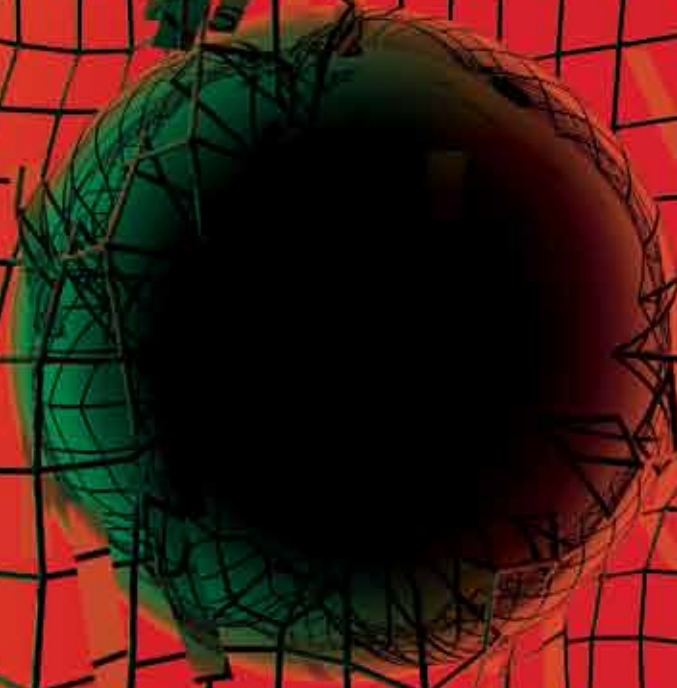


U P O R A B N A

# I N F O R M A T I K A

2011 ŠTEVILKA 4 OKT/NOV/DEC LETNIK XIX



# Izpitni centri ECDL

**ECDL** (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščen ustanova ECDL Foundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 11,6 milijona indeksov, v Sloveniji več kot 17.000 in podeljenih več kot 11.000 spričeval. Za izpitne centre v Sloveniji je usposobljenih 15 organizacij, katerih logotipe objavljamo.



# U P O R A B N A I N F O R M A T I K A

2011 ŠTEVILKA 4 OKT/NOV/DEC LETNIK XIX ISSN 1318-1882

## Uvodnik

## Znanstveni prispevki

- Matej Rebec, Cene Bavec:  
**Nekateri varnostni vidiki informacijskih tehnologij na področju turizma v Sloveniji** 185
- Boris Ovčjak, Gregor Jošt, Gregor Polančič, Marjan Heričko:  
**Primerjava učinkovitosti delovanja aplikacij v oblaku in v lokalnem okolju** 193
- Ivana Dvorski Lacković:  
**Human Resource Information Systems in Croatian Banks: current practice and trends** 207

## Pregledni znanstveni prispevki

- Saša Kuhar, Marjan Heričko:  
**Uporabnost metafor v informatiki** 215

## Informacije

- Sklepi konference Informatika v javni upravi 2011** 227
- Iz Islovarja** 229

#### Ustanovitelj in izdajatelj

Slovensko društvo INFORMATIKA  
Revija Uporabna informatika  
Vožarski pot 12, 1000 Ljubljana

#### Predstavniki

Niko Schlamberger

#### Odgovorni urednik

Jurij Jaklič

#### Uredniški odbor

Marko Bajec, Vesna Bosilj Vukšič, Gregor Hauc, Jurij Jaklič, Milton Jenkins, Andrej Kovačič, Katarina Puc, Vladislav Rajkovič, Heinrich Reinermann, Ivan Rozman, Rok Rupnik, Niko Schlamberger, John Taylor, Mirko Vintar, Tatjana Welzer Družovec

#### Recenzenti

Marko Bajec, Marko Bohanec, Vesna Bosilj Vukšič, Dušan Caf, Srečko Devjak, Tomaž Erjavec, Matjaž Gams, Izidor Golob, Tomaž Gornik, Janez Grad, Miro Gradišar, Jozsef Györkös, Marjan Heričko, Mojca Indihar Štemberger, Jurij Jaklič, Milton Jenkins, Andrej Kovačič, Jani Krašovec, Katarina Puc, Vladislav Rajkovič, Heinrich Reinermann, Ivan Rozman, Rok Rupnik, Niko Schlamberger, Tomaž Turk, Mirko Vintar, Tatjana Welzer Družovec, Lidija Zadnik Stirn, Alenka Žnidaršič

#### Tehnična urednica

Mira Turk Škraba

#### Oblikovanje

Bons  
Ilustracija na ovitku: Luka Umek za BONS

#### Prelom in tisk

Boex DTP, d. o. o., Ljubljana

#### Naklada

600 izvodov

#### Naslov uredništva

Slovensko društvo INFORMATIKA  
Uredništvo revije Uporabna informatika  
Vožarski pot 12, 1000 Ljubljana  
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljni izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revijo sofinancira Javna agencija za knjigo RS.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

© Slovensko društvo INFORMATIKA

## Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne članke domačih in tujih avtorjev z najširšega področja informatike v poslovanju podjetij, javni upravi in zasebnem življenju na znanstveni, strokovni in informativni ravni; še posebno spodbujamo objavo interdisciplinarnih člankov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov [ui@drustvo-informatika.si](mailto:ui@drustvo-informatika.si) ali po pošti na naslov Slovensko društvo INFORMATIKA, Vožarski pot 12, 1000 Ljubljana.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, objavljena v nadaljevanju.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Članki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni članek ponovno prejmejo v pregled. Uredništvo pa lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če članek ne ustreza kriterijem za objavo v reviji.

Pred objavo članka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost članka in dovoljuje prenos materialnih avtorskih pravic. Nenaročenih prispevkov ne vračamo in ne honoriramo. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke.

S svojim prispevkom v reviji Uporabna informatika boste prispevali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo.

Uredništvo revije

## Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in – kjer je mogoče – njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznih priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika Islovar ([www.islovar.org](http://www.islovar.org)).

Znanstveni članek naj obsega največ 40.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Članek naj bo praviloma predložen v urejevalniku besedil Word (\*.doc ali \*.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en prazen prostor, pri odstavkih ne uporabljajte zamika.

Naslovu članka naj sledi za vsakega avtorja polno ime, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir članka. Pred povzetkom v angleščini naj bo še angleški prevod naslova, prav tako pa naj bodo dodane ključne besede v angleščini. Obratno pa velja v primeru predložitve članka v angleščini.

Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Vsako sliko in tabelo razložite tudi v besedilu članka. Če v članku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slik zaslonov ne objavljamo, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema APA navajanja bibliografskih referenc, najpogostejše torej v obliki: (Novak & Kovač, 2008, str. 235). Na koncu članka navedite samo v članku uporabljeno literaturo in vire v enotnem seznamu po abecednem redu avtorjev, prav tako v skladu s pravili APA. Več o APA sistemu, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani <http://owl.english.purdue.edu/owl/resource/560/01/>.

Članku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

*Spoštovane bralke in spoštovani bralci,*

*leto končujemo s štirimi znanstvenimi prispevki z različnih področij informatike. Vsem je skupna značilnost, da je mogoča neposredna uporaba ugotovitev raziskovalnega dela v poslovni praksi. Avtorje znanstvenih prispevkov vedno usmerjamo tako, da svoje raziskave predstavijo sicer skladno z načeli znanstvenoraziskovalnega dela, vendar s posebnimi poudarki njihove uporabnosti. S tako usmeritvijo revija Uporabna informatika upravičuje svoje ime, omogoča objavo rezultatov raziskav v slovenskem jeziku v obliki znanstvenih člankov, hkrati pa se skuša približati široki množici svojih bralcev. Verjamemo, da tudi s tem povezujemo raziskovalce in gospodarstvo v Sloveniji.*

*Pomembno poslanstvo revije Uporabna informatika je skrb za uporabo lepega slovenskega jezika v okviru naše stroke. Temu namenjamo veliko pozornost, saj so vsi prispevki skrbno jezikovno pregledani, avtorje pa usmerjamo na uporabo terminološkega slovarja informatike Islovar, za katerega skrbi jezikovna sekcija Slovenskega društva INFORMATIKA. Že nekaj let v vsaki številki revije tudi redno objavljam izbrane sklope izrazov iz Islovarja. Še naprej vas vabimo k sodelovanju pri skupnem oblikovanju Islovarja.*

*Vam, bralcem, želim veliko zanimivega branja, Uporabni informatiki pa srečno v jubilejnem letu 2012.*

*Jurij Jaklič,  
odgovorni urednik*

# Najava

## konference Dnevi slovenske informatike 2012

Spoštovani,

obveščamo vas, da bo **od 16. do 18. aprila 2012** v Kongresnem centru Grand hotela Bernardin v Portorožu potekala že 19. konferenca **Dnevi slovenske informatike**.

Prireditelj konference je **Slovensko društvo Informatika**.

Vabimo vas k **oddaji prispevkov** za konferenco. Več informacij o konferenci (in oddaji prispevkov) bo v kratkem na voljo na spletni strani [www.dsi2012.si](http://www.dsi2012.si).

Slovensko društvo Informatika

# █ Nekateri varnostni vidiki informacijskih tehnologij na področju turizma v Sloveniji

<sup>1</sup>Matej Rebec, <sup>2</sup>Cene Bavec

<sup>1</sup>Ministrstvo za notranje zadeve, Policija; <sup>2</sup>Univerza na Primorskem, Fakulteta za management Koper  
matej\_rebec@yahoo.com; cene.bavec@guest.arnes.si

## Izvleček

V članku so povzeti rezultati empirične raziskave o varovanju informacijskih tehnologij v organizacijah s področja nastanitvenega turizma. Pokazalo se je, da so razlogi, da se le malo anketiranih turističnih podjetij odloča za uporabo posebnih programskih rešitev za menedžment odnosov s strankami, tudi v dilemah, povezanih z varnostjo sistema, zato smo podrobneje analizirali stanje na tem področju. V anketiranih podjetjih je s tehnološke plati kakovost varovanja pod slovenskim in tudi evropskim povprečjem. Posebno preseneča dejstvo, da nekatera podjetja ne uporabljajo najosnovnejših varnostnih ukrepov, povezanih z virusi in vohunskimi programi. Rezultati med drugim potrjujejo tudi hipotezo, da imajo podjetja z lastnimi strokovnjaki za informatiko bolj kakovostno varovanje in manj varnostnih težav, kot jih imajo druga.

**Ključne besede:** varovanje opreme, varnost informacijske tehnologije, varovanje podatkov, varnostne težave, turistične nastanitvene organizacije.

## Abstract

### Some Aspects of Information Technology Security in Tourist Organizations in Slovenia

In the paper we present results of empirical research on IT security issues in touristic organizations in Slovenia. The results indicate that one of the reasons that the majority of organizations do not use specialized Customer Relations Management systems lies among others in their concern over data protection and security of information systems. For that reason we analyzed present security issues. We confirmed that the level of security in tourist organizations was under Slovenian and the EU average. Some organizations do not use even basic technological security measures concerning virus and anti-spy programs. The results also confirm the hypothesis that organizations with their own IT staff have better system security and less security threats than others.

**Key words:** IT security, data security, security threats, tourist organizations.

## 1 UVOD

Svetovni trendi kažejo, da je turizem panoga, ki nenehno raste. To dejstvo potrjujejo podatki Svetovne turistične organizacije (UNWTO, 2011), ki za leto 2010 poroča o 935 milijonih mednarodnih turistov, do leta 2020 pa naj bi se to število skoraj podvojilo in preseгло milijardo in pol turistov. Podoben trend se bo zanesljivo nadaljeval tudi pri nas kljub 5,8-odstotnem upadu tujih gostov v letih 2008 in 2009 kot posledici svetovne gospodarske krize (STO, 2010). Čeprav je turizem naglo se razvijajoča gospodarska panoga, pa turistična podjetja delujejo v specifičnem in izjemno konkurenčnem okolju. Posebnost turističnega gospodarstva je njegova neposredna usmerjenost k strankam, pa naj gre za organizirane ali posamezne turiste (Kalakota in Whinston, 1997). Turistične organizacije seveda ugotavljajo, da lahko

dvignejo svojo konkurenčnost in si zagotovijo dolgoročen obstoj na trgu turističnih storitev predvsem s tem, da povečajo svojo vidnost na internetu in posvetijo izrazito individualno pozornost svojim strankam. Zato je postal menedžment odnosov s strankami (CRM – Customer Relationship Management) pomemben del poslovne strategije celotnega turističnega gospodarstva (Fox in Stead, 2001; Bull, 2003). Dolgoročno gledano pa uvedba sistema CRM zmanjšuje tudi stroške prodaje kapacitet in storitev posameznega namestitvenega podjetja z natančnejšim usmerjanjem v določeno stranko, hkrati pa omogoča hitro prilagajanje tržnim pogojem (Gray in Byun, 2001; Buhalis 2003).

Učinkovito sodelovanje s strankami seveda zahteva ustrezno informacijsko podporo. V preprostejših primerih zadostujejo preprostejše računalniške rešitve,

vendar se večje turistične organizacije v tujini praviloma odločajo za integralno informacijsko podporo, ki jo omogočajo specializirani informacijski sistemi za menedžment odnosov s strankami (CRM). Njegov temelj je enotna osrednja zbirka podatkov, ki je dostopna vsem, ki jo potrebujejo ali lahko vanjo prispevajo potrebne podatke (Rigby, Reichheld in Berez, 2002). Podjetje zapisuje vse demografske, pa tudi psihografske podatke o strankah ter odnose in stike zaposlenih z vsako stranko. S tem podjetje gradi informacijske temelje za bolj osebni pristop k stranki. Problemi menedžmenta odnosov s strankami v turizmu so podobni problemom v drugih panogah, ki obdelujejo veliko število strank, pa naj gre za organizacije ali posameznike. Pomembna razlika pa je v izboru osebnih podatkov, ki v turizmu bistveno presega običajne identifikacijske podatke. Turistične organizacije praviloma obdelujejo širok spekter osebnih podatkov, ki so povezani s konkretnimi potovanji ter z osebnimi željami in s preferencami posameznikov, kar omogoča oblikovanje individualnih ponudb, ki so v celoti prilagojene stranki.

Jedro informacijskih sistemov v turističnih in še posebno v nastanitvenih organizacijah, pa naj gre za enostavne evidence ali pa integrirane sisteme za menedžment odnosov s strankami, so osebni podatki, povezani s preferencami posameznega turista. Na videz ti podatki niso posebno občutljiva kategorija podatkov, vendar ta videz precej vara. Razkritje ali nedovoljena izmenjava teh podatkov je lahko za posameznike izrazito neprijetna. Tega se zavedajo tudi turistične organizacije, ki se med drugim izogibajo morebitnim odškodninskim zahtevkom (Rosen, 2000; Istraturist, 2011). Veliko občutljivih osebnih podatkov postavlja te organizacije in njihove informacijske sisteme pred zahtevne varnostne izzive. Na načelni ravni se postavlja že vprašanje, koliko podatkov sploh zbirati in katere, da se po nepotrebnem in pretirano ne poseže v zasebnost strank (Zorman, 2001). Večja kot je količina osebnih podatkov, večjo zakonsko in etično odgovornost si nalagajo turistične organizacije, kar se odraža predvsem v vedno večjih zahtevah po ustreznih varnostnih ukrepih. Pri tem ne kaže pozabiti, da je zbiranje osebnih podatkov zakonsko urejeno, zato njihova nenamenska uporaba in slabo varovanje lahko vodita v materialne in poslovne posledice za organizacijo, ki ji dokažejo kakršno koli zlorabo ali malomarnost.

Vendar vsa turistična podjetja tega področja ne obvladujejo tako, kot bi pričakovali in želeli. To velja tako za organizacijske kot za tehnološke varnostne ukrepe (Bin, 2010). Varovanje informacijskih sistemov je kompleksno področje, ki pa je zanesljivo le toliko, kot je zanesljiv najšibkejši člen v verigi varovanja (Bottitta, 2005). Pogosto se namreč pokaže, da se lahko kompleksni in na videz zelo zanesljivi sistemi zrušijo zaradi enega samega slabega člana v omenjeni verigi. Zato ne kaže podcenjevati tudi nekaterih na prvi pogled trivialnih problemov ali izzivov. Posebno je treba poudariti, da veliko organizacij – ne samo turističnih – ni sposobnih vpeljati niti temeljnih varnostnih ukrepov, za katere bi povprečen računalniški strokovnjak mislil, da so razumljivi sami po sebi. V to kategorijo spada elementarna zaščita pred virusi, trojanskimi konji in podobnimi zlonamernimi kodami, šifriranje podatkov in tudi elektronski podpisi. Če se omejimo predvsem na tehnološke vidike, potem lahko naštejemo celo vrsto razlogov, ki zmanjšujejo varnostno raven. Perše (2000) navaja preslabo varnostno in tehnološko opremljenost in nezmožnost uvažanja novih varnostnih standardov. Lawrence, Loeb, Lucyshyn in Richardson (2004) omenjajo zaščito podatkov in fizičnih komponent sistema pred nenaumno ali namerno zlorabo. Acar in Michener (2002) izpostavljata problem, da še vedno mnoga podjetja o varnosti razmišljajo kot o kategoriji, ki bo dodana, ko bodo imeli čas ter ustrezna sredstva ipd.

Samo kot primer omenimo raziskavo o osnovni zaščiti informacijske in komunikacijske tehnologije in težavah podjetij z varnostjo, ki jo je objavil Eurostat (2005), v kateri je ocenjena tudi Slovenija. Po tej raziskavi je bila Slovenija skoraj po vseh merilih nekje v povprečju EU, kljub temu pa so nekateri podatki zaskrbljujoči. Le 78 odstotkov podjetij, ki imajo več kot 10 zaposlenih in ne spadajo v finančni sektor, je v zadnjih treh mesecih obnovilo sistemsko zaščito. Posledica je, da je v Sloveniji 31 odstotkov podjetij imelo varnostne težave, precej jih je izgubilo vsaj nekaj podatkov zaradi virusov, med 1 in 2 odstotkoma pa jih je zabeležilo zlorabo osebnih podatkov, zlorabe pri plačilu, nepooblaščen dostope in izsiljevanje in grožnje. Čeprav je od raziskave preteklo že precej let, so rezultati indikativni in kažejo presenetljivo visok odstotek elementarnih varnostnih težav. Predvidevanja strokovnjakov za leto 2007 kažejo celo na povečanje števila kršitev na področju varovanja podatkov. Problem varovanja podatkov naj bi se celo povečal



leta 2008. Ocenjujejo, da je leta 2007 utrpelo težave z varovanjem podatkov 79 do 162 milijonov uporabnikov (Coenen, 2008).

Omenjene ugotovitve so nas spodbudile, da smo opravili raziskavo o nekaterih vidikih varovanja opreme in podatkov v izbranih turističnih organizacijah v Sloveniji. Zanimalo nas je, kaj mislijo o specializiranem sistemu za vodenje odnosov s strankami in kje vidijo zadržke, probleme in morebitne omejitve. Na podlagi predhodnih neformalnih pogovorov v turističnih organizacijah smo namreč dobili vtis, da je eden od temeljnih razlogov za omahovanje pri uvajanju kompleksnejših informacijskih rešitev očitno nezaupanje v varnostne mehanizme in občutek tveganja. Zato smo želeli preveriti, kako v sedanjih razmerah varujejo svojo opremo in posledično osebne podatke ter kakšni so trenutno njihovi varnostni izzivi in problemi.

## 2 PREDSTAVITEV REZULTATOV

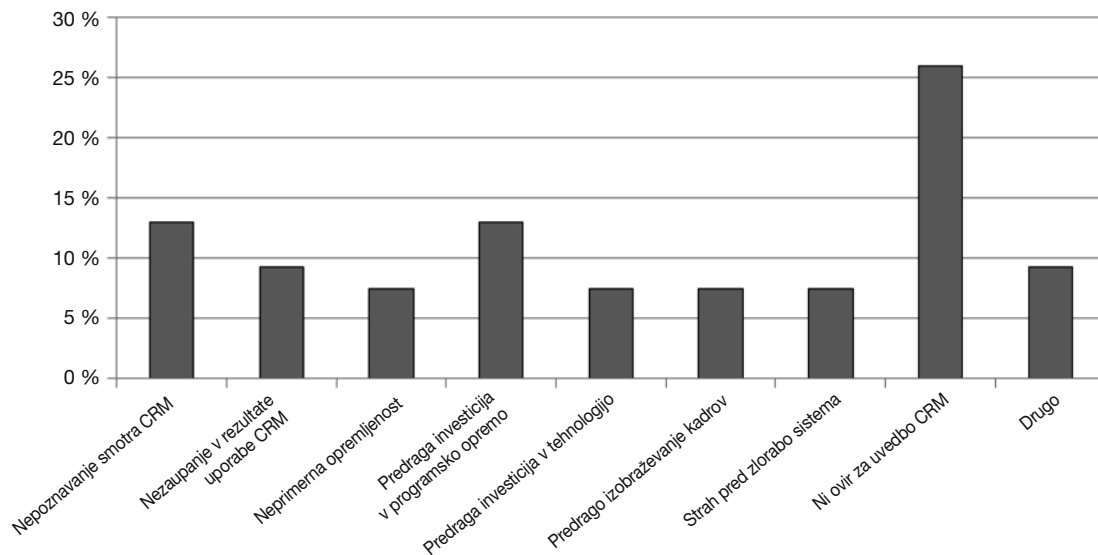
V raziskavi smo se omejili le na tisti del organizacij, ki so ponudnice turističnih namestitev (hoteli, moteli in penzioni). Anketni vprašalnik je bil sestavljen iz osemnajstih vprašanj zaprtega tipa in petih sklopov: splošni podatki o podjetju, stanje informacijske opreme v podjetju, varovanje elektronskega poslovanja v podjetju, podatki o sistemu CRM ter glavne ovire in prednosti za vzpostavitev sistema CRM. Od 184 ponudnikov turističnih namestitev je anketni vprašalnik v celoti izpolnilo le 43 podjetij. Izbrani vzorec je vključeval ponudnike turističnih namesti-

tev v Sloveniji, objavljene v hotelskem ceniku za leto 2007. V raziskavo smo vključili samo ponudnike, ki imajo med podatki navedeno elektronsko pošto, prek katere smo anketirali.

Pri tem smo se srečali s podobnim problemom kot druge raziskave, ki so se spuščale v razmeroma občutljive podrobnosti in probleme, ki jih podjetja nerada razkrivajo. Poleg tega so bila vprašanja precej tehnično obarvana, kar je brez dvoma povzročalo probleme tistim organizacijam, ki nimajo svojih strokovnjakov s področja informatike. Kljub vsemu je bil vzorec dovolj velik, da smo lahko opravili signifikantne statistične obdelave. Del podatkov pa smo pridobili iz sekundarnih informacijskih virov, predvsem EU.

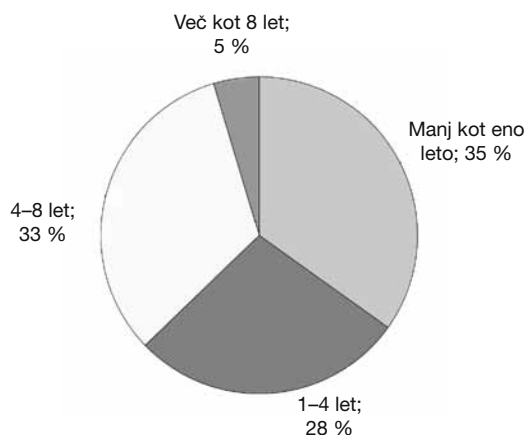
Anketirancem smo morali pojasniti, kaj si predstavljamo pod izrazom menedžment odnosov s strankami. Zavedali smo se, da imajo določene podatke o strankah vse anketirane organizacije in da je za marsikoga že enostavna evidenca zadostna informacijska podpora menedžmentu odnosov s strankami. Druga skrajnost pa so specializirani informacijski sistemi za menedžment odnosov s strankami. Zanje smo uporabili kratico CRM, kot je to običajno v jeziku ponudnikov takih rešitev.

Glede na to, da smo že pred raziskavo vedeli, da le malo turističnih namestitvenih organizacij uporablja specializirane sisteme, so nas najprej zanimali vzroki za tako stanje. Ugotovili smo, da le 7 odstotkov anketiranih podjetij uporablja omenjene sisteme. Poskušali smo analizirati, kje so razlogi za tako zadržanost (slika 1). Le 26 odstotkov organizacij

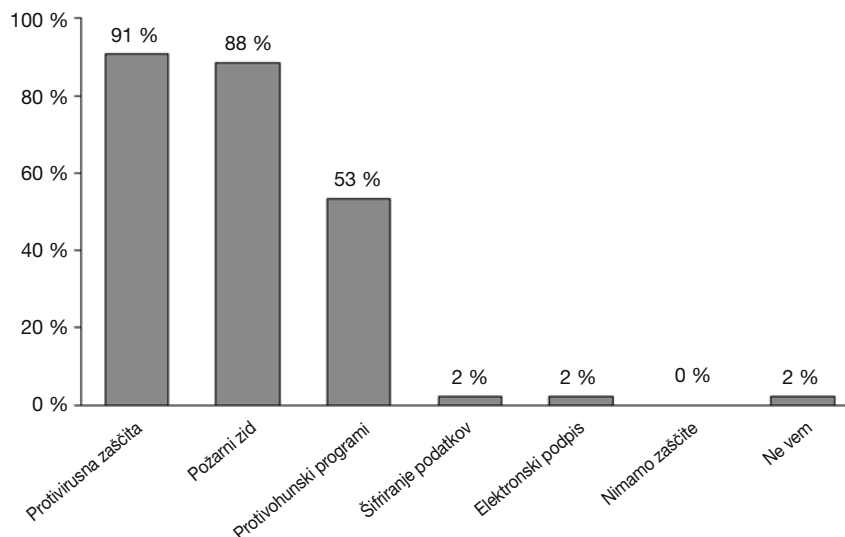


Slika 1: Razlogi za neuporabo posebnih programskih rešitev za vodenje CRM (Vir: raziskava)

ni omenjalo ovir za uvedbo CRM, ostale tri četrtine so omenjale različne razloge, ki so podjetja odvrčali od uvedbe CRM. Če odštejemo 13 odstotkov podjetij, ki so kot razlog za neuporabo navedla nepoznavanje, so vsa druga tako ali drugače omenjala probleme z opremo in s kadri ter nezaupanje in strah pred zlorabo. Vsi omenjeni zadržki so neposredno ali pa vsaj posredno povezani z varnostnimi vidiki. To ne velja le za računalniško opremo, temveč tudi za kadre, brez katerih seveda ne moremo izvajati potrebnih varnostnih ukrepov. Odgovori so razmeroma enakomerno porazdelili probleme z neprimernostjo obstoječe računalniške opreme, predrago investicijo v programsko in strojno opremo ter kadre, strah pred zlorabo in nezaupanje v rezultate uporabe CRM.



Slika 2: Povprečna starost računalniške opreme v anketiranih podjetjih (Vir: raziskava)

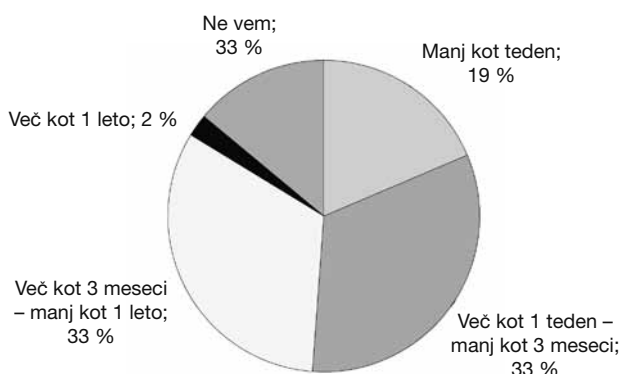


Slika 3: Varovanje računalniške opreme v anketiranih podjetjih (Vir: raziskava)

Tehnološko plat raziskave smo začeli z analizo starosti opreme, saj smo predpostavljali, da je starost opreme eden od dejavnikov, ki vpliva na varnost (slika 2). Več kot tretjina obstoječe opreme je starejša od štirih let, tretjina pa je stara manj kot leto. Ker v raziskavi nismo šli v podrobnosti, katera oprema se za kaj uporablja, lahko le sklepamo, da je starejša oprema predvsem v funkciji osebne uporabe in ne strežnikov, ki so nosilci skupnih podatkov. Vendar tega nismo posebej preverjali. Čeprav gre za subjektivno oceno, lahko vseeno ugotovimo, da je oprema v anketiranih podjetjih razmeroma stara in lahko pomeni težavo pri vzpostavljanju ustreznih varnostnih mehanizmov.

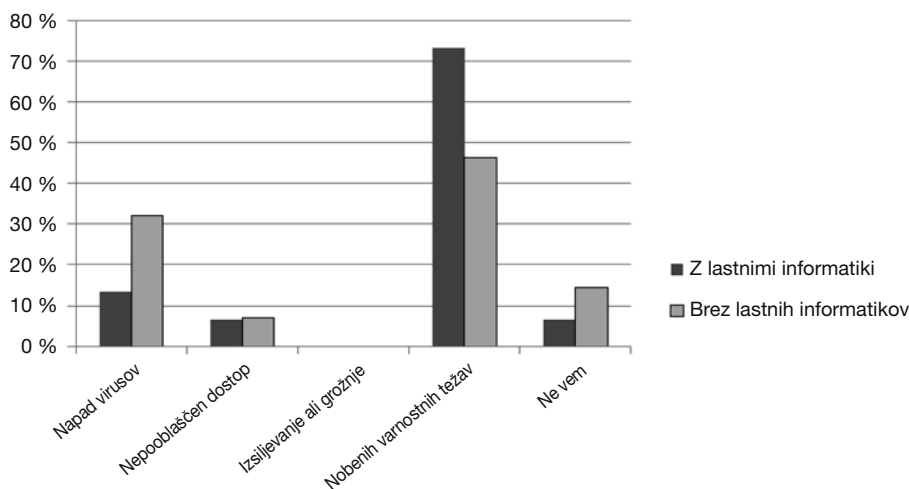
Seveda pa je varnost odvisna tudi od varnostne programske opreme. Držali smo se metodologije EU, da smo lahko rezultate anketiranih turističnih nastanitvenih podjetij postavili v širši kontekst. Slika 3 kaže, da je uporaba protivirusne zaščite in požarnih zidov na videz zelo široka, kljub temu pa presenetljivo veliko podjetij ne uporablja niti najbolj osnovnih tehnik varovanja. To je težko sprejeti ali razumeti, saj gre v njihovem primeru za obdelavo osebnih podatkov. Če k temu dodamo, da le malo več kot polovica podjetij uporablja zaščito proti vohunskim programom, lahko ugotovimo, da je varnost precej problematična. Zanimivo je tudi, da skoraj nihče ne šifrira podatkov, s čimer bi lahko bistveno povečali varnost osebnih podatkov (Min, 2010). Zato lahko podvomimo v učinkovito varovanje podatkov v večini anketiranih podjetij. Ti rezultati lahko pojasnijo tudi strah anketirancev pred zlorabo sistema CRM (slika 1), saj se očitno zavedajo, da njihova trenutna oprema ni dovolj varovana.

Statistika EU iz leta 2005 je pokazala, da v Sloveniji skoraj četrtina podjetij v zadnjih treh mesecih ni obnovila zaščite. V primeru anketiranih podjetij pa je stanje celo slabše, saj jih v zadnjih treh mesecih kar tretjina ni obnovila zaščite, kar ponovno kaže, da je stanje na področju varovanja pod slovenskim in seveda tudi evropskim povprečjem.



Slika 4: Nazadnje obnovljena zaščita v anketiranih podjetjih (Vir: raziskava)

V končni fazi pa so nas zanimale posledice omejenih varnostnih problemov in konkretne varnostne težave. Pokazalo se je, da je imelo v celoti 33 odstotkov anketiranih podjetij varnostne težave, kar je zopet občutno slabše od slovenskega povprečja. Omeniti pa je treba, da so imela podjetja z lastnimi zaposlenimi informatiki bistveno manj varnostnih težav kot podjetja brez svojih informatikov (slika 5). Ta rezultat je presenetljiv, saj zunanje izvajanje teh dejavnosti običajno ne postavlja v ospredje večjega varnostnega tveganja. Lahko le predpostavljamo, da anketirana podjetja nimajo ustrezno urejenih dogovorov in odnosov z zunanjimi strokovnjaki s področja informatike.



Slika 5: Odstotek anketiranih podjetij z zaposlenimi informatiki in podjetji brez lastnih informatikov, ki so zaznale varnostne težave (Vir: raziskava)

V slikah 1 do 5 smo predstavili neposredne rezultate raziskave. Vendar smo poskušali analizirati tudi določene vzročne in funkcijske povezave. Zbrane podatke smo uporabili zato, da smo analizirali povezave med varnostnimi težavami, starostjo opreme ter ravno in kakovostjo zaščite. Vpeljali smo štiri nove agregatne spremenljivke, ki smo jih izračunali iz osnovnih podatkov, pridobljenih z anketo:

- *varnostne težave* – podjetja smo razvrstili v dve skupini. V prvi so podjetja, ki niso ugotovila no-

benih varnostnih težav, v drugi pa tista, ki so ugotovila katero od težav (slika 5);

- *varnostna zaščita* – tudi v tem primeru so podjetja razdelili v dve skupini. V prvi so podjetja z dobro varovano opremo, ki uporabljajo vsaj tri varnostne metode (slika 3), kot slabo varovano opremo pa smo šteli primere, ko ima podjetje le dva zaščitna ukrepa ali manj;
- *pogostnost obnavljanja zaščite* – v skladu z metodologijo EU je ena skupina podjetij redno obnavljala

zaščito, če so to storila manj kot v treh mesecih, in neredno, če tega niso storila v zadnjih treh mesecih (slika 4);

- *starost opreme* – podjetja, ki imajo opremo, mlajšo od štirih let, so bila v eni skupini, v drugi pa podjetja z opremo, ki je starejša od štirih let (slika 2). Soodvisnost izbranih spremenljivk smo ocenjevali s hi-kvadrat ( $\chi^2$ ) testom. Ugotovili smo:

- soodvisnost med *varnostnimi težavami* in *varnostno zaščito* je statistično značilna ( $p < 0,01$ ), saj je kontingenčna tabela stabilna, ker znaša frekvenca 5,97 in zadošča pogoju o minimalni frekvenci večji od 5. S tem smo pokazali, da tudi v primeru anketiranih turističnih podjetij velja, da boljša varnostna zaščita zmanjša varnostne težave in seveda obratno.
- statistično je značilna tudi soodvisnost med *varnostnimi težavami* in *pogostnostjo obnove zaščite* ( $p < 0,01$ ), saj je tudi v tem primeru kontingenčna tabela stabilna (frekvenca znaša 6,12). Tako smo pokazali, da se podjetja, ki neredno obnavljajo opremo, pogosteje srečujejo z varnostnimi težavami in obratno.
- Kot zadnji test smo analizirali soodvisnost med *starostjo opreme* in *pogostnostjo obnavljanja zaščite*. Test je potrdil soodvisnost (frekvenca znaša 7,78 in  $p < 0,05$ ), kar pomeni, da uporabniki z novejšo računalniško opremo pogosteje obnavljajo zaščito informacijskega sistema. Predpostavljamo, da se novejša oprema uporablja za zahtevnejše in posledično bolj varovane obdelave.

Omenjene soodvisnosti so sicer pričakovane in jih zasledimo tudi v drugi panogah, vendar se nam je zdelo pomembno, da jih kljub temu potrdimo tudi za anketirana turistična podjetja, saj pomenijo pomemben argument za celovitejši pristop k uvajanju in izvajanju ustreznih varnostnih ukrepov.

### 3 POVZETEK UGOTOVITEV

Slovenska nastanitvena turistična podjetja skoraj ne uporabljajo specializiranih sistemov CRM in kažejo vrsto zadržkov proti njihovi uporabi, saj je ta v precejšnji meri povezana tudi z varnostnimi vidiki. Vendar je glavni vzrok še vedno dejstvo, da so to globalno gledano razmeroma majhna in nepovezana podjetja, ki imajo temu primerne tudi finančne in tehnološke omejitve. To se odraža v razmeroma velikem odstotku podjetij brez lastnih kadrov s področja informatike in zanje drage programske ter strojne opreme. Kljub temu da ne uporabljajo integriranih

sistemov CRM, pa vodijo veliko evidenc o svojih strankah, vendar to počno na preprostejši način, ob uporabi neintegriranih programskih rešitev. Tako razdrobljene aplikacije praviloma pomenijo tudi težje vzdrževanje in predvsem težji nadzor nad njihovo uporabo. Med te probleme lahko uvrstimo tudi varnostne ukrepe. Ne glede na to, kakšno programsko opremo uporabljajo, raziskava kaže, da v povprečju varnostnim vidikom ne posvečajo dovolj pozornosti. Varnostna zavest je na splošno pod slovenskim povprečjem. Poudariti pa je treba, da je med anketiranimi podjetji tudi nekaj takih, za katere lahko trdimo, da so odlična na področju varovanja.

Ker je večina anketiranih turističnih namestitvenih podjetij manjših in si ne morejo privoščiti lastnih računalniških strokovnjakov, mora menedžment ustrezno urediti sodelovanje z zunanjimi izvajalci in njihovimi strokovnjaki. Rezultati naše raziskave, ki kažejo izrazito manj varnostnih težav v podjetjih z lastnimi računalniškimi strokovnjaki, kot v podjetjih, ki jih nimajo, pa odpirajo vrsto problemov in vprašanj, povezanih z menedžmentom zunanjega izvajanja. Seveda ne kaže podcenjevati prednosti kakovostnih lastnih strokovnjakov, vendar si tega očitno ne morejo privoščiti vsi. Delo z zunanjimi izvajalci ne bi smelo vplivati na kakovost storitev ali na raven varovanja. Po naši oceni je temeljni problem v večini analiziranih podjetij v organizaciji zunanjega izvajanja, kar se v končni fazi odraža na povsem tehnoloških področjih.

Z računalniškimi strokovnjaki je povezan problem uporabljene varnostne programske opreme, njeno posodabljanje in vzdrževanje. Rezultati raziskave kažejo, da v veliko podjetjih računalniški strokovnjaki svoje funkcije ne opravljajo prav vzorno. Samo sklepamo lahko, da je problem tudi v njihovi usposobljenosti. Razlago lahko najdemo tudi v izrazito nizkih investicijah v opremo, 21 odstotkov podjetij omenja predrago tehnologijo, saj s staro in neustrezno opremo ni mogoče zagotoviti potrebne varnosti. Zanimiva je povezava med starostjo opreme in pogostnostjo posodabljanja programske opreme, saj potrjuje domnevo, da anketirana podjetja generalno podcenjujejo pomen informacijske tehnologije za svoje poslovanje ali pa so tako na robu rentabilnosti, da si ne morejo privoščiti sodobne opreme in ustreznih kadrov. To, da je tretjina opreme starejše od štirih let, samo potrjuje navedeno domnevo. Posebno so presenetljivi tisti rezultati, ki kažejo, da mnoga

podjetja ne uporabljajo tudi najbolj osnovne varnostne zaščite (protivirusni programi, požarni zidovi in protivohunski programi), ki v splošnem predstavlja izjemno majhen strošek. To pa je, brez dvoma, napačna računalniških strokovnjakov.

Če poskušamo umestiti rezultate naše raziskave v širši slovenski in evropski kontekst, lahko omenimo raziskavo, ki jo je opravilo dansko podjetje Secunia (RIS, 2008), ki ugotavlja, da je večina računalnikov ogroženih prav zaradi pomanjkljivega posodabljanja programske opreme z izboljšanimi različicami, saj so virusi zadnje generacije sposobni obiti mnoge protivirusne zaščite, jih celo izključiti oz. onesposobiti (Androutsellis, Spinellis in Vlachos, 2010). Sicer jih protivirusna zaščita odkrije, vendar ne vsebuje orodij za preprečitev škodljivih posledic. Po mnenju Secunie bi morali posodobitve nameščati vsaj enkrat mesečno. Tega merila pa ne izpolnjuje več kot polovica anketiranih podjetij. To pomeni, da je bil naše trimesečno merilo za sprejemljivo pogostnost obnavljanja zaščite opazno preblago. Če bi se odločili, da je meja te sprejemljivosti le en mesec, bi bili rezultati raziskave precej bolj neugodni. Vendar je to razmeroma lahko rešljiv problem, saj ne zahteva posebnih vlaganj, zahteva samo urejenost na tehnološki in organizacijski ravni.

#### 4 SKLEP

Ko smo začeli z raziskavo, smo predpostavljali, da vidijo slovenska turistična podjetja v sistemih CRM rešitev svojih tehnoloških in organizacijskih problemov, povezanih s poslovanjem s strankami. Pokazalo se je, da je za večino podjetij to le oddaljena možnost, na katero v bližnji prihodnosti ne računajo. Njihova organizacijska razdrobljenost in razmeroma draga tehnologija sta glavni oviri. Če bodo želela dolgoročno izrabiti tržne prednosti, ki jih potencialno dajejo sistemi CRM, bodo najbrž morala iskati rešitve tudi v medsebojnem sodelovanju na tehnološki ravni. Vendar bodo morala pri tem veliko večjo pozornost posvetiti konceptualnim vprašanjem, povezanim z informacijskimi sistemi in ustrezno tehnologijo. Med ta vprašanja spada tudi varnost opreme in predvsem osebnih podatkov, saj trenutno stanje v večini analiziranih turističnih organizacij ni zadovoljivo.

Čeprav smo v raziskavi analizirali predvsem tehnološke vidike sistemov ter varovanja opreme in podatkov, pa je varovanje v bistvu zadeva vodstva podjetja. Po eni strani mora vodilni menedžment

zagotoviti organizacijske in seveda finančne pogoje, da sistem varovanja sploh lahko zaživi, ne glede na tehnološke rešitve. Po drugi strani pa mora problem v okviru svojih kompetenc tudi razumeti in nadzorovati. Menedžment mora sprejeti dejstvo, da sta razvoj in delovanje informacijskih sistemov, ki vsebujejo individualne osebne podatke, kočljiva zadeva, ki je med drugim tudi zakonsko regulirana in sankcionirana. Te odgovornosti pa ne more prenesti na računalniške strokovnjake. Nepoznavanje in nezaupanje v informacijski sistem ter strah pred varnostnimi problemi in potencialno zlorabo je seveda nekaj, s čimer se mora poleg informatikov spopasti predvsem menedžment. Informatiki morajo zagotoviti samo ustrezno tehnično zaščito in jo vgraditi v delovne procese.

Za turistična podjetja je značilno, da so komunikacije med njimi in strankami pogosto dvosmerne, saj v optimalnih razmerah omogočajo stranki, da si sama kreira svoj profil in določa svoje individualne želje. To pa je seveda mogoče le prek interneta in razmeroma odprtih in uporabniško prijaznih programskih rešitev. Taki sistemi pa so notorično varnostno ranljivi, še posebno v primeru manjših turističnih podjetij, ki si ne morejo privoščiti vrhunskih programskih in strojnih zaščit. Zato je očitno, da je treba najti kompromis med varnostjo in stroški obratovanja, ki bo zagotovil potrebne varnostne standarde. Marsikaterih ukrepov, ki ne zahtevajo velikih vlaganj, bi pa bistveno povečali varnost sistemov, velika večina podjetij ne uporablja. Omenimo samo šifriranje podatkov in ne nazadnje tudi varne elektronske podpise.

Kakovostno varovanje opreme in podatkov pa ima tudi drugo dimenzijo. Posamezniki so vedno bolj osveščeni o varnostnih vidikih interneta in na njem zasnovanih storitev, zato se glas o neustreznem varovanju lahko kaj hitro razširi. Izkušnje kažejo, da se v takih primerih zanimanje za določeno podjetje dramatično zmanjša, posledično pa se bistveno zmanjša poslovna uspešnost. Zato je dobro varovanje tudi dobra marketinška poteza, ki jo podjetja, ki delujejo v tako konkurenčnem okolju, ne bi smela zanemariti.

V raziskavi se s tem problemom sicer nismo posebej ukvarjali, vendar je z varovanjem podatkov posredno povezan tudi izbor in obseg individualnih podatkov, ki jih zbira in obdeluje posamezno podjetje. Po eni strani se s povečanim obsegom individualnih podatkov povečuje možnost zlorab v primeru vdora v sistem, povečuje pa se tudi možnost zlorab

v podjetju. S tem pa se že dotikamo poslovne filozofije in etike podjetij, ki je povezana z namenom in načinom uporabe podatkov ter seveda tudi z implementacijo sistemov CRM. Če k temu dodamo različne marketinške in tehnološke trike, povezane z iskanjem podatkov na internetu, s katerimi se samodejno zbirajo informacije o potencialnih in tudi konkretnih strankah, potem postaja vedno bolj jasno, da je problem zbiranja in varovanja osebnih podatkov v turističnih organizacijah širok in resen. Tehnološki vidiki varovanja so le vrh ledene gore in predstavljajo le manjši del problemov, s katerimi se srečujejo turistične organizacije. Ustrezna tehnologija, ki jo simbolizirajo sistemi CRM, pa lahko pomaga pri učinkovitem reševanju teh problemov.

Omeniti je treba tudi dve pomembni omejitvi naše raziskave. Najprej je bila omejena na razmeroma majhno število organizacij, ki so sodelovale v njej, saj je bila iz že omenjenih razlogov stopnja odgovarjanja razmeroma nizka. Rezultati so sicer statistično signifikantni, vendar bi večji vzorec omogočal podrobnejši vpogled v strukturo posameznih odgovorov. Druga značilnost izbranega vzorca pa je velikost organizacij, saj so individualni ponudniki turističnih nastanitvenih storitev praviloma manjše organizacije, ki imajo omejene možnosti za uvajanje zahtevnejših tehnoloških rešitev, kot so integrirani sistemi CRM. Zato se sama po sebi ponuja mogoča rešitev v združevanju sil na tem področju, zunanem izvajanju v skupnih centrih, v uporabi enotnih programskih rešitev, skupnem izobraževanju, svetovanju ipd. Vendar bi bilo treba tudi v tem primeru vložiti precej naporov v iskanje enotnih organizacijskih in tehničnih rešitev, ki bi ustrezale vsem udeležencem, predvsem pa rešitev, ki bi zagotovile potrebno varovanje podatkov.

## 5 VIRI IN LITERATURA

- [1] Acar, Tolga in John R. Michener. 2002. *Risks in Features vs. Assurance*. CACM Inside Risks. <http://www.csl.sri.com/users/neumann/insiderisks.html>.
- [2] Androutsellis, Stefanos, Diomidis Spinellis in Vasileios Vlachos. 2010. Biological Aspects of Computer Virology. *Social Informatics and Telecommunications Engineering* 26: 209–219.
- [3] Bin Ibrahim, Al-Fattah. 2010. *Customer profile management system using CRM*. Faculty of Computer Science and Information Systems, Technological University of Malaysia.
- [4] Bottitta, Silvestro. 2005. *Una Revisione, Caratterizzazione ed Implementazione per la Comprensione del Concetto di Trust*. [http://homepages.inf.ed.ac.uk/mfelici/TrustGames/tesi\\_sb.pdf](http://homepages.inf.ed.ac.uk/mfelici/TrustGames/tesi_sb.pdf).
- [5] Buhalis, Dimitrios. 2003. *eTourism: information Technology for strategic Tourism Marketing*. Upper Saddle River (New Jersey): Prentice-Hall.
- [6] Bull, Christopher. 2003. Strategic issues in customer relationship management (CRM) implementation. *Business process management journal* 9(5): 592–602.
- [7] Coenen, Tracy. 2008. *More Data Security Problems Expected in 2008*. <http://www.allbusiness.com/crime-law-enforcement-corrections/criminal-offenses-fraud/5843128-1.html>.
- [8] Eurostat. 2005. *Raziskava o IKT*. <http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/>.
- [9] Fox, Tricia in Steve Stead. 2001. *Customer Relationship Management: Delivering the Benefits*. <http://www.crmuk.co.uk/downloads/CRM01.pdf>.
- [10] Gray, Paul in Jongbok Byun. 2001. *Customer Relationship Management*. Centre for Research on Information Technology and Organisations. California: University of California.
- [11] Istraturist. 2011. *Privacy Policy on the Website*. [http://www.istraturist.com/en/info/privacy\\_policy](http://www.istraturist.com/en/info/privacy_policy).
- [12] Kalakota, Ravi in Andrew Whinston. 1997. *Electronic commerce. A Managers guide*. London: Adison Wesley Longman Inc.
- [13] Lawrence, A. Gordon in drugi. 2004. *Computer Crime and Security Survey*. San Francisco: Computer Security Institute Publications.
- [14] McAllister, Thomas. 2004. *Customer Relationship Management – A Case for e-business Strategy*. Maryland: University of Maryland.
- [15] Min, Wei. 2010. Construction of Marketing Management Information System of Travel Agency Based on Customer Relationship Management. *Journal of Convergence Information Technology* 5 (8).
- [16] Perše, Zoran. 2000. Varstvo in zaščita osebnih podatkov pri elektronskem poslovanju. *Gospodarski vestnik* 11.
- [17] Raba interneta v Sloveniji (RIS). 2008. *Uporabniki računalnikov ne skrbijo dovolj za posodabljanje programske opreme*. <http://www.ris.org/index.php?fl=2&lact=1&bid=8442&menu=0>.
- [18] Rebec, Matej. 2009. *Management ravnanja s kupci na področju turizma v Sloveniji z vidika elektronskega varovanja podatkov*. Magistrsko delo. Koper: Fakulteta za management.
- [19] Rigby, Darrell, Fred Reichheld in Steve Berez. 2002. Custom fit. *Optimize*, December 2002, 26–36.
- [20] Rosen, Charles. 2000. *Customer Intelligence Gets Smarter*. New York: InformationWeek. <http://www.informationweek.com/bizint/biz804/customer.htm>.
- [21] Slovenska turistična organizacija (STO). 2010. *Slovenski turizem v številkah 2009*. [http://www.slovenia.info/pictures%5CTB\\_board%5Catachments\\_1%5C2010%5CstoVStevilkah\\_09\\_web\\_9710.pdf.pdf](http://www.slovenia.info/pictures%5CTB_board%5Catachments_1%5C2010%5CstoVStevilkah_09_web_9710.pdf.pdf).
- [22] UNWTO. 2011. *2010 International Tourism Results and Prospects for 2011*. [http://www.unwto.org/facts/eng/pdf/barometer/UNWTO\\_HQ\\_Fitur11\\_JK\\_1pp.pdf](http://www.unwto.org/facts/eng/pdf/barometer/UNWTO_HQ_Fitur11_JK_1pp.pdf).
- [23] Zorman, Marijana. 2001. Vas skrbi zloraba osebnih podatkov? *Gospodarski vestnik* 39: 64–65.

Matej Rebec je diplomiral na Ekonomski fakulteti v Mariboru, leta 2009 pa je na Fakulteti za management Koper magistriral s področja elektronskega varovanja podatkov. Zaposlen je v policiji kot kriminalistični inšpektor specialist in pokriva delovno področje računalniške kriminalitete. Aktivno se udeležuje znanstvenih konferenc, predvsem s področja informatike ter turizma.

Cene Bavec je redni profesor na Fakulteti za management Univerze na Primorskem in pokriva področje splošnega menedžmenta in menedžmenta informacijskih tehnologij. Redno predava tudi na Fakulteti za matematiko, naravoslovje in informacijske tehnologije Univerze na Primorskem.

# Primerjava učinkovitosti delovanja aplikacij v oblaku in v lokalnem okolju

Boris Ovcjak, Gregor Jošt, Gregor Polančič, Marjan Heričko

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko, Smetanova 17, 2000 Maribor  
{boris.ovcjak, gregor.jost, gregor.polancic, marjan.hericko}@uni-mb.si

## Izvleček

Računalništvo v oblaku je vse bolj priljubljena metoda pri razvoju, izvajanju in uporabi informacijskih rešitev. Hitra vzpostavitev izvajalnega okolja, skalabilnost in poslovna agilnost so le nekatere prednosti, ki jih obljublja t. i. oblachna infrastruktura. Zato postaja računalništvo v oblaku vse bolj konkurenčno tradicionalnim metodam pri gostovanju in uporabi informacijskih rešitev.

Namen prispevka je primerjati lokalno strežniško okolje in okolje v oblaku z vidika učinkovitosti ter ugotoviti, pri kakšni konfiguraciji dosegeta okolji podobno učinkovitost. Zato smo izvedli preizkuse obremenitve testne aplikacije v lokalnem okolju in v različnih konfiguracijah t. i. oblachnega okolja. Pri tem smo učinkovitost opredelili s sedmimi metrikami, ki smo jih pridobili iz standarda ISO/IEC 9126-2 in orodja za izvajanje obremenitev JMeter.

Rezultati testov obremenitev so pokazali, da je platforma kot storitev v primeru večjega števila primerkov učinkovitejša od lokalnega okolja. V primeru manjšega števila primerkov se je lokalno okolje izkazalo za učinkovitejše, zato smo v sklopu raziskave iskali tudi konfiguracijo, pri kateri se izenačita učinkovitosti delovanja obeh okolij. Pomembno vlogo pri zagotavljanju ustrezne učinkovitosti ima skalabilnost okolja, saj omogoča optimalno delovanje aplikacij ne glede na okoliščine; pri tem moramo upoštevati tudi stroške, ki naraščajo z zmogljivostjo platforme.

**Ključne besede:** računalništvo v oblaku, platforma kot storitev, učinkovitost, programska oprema kot storitev, Windows Azure.

## Abstract

### Comparison of Local and Cloud Based Applications Efficiency

Cloud computing represents an increasingly popular approach to the development, implementation and use of information technology (IT) solutions. Fast establishment of runtime environments, scalability and business agility are just some of the advantages that the cloud infrastructure promises to ensure. Therefore, cloud computing is becoming a competitive approach to the traditional hosting and usage of IT solutions.

The purpose of this paper was to compare the local server environment with the cloud environment in terms of effectiveness and to determine the configuration where both environments achieve similar efficiency. Therefore, we performed a load testing of the test application in the local environment and in different configurations of the cloud environment. We measured efficiency with seven performance metrics from the ISO/IEC 9126-2 standard and the load testing tool, JMeter.

The results of load testing indicate that in case of more instances, the platform as a service (PaaS) performed better than the local environment. However, the local environment has proven to be more efficient compared to low number of PaaS instances. While we compared the efficiency of both environments, we found that the local environment ranged between four and five instances of the cloud counterpart. We found that scalability has an important role in ensuring appropriate efficiency as it allows optimal application performance, regardless of the circumstances; however we need to take into consideration the costs of using the platform which grows with the capacity of the platform.

**Key words:** cloud computing, platform as a service, efficiency, software as a service, Windows Azure.

## 1 UVOD

Računalništvo v oblaku se omenja od leta 2007<sup>1</sup> in je še vedno najbolj aktualno področje informatike. Ključna prednost novega načina uporabe informacijske tehnologije je zmanjšanje investicij v strojno in programsko opremo, saj do storitev dostopamo kar prek omrežja (Gospodarska zbornica Slovenije, 2011).

Zaradi posrednega najema infrastrukture običajno nimamo stroškov z nadgradnjami in vzdrževanjem strojne opreme, kar zmanjša tudi obseg vzdrževalnega osebja. Sodoben tehnološki pristop omogoča razvoj novih in inovativnih storitev, kar posledično dviguje konkurenčnost.

<sup>1</sup> Razvidno iz grafa, pridobljenega s <http://www.google.com/trends?q=cloud+computing>.

O priljubljenosti računalništva v oblaku priča študija, ki jo je opravilo podjetje IBM maja 2011, ki je zajela več kot 3.000 globalnih direktorjev informatike (angl. CIO), med njimi jih je bilo deset tudi iz Slovenije. Rezultati so pokazali, da je 60 odstotkov organizacij pripravljenih sprejeti računalništvo v oblaku v naslednjih petih letih za potrebe poslovnega napredka in doseganja konkurenčnosti, kar je dvakrat več, kot je pokazala enaka študija iz leta 2009 (O'Malley & Tomasco, 2011).

Pomembnost paradigme je prepoznalo tudi raziskovalno podjetje Gartner Inc., saj je na seznam področij informacijske tehnologije, ki bodo imela v letu 2011 največjo strateško vlogo v poslovanju, na prvo mesto uvrstilo računalništvo v oblaku. V prihodnjih treh letih naj bi bili priča storitveno usmerjenim pristopom v oblaku. Po Gartnerjevih napovedih bodo ponudniki storitev ponujali zasebne oblake, ki bodo omogočali vključitev storitvenih tehnologij in metodologij v obliki implementacije znotraj podjetij (Petty, b. d.).

Tudi v Sloveniji vse bolj spodbujajo razvoj aplikacij in storitev računalništva v oblaku. Pošta Slovenije je nedavno predstavila informacijsko storitev Digitalna pisarna, ki je namenjena poslovnim uporabnikom in je zasnovana na konceptu programske opreme kot storitve (Pošta Slovenije, 2011). Tudi Skupina Telekom Slovenije se strateško vključuje v računalništvo v oblaku z zagotavljanjem infrastrukture kot storitve (najem in upravljanje strežnikov) in programske opreme kot storitve (CRM, elektronska pošta itd.) (Siol.net, 2011). Gospodarska zbornica Slovenije (Združenje za informatiko in telekomunikacije) spodbuja razvoj rešitev za računalništvo v oblaku; sodeluje pri kompetenčnem centru Zavod e-Oblak, ki je tudi član evropskega združenja Euro Cloud Europe. Cilj delovanja Zavoda je razvoj mednarodno priznane blagovne znamke in uspešno trženje storitev na mednarodnih trgih (Gospodarska zbornica Slovenije, 2011).

Kljub prednostim, ki jih prinaša računalništvo v oblaku, je vprašanje selitve v oblak še vedno aktualno. Glavni razlogi za selitev so hitra postavitve, manjši stroški in zmožnost skalabilnosti<sup>2</sup> (angl. scalability) aplikacij (Subashini & Kavitha, 2011), ki se prilagajajo potrebam. V prispevku želimo ugotoviti, kako prednosti računalništva v oblaku (predvsem skalabilnost) vplivajo na učinkovitost delovanja aplikacij v primerjavi z lokalnim okoljem.

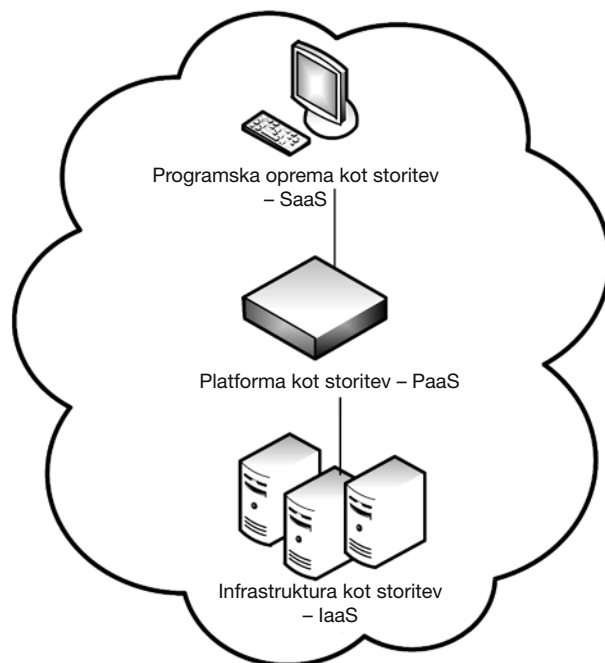
<sup>2</sup> Skalabilnost je sposobnost sistema, da prilagaja računalniške vire glede na trenutne potrebe aplikacije.

Prispevek je urejen po razdelkih: v drugem smo definirali računalništvo v oblaku in platformo kot storitev ter podali pogled nekaterih večjih ponudnikov na omenjenem področju. Izpostavili smo ponudnika, ki smo ga vključili v raziskavo. V tretjem razdelku smo predstavili postopek primerjalne analize, vključno z aplikacijo, ki smo jo testirali v različnih konfiguracijah in okoljih. Dobljene rezultate smo interpretirali in podali sklepe. V zadnjem razdelku smo podali omejitve in uporabno vrednost raziskave ter načrte za naše nadaljnje delo.

## 2 RAČUNALNIŠTVO V OBLAKU

Čeprav uradne definicije računalništva v oblaku še ni (Mell & Grance, 2011), je inštitut The National Institute of Standards and Technology (v nadaljevanju NIST), ki skrbi za razvoj standardov, definiral računalništvo v oblaku kot »model, ki omogoča vseprisotno, priročno in na zahtevo pridobljen mrežni dostop do bazena računalniških virov (npr. omrežja, strežnikov, šrambe, aplikacij in storitev)« (Mell & Grance, 2011). Med prednosti računalništva v oblaku štejemo zmanjšanje stroškov, skalabilnost in neprekinjeno delovanje (Furht & Escalante, 2010).

Čeprav računalništvo v oblaku obravnava vse kot storitev (angl. Everything-as-a-Service), štejemo med glavne sloje arhitekture (Furht & Escalante, 2010): 1) infrastrukturo kot storitev (angl. Infrastructure-



Slika 1: Sloji arhitekture računalništva v oblaku



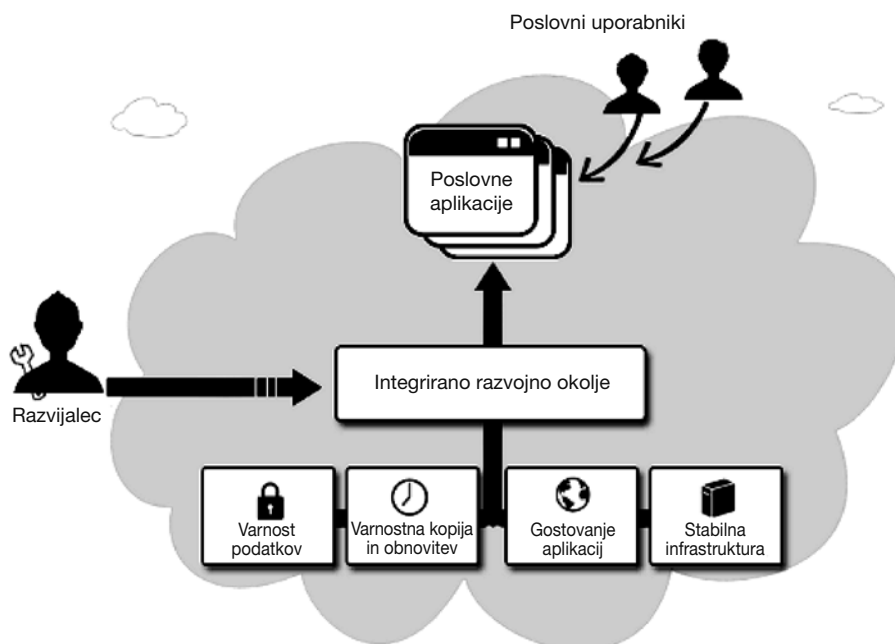
-as-a-Service, v nadaljevanju IaaS), 2) platformo kot storitev (angl. Platform-as-a-Service, v nadaljevanju PaaS) in 3) programsko opremo kot storitev (angl. Software-as-a-Service, v nadaljevanju SaaS) (slika 1).

IaaS je najnižji sloj, ki ponuja računalniške vire v obliki storitev. Sem štejemo virtualizacijo računalnikov z zajamčeno zmogljivostjo procesorja in pasovno širino za shrambo (angl. Storage bandwidth) ter dostop do omrežja. PaaS je nadgradnja IaaS, saj vključuje tudi operacijski sistem in storitve za specifične aplikacije. Na najvišjem sloju arhitekture računalništva v oblaku se nahajajo storitve informacijskih tehnologij, ki jim pravimo tudi SaaS. Te omogočajo, da uporabniki poganjajo aplikacije neposredno iz sloja oblaka PaaS (Furht & Escalante, 2010).

Ker bomo v sklopu prispevka primerjali učinkovitost delovanje aplikacije v oblaku in v lokalnem okolju, se bomo osredotočili le na PaaS.

## 2.1 Platforma kot storitev

NIST definira platformo kot storitev (v nadaljevanju PaaS) kot »možnost, da v infrastrukturo računalništva v oblaku postavimo uporabniško ustvarjene ali pridobljene aplikacije z uporabo programskih jezikov in orodij, ki jih podpira ponudnik. Uporabnik načeloma ne nadzoruje ali upravlja infrastrukture (mreža, strežniki, operacijski sistemi in shrambe), ima pa nadzor nad postavljenimi aplikacijami in potencialno tudi nad okolji za njihovo gostovanje« (Mell & Grance, 2011). Slika 2 prikazuje koncept PaaS s pripadajočimi vlogami.



Slika 2: Vloge v PaaS (ZOH Creator, b. d.)

PaaS omogoča razvijalcem popoln nadzor nad razvojem in postavitvijo aplikacij. Pri tem je proces izdaje (angl. release) programskih rešitev enostavnejši in hitrejši, saj je mnogo opravil, povezanih z izdajanjem programskih rešitev (npr. nastavitve gostovanja, strežnikov, podatkovnih baz in procesov uporabniške interakcije), že vnaprej pripravljenih (ZOH Creator, b. d.). Glede na ponujene funkcionalnosti lahko definiramo dva tipa PaaS (Pavlinek, Košič, Jošt, Gradišnik & Ovčjak, 2011):

- 1) platforme za računalništvo v oblaku, ki prinašajo razvojna okolja in jih mora razvijalec še vedno lokalno namestiti, preden jih uporablja;
- 2) platforme, ki razvojna orodja selijo v oblak, dostopna pa so kot novodobna razvojna okolja.

## 2.1 Ponudniki platforme kot storitve

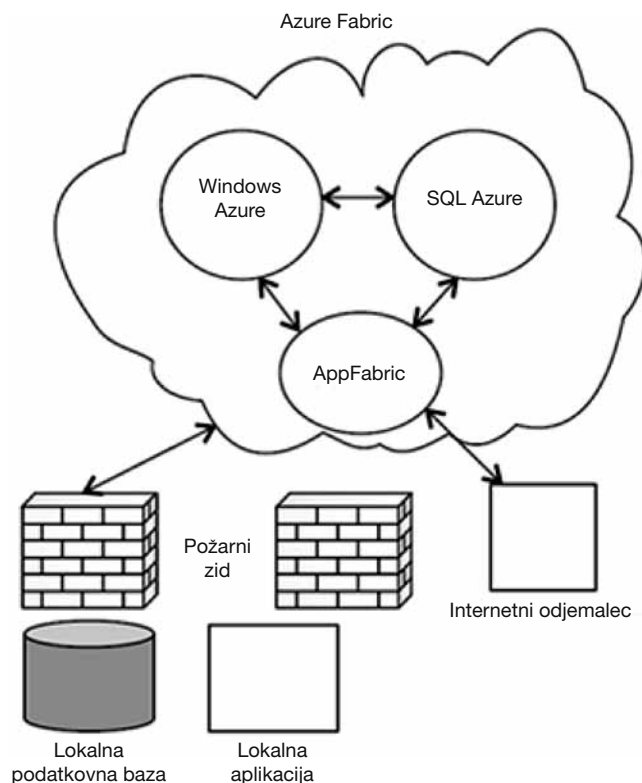
Trenutno se na trgu pojavlja mnogo ponudnikov platform kot storitev, med katerimi med bolj prepoznavne štejemo (SearchCloudComputing.com, b. d.):

- 1) *Amazon Web Services (AWS)*, ki je zbirka spletnih storitev, ki skupaj sestavljajo platformo kot storitev in vključujejo Amazon EC2, Amazon S3, Amazon Simple DB, Amazon SQS, Amazon FPS idr. AWS prinaša fleksibilno, skalabilno in cenovno ugodno računalniško platformo za podjetja vseh velikosti. Z AWS pridobimo dostop do zanesljive in varne tehnološke platforme, ki poganja tudi Amazonovo globalno spletno lastnino. Do AWS dostopamo prek protokolov HTTP ali SOAP, vse storitve pa zaračunavajo po porabi. Z uporabo platforme pridobimo dostop do infrastrukture podjetja Amazon in najamemo vire, ki jih potrebujemo za skaliranje svoje aplikacije. AWS ponuja primerke na zahtevo, pri čemer plačamo za računalniške vire po uri (AWS, b. d.);
- 2) *Google App Engine (GAE)* je platforma za razvoj in gostovanje spletnih aplikacij v Googlovih podatkovnih in aplikacijskih centrih. Omogoča izgradnjo in upravljanje aplikacij, zagotavlja podatkovne shrambe in skalabilnost v primeru rasti prometa. Platforma odpravi skrbi glede vzdrževanja infrastrukture in omogoča razvijalcem, da skrbijo izključno za svoje aplikacije. GAE trenutno podpira programske jezike, kot so Python, Java in Go ter posledično tudi druge jezike, ki temeljijo na JVM (angl. Java Virtual Machine). Mednje štejemo Groovy, JRuby, Scala, Clojur in Jython. GAE temelji na modelu plačila storitve glede na uporabo (angl. pay-as-you-go pricing), pri čemer uporabniki plačujejo le vire, ki jih porabljajo (Google, b. d.);
- 3) *Salesforce.com – Force.com* je produkt PaaS podjetja Salesforce. Začeli so kot ponudnik programske opreme kot storitve za upravljanje odnosov s strankami (angl. Customer relationship management, CRM), kasneje pa so razvili tudi platformo, imenovano Force.com. Lastnosti in filozofija Force.com se razlikujejo od ostalega tržišča ponudnikov PaaS, saj je Force.com ustvarjen specifično za poslovne aplikacije in temu primerno je prirejena tudi infrastruktura. Platforma vključuje tudi svoj programski jezik, imenovan Apex. Razvijalcem omogoča skriptiranje interakcij z drugimi funkcijami platforme, vključno z uporabniškim vmesnikom. Življenjski cikel razvoja aplikacij za Force.com je podoben razvoju spletnih aplikacij v lokalnem okolju. Za razliko od GAE Salesforce.com ponuja vnaprej zakupljene količine virov (Ouellette, b. d.);

- 4) *Windows Azure* (v nadaljevanju Azure) je Microsoftova rešitev na področju računalništva v oblaku. Ponuja dve možnosti obračunavanja: naročniško (angl. Subscription offers) in plačilo storitve glede na uporabo (Microsoft Corporation, 2011a). Ker smo se v eksperimentalnem delu članka omejili na Azure, bomo v nadaljevanju omenjeno platformo predstavili podrobneje.

## 2.2 Platforma Windows Azure

Platforma Azure se nahaja v zmogljivih Microsoftovih podatkovnih centrih. Platformo sestavljajo tri komponente (Dudley, 2010): operacijski sistem Windows Azure, SQL Azure in Windows Azure AppFabric. Slika 3 prikazuje medsebojne relacije komponent.



Slika 3: **Komponente platforme Windows Azure (Dudley, 2010)**

### 2.2.1 Operacijski sistem Windows Azure

Windows Azure je operacijski sistem platforme, ki omogoča poganjanje skalabilnih aplikacij na Microsoftovih strežnikih in podatkovnih centrih. Operacijski sistem v danem kontekstu pomeni, da Windows Azure opravlja dela, ki so značilna za konvencionalni operacijski sistem (Hay & Prince, 2010), med katera lahko štejemo gostovanje in izvajanje aplikacij, abstrakcijo strojne opreme in zagotavljanje vmesnika

med uporabniki ter aplikacijami. Glavna razlika med Windows Azure in konvencionalnim operacijskim sistemom je pri dodeljevanju procesorske moči in pomnilnika. Aplikacije, gostujoče na Windows Azure, se ne izvajajo vedno na enem strežniku, zaradi česar za dodeljevanje virov ne more biti zadolžen operacijski sistem platforme. To nalogo opravlja virtualni stroj, ki skrbi za porazdelitev storitev na različne strežnike.

Windows Azure v bistvu omogoča gostovanje spletnih aplikacij ali aplikacij, ki obdelujejo podatke v ozadju. Prav tako pa zagotavlja storitve, namenjene obdelavi (angl. Windows Azure Compute Services) in shrambi (angl. Windows Azure Storage Services) (Microsoft Corporation, 2010).

V primeru obdelave podatkov imamo opravka z dejansko programsko kodo [4]. Aplikacije so strukturirane v obliki vlog, ki jih poganjamo v izvajalnem okolju. V vsakem trenutku imamo lahko eno ali več primerkov določene vloge. Windows Azure podpira (Hay & Prince, 2010; Microsoft Corporation, 2010):

- 1) *spletno vlogo* (angl. Web Role), ki je namenjena razvoju spletnih aplikacij, skladno z IIS (Internet Information Services) 7 in ASP.NET;
- 2) *delovno vlogo* (angl. Worker Role), ki je uporabna za splošni razvoj in lahko skrbi za procesiranje (v ozadju) za potrebe spletne vloge. Delovna vloga je primerljiva s storitvami na lokalnih operacijskih sistemih;
- 3) *vlogo virtualne naprave* (angl. Virtual Machine Role), ki zagotavlja sliko (angl. image) sistema Windows Server 2008 R2, ki olajša selitev obstoječe aplikacije na Windows okolje Azure.

Storitve, namenjene shrambi, zagotavljajo trajno shrambo v oblaku. Med temeljne storitve štejemo (Hay & Prince, 2010; Microsoft Corporation, 2010):

- 1) zbirke binarnih podatkov (angl. Blob services), namenjene hranjenju besedilnih ali binarnih podatkov;
- 2) vrste (angl. Queue service), ki skrbijo za zanesljivo pošiljanje in izmenjavo sporočil med storitvami;
- 3) tabele (angl. Table services), ki so namenjene strukturirani shrambi podatkov. Teh storitev ne smemo enačiti z relacijsko podatkovno bazo, saj standardni jezik za povpraševanje (SQL) ni podprt [4].

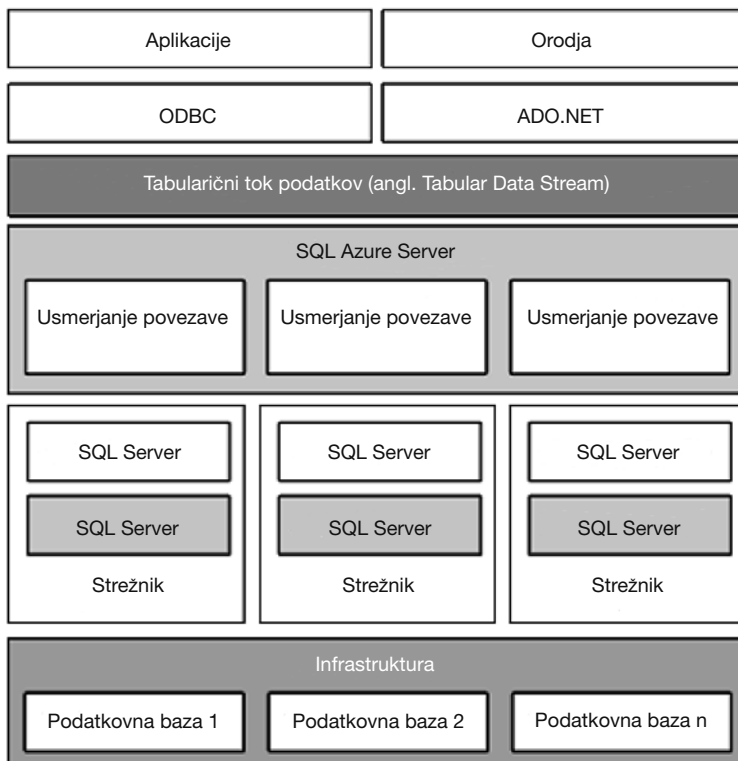
## 2.2.2 Podatkovne storitve SQL Azure

SQL Azure je skoraj popolna implementacija strežnika SQL Server 2008 [4] in ponuja storitve relacijske podatkovne baze. Nahaja se na platformi Azure in podpira relacijski model podatkovne baze, ki temelji na jeziku T-SQL (Microsoft Corporation, 2011b).

Podatkovno bazo SQL Azure lahko upravljamo prek 1) portala SQL Azure Portal, 2) ukazne vrstice `sqlcmd` ali 3) v okolju SQL Server Management Studio 2008 R2. Za uporabo SQL Azure ni treba lokalno ali v oblaku nameščati, nastavljanje ali vzdrževanje dodatne programske opreme (Microsoft Corporation, 2011b).

Skladno s koncepti platforme kot storitve omogoča SQL Azure tudi večnajemništvo (angl. multi-tenancy), plačilo storitve glede na uporabo in minimizacijo administracije infrastrukture (Microsoft Corporation, 2011b).

Izvedba SQL Azure se razlikuje od klasičnega sistema SQL Server. Pri povezavi na SQL Azure se ne povezujemo na fizični SQL Server, ampak na simulacijo strežnika. Gre torej za oblačne storitve, ki zagotavljajo večino značilnosti podatkovne baze SQL Server. SQL Azure deluje tako, da ovija (angl. wrapper) fizično infrastrukturo (slika 4) (Hay & Prince, 2010).



Slika 4: Fizični pogled na strukturo SQL Azure (Hay & Prince, 2010)

Do SQL Azure lahko dostopamo z ADO.NET, ODBC in programskim jezikom PHP. Kljub temu da SQL Azure temelji na SQL Server, se soočamo z nekaterimi omejitvami. Zaradi nivoja abstrakcije fizične implementacije npr. ne moremo uporabiti ukaza USE, prav tako niso podprti nekateri ukazi T-SQL, ki se sklicujejo na fizični nivo (npr. BACKUP).

Z omejitvami se soočamo tudi pri povezavi na SQL Azure. Ta namreč omogoča dostop le prek vrat (angl. port) 1433 (Hay & Prince, 2010). SQL Azure v splošnem podpira transakcije, vendar mu manjka podpora porazdeljenim transakcijam, saj vrata za njihovo izvedbo niso podprta.

Omejen je tudi časovni interval izvedbe ukazov in poizvedb. Ti ne smejo trajati dlje kot 30 minut, kolikor je privzeta časovna omejitev. Vse transakcije, ki trajajo dlje, se prekličejo (Hay & Prince, 2010).

### 2.2.3 Komponenta AppFabric

AppFabric je najstarejša komponenta Azure. Prvotno se je AppFabric imenoval BizTalk Services in pomeni dopolnilno oblačno ponudbo strežniku BizTalk.<sup>3</sup> Kasneje so BizTalk Services preimenovali v .NET Services in nato v Azure AppFabric z namenom, da bi komponento bolje povezali z Azure platformo (Hay & Prince, 2010).

AppFabric je knjižnica storitev in pomaga pri pogonjanju oblačnih storitev ter povezovanju z zunanjimi programskimi okolji. Je varen most med obstoječimi (angl. legacy) aplikacijami in oblačnimi storitvami. AppFabric je sestavljen iz dveh ključnih delov (Dudley, 2010): 1) storitvenega vodila (angl. Service Bus) in 2) nadzora dostopa (angl. Access Control, v nadaljevanju ACS).

*Storitveno vodilo* zagotavlja vodilo v oblaku, ki služi za povezovanje storitev z odjemalci. Skladno s karakteristikami storitvenih vodil omogoča poimenovanje, odkrivanje, izpostavljanje in orkestracijo storitev neodvisno od njihovih fizičnih lokacij (Dudley, 2010). Storitveno vodilo pomeni način, kako povezati storitve in usmerjati sporočila. Prednost takšnega pristopa je, da lahko storitve povezujemo brez poznavanja povezovalne tehnologije (Hay & Prince, 2010). Storitveno vodilo abstrahira fizično lokacijo storitve in ponuja URI, prek katerega lahko priključimo storitev.

V storitvenem vodilu potrebujemo še mehanizem, kako omejiti dostop do aplikacij, kar zagotavlja komponenta AppFabric ACS (angl. Access Control Service). ACS skrbi za varnost naših "oblačnih storitev" in aplikacij, saj abstrahira overjanje (Hay & Prince, 2010). S pomočjo ACS definiramo pravila (z uporabo storitev, ki temeljijo na Windows Communication Foundation), preko katerih lahko do oblačnih storitev dostopajo tudi drugi. ACS je podoben storitvam, kot so Open ID ali Live ID (Dudley, 2010).

## 3 PRIMERJALNA ANALIZA DELOVANJA APLIKACIJE V OBLAKU IN V LOKALNEM OKOLJU

V okviru primerjalne analize smo izvedli test obremenitve (angl. Stress testing) testne aplikacije v lokalnem okolju in v okolju Azure. Test obremenitve je namenjen ugotavljanju robustnosti, dosegljivosti in zanesljivosti aplikacije v skrajnih okoliščinah. Med te okoliščine štejemo velike obremenitve, sočasno delovanje in omejene vire. Namen testa obremenitve je identificiranje težav, povezanih z viri, s prioritetai itd. (Microsoft Corporation, 2007).

Pri testu obremenitve običajno izvajamo simulacije več scenarijev v različnih okoliščinah. V našem primeru smo se osredinili na test obremenitve spletnih aplikacij z namenom, da opazujemo učinkovitost delovanja aplikacij v skrajnih okoliščinah. Slika 5 prikazuje proces poteka primerjalne analize v notaciji BPMN.<sup>4</sup>

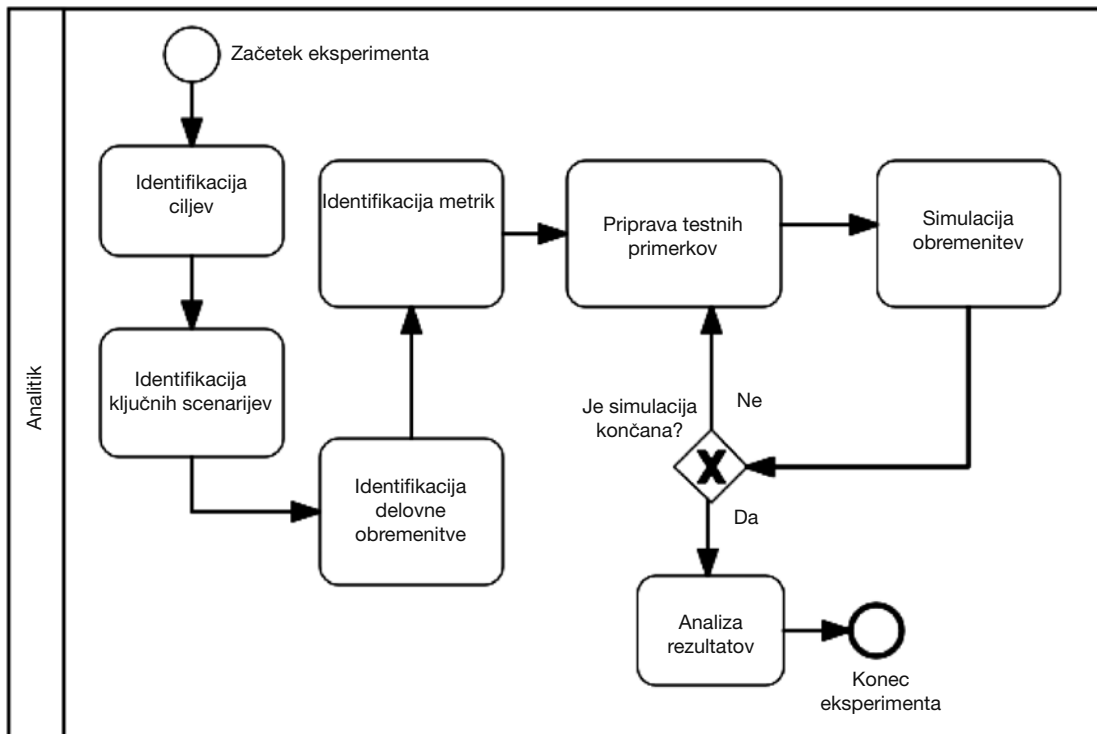
### 3.1 Opredelitev testne aplikacije

Za testiranje zmogljivosti različnih platform in konfiguracij smo pripravili testno aplikacijo, ki ima sicer enostavne, a procesorsko potratne funkcionalnosti. Glavni delež procesiranja smo tako delegirali na razčlenjevalnik XML, saj je obdelovanje dokumentov XML v splošnem procesorsko zahtevno opravilo (Nicola & John, 2003). Za potrebe testne aplikacije smo izbrali podatkovno zbirko iz portala SI-STAT, nad katero smo z uporabo omenjenega razčlenjevalnika izvajali operacije in s tem simulirali različne obremenitve.

Testna aplikacija ASP.NET je izdelana s programskim jezikom C#. V primeru lokalne infrastrukture smo lahko datoteko XML shranili na fizični lokaciji

<sup>3</sup> BizTalk je strežnik za integracijo in izvajanje poslovnih procesov.

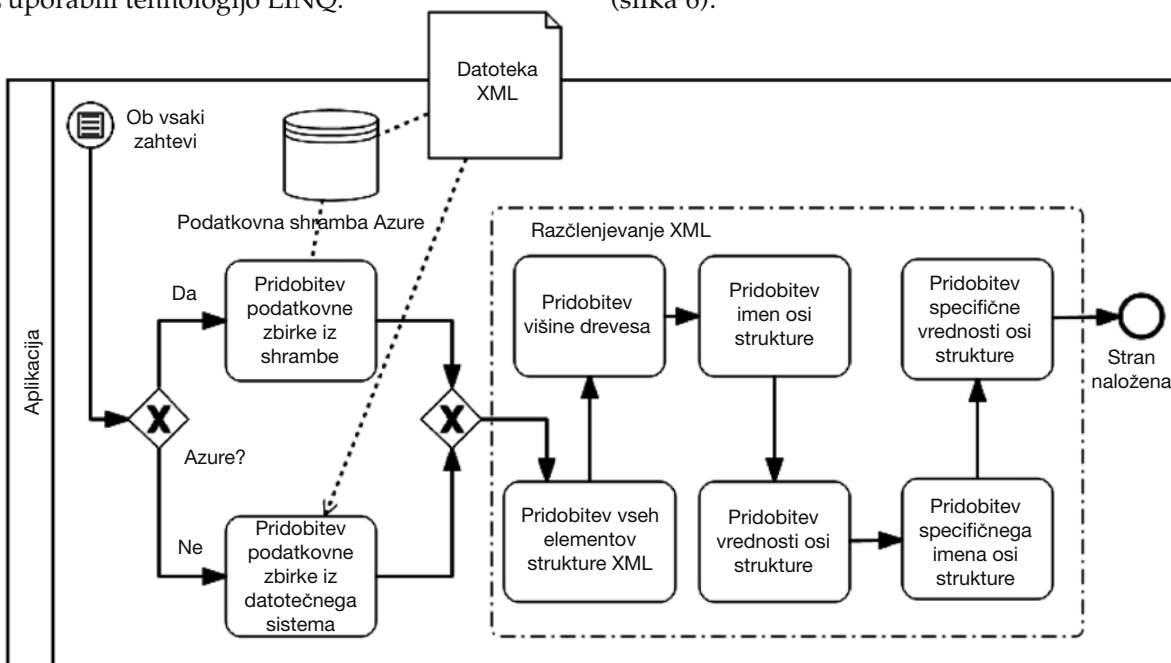
<sup>4</sup> BPMN (BusinessProcessModellingNotation) je standard za grafično modeliranje poslovnih procesov. Vir: <http://www.bpmn.org/>



Slika 5: **Model primerjalne analize BPMN**

(podatkovnem disku), za razliko od lokalne infrastrukture Azure, ki ne omogoča fizičnega shranjevanja. Datoteko XML smo zato shranili v nestrukturirano podatkovno shrambo (Azure Storage). V obeh primerih smo za prebiranje podatkov iz datoteke XML uporabili tehnologijo LINQ.

Postopek izvajanja primerjave je potekal tako, da je vsak od simuliranih uporabnikov izvedel poizvedbe HTTP na domačo stran spletne aplikacije in s tem sprožil pridobitev podatkovne zbirke ter šest različnih poizvedovalnih operacij po zbirki (slika 6).



Slika 6: **Model izvajanja testne aplikacije BPMN**

### 3.2 Predstavitev testnih okolij

Za potrebe izvedbe testov obremenitev potrebujemo dve izvajalni okolji za omenjeno aplikacijo. Ker je namen raziskave primerjati učinkovitost delovanja aplikacij v lokalni in oblaki infrastrukturi, smo želeli primerjati rešitve istega ponudnika, saj tako dobimo najbolj primerljive rezultate. Kot predstavnik tehnologije računalništva v oblaku smo izbrali platformo Azure, medtem ko je lokalno okolje predstavljala infrastruktura z operacijskim sistemom Windows Server 2008 R2 64-bit.

Definicija platforme kot storitve pove, da razvijalci oz. uporabniki nimajo neposrednega vpogleda v infrastrukturo, v kateri tečejo aplikacije oz. storitve. Te se fizično nahajajo na virtualnih napravah, pri čemer vsaka virtualna naprava gosti en primerek aplikacije. Azure ima na voljo pet različnih velikosti virtualnih naprav: ExtraSmall, Small, Medium, Large in Extra Large. Vsaka velikost ima vnaprej določene vire skupaj z jasno definiranimi specifikacijami (število jeder, količina spomina, diskovna kapaciteta in pasovna širina). Da bi zmanjšali vpliv nastavitve na rezultate primerjalne analize, smo tako v lokalnem okolju kot v primeru Azure uporabili privzeto nastavitve (velikost primerka Small). Ta zagotavlja za vsak primerek aplikacije oz. storitve procesorsko jedro s frekvenco 1,6 GHz, 1,75 GB spomina, 225 GB diskovne kapacitete in 100 Mbit pasovne širine (Microsoft Corporation, b. d.).

Za razliko od PaaS, pri kateri smo izbrali evropsko regijo, imamo v primeru lokalnega strežnika popoln nadzor nad infrastrukturo, operacijskim sistemom in njegovo optimizacijo. Za potrebe simulacije smo izbrali osebni računalnik z operacijskim sistemom Windows Server 2008 R2 64-bit. Tabela 1 prikazuje primerjavo konfiguracij obeh testnih okolij.

Tabela 1: **Primerjava konfiguracij infrastrukture okolja v oblaku in lokalnega okolja**

Atributi	Okolje v oblaku	Lokalno okolje
Število procesorskih jeder	2, 10 in 20	4
Zmogljivost procesorskih jeder	1,6 GHz	2,66 GHz
Pomnilnik (RAM)	1,75 GB	4 GB
Diskovna kapaciteta (GB)	225 GB	500 GB

### 3.3 Predstavitev metrik učinkovitosti

Pri definiranju metrik učinkovitosti smo se zgledovali po standardu ISO/IEC 9126-2. Standard se osredinja

na ocenjevanje kakovosti produkta, in sicer na zunanje metrike, ki naslavljajo učinkovitost programske opreme. Za potrebe analize smo izbrali metrike (International Organization for Standardization, b. d.):

- **metrike časovnega obnašanja** (angl. Time behaviour metrics):
  - **odzivni čas (ms)** (angl. Response time) meri povprečni čakalni čas, ki preteče od podane zahteve do pridobitve rezultata pod specifičnimi obremenitvami sistema v smislu sočasnih nalog in sistemske uporabe (angl. utilisation);
  - **prepustnost (št./s)** (angl. Throughput) meri povprečno število sočasnih nalog, ki jih lahko sistem obdela v določenem časovnem intervalu;
- **metrike uporabe virov** (angl. Resource utilisation metrics):
  - **delež napak pri prenosu v določenem času (%)** (angl. Mean of transmission error per time) meri delež napak med prenosom v določenem časovnem intervalu in pri specifični sistemske uporabi.

Poleg navedenih metrik smo uporabili še metrike izbranega simulacijskega orodja (Apache Jakarta Project, b. d.):

- **mediana (ms)** – je srednji čas množice rezultatov. Polovica primerov je pod to vrednostjo, druga polovica pa nad njo;
- **črta 90 % (ms)** – pove, da je 90 odstotkov primerkov vzelo vsaj toliko časa, kot je definirana vrednost, preostalih 10 odstotkov pa je trajalo dlje;
- **minimum (ms)** – je najkrajši čas zahteve množice primerkov;
- **maksimum (ms)** – je najdaljši čas zahteve množice primerkov.

### 3.4 Izvedba testov obremenitev

V fazi izvedbe testa obremenitve smo uporabili različno število primerkov aplikacije in primerjali učinkovitost delovanja pod različnimi obremenitvami. Naš uporabniški račun Azure je omogočal izkoriščanje dvajsetih procesorskih jeder. Na podlagi analize platforme smo za testne namene definirali: 1) dva primerka – Azure priporoča vsaj toliko za posamezno aplikacijo, saj s tem ob morebitnih težavah zagotovimo nemoteno delovanje aplikacij, 2) deset primerkov in 3) dvajset primerkov.

Za simulacijo obremenitev aplikacije smo uporabili programsko orodje *Apache JMeter* (v nadaljevanju JMeter). JMeter je odprtokodno orodje za merjenje obremenitev in učinkovitosti delovanja. Uporablja se za testi-

ranje delovanja statičnih in dinamičnih virov (datotek, servletov, Java objektov, podatkovnih baz itd.). Prav tako lahko izvajamo simuliranje obremenitev strežnika, omrežja ali objektov z namenom opazovanja stabilnosti in analize učinkovitosti delovanja pod različnimi obremenitvami (Apache Jakarta Project, b. d.).

V okviru analize smo pripravili več različnih scenarijev. Za vsako od okolij, v katerem je delovala aplikacija, smo simulirali obremenitev pri različnem številu odjemalcev (tabela 2).

Tabela 2: Konfiguracije obremenitve aplikacije

Atributi	Vrednosti
Število uporabnikov	200, 500, 800, 1000
Število zahtev	20.000
Zamik	5 sekund

Kot je razvidno iz tabele, smo simulirali 200, 500, 800 in 1000 odjemalcev. Za potrebe testiranja apli-

kacije smo najprej ugotovili maksimalno vrednost hkratno prisotnih uporabnikov, ki znaša 1000 (točka, na kateri so se začele pojavljati napake). Vrednost 500 smo določili kot polovično vrednost maksimalne (za lažjo primerjavo), medtem ko smo se z vrednostma 200 in 800 želeli bolj približati skrajnima vrednostma (0 in 1000) in obenem tudi zadostiti, da je med številoma faktor potence števila 2.

Za vsakega smo določili petsekundno zakasnitev med pošiljanjem zahtev, kar se ujema z običajnim časom zadrževanja uporabnika na spletni strani.<sup>5</sup>

Ker smo morali predhodno definirati število zahtev, smo najprej na lokalnem okolju izvedli simulacijo, ki je trajala 10 minut, kar smo postavili kot izhodiščno vrednost pri nadaljnjih testih.

### 3.5 Rezultati raziskave

Tabela 3 prikazuje rezultate testov obremenitev in je razdeljena na štiri sklope: lokalno okolje in okolje Azure za dve, deset in dvajset primerkov.

Tabela 3: Rezultati meritev

Atributi	Vrednosti			
<b>Izvajalno okolje</b>	<b>Lokalno okolje – IIS 7</b>			
<b>Število primerkov/procesorskih jeder</b>	<b>Štiri procesorska jedra</b>			
<b>Število uporabnikov</b>	<b>200</b>	<b>500</b>	<b>800</b>	<b>1000</b>
Povprečni odzivni čas (ms)	5.217,67	14.415,00	20.374,33	28.301,33
Napaka pri prenosu (%)	0,00	0,00	0,00	0,00
Prepustnost (št. zahtev/s)	38,65	34,43	39,00	35,80
Mediana (ms)	5.211,00	14.291,00	20.657,67	28.765,00
Črta 90 % (ms)	5.570,67	16.018,33	21.120,00	30.449,33
Minimalni odzivni čas (ms)	131,33	58,67	240,50	87,67
Maksimalni odzivni čas (ms)	8.829,67	22.540,00	25.509,67	36.326,33
<b>Izvajalno okolje</b>	<b>Azure</b>			
<b>Število primerkov/procesorskih jeder</b>	<b>Dva primerka</b>			
<b>Število uporabnikov</b>	<b>200</b>	<b>500</b>	<b>800</b>	<b>1000</b>
Povprečni odzivni čas (ms)	12.320,50	30.374,50	47.156,50	115.438,00
Napaka pri prenosu (%)	0,00	0,00	0,00	84,00
Prepustnost (št. zahtev/s)	14,75	16,10	16,90	8,50
Mediana (ms)	13.032,00	30.907,50	47.634,00	123.595,00
Črta 90 % (ms)	16.255,00	32.643,00	51.446,50	127.991,00
Minimalni odzivni čas (ms)	176,50	545,50	482,50	213,00
Maksimalni odzivni čas (ms)	75.466,50	25.892,50	117.318,50	169.093,00

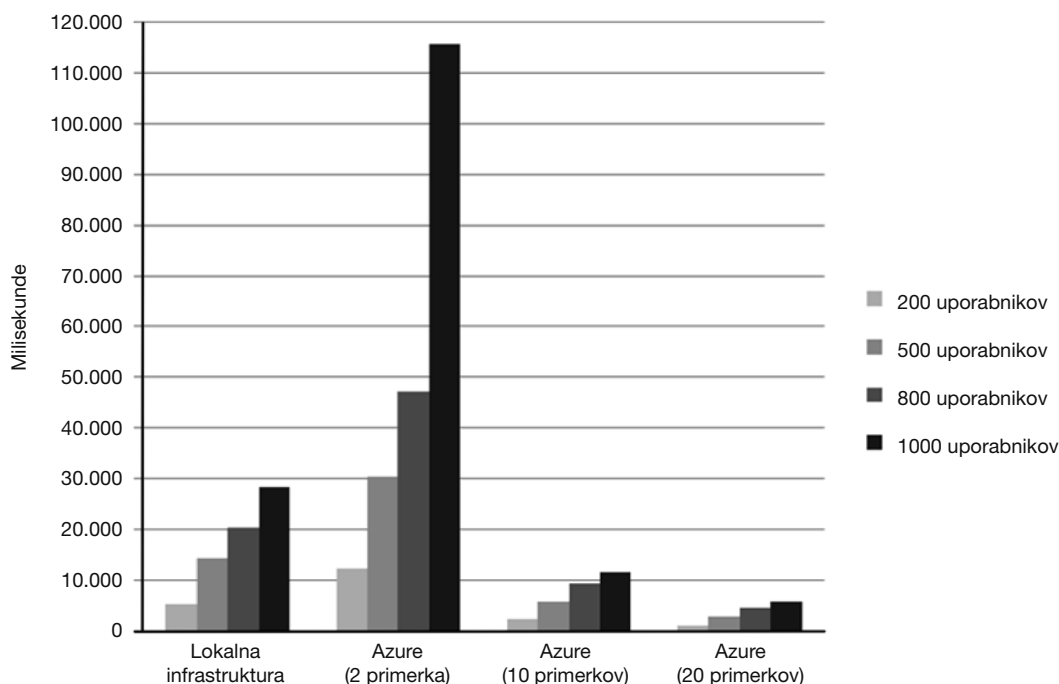
<sup>5</sup> <http://ezinearticles.com/?Reasons-Why-Your-Site-Has-Only-5-Seconds-to-Get-Someones-Attention&id=5674345>.

Atributi	Vrednosti			
<b>Izvajalno okolje</b>	<b>Azure</b>			
<b>Število primerkov/procesorskih jader</b>	<b>Deset primerkov</b>			
<b>Število uporabnikov</b>	<b>200</b>	<b>500</b>	<b>800</b>	<b>1000</b>
Povprečni odzivni čas (ms)	2.309,00	5.843,00	9.315,00	11.688,67
Napaka pri prenosu (%)	0,00	0,00	0,00	0,00
Prepustnost (št. zahtev/s)	81,80	81,10	75,80	76,97
Mediana (ms)	2.265,00	5.716,00	8.968,33	11.325,33
Črta 90 % (ms)	2.871,00	7.147,67	12.404,33	14.782,00
Minimalni odzivni čas (ms)	154,67	147,33	163,33	154,00
Maksimalni odzivni čas (ms)	15.819,67	57.620,67	90.997,00	107.191,67
<b>Izvajalno okolje</b>	<b>Azure</b>			
<b>Število primerkov/procesorskih jader</b>	<b>Dvajset primerkov</b>			
<b>Število uporabnikov</b>	<b>200</b>	<b>500</b>	<b>800</b>	<b>1000</b>
Povprečni odzivni čas (ms)	1.175,00	2.881,33	4.587,67	5.838,67
Napaka pri prenosu (%)	0,00	0,00	0,00	0,00
Prepustnost (št. zahtev/s)	108,33	160,40	156,00	156,30
Mediana (ms)	1.159,67	2.804,00	4.418,00	5.567,67
Črta 90 % (ms)	1.501,00	3.811,00	5.993,67	7.776,00
Minimalni odzivni čas (ms)	145,67	144,67	147,67	150,33
Maksimalni odzivni čas (ms)	5.713,33	24.632,33	44.304,50	59.166,00

Kot je razvidno iz tabele, so med izvajanjem testa obremenitve nastopile napake le pri Azure v primeru dveh primerkov v skrajnih okoliščinah (1000 zahtevkov v razmiku petih sekund). Temu botruje privzeto nastavljena časovna omejitev na zahtevo platforme. Preostali podatki, ki smo jih pridobili s

testi obremenitev, so podrobneje razčlenjeni v nadaljevanju.

Pri simulaciji obremenitev aplikacije smo najprej spremljali povprečni odzivni čas glede na število uporabnikov. Slika 7 prikazuje, kako se je povečeval odzivni čas z naraščanjem števila uporabnikov.



Slika 7: Povprečni odzivni čas v odvisnosti od števila uporabnikov



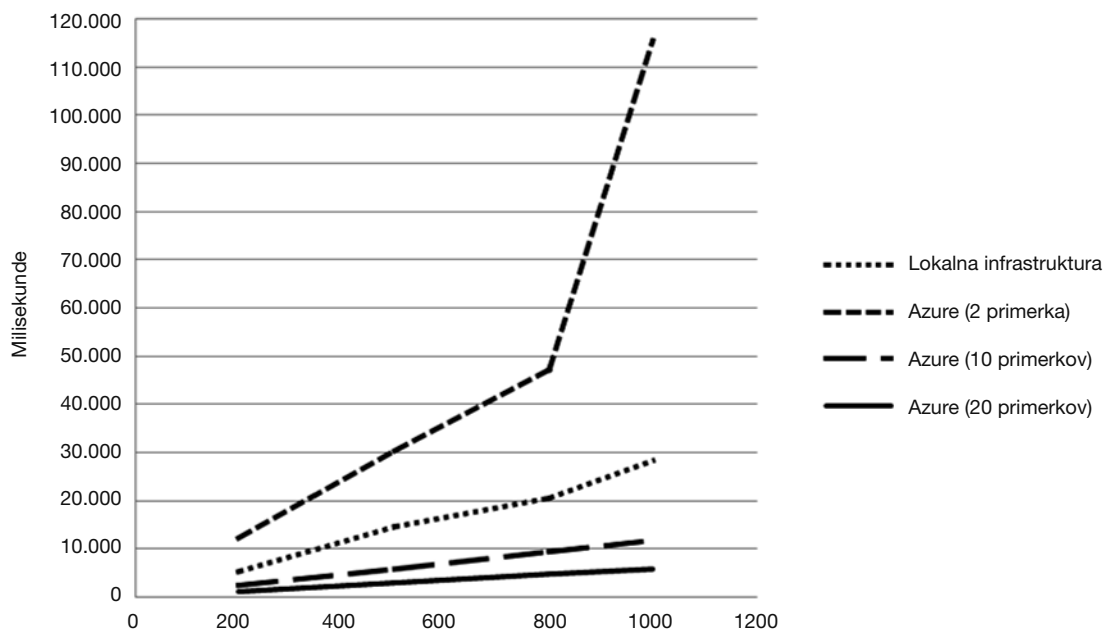
Iz slike 7 je razvidno, da je Azure v primeru dveh primerkov dosegel najvišji odzivni čas in se v skrajnem primeru približal meji dveh minut. Pri tem je treba upoštevati dejstvo, da dva primerka v našem primeru pomenita dvoje procesorskih jeder, kar je polovico manj kot v lokalnem okolju. Opazimo lahko, da se s številom primerkov (procesorskih jeder) odzivni čas opazno zmanjšuje in pri dvajsetih primerkih doseže signifikantno boljše rezultate tudi v primeru skrajnih obremenitev. Pri 1000 uporabnikih je povprečni odzivni čas v primeru Azure 5,83 sekunde, za razliko od infrastrukture lokalnega okolja, ki je pri enakih pogojih potrebovala 28,3 sekunde. Tudi v primeru desetih primerkov je bil povprečni odzivni čas 142 % krajši od lokalnega okolja (11,68 proti 28,3 sekunde). Tabela 4 prikazuje spremembe koeficientov povprečnega odzivnega časa.

Tabela 4: Tabela sprememb koeficientov

	Koeficient spremembe povprečnega odzivnega časa			
Primerjava števila primerkov	200	500	800	1000
2 : 10	5,34	5,20	5,06	9,88
10 : 20	1,97	2,03	2,03	2,00

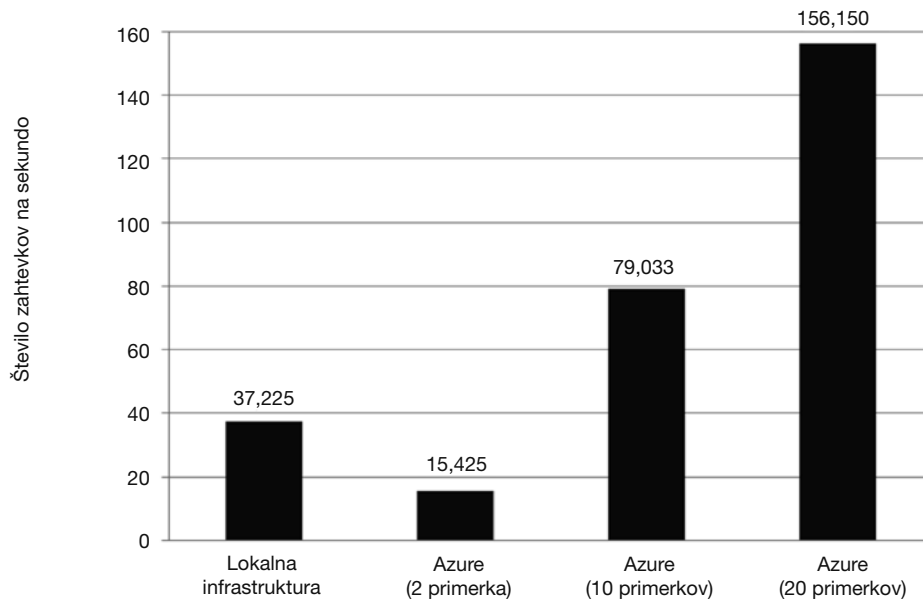
Kot je razvidno iz tabele, se pri petkratnem povečanju števila primerkov tudi koeficient spremembe povprečnega odzivnega časa zmanjša za faktor pet. Pri prehodu z 2 na 10 primerkov lahko opazimo, da je pri dveh primerkih prišlo do zasičenja, kar se odraža kot anomalija v rezultatih.

Slika 8 prikazuje linearno rast odzivnega časa v odvisnosti od število uporabnikov. Izjema je Azure pri dveh primerkih, pri katerih je odzivni čas pri 1000 uporabnikih zaradi preobremenitve presegel prednastavljeno časovno omejitev zahteve. Podatki so posledično izgubili zanesljivost, kar se odraža v obliki presežka odzivnega časa (število napak pri simulaciji).



Slika 8: Povprečni odzivni čas za posamezna okolja

Večanje števila primerkov aplikacije veča tudi prepustnost, ki raste sorazmerno s številom primerkov (slika 9).



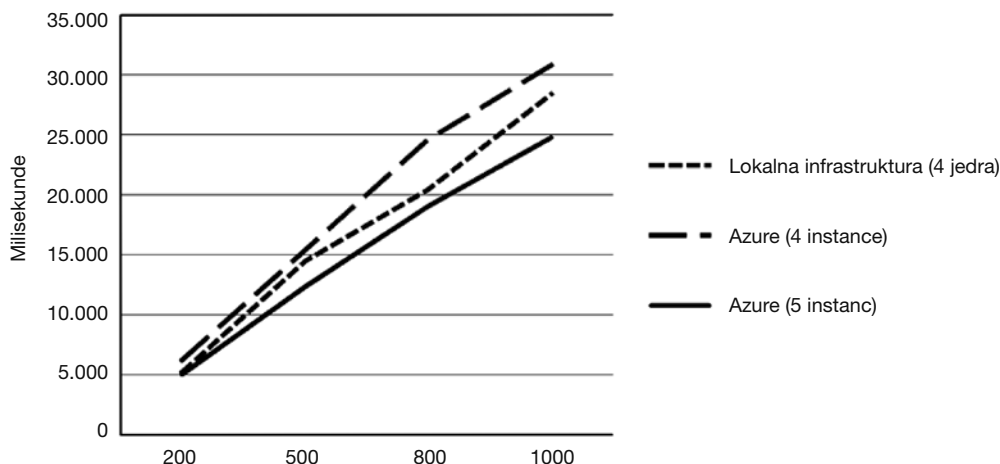
Slika 9: **Mediana prepustnosti**

Slika 9 prikazuje mediano (srednja vrednost) prepustnosti rezultatov simuliranih uporabnikov (200, 500, 800 in 1000). Največjo prepustnost je mogoče opaziti pri Azure s konfiguracijo 20 primerkov, pri čemer je platforma sposobna obdelati 156 zahtevkov na sekundo. Lokalno okolje se je izkazalo za boljše (37 zahtevkov na sekundo) od Azure pri konfiguraciji dveh primerkov (15 zahtevkov na sekundo), vendar pa je bil Azure pri konfiguraciji desetih primerkov za 114 % boljši (79 zahtevkov na sekundo).

V sklopu primerjave smo izvedli tudi simulacijo, s katero smo se želeli s konfiguracijo platforme kot sto-

ritve čim bolj približati zmogljivosti infrastrukture lokalnega okolja. Želeli smo ugotoviti konfiguracijo, pri kateri se obe okolji izenačita z vidika učinkovitosti delovanja. Upoštevati moramo, da je ena izmed omejitev primerjave, da nimamo neposrednega vpogleda v infrastrukturo platforme kot storitve. Zanašali smo se torej le na število procesorskih jeder, saj njihove zmogljivosti ne moremo preveriti (Microsoft Corporation, b. d.).

Infrastruktura lokalnega okolja uporablja štiri jedra, zato smo na Azure izbrali konfiguraciji s štirimi in petimi primerki ter primerjali učinkovitost delovanja (slika 10).



Slika 10: **Primerjava odzivnega časa konfiguracij okolij**

Iz slike 10 je razvidno, da je pri tisoč uporabnikih odzivni čas lokalnega okolja 28,3 sekunde, kar je za približno dve sekundi boljše od Azure pri konfiguraciji štirih primerkov (30,7 sekunde), čeprav bi naj imeli obe okolji v tem primeru enako število procesorskih jeder. Pri konfiguraciji petih primerkov je razlika v prid Azure, pri čemer je za približno 4 sekunde krajši povprečni odzivni čas (24,1 sekunde) od infrastrukture lokalnega okolja (28,3 sekunde). Tabela 5 prikazuje rezultate primerjav simulacije števila uporabnikov, iz katerih je razvidno, da je trend enak kot v primeru tisočih uporabnikov.

Tabela 5: **Primerjava odzivnih časov za posamezna okolja**

Izvajalno okolje	Povprečni odzivni čas (ms)			
	Število uporabnikov			
	200	500	800	1000
Lokalna okolje – IIS	5.217,67	14.415,00	20.374,33	28.301,33
Azure (4 primerki)	6.234,00	15.419,00	24.601,00	30.773,00
Azure (5 primerkov)	4.986,00	12.316,00	18.995,00	24.136,00

Iz grafa (slika 10) in tabele 5 je razvidno, da se pri opazovanju odzivnega časa aplikacija na infrastrukturi lokalnega okolja uvršča med aplikaciji v konfiguraciji oblaka s štirimi in petimi primerki. Rezultati kažejo, da smo lahko le približno ocenili neposredno primerjavo med okoljema.

### 3.3 Omejitve primerjalne analize

V prispevku smo se omejili na Windows Azure, ker smo imeli na voljo dostop do platforme, prav tako pa smo imeli možnost gostovati aplikacijo na infrastrukturi Windows Server 2008 R2. Tehnologiji istega ponudnika sta omogočali bolj enakovredno primerjavo med lokalnim in oblačnim okoljem.

Pri interpretaciji rezultatov moramo upoštevati, da bi lahko drugačne konfiguracije okolja Azure in lokalnega okolja vplivale na končni rezultat primerjalne analize. Kot smo že omenili, smo uporabili privzeto nastavitve Azure (majhna virtualna naprava), ki gosti posamezen primerek aplikacije.

Pri primerjavi smo se omejili le na razčlenjevanje datotek XML, zato ne moremo podajati ugotovitev za primere kakršnih koli drugih opravil. Prav tako nismo vključili podatkovne baze Azure SQL. Do datotek smo v primeru Azure dostopali prek shrambe Azure (Azure ne omogoča običajnega dostopa do datotek), medtem ko smo v primeru lokalnega okolja

dostopali do datotek na disku. Upoštevati moramo, da lahko različna pristopa zajemanja podatkov vplivata na rezultate testa obremenitve.

## 4 SKLEP

Paradigma računalništva v oblaku bo tudi v prihodnje imela pomembno vlogo na področju informacijskih tehnologij. V njen prid govorijo predvsem te lastnosti: zniževanje stroškov, večja svoboda uporabnikov, centraliziranost, agilnost, varnost in preprostost upravljanja.

Namen članka je bil primerjati učinkovitost lokalnega okolja z okoljem v oblaku in preveriti, ali je skaliranje linearno. Porazdeljevanje prometa med primerki bi namreč lahko vplivalo na linearnost (Amdahllov zakon), zaradi česar ne moremo posplošiti, da je skaliranje linearno. Kljub temu smo pri dvajsetih primerkih ugotovili, da linearnost še vedno drži.

V raziskavi smo si prizadevali, da bi bili specifikaciji obeh okolij primerljivi, vendar zaradi različnih konfiguracij tega nismo uspeli doseči. To je narekovalo izbor privzete velikosti primerka, ki vključuje eno procesorsko jedro. Na podlagi tega smo želeli izvesti neposredno primerjavo glede na število procesorskih jeder. Prišli smo do spoznanja, da Azure v primeru štirih primerkov privzete konfiguracije ni ekvivalenten lokalnemu okolju, ki vsebuje štiri procesorska jedra, v primeru petih primerkov pa je njegova zmogljivost višja. Pri tem moramo upoštevati, da pri Azure vsak primerek aplikacije teče na svojem ločenem, oddaljenem strežniku, za razliko od lokalne infrastrukture, pri kateri imamo opravka z eno napravo.

Iz rezultatov je prav tako razvidno, da aplikacija, ki se izvaja na Azure, prinaša boljše rezultate v primeru večjega števila primerkov. Pri majhnem številu primerkov (dva primerka) se je učinkovitost delovanja aplikacije ustrezno zmanjšala. Za zagotavljanje optimalnega delovanja aplikacije moramo najti ravnovesje med številom primerkov in stroški uporabe platforme. V sklopu analize smo iskali tudi primerljivo konfiguracijo platforme kot storitve v primerjavi z infrastrukturo lokalnega okolja. Ugotovili smo, da sta okolji primerljivi pri petih primerkih privzete konfiguracije platforme Azure, vendar je podrobnejša primerjava otežena zaradi omejenih informacij o specifikaciji virov infrastrukture v oblaku.

V prihodnje nameravamo raziskave na področju učinkovitosti PaaS razširiti z vključevanjem podatkovne baze SQL Azure. Prav tako bo testna aplikacija

predstavlja realno poslovno okolje, učinkovitost delovanja pa bomo merili z uporabo kvalitativnih metrik. Na ravni platforme načrtujemo preizkusiti različne velikosti virtualnih naprav posameznih primerkov. Prav tako nameravamo izvesti primerjalno analizo učinkovitosti in lastnosti preostalih ponudnikov PaaS.

## 5 VIRI

- [1] Apache Jakarta Project (b. d.)a. JMeter – User's Manual: Component Reference. Pridobljeno 25. 5. 2011 s [http://jakarta.apache.org/jmeter/usermanual/component\\_reference.html#Aggregate\\_Report](http://jakarta.apache.org/jmeter/usermanual/component_reference.html#Aggregate_Report).
- [2] Apache Jakarta Project (b. d.)b. JMeter – Apache JMeter. Pridobljeno 25. 5. 2011 s <http://jakarta.apache.org/jmeter/>.
- [3] AWS (b. d.). AWS Free Usage Tier. Pridobljeno 25. 5. 2011 s <http://aws.amazon.com/free/>.
- [4] Dudley, R. (2010). *Microsoft Azure enterprise application development: straight talking advice on how to design and build enterprise applications for the cloud*. Birmingham: Packt Publishing.
- [5] Furht, B. & Escalante, A. (ur.) (2010). *Handbook of Cloud Computing*. Boston, MA: Springer US. Pridobljeno s <http://www.springerlink.com/index/10.1007/978-1-4419-6524-0>.
- [6] Google (b. d.). What Is Google App Engine? Pridobljeno 25. 5. 2011 s <http://code.google.com/intl/sl-SI/appengine/docs/whatisgoogleappengine.html>.
- [7] Gospodarska zbornica Slovenije (2011). Računalništvo v oblaku. Pridobljeno 20. 6. 2011 s [http://www.gzs.si/slo/o\\_gzs/gzs\\_jeva\\_zgodba\\_v\\_casu/jutri/racunalninstvo\\_v\\_oblaku](http://www.gzs.si/slo/o_gzs/gzs_jeva_zgodba_v_casu/jutri/racunalninstvo_v_oblaku).
- [8] Hay, C. & Prince, B. H. (2010). *Azure in action*. Greenwich Conn.; London: Manning; Pearson Education [distributor].
- [9] International Organization for Standardization (b. d.). ISO – International Organization for Standardization. Pridobljeno 25. 5. 2011 s <http://www.iso.org/iso/home.html>.
- [10] Mell, P. & Grance, T. (2011). The NIST Definition of Cloud Computing (Draft). Pridobljeno s [http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf).
- [11] Microsoft Corporation (2007). Stress Testing Web Applications. Pridobljeno 25. 5. 2011 s <http://msdn.microsoft.com/en-us/library/bb924374.aspx>.
- [12] Microsoft Corporation (2010). About Windows Azure. Pridobljeno 20. 6. 2011 s <http://msdn.microsoft.com/en-us/library/dd179442.aspx>.
- [13] Microsoft Corporation (2011a). Windows Azure Pricing. Pridobljeno 5. 5. 2011 s <http://www.microsoft.com/windowsazure/pricing/>.
- [14] Microsoft Corporation (2011b). The Next Generation Cloud Database. Pridobljeno 20. 6. 2011 s <http://www.microsoft.com/windowsazure/sqlazure/database/>.
- [15] Microsoft Corporation (b. d.). Windows Azure Compute. Pridobljeno 31. 8. 2011 s <http://www.microsoft.com/windowsazure/features/compute/>.
- [16] Nicola, M. & John, J. (2003). XML parsing. *Proceedings of the twelfth international conference on Information and knowledge management – CIKM '03* (str 175). Predstavljeno na The Twelfth International Conference, New Orleans, LA, USA. doi:10.1145/956863.956898.
- [17] O'Malley, T. & Tomasco, S. (2011). New Global IBM Study Confirms Cloud Computing Poised to Take Off at Companies. Pridobljeno 3. 6. 2011 s <http://www.03.ibm.com/press/us/en/pressrelease/34530.wss>.
- [18] Ouellette, J. (b. d.). Salesforce force.com platform. Pridobljeno 25. 5. 2011 s <http://thecloudtutorial.com/salesforces-forcecom-platform.html>.
- [19] Pavlinek, M., Košič, K., Jošt, G., Gradišnik, M. & Ovcjak, B. (2011). Platforma kot storitev – med infrastrukturo in aplikacijo (str. 37–46). Predstavljeno na OTS 2011, Maribor.
- [20] Pettey, C. (b. d.). Gartner Identifies the Top 10 Strategic Technologies for 2011. Pridobljeno 25. 5. 2011 s <http://www.gartner.com/it/page.jsp?id=1454221>.
- [21] Pošta Slovenije (2011). Informacijske storitve Pošte Slovenije – Računalništvo v oblaku. Pridobljeno 20. 6. 2011 s <http://www.posta.si/novica/26996/Informacijske-storitve-Poste-Slovenije-Racunalninstvo-v-oblaku?nodeid=493&page=1&year=0>.
- [22] Search Cloud Computing.com (b. d.). Verizon: 2010 top cloud computing provider. Pridobljeno 25. 5. 2011 s <http://searchcloudcomputing.techtarget.com/feature/Top-cloud-computing-providers-Verizon#slideshow>.
- [23] Siol.net (2011). Računalništvo v oblaku – nebo je meja. Pridobljeno 20. 6. 2011 s [http://www.siol.net/tehnologija/racunalninstvo/2011/05/telekom\\_racunalninstvo\\_v\\_oblaku.aspx](http://www.siol.net/tehnologija/racunalninstvo/2011/05/telekom_racunalninstvo_v_oblaku.aspx).
- [24] Subashini, S. & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1), 1–11. doi:10.1016/j.jnca.2010.07.006.
- [25] ZOH Creator (b. d.). Platform as a Service, Cloud Computing Platform. Pridobljeno 25. 5. 2011 s <http://www.zoho.com/creator/paas.html>.

■  
Boris Ovcjak je zaposlen na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Med njegove interesne dejavnosti spadajo področja odprtih podatkov javne uprave, računalništva v oblaku in mobilne tehnologije.

■  
Gregor Jošt je raziskovalec in doktorski študent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Med njegove interesne dejavnosti spadajo področja upravljanja poslovnih procesov, računalništva v oblaku in storitveno usmerjene arhitekture.

■  
Gregor Polančič je docent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Med njegova interesna področja spadajo tehnološki in netehnološki vidiki sistemov za komuniciranje, sodelovanje in upravljanje informacijskih procesov vključno z implikacijami sodobnih storitveno usmerjenih informacijskih rešitev z omenjena področja.

■  
Marjan Heričko je redni profesor za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Je namestnik predstojnika Inštituta za informatiko ter vodja Laboratorija za informacijske sisteme. Njegovo raziskovalno delo zajema vsa področja razvoja sodobnih informacijskih rešitev in storitev s poudarkom na naprednih pristopih k modeliranju in načrtovanju informacijskih sistemov, načrtovalskih vzorcih in metrikah. V zadnjem obdobju se posveča tudi področjem storitvene znanosti in storitvenega inženirstva ter uporabniško usmerjenega razvoja.

# Human Resource Information Systems in Croatian Banks: Current Practice and Trends

Ivana Dvorski Lacković  
PBZ stambena štedionica, d. d., Kukuljevićeva 17, Varaždin, Hrvatska  
ivana.dvorski-lackovic@pbz.hr; dvorski.ivana@gmail.com

## Abstract

Human resources are considered to be organizations' most valuable asset, therefore its adequate management is required. Information technology is believed to be crucial in human resource management in contemporary organizations. Banks are owing to their characteristics especially sensitive to human resource management: careful selection of employees is required, their training and performance need to be measured and tracked, legislative requirements regarding human resource management need to be met, but most importantly, employees need to be motivated in order to give better results. In this article the author gives a general overview of human resource information systems (HRIS) and their application in Croatian banks. A survey was conducted in order to explore current practice of HRIS usage in Croatian banking industry. Some of the researched areas include modality of HRIS development, functionalities it supports, general level of satisfaction with implemented HRIS and target aims when developing it. The results of the research support the findings of other similar researches in the world.

**Key words:** Human Resource Information Systems, Banking industry, Human Resource Management.

## Izvlaček

### Informacijski sistemi za upravljanje človeških virov v hrvaških bankah: sedanja praksa in trendi

Človeški viri so najbolj dragocena dobrina vsake organizacije, zato jih je treba primerno upravljati. Pri tem je v sodobnih organizacijah vloga informacijske tehnologije ključnega pomena. Upravljanje človeških virov je v bankah zaradi njihovih posebnih lastnosti še posebej občutljivo; potrebujejo namreč skrben postopek izbire pri zaposlovanju, spremljanje usposabljanja in merjenje uspešnosti svojih zaposlenih ob upoštevanju zakonitosti, predvsem pa bi moralo upravljanje človeških virov motivirati zaposlene za doseganje boljših rezultatov.

V prispevku avtorica predstavi splošen pregled informacijskih sistemov (HRIS) za upravljanje človeških virov in njihovo uporabo v hrvaških bankah. V ta namen je bila izpeljana raziskava, ki ugotovi, kakšno je trenutno stanje glede uporabe HRIS v hrvaškem bančništvu. Nekatera raziskana področja vključujejo modaliteto razvoja HRIS, funkcionalnosti, ki jih podpira, splošno stopnjo zadovoljstva z implementacijo HRIS in doseženimi cilji. Rezultati raziskave so skladni z izsledki podobnih raziskav v svetu.

**Ključne besede:** informacijski sistem za upravljanje človeških virov, bančništvo, upravljanje človeških virov.

## 1 INTRODUCTION

Human resources are considered to be organizations' most valuable asset. Satisfied employees are a precondition for satisfied customers (Sinčić, Pološki, 2007). This statement is of special importance in banking industry where employees are exposed to interaction with clients on daily basis. The field of human resource management (HRM) can be characterized as having encountered frequent and numerous innovations in technology (Ngai et al., 2008). One of these is the implementation of Human Resource Information Systems (HRIS) in order to support HRM processes. Experience shows there are many advantages coming out of

the adoption of HRIS, but also obstacles to its implementation. So far there are few research paper topics for HRIS in Croatia: HRIS practice in large Croatian companies (Markić et al., 2011), HRIS development at universities (Bambir et al., 2010) and HRIS in hotels at the Adriatic coast (Arnerić et al., 2007). In this article the author discusses different aspects of HRIS implementation in Croatian banking industry. In order to support the theory, a survey has been conducted among Croatian banks. This paper is divided in seven parts.

After introductory notes, the author gives general overview of Human Resource Management and contemporary tendencies in the field. Following are the

endencies in HRIS development and its implementation in banking. After presenting the conducted survey and discussing its results, conclusions are made.

## 2 HUMAN RESOURCE MANAGEMENT

Human resources are total knowledge, skills, creative abilities, motivation and loyalty an organization has on disposal. It is total intellectual and psychic energy the organization can employ on goals achievement and business development (Bahtijarević – Šiber, 1999, p. 16).

Human Resource Management (HRM) is commonly defined as a process of acquiring, training, appraising and compensating employees, and attending to their labor relations, health and safety and fairness concerns (Dessler, 2005, p. 4).

HRM is a sequence of interconnected activities and managerial tasks focused on insurance of adequate employee number and structure, their knowledge, skills, interests, motivation and behavior modalities necessary for the achievement of current, developmental and strategic organizational goals (Bahtijarević – Šiber, 1999, p. 17).

HRM aims are to (Armstrong, 1999, p. 4): provide a range of services which support the achievement of corporate objectives as a part of the process of running the organization; enable the organization to obtain and retain the skilled, committed and well-motivated workforce it needs; enhance and develop the inherent capacities of people – their contributions, potential and employability – by providing learning and continuous development opportunities; create a climate in which productive and harmonious relationships can be maintained between management and employees and in which feelings of mutual trust can be developed; develop an environment in which teamwork and flexibility can flourish; help the organization to balance and adapt to the needs of its stakeholders (owners, government bodies or trustees, management, employees, customers, suppliers and the public at large); ensure that people are valued and rewarded for what they do and achieve; manage a diverse workforce, taking into account individual and group differences in employment needs, work style and aspiration; ensure that equal opportunities are available to all; adopt an ethical approach to managing employees which is based on concern for people, fairness and transparency; and maintain and improve the physical and mental wellbeing of employees.

HRM, due to the quantitative complexity of the profession, coupled with its responsibility for enormous record-keeping, has a very long history of information technology experience in organizations and continues to be a leader in the implementation and use of IT (Townsend and Bennett, 2003).

## 3 GENERAL OVERVIEW OF HUMAN RESOURCE INFORMATION SYSTEMS

Human Resource Information System (HRIS) is defined as the composite of databases, computer applications and hardware and software necessary to collect/record, store, manage, deliver, present and manipulate data for human resources (Broderick and Boudreau, 1992).

HRIS is used to gather and maintain the data that describe human resources (HR), transforming data into information and then reporting the information to users (Ngai et al, 2008). HRIS is a concept which utilizes the development of Information Technology for effective management of the Human Resource functions and applications. It has emerged as an important interdisciplinary tool to achieve vital organizational Human Resource objectives. HRIS is applied in personnel administration, salary administration, leave/absence recording, skill inventory, medical history, performance appraisal, training and development, HR planning, recruitment, career planning, negotiations etc. (Singh et al., 2011). Nowadays HRIS is considered to be an integral part of Enterprise Resource Planning (ERP) system and thus supports a process-oriented view of organization and standardizes business processes across the enterprise (Nah et al., 2003). ERP system is a widely accepted solution to achieve an integrated enterprise information system (Bajgorić and Moon, 2009) because it integrates key functions of an organization, such as accounting, finance, human resources and support functions into one universal database.

The basic purpose of HRIS is:

- Insurance of an adequate information base for decision making process in human resources area;
- Development of a selection program, education, training, promotion, career planning, communication;
- Analysis of working abilities and employees' development potential;
- Complying working claims with individual possibilities and interests;

- Establishing needs for further qualification or knowledge renewal;
- Career planning;
- Ensure continuous analysis of structure, success and movement of human resources;
- Making reports on human resources;
- Enabling organizational analysis. (Bahtijarević – Šiber, 1999, p. 221)

Bahtijarević – Šiber (1999, p. 223) also points out that an organization must ensure data on working places, their basic characteristics and claims in order to be able to continuously comply working claims with individual possibilities.

For the purpose of this article HRIS is defined and will be considered as an information system whose basic purpose is the support of human resource management through facilitated processes of data collection, objective informing and decision making.

According to Panayotopoulou et al. (2007) there are several effects of technology on six key HR processes, namely:

- HR planning;
- Acquiring HR (recruitment and selection);
- HR evaluation (performance appraisal);
- Communication;
- Rewarding HR (performance appraisal, compensation and benefits); and
- Developing HR (training and development, career management).

Some authors discuss that the nature of HRIS varies among organizations in relation to their size. In small organizations, it tends to be informal, whereas in large organizations it is more formal and coordinated (Singh et al., 2011). According to Ball (2001) the organizational size is a determinant, whether an organization has HRIS at all and whether it adopts certain modules over others (core personnel administration over training and recruitment). Ngai and Wat (2006) tried to determine the relationship between company size and HRIS adoption, and there were positive, statistically significant differences between the adoption of HRIS and company size, what shows that large companies are more likely to adopt HRIS. According to organizational size HRIS can be implemented at three different levels: publishing of information, automation of transactions or transformation of HR into a strategic partner with the line business (Lengnick-Hall and Moritz, 2003).

Many companies have problems when implementing new technologies, including HRIS, due to lack of

sufficient capital and skills (Ngai and Wat, 2006). Therefore, companies are reluctant to implement HRIS, unless they are convinced in benefits the information system would bring to their organizations. The common benefits of HRIS include enhancement in executive decision making, employee training, technology usage, interdepartmental integration and better reporting structures (Mayfield, Mayfield and Lunce, 2003).

Beckers and Bsat (2002) pointed out at least five reasons why companies should use HRIS. This is what HRIS can do:

- Increase competitiveness by improving HR operations;
- Produce a greater number and variety of HR-related reports;
- Shift the focus of HR from the processing of transactions to strategic HRM;
- Make the employees part of HRIS; and
- Reengineer the entire HR function of companies.

A survey conducted by the Institute of Management and Administration in the year 2002 indicated that the biggest problems in managing HRIS include: lack of staff, lack of budget, problems with time management, need to work with other departments and lack of information technology support (Ngai and Wat, 2006). Since these are rather general obstacles that can relate to any information system, a list of more specific obstacles, when implementing and managing HRIS, would include:

- *Responsibility for HRIS design*: HRIS presents a field of intersection between human resource management and information and communication technology (Arnerić et al., 2007) and due to its interdisciplinary character in many cases it is yet unclear, who are key persons with adequate level of knowledge within the organization, responsible for basic HRIS design;
- *Compliance with legislation*: the field of HRM is regulated by several national laws and regulations. When making HRIS, all specific claims need to be taken into consideration in order to eliminate compliance risk;
- *Internal policies*: besides general “roof” policies that regulate information system management, specific policies about HRIS management need to be created and put into practice in the organization. This is especially important because of the sensitivity of the data that HRIS encompasses (personal data protection);

- *Measuring Return on Investment (ROI) of HRIS*: measuring ROI of HRIS is rather intangible. Therefore, management should develop a list of key indicators or ROI measurement methodology that would enable it to estimate benefits of HRIS more easily.

#### 4 HUMAN RESOURCE INFORMATION SYSTEMS IN BANKING INDUSTRY

Banking sector started the technology adoption process early on and before other sectors (Panayotopoulou et al., 2007) owing to great global competition and struggle for clients. A satisfactory service level is the banks' competitive advantage. A contemporary client is no longer satisfied only with the availability of a product, he seeks active communication with the bank's employees. Therefore, the banks are starting to focus on the values they can get from their employees. Also, one must consider that there are more and more back office employees, who support front office operations and the employees exposed to the public.

HRIS has transformed banks' financial as well as non-financial processes from manual to automatic computerized systems (Singh et al., 2011). To be more precise, at the beginning of its development HRIS included mostly data on salary and financial benefits (financial processes). With time HRIS has developed and nowadays supports testing, recruiting, training and learning (non-financial processes). This statement is supported by several sources, who claim that HRIS is now being used not only for administrative purposes but also for strategic and business decision-making purposes (Broderick and Boudreau, 1992; Kossek et al., 1994; Kovach et al., 2002; Ngai and Wat, 2006). Having this in mind, it becomes clear, why HRIS is relevant to contemporary banks and their management.

There are many questions related to HRIS implementation in banking that are awakening interest. Does HRIS depend on banks' size, measured in number of employees? Is banks' size one of prerequisites for the investment of larger amount of funds into HRIS? The greatest barrier to the adoption of HRIS is according to some researches the insufficient financial support (Ngai and Wat, 2006). How is HRIS developed, internally or externally? IS personnel possess some technical skills, derived from training and experience, that allow them to perform IS functions, i.e. develop and implement new IS applications, operate the IS to provide the services required, maintain

the IS running and/or conceive, plan and exploit IS applications to support and enhance other business functions (Alvarez-Suescun, 2007).

What functionalities are supported by HRIS? Are employees active users of HRIS or are they passive information givers for the needs of management data collection?

#### 5 RESEARCH: HUMAN RESOURCE INFORMATION SYSTEMS IN CROATIAN BANKS

In order to research the state of the art of HRIS implementation and its usage in Croatian banks, a survey was sent to all Croatian banks ( $n=33$ ) during September 2011 with a request to forward the survey to the employee in charge of human resource management. The response rate was 27.27 % (the absolute number of surveyed banks was 9), which is within the expected range. A similar research achieved a response rate of 29 % (Ngai et al., 2006; Ngai et al., 2008) and 24.4 % (Ball, 2001).

The banks that answered survey questions for the research were divided according to two criteria. The first was the criterion of Croatian National Bank that classifies banks as small, medium and big. One survey question was the number of bank employees and that was the second criterion for bank classification. The structure of the surveyed banks according to the first criterion and its comparison to the structure of all Croatian banks according to their size is shown in Table 1.

Most of the surveyed banks are considered small (5), follow big banks (3) and finally a medium size bank (1). This distribution is considered normal and satisfactory for the needs of the research since the structure of banks' size in Croatia is similar.

Table 1: Comparison of all Croatian banks and surveyed banks' size

Bank size	All Croatian banks	Surveyed banks
Small	24	5
Medium	3	1
Big	6	3
TOTAL	33	9

Table 2 displays the number of employees in the surveyed banks. 4 surveyed banks are within the category of 200 to 500 employees. The number of the surveyed banks' employees can be considered compliant with the bank's size.



Table 2: Structure of the surveyed banks according to the number of employees

Number of employees	Number of banks
Less than 200	2
200–500	4
501–1000	0
1001–2000	1
2001–3000	1
More than 3000	1
TOTAL	9

The findings of the research indicate that all the surveyed banks have a separate organizational unit in charge of HRM. Also all the banks have implemented HRIS. The year of HRIS implementation is shown in Table 3.

Table 3: Structure of the surveyed banks according to the year of HRIS implementation

Year of HRIS implementation	Number of banks
2000	3
2002	2
2003	1
2004	1
2007	1
2008	1
TOTAL	9

All three big banks and one medium size bank from the sample stated that the year of HRIS implementation was in the period from 2000 – 2003. In the same period only two banks from the sample that are considered small implemented HRIS. This leads to the expected conclusion that big banks had the opportunity to develop HRIS prior to small banks.

Another interesting finding of the research is that no significant connection has been found between bank size and modality of HRIS development: 5 surveyed banks outsourced HRIS development compared to 2 banks that developed it within banks organizational units and 2 banks that used cooperation of an outsourced company and internal resources (Table 4). Outsourcing was used by three small, one big and one medium size bank. One big and one small bank from the sample developed HRIS internally. Also one big and one small bank from the sample used cooperation of internal and external resources for HRIS development.

Table 4: Structure of the surveyed banks according to modality of HRIS development

Modality of HRIS development	Number of banks
Internally	2
Externally	5
Cooperation	2
TOTAL	9

Alvarez – Suescun (2007) discussed that firm's internal factors, such as HRIS implementation capability and the strategic contribution of HRIS, determine whether that activity is undertaken internally or outsourced, whereas the technical skill set and the firm's size do not affect that decision and that seems to be confirmed by this research since both big and small banks used all three modalities of HRIS development and no statistically relevant correlation between the bank's size and the decision whether to outsource HRIS development had been found. Alvarez – Suescun (2007) also points out that possessing a HRIS implementation capability is a necessary condition for insourcing this function, but is not sufficient inducement to initiate the process. The capability needs to be superior (relative to other firms), i.e. enable the firm to implement the HRIS cheaper and faster while obtaining an application better adapted to and more useful for its needs, in order to turn into a key determinant of IS sourcing decisions.

A very significant aspect of HRIS are its functionalities. They can offer basic features or be a strategic decision making HRM tool. In order to research the "content" of HRIS, a list of functionalities had been made. This list is a synthesis of relevant employees' data file (Bahtijarević – Šiber, 1999, p. 222), Croatian Labor Law and Credit institutions act claims. To explain this in detail, employees' data file should include their personal data, working experience, education, rotation within bank, salary data, data on employees' work success, working plans for future periods, benefits and stimulation data, data on absenteeism and disciplinary actions, psychological testing results, education plans, data on development potential, identification of highly potential employees, employees' specific working preferences and health data. According to Croatian Labor Law organizations have the obligation to monitor the data on vacation and working hours and according to Credit institutions' act all employees need to have access to

working places systematization and internal acts related to human resource management. Additionally, the author added two functionalities: self service for employees and internal job market.

Table 5: **Distribution of HRIS features in the surveyed banks**

Functionality	Number of banks
Employees' personal data	9
Data on employees' work experience	9
Data on employees' education	9
Data on employees' rotation within bank	9
Data on vacation	9
Salary data	9
Data on working hours	8
Data analysis and making reports	8
Working places systematisation	6
Data on grading employees' work success	5
Making employees' working plans for future periods	5
Benefits and stimulation data	5
Potential future employees' data (testing, selection)	5
Data on absenteeism and disciplinary actions towards employee	4
Employees' psychological testing results	4
Self service for employees	4
Education plans	3
Data on employees' development potential	3
Identification of highly potential employees	3
Internal job market	3
Internal acts on human resources and employees' rights	2
Employees' specific working preferences	0
Employees' health data	0

As can be seen from Table 5, all the surveyed banks included demographic data and data on vacation and salary in their HRIS. Some basic information systems prerequisites, like making reports and data analysis, are one of the basic features. About a half of the surveyed banks' HRIS includes specific data about employees, like grading their success, making working plans for future period, data on absenteeism or data on potential future employees. What is rather significant is the fact that most of the banks do not offer advanced HRIS features, for example education plans for employees, data on their development potentials, internal job market that would make employment process easier. The advanced

HRIS features are used mostly in big banks, which is consistent with the results of the prior research that shows connection between company size and HRIS usage (Ball, 2001; Ngai et al, 2006).

Three big banks, one medium and one small bank from the sample stated that all the employees have access to some HRIS functionalities, what might be an indicator of the bank's usage of HRIS for active HRM that enables employees to be active participants in their own career management.

The following researchers' interest has been also to find out, whether the banks had identified different levels of authorization for the employees' HRIS usage. 8 surveyed banks have confined the rights to 'write' and 'see' because there are different levels of information an individual employee can access. As for data protection, all the banks answered that employees are familiar with the process of information collection, their purpose and use.

Another research question was a number of applications from which HRIS as centralized integrated database "pulls" its data. Most small banks are using one application as data source while big banks are getting data from more applications. This is consistent with previous findings on functionalities HRIS supports. To explain, small banks are focused on basic HRIS functionalities like payroll, personal data, working hours and big banks encompassed also advanced functionalities, like education plans or identification of highly potential employees. Therefore, it is necessary and expected for big banks to pull data from more sources. This distribution is shown in Table 6.

Table 6: **Banks' distribution according to the number of applications used for HRM**

Number of applications	Number of banks
1	3
2	3
3	0
4	2
5	1
TOTAL	9

When being asked about reasons for HRIS development, all three big banks and one small bank from the sample answered it was the wish for internal improvement of processes that was critical. Four small banks and a medium size bank said legal regulati-

ons were what made them implement HRIS. Among other reasons for HRIS implementation one big bank stated that the wish for a better quality HRM was also their guideline. The answers to this survey question are a nice indicator how big banks are more likely to develop HRIS for the reasons beyond pure legislative force.

As for the banks' satisfaction with HRIS, seven banks are satisfied with the implemented HRIS and have noticed improvement in HRM process.

## 6 CONCLUSION

Basic purpose of this article was to give general overview of HRIS state of the art and implementation in Croatian banks. Possible limitation of the research is that only 9 of 33 banks answered the survey, but since the survey response rate and the surveyed banks size distribution are found satisfactory, the survey can be considered a good indicator of current practice in Croatian bank industry. HRM is one of the most important strategic areas for the development of banks since their front office employees are in continuous interaction with clients and back office employees are responsible for the tasks oriented to constant service, product improvement and competition with other banks. This is where adequate HRIS is significant. It can be more than a tool for following the employees' basic demographic data and working hours. It can be a tool for identifying highly potential employees and directing them to areas of their working interest. It can give the management the information of not only what was done, but who had done it. It also enables it to give feedback and not just take disciplinary, but also rewarding actions towards employees.

The findings of the research are the following: big banks are more likely to develop advanced HRIS features than small ones (what is consistent with the findings of Ball, 2001 and Ngai and Wat, 2006), the bank size is not a relevant factor for outsourcing HRIS development (comparable with Alvarez – Suescun, 2007), HRIS in big banks gathers data out of more applications than HRIS in small banks, big ones are impelled by internal improvement and not just by legislative to develop HRIS.

The findings of this research can be used in business practice in small banks. They can use a list of HRIS functionalities listed in Section 5 and use it as a template for their HRIS improvement and its upgrading with advanced functionalities.

The future research related to this topic might include comparison of HRIS usage in the banks of the region, development of universal key indicators that can be used for measuring ROI on HRIS in banks or researching what factors influence the banks' decision on outsourcing HRIS development.

## 7 REFERENCES

- [1] Alvarez-Suescun, E. (2007), Testing resource – based propositions about IS sourcing decisions, *Industrial Management and Data Systems*, Vol. 107 No. 6, pp. 762–779.
- [2] Armstrong, M. (1999), *Human Resource Management Practice*, 7th ed., Kogan Page, London.
- [3] Arnerić, J., Garbin Praničević, D., Tadić, I. (2007), The Use of human resource information systems (HRIS) in Croatian hotels on the Adriatic coast: A survey, *Global Business and Economics Antology*, Vol. 1, pp. 110 – 121.
- [4] Bahtijarević – Šiber, F. (1999), *Management ljudskih potencijala*, Golden marketing, Zagreb.
- [5] Bajgorić, N. and Moon, Y. B. (2009), Enhancing systems integration by incorporating business continuity drivers, *Industrial Management and Data Systems*, Vol. 109 No. 1, pp. 74 – 97.
- [6] Ball, K. (2001), The use of human resource information systems: a survey, *Personnel Review*, Vol. 30 No. 6, pp. 677–693.
- [7] Bambir, D., Hutinski, Ž., Drozdova, M. (2010), A Review of Some Basic Prerequisites for Human Resource Management Information Systems Implementation at the Universities, *Proceedings of the 21<sup>st</sup> Central European Conference on Information and Intelligent Systems*, Varaždin: Fakultet organizacije i informatike, pp. 297–303.
- [8] Beckers, A. M. and Bsat, M. Z. (2002), A DSS classification model for research in human resource information systems, *Information Systems Management*, Vol. 19 No. 3, pp. 41–50.
- [9] Broderick, R. and Boudreau, J. W. (1992), Human resource management, information technology and the competitive edge, *Academy of Management Executive*, Vol. 6 No. 2, pp. 7–17.
- [10] Croatian National Bank (2011), *Banks Bulletin*, No. 22.
- [11] Dessler, G. (2005), *Human Resource Management*, 10th ed., Pearson Education, Inc., Upper Saddle River.
- [12] Kossek, E. E., Young, W., Gash, D. C. and Nichol, V. (1994), Waiting for innovation in the human resources department: Godot implements a human resource information system, *Human Resource Management*, Vol. 33 No. 1, pp. 135–59.
- [13] Kovach, K. A., Hughes, A. A., Fagan, P. and Maggitti, P. G. (2002), Administrative and strategic advantages of HRIS, *Employment Relations Today*, Vol. 29 No. 2, pp. 43–8.
- [14] Lengnick-Hall, Mark L. and Moritz Steve (2003), The Impact of e-HR on the Human Resource Management Function, *Journal of Labor Research*. 24(3), pp. 365–379.
- [15] Markić, D., Horvat, J., Oreški, D. (2011), Human Resource Information Systems in Croatian Companies, *Proceedings of ITI 2011*, Zagreb: Sveučilišni računski centar, pp. 415–420.
- [16] Mayfield, M., Mayfield, J. and Lunce, S. (2003), Human resource information systems: A review and model development, *Advances in Competitiveness Research*, Vol. 11 No. 1, pp. 139–152.
- [17] Nah, F., Zuckweller, K. and Lau, J. (2003), ERP Implementation: Chief Information Officers' Perception of Critical Success Factors, *International Journal of Human – Computer Interaction*, Vol. 16 No. 1, pp. 5–22.

- [18] Narodne novine (2009), Zakon o kreditnim institucijama, Zagreb: Narodne novine d.d. 153.
- [19] Narodne novine (2009), Zakon o radu, Zagreb: Narodne novine d.d. 149.
- [20] Ngai, E., Law, C., Chan, S., Wat, F. (2008), Importance of the Internet to human resource practitioners in Hong Kong, *Personnel Review*, Vol 37, No. 1, pp. 66–84.
- [21] Ngai, E., Wat F. (2006), Human Resource Information Systems: a review and empirical analysis, *Personnel Review*, Vol 35, No. 3, pp. 297–314.
- [22] Panayotopoulou, L., Vakola, M., Galanaki, E. (2007), E–HR adoption and the role of HRM: evidence from Greece, *Personnel Review*, Vol. 36 No. 2, pp. 277–294.
- [23] Sinčić, D., Pološki Vokić, N. (2007), Integrating internal communications, human resource management and marketing concepts into the new internal marketing philosophy, Working paper series, Paper No. 07–12, Faculty of economics and business, Zagreb.
- [24] Singh, H. P., Jindal, S., Samim, S. A. (2011), Role of Human Resource Information Systems in Banking Industry of Developing Countries, The First International Conference on Interdisciplinary Research and Development, 31 May – 1 June 2011, Thailand.
- [25] Townsend, A. M. and Bennett, J. T. (2003), Human resources and information technology, *Journal of Labor Research*, Vol. 24 No. 3, pp. 361–3.

■

Ivana Dvorski Lacković graduated at Faculty of Economics and Business in Zagreb and is currently a PhD student of direction Information Science at Faculty of Organization and Informatics in Varaždin. The author has over five years working experience in Croatian banking sector. She is the author of several scientific papers in the interdisciplinary area of information science and economics.

# Uporabnost metafor v informatiki

Saša Kuhar, Marjan Heričko

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova ul. 17, 2000 Maribor

sasa.kuhar@uni-mb.si; marjan.hericko@uni-mb.si

## Izvleček

Metafore pomagajo pri razumevanju novih in abstraktnih pojmov, zaradi česar jih zasledimo v večini informacijskih rešitev. Vse od njihove prve uporabe v grafičnih uporabniških vmesnikih so bile predmet pozitivnih in negativnih kritik. Metafore uporabljamo tudi v programskem inženirstvu, v katerem so uporabno orodje v ekstremnem in objektno orientiranem programiranju.

S pregledom literature smo ugotovili, da imajo metafore v informatiki pomembno vlogo. Raziskava se osredinja na prednosti in slabosti uporabe metafor pri zasnovi uporabniških vmesnikov. Za uporabnost uporabniškega vmesnika sta izbira primerne metafore in njena implementacija ključnega pomena, zato podamo napotke za izbiro metafor in njihovo vizualizacijo. Na podlagi rezultatov raziskave ugotovimo, da so metafore pomemben pripomoček pri prepoznavanju sistemske funkcionalnosti.

**Ključne besede:** metafore, uporabniški vmesniki, programsko inženirstvo, vizualizacija.

## Abstract

### On Usability of Metaphors in Informatics

Metaphors help to understand new and abstract concepts and have been for that reason implemented in nearly all computer systems. Since their very first appearance in graphic user interfaces and because of some inappropriate implementations, the metaphors have been a topic of diverse opinions about the benefits they supposedly produce in human-computer interaction. Metaphors have their role also in program engineering, as they are a helpful tool in extreme programming and object-oriented programming.

Literature review showed the important role metaphors play in user interfaces. In the article we show the benefits and disadvantages of metaphors used in informatics. Since the choice of a good metaphor is crucial for the success of it, we provided directions for choosing an appropriate metaphor as well as for visualizing it. With the help of a research we found out that metaphors do help users interact with the system.

**Key words:** metaphors, user interfaces, program engineering, visualization.

## 1 UVOD

Metafore se pogosto pojavljajo v verbalni komunikaciji ter v naših mislih, čeprav se jih večinoma ne zavedamo. Že v stari Grčiji so ugotovili, da so koristen pripomoček pri učenju in spoznavanju novih stvari ter konceptov. Metafore s pomočjo znanega pojma predstavijo lastnosti in značilnosti nove, še nepoznane stvari ali abstraktnega koncepta ter tako olajšajo pojmovanje in kvantificiranje abstraktnih stvari (Barr, 2003). V informatiki se pojavlja veliko abstraktnih in marsikdaj nepoznatih stvari. Za predstavitev teh pojmov in njihovega delovanja bi bilo treba veliko napora, če ne bi poznali metafor. Že ob nastanku prvega namiznega računalnika, ki je uporabljal grafični uporabniški vmesnik, se pojavi metafora »namizje«, danes pa so metafore implementirane v vse uporabniške vmesnike. Kategoriziramo jih v tri skupine: orientacijske (gumbi naprej in nazaj, količinski drsniki, vrstice stanja), ontološke (poimenovanje in shranjevanje dokumentov, definiranje njihove velikosti) in – najpogostejše – strukturne metafore (namizje, koš, mapa, kalkulator itd).

Po začetnem navdušenju nad uporabo metafor v uporabniških vmesnikih so jih začeli načrtovalci uporabljati v prevelikem obsegu (Blackwell, 2006). Polom aplikacij, v katerih metafore niso bile primerno implementirane, je privedel do kritik njihove uporabe ter do prvih obširnejših raziskav na tem področju.

Metafore se v informatiki uporabljajo tudi pri programiranju. V nekaterih primerih se pojavljajo kot poimenovanja problemov, principov in metod v programiranju ali pa so pripomoček pri načrtovanju sistemov. V ekstremnem programiranju uporabljajo sistemske metafore, ki olajšajo komunikacijo med naročniki, vodstvom in razvijalci (Khaled & Noble, 2005).

V članku se najprej posvetimo teoriji metafor in te definiramo. Sledita razdelka o metaforah v programiranju in v uporabniških vmesnikih. Podamo napotke za izbiro primernih metafor ter za njihovo vizualizacijo. Članek sklenemo s predstavitevjo

rezultatov raziskave o prepoznavanju funkcionalnosti sistemov s pomočjo metafor.

## 2 KAJ SO METAFORE

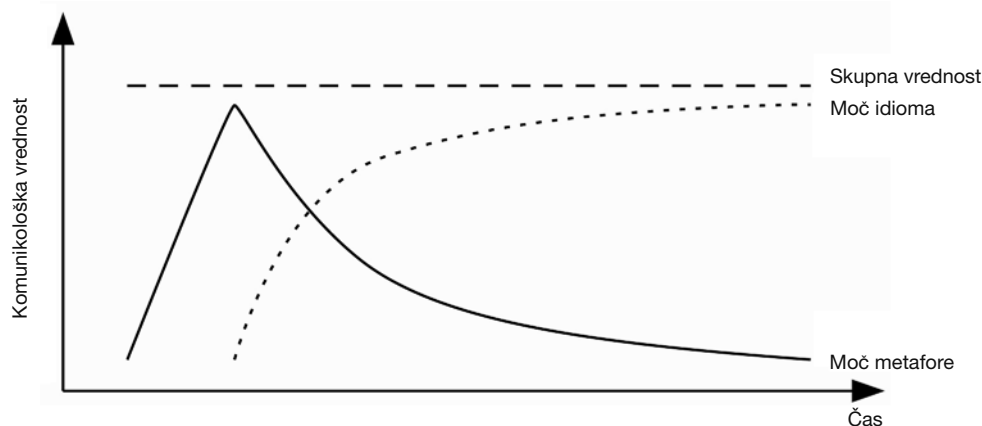
### 2.1 Definicije metafor

Že Aristotel je spoznal pomembnost metafor ter jih imel za ključne pri spoznavanju in učenju novih stvari in konceptov (Honeycutt, 2004). Moderne teorije (Lakoff & Johnson, 2003; Reddy, 1993), so prinesle nova spoznanja o našem mentalnem sistemu, ki naj bi bil že po naravi metaforičen: metafore ne obstajajo samo v jeziku, ampak tudi v naših mislih »na način, kako konceptualiziramo eno domeno skozi pomen druge« (Lakoff & Johnson, 2003).

Lakoff in Johnson (Lakoff & Johnson, 2003) sta metaforo razdelila na dva dela: razlagajoči del metafore je izvorna, del, ki potrebuje razlago, pa ciljna domena. Definicijo lahko podrobneje razložimo na Shakespearovi metafori »Julija je sonce«, v kateri je sonce izvorna domena, Julija pa ciljna. Julija je tako topla, svetla, središče sveta itd. Avtorja sta poudarila, da je metafora samo delna, saj se ne prenese celotnega pomena ene domene na drugo. Julija namreč ni ogromna ali žgoča, kot je sonce.

Priročnik *The Oxford Companion to Philosophy* (Honderich, 1995) razkrije pomembno lastnost metafor. Pravi namreč, da ljudi interpretiramo metafore ter da jih interpretiramo različno. Metafore niso interpretirane samo z vidika posameznika, ampak v povezavi z njegovim kulturnim kontekstom, v katerem lahko kultura pomeni lokacijo bivanja, religijo, izobrazbo, pripadnost socialnim skupinam itd. Ljudje z različnim kulturnim ozadjem imajo lahko različne interpretacije enakih metafor. Število ljudi, ki razume metaforo, je odvisno od stopnje znanja (prirojeno, senzorno-motorično, kulturno ali strokovno), ki je potrebno za interpretiranje določene metafore.

Metafore imajo življenjski cikel, ki ima pet stopenj: rojstvo, pronicljivost, polna moč, splošnost in smrt. Ko je metaforični izraz interpretiran prvič, se metafora rodi. S tem aktivira interpretorja (ali izvor) in omogoči vpogled v lastnosti ciljne domene. Po takšni interpretaciji ima metafora največjo moč. Kmalu prvi vtis zaradi izkušenj postopoma zbledi, metafora postane splošna in na koncu se spremeni v idiom ali splošno poznano frazo (Pirhonen, 2005).



Slika 1: **Moč metafore skozi življenjski cikel (povzeto po Pirhonen, 2005)**

Ker metafore obstajajo v naših mislih in niso omejene na jezik, lahko zavzamejo različne oblike. Metafore so lahko verbalne, vizualne, glasovne, lahko so v obliki gest itd. (Barr, 2003).

Barr (Barr, 2003) je definiral štiri ključne lastnosti metafor:

1. metafore so po definiciji napačne;
2. metafore so lahko različno interpretirane, kar je odvisno od posameznikovega znanja in izkušenj ter kulturne pripadnosti;
3. metafore naj bi bile ena izmed ključnih metod za razumevanje novih pojmov;
4. metafore ne obstajajo samo v jeziku, ampak delujejo znotraj sistema mišljenja.

Pri pregledu literature smo seznamu dodali nadaljnji dve:

5. metafore čez čas izgubijo moč in se spremenijo v idiom;
6. metafore lahko zavzamejo različne oblike, tudi vizualne.

Definicijo metafore bi lahko povzeli v stavku: Metafora je besedna figura, v kateri se s procesom enakega poimenovanja pomen ene stvari prenese na kakšno drugo stvar.

### 3 METAFORE V PROGRAMSKEM INŽENIRSTVU

Metafore se v programskem inženirstvu pojavljajo kot poimenovanja problemov in principov (problem trgovskega potnika, nevronske mreže, odločitvena drevesa, evlucijsko in genetsko programiranje, podatkovno rudarjenje ter skladiščenje itd.) ali pa kot metode in pripomočki v programiranju (metode v ekstremnem in objektno orientiranem programiranju).

V nadaljevanju se osredinjamo predvsem na metafore kot metode, saj je poimenovanje principov s pomočjo metafor samo jezikovni aspekt in ga ne moremo uporabiti za namen razvoja aplikacij.

#### 3.1 Sistemske metafore

Ena izmed dvanajstih temeljnih praks v ekstremnem programiranju je sistemska metafora. To je zgodba, s katero lahko vsak (stranke, programerji in vodstvo) opiše delovanje sistema (Beck, 2005).

Sistemska metafora je sredstvo za komunikacijo v projektu, ki jo razumejo naročniki in razvijalci in ki ne zahteva vnaprejšnje priprave ali strokovnega izobraževanja. S tem ko uveljavlja skupno besedišče (ki je poznano vsem udeležencem v projektu), olajša razpravo o sistemskih problemih in rešitvah. Sistemska metafora usmerja mentalne modele udeležencev projekta, s čimer pripomore h konsistentnemu poimenovanju elementov programa in izoblikovanju logične arhitekture sistema.

Najbolj poznana sistemska metafora je »plačilni sistem je montažna linija«. Ta metafora uporablja koncepte proizvodnje (zabojniki, linije, postaje in deli) za ponazoritev plačilnega sistema. Potek plačilnega sistema razloži tako: Plača osebe je kombinacija »delov«, pri čemer se deli nanašajo na delovne ure, iz česar se izračuna osnovna plača. Deli so vstavljeni v »vhodne zabojnike«, ki se pomikajo po »montažni liniji«, po kateri so prepeljani do »postaj«. Vse postaje si sledijo v določenem »zaporedju« in »obdelujejo« posamezne dele – od osnovne plače dodajajo ali odvezemajo zneske (davek, pokojnina, nadure, članarina v sindikatu itd). Po obdelavi na posamezni postaji je del vstavljen v »izhodni zabojnik«, ki ga linija odpelje do naslednje postaje. Končna plača je tako sestavljena iz vseh izhodnih zabojnikov, ki so nastali v procesih obdelav na liniji. Ta metafora je tipičen primer za poenostavitev razlage delovanja programa ter funkcij, ki jih mora vsebovati program (Khaled & Noble, 2005).

#### 3.2 Usmerjevalne metafore

Nekateri avtorji ponujajo metaforo za pomoč v objektno orientiranem programiranju. Usmerjevalna metafora, kot jo poimenuje Heinz Züllighoven (Züllighoven, 2005), služi za razlago problema, kako ustvariti določen sistem ter kakšna podpora je potrebna ob tem. Metafora pomaga pri osnovni orientaciji vseh udeležencev v vsem procesu razvoja, podpira oblikovanje, uporabo ter ovrednotenje programa, lahko pa pomaga tudi pri učenju uporabnikov.

Züllighoven poda štiri primere metafor, ki jih lahko uporabimo v programskem inženirstvu, in jih aplicira na elemente razvoja (tabela 1).

Tabela 1: **Uporaba usmerjevalnih metafor v programskem inženirstvu (Züllighoven, 2005)**

Usmerjalna metafora	Oblikovalski cilj	Vloga uporabnikov	Vloga razvijalcev
Objektni svetovi	Virtualni svetovi, naseljeni z aktivnimi objekti (agenti)	Sprožilci začetnih navodil	Oblikovalci minisvetov
Neposredna manipulacija	Manipulacijski objekti	Neodvisni delavci	Konstrukcija artefaktov
Tovarna	Vodenje in upravljanje z delom kot na montažni liniji	Upravljalci strojev	Izdelovalci strojev
Delovno okolje strokovnjakov	Podpora strokovnjakom z razvojem primerne opreme	Strokovnjaki	Oblikovalci orodij in okolja

#### 3.3 Metafore v programskih knjižnicah

Znanje o določenem programskem jeziku je odvisno od razumevanja in aplikacije standardnih komponentnih knjižnic. Alan F. Blackwell (Blackwell, 1995) predlaga, da bi spoznanja iz kognitivne psihologije

prenesli na komunikacijo med razvijalci jezika in programerji – specifično na knjižnice.

Blackwell je sistematsko preučeval metafore (kot temelje mentalnih modelov) v odnosu z javanskimi knjižnicami. Osredinil se je na besednjak, ki so ga

uporabljali avtorji javanskih dokumentov, s čimer je identificiral mogoče kandidate za metafore. Nekatere izmed metafor, ki jih je identificiral Blackwell, so »programi delujejo v preteklosti«, »programsko stanje je lahko kvantificirano«, »komponente so člani družbe«, »metode klicanja so dejanja govora«, »komponente imajo prepričanja in namene«, »programi operirajo v prostorskem svetu, ki ima notranje in zunanje okolje«.

Podobno je raziskoval Nikolas S. Boyd (Boyd, 2003), ki je preučeval metafore v programski kodi. Ugotovil je, da so metafore zmeraj prisotne, ne samo v programskih jezikih, ampak na splošno v besednjaku razvijalcev, kadar govorijo o sistemih.

## 4 METAFORE V UPORABNIŠKIH VMESNIKIH

Metafore imajo znotraj uporabniških vmesnikov nekoliko drugačne značilnosti kot v teoretičnem, jezikovnem smislu, vendar so osnovne značilnosti enake. Najbolj očitna vloga metafore v uporabniških vmesnikih je podobna kot pri splošni metafori: predstavljanje neke druge stvari; metafora je sredstvo za razlago ene stvari, s tem ko jo identificira z drugo.

Islovar (Koren, 2010) navaja definicijo metafore vmesnika tako: je »metafora, ki jo simulira vmesnik s prikazovanjem uporabniku že znanega z drugega področja in tako olajša delo«. Izvorna domena je običajno stvar iz realnega sveta, njene karakteristike pa opisujejo sistemsko funkcionalnost. Osrednja prednost metafor v uporabniških vmesnikih je, da uporabniku pomagajo ustvariti pričakovanja in spodbujajo predstave o obnašanju sistema.

Metafore v uporabniških vmesnikih lahko predstavimo s pomočjo različnih čutov in ne samo v jezikovnem smislu. Računalniške metafore so lahko slikovne (ikona mape) ali zvočne upodobitve (zvonjenje ob klicu), s pomočjo miške uporabljamo kinestetične metafore (vleci in spusti, angl. drag and drop), novejši uporabniški vmesniki in vnosne naprave, ki omogočajo upravljanje z večprstnim dotikanjem (angl. multi-touch), pa tudi izkoriščajo tip in geste (povečaj ali zavrti, angl. zoom in rotate) (Barr, 2003).

Predpostavka, da so metafore interpretirane, je z vidika uporabniških vmesnikov precej pomembna. Ko se namreč uporabniki srečajo z metaforo, si jo razložijo in interpretirajo znotraj svojega individualnega konteksta. Človeški miselni kontekst je pogojen z izkušnjami, s kulturo, z izobrazbo, s statusom ter z drugimi značilnostmi, ki izoblikujejo posameznika v

specifično osebo. Na to morajo biti pozorni načrtovalci uporabniških metafor, saj posamezne uporabniške vmesnike uporablja veliko število ljudi, kar lahko posledično pripelje do različnih ali napačnih interpretacij (Barr, 2003).

### 4.1 Tipi metafor v uporabniških vmesnikih

V uporabniških vmesnikih najdemo vse tri tipe metafor, kot sta jih definirala (Lakoff & Johnson, 2003): orientacijske, ontološke in strukturne.

#### 4.1.1 Orientacijske metafore

Uporabniški vmesniki obstajajo v dvojnopolnodimenzionalnih zaslonih (okna se prekrivajo), zato lahko razvijalci s pridom uporabljajo orientacijske metafore. Orientacijske metafore služijo v sistemu dvema namenoma: navigacija in kvantifikacija.

V programski opremi se uporabnik večkrat sreča z navigiranjem po vsebini. Velikokrat se pojavita gumba »naprej« in »nazaj« (angl. next in back), ki sta postavljena drug zraven drugega. Običajno imata vsak svojo manjšo puščico, ki vizualno nakaže smer. Gumba »naprej« in »nazaj« sta bila za potrebe dvo-dimenzionalnega prostora preoblikovana v »levo« in »desno« (slika 2).

Količinska opredelitev je v interakciji človeka z računalnikom pomemben koncept, zato je tudi vizualna ponazoritev količine, ki je predstavljena večinoma metaforično, pogost pojav. Eden izmed najpogostejših primerov količinske ponazoritve je drsnik (angl. slider). Vertikalni drsniki delujejo po principu »več je gor«, horizontalni drsniki pa so orientirani levo–desno, pri čemer desno pomeni več (slika 3). Razloge za takšno orientiranost lahko najdemo v preoblikovanju osnovne metafore »več je gor« za potrebe drugih dimenzij.

Orientiranost se navezuje na pozicijo posameznika na časovnem traku. Uporabniki si ob poteku določene operacije želimo informacije o tem, kje se trenutno nahajamo. Vmesniki zato s pomočjo vrstic napredka (angl. progress bar) ponudijo informacijo o časovnem poteku, ki deluje kot časovni trak, na katerem je označeno trenutno stanje.



Slika 2: Gumba Nazaj in Naprej v operacijskem sistemu Kubuntu



Slika 3: **Orsnik za prilagoditev glasnosti v programu iTunes**

#### 4.1.2 Ontološke metafore

Sistemske aplikacije so večinoma neoprijemljivi abstraktni koncepti, zato so lahko ontološke metafore v uporabniških vmesnikih zelo uporabne.

Eden izmed namenov ontoloških metafor je nanašanje na nekaj. S tem ko prepoznamo sistemske koncepte kot entitete in objekte, lahko uporabnikom bolj preprosto predstavimo njihov obstoj in značilnosti. Dokumenti imajo tako imena, nahajajo se na določeni lokaciji, definiramo jim lahko velikost, uporabniki jih lahko kopirajo, prilepijo in prestavijo v koš (slika 4).

Pomemben del uporabniških vmesnikov je obveščanje uporabnika o procesih, pri čemer je običajno vključena tudi pojasnitev vzroka programskih dogodkov. V računalniških okoljih dojemamo koncepte kot resnične fizične objekte in entitete, ki nam kot taki dobro služijo za obrazložitev vzrokov. Programi lahko tako vsebujejo napake, sistem pa nam lahko ponudi razlago vzrokov različnih operacij.

Metafore vsebnikov so v uporabniških vmesnikih prav tako precej pogoste. Za primer lahko podamo »vnosno polje« (angl. text-box), »v« katerega uporabnik vpiše besedilo, nato pa polje »shrani« vpisano besedilo.

Slika 4: **Prikaz lastnosti slike v operacijskem sistemu Ubuntu**

#### 4.1.3 Strukturne metafore

Kadar govorimo o metaforah v uporabniških vmesnikih, imamo običajno v mislih strukturne metafore. Strukturne metafore so znotraj uporabniških vmesnikov ustvarjene iz koncepta fizičnih stvari, pri čemer se del strukture realne stvari uporabi za lažje razumevanje abstraktnega programskega koncepta.

Primer računalniške metafore, ki je zlahka prepoznan objekt iz realnega sveta, je »koš za smeti« (angl. trashcan ali trash). Koš pooseblja strukturno metaforo, ko »datoteko izbrisemo tako, da jo vržemo v koš«.

Čeprav so strukturne metafore najbolj podobne realnim stvarim, pa vseh konceptov in atributov realne stvari vendarle ne moremo prenesti na abstraktni sistemski element. Računalniški koš tako nima vseh atributov fizičnega koša: ne more se npr. prevrniti (slika 5). Prav tako pa realni koncepti nimajo vseh lastnosti, ki jih imajo metafore v vmesnikih. V Appleovem operacijskem sistemu Mac OS lahko uporabnik zunanji pomnilnik izvrže tako, da ikono pomnilnika premakne v koš. Takšne ali podobne funkcije nima resničen koš.

Ker so strukturne metafore najbolj specifične pri predstavljanju novih konceptov, so lahko posledično tudi najbolj občutljive za kulturne razlike.

#### 4.2 Kritike uporabe metafor v uporabniških vmesnikih

Kritike metafor so se začele pojavljati že ob njihovi prvi uporabi v uporabniških vmesnikih. V nadaljevanju navajamo pet temeljnih kritik uporabe metafor v uporabniških vmesnikih, nato pa povzemamo še mnenja zagovornikov metafor.

1. Metafore uporabniških vmesnikov ne morejo nikoli prenesti celotnega pomena koncepta realnega sveta na sistemski koncept. To lahko povzroči nepravilno dožemanje funkcionalnosti, saj lahko uporabniki pričakujejo, da so abstraktni objekti enako zmogljivi kot realni. Metafore so torej zavajajoče (Saffer, 2005).
2. Ciljna domena lahko vsebuje več funkcionalnosti, kot jih ima izvorna domena. Posledično lahko uporabniki spregledajo nekatere značilnosti vmesnika, saj jih izbrana metafora morda ne sporoča (Carroll, 1984).
3. Metafore se ne morejo razširiti ali zmanjšati. Čeprav so lahko trenutne funkcije aplikacije smiselne znotraj določene metafore, se lahko zgodi, da morebitne nove funkcije ne bodo več (Neale & Carroll, 1997).

4. Po določenem času metafore zgubijo svojo moč in se spremenijo v idiom.
5. Metafore so v sisteme implementirane prepogosto. Čeprav morda olajšajo delo novim uporabnikom, pa lahko ovirajo napredne uporabnike (Neale & Carroll, 1997).

Zagovorniki metafor v uporabniških vmesnikih (Lakoff & Johnson, 2003; Reddy, 1993) pravijo, da so metafore del našega mentalnega sistema in bi zato brez njih težko izražali naše (abstraktne) misli. Računalniki so večinoma abstraktni sistemi, zato so metafore naravno orodje pri razlagi sistemskih značilnosti.

Če ne bi metafor v vmesnike implementirali že razvijalci, lahko predvidevamo, da bi si jih za razlago sistemskih značilnosti izmislili uporabniki sami. Vprašanje je, ali bi bile metafore uporabnikov enako primerne, kot so metafore razvijalcev. Uporabniki običajno namreč niso strokovnjaki na področju razvoja uporabniških vmesnikov (Cooper, 1995).

Alty (Alty, Knott, Anderson, & Smyth, 2000) pretvori kritiko, ki zavrača metafore zaradi neenakosti funkcij in lastnosti med izvorom in ciljem, v pripomoček pri ugotavljanju primernosti metafor. Avtor je na primeru definiral, koliko se lastnosti izvorne domene prenesejo na ciljno domeno ter koliko se lastnosti cilja prekrivajo z lastnostmi izvora. S tem je definiral tri kazalnike: 1) v kolikšnem obsegu bo metafora povzročala nepravilno sklepanje o sistemski funkcionalnosti, 2) koliko funkcionalnost programa sovпада z lastnostmi metafore ter 3) v kolikšnem obsegu se funkcionalnost programa izraža v metafori.

Pirhonen (Pirhonen, 2005) trdi, da idiomi niso problem. Mrtve metafore lahko namreč koristijo novim uporabnikom (ko jim ponudijo znanje o abstraktnih konceptih), napredni uporabniki pa jih lahko uporabljajo kot idiome (ki jih ne ovirajo eksplicitno).

Kljub kritikam metafore še zmeraj veljajo za uporabno orodje v interakciji človeka z računalnikom in so zaradi tega implementirane v večino vmesnikov.

## 5 IZBIRA IN VIZUALIZACIJA METAFOR

### 5.1 Napotki za izbiro primernih metafor

Veliko avtorjev ponuja napotke za izbiro primernih metafor v uporabniških vmesnikih. Kot je omenjeno v tretjem razdelku, se metafore ne nanašajo samo na vizualne objekte v uporabniških vmesnikih, ampak razkrivajo tudi funkcionalnosti in delovanje sistema.

Zato lahko napotke, ki jih povzemamo v nadaljevanju, uporabimo tudi v programskem inženirstvu.

Neale in Carrol (Neale & Carroll, 1997) priporočata tri korake za izbiro metafore, in sicer:

1. razmislek o mogočih kandidatih za metaforo (analiza obstoječih programov, opazovanje uporabnikov ali tvorjenje nove metafore),
2. opredelitev primernosti kandidatov za metafore (definiranje neujemanj med izvorno in ciljno domeno),
3. razrešitev neujemanj.

Erickson (Erickson, 1995) priporoča drugačen pristop, s katerim lahko razvijalci poiščejo primerno metaforo za vsak del sistema, v katerem se pojavi problem v funkcionalnosti. Za ugotavljanje primernosti ponudi definiranje petih atributov: količine strukture, aplikativnosti strukture, reprezentativnosti, uporabniške primernosti in razširljivosti.

Alty (Alty et al., 2000) predlaga šest korakov za načrtovanje uporabniških vmesnikov:

1. identifikacija sistemske funkcionalnosti,
2. tvorjenje in opisovanje potencialnih metafor,
3. analiza parov metafora–sistem,
4. implementacija,
5. vrednotenje,
6. pridobivanje povratnih informacij uporabnikov.

Za drugi korak (tvorjenje in opisovanje potencialnih metafor) predlaga avtor več tehnik: oblikovanje novih metafor, razširjanje obstoječih metafor, viharjenje možganov (angl. brainstorming), analizo trga in študijo delovnih mest.

Madsen (Madsen, 1994) ponudi za oblikovanje primernih metafor zbirko hevrističnih metod. Za generiranje metafor predlaga opazovanje uporabnikov, nadgradnjo obstoječih metafor, uporabo predhodnika artefakta ali iskanje dogodkov v realnem svetu, ki izražajo ključne aspekte funkcionalnosti. V procesu vrednotenja naj bi razvijalci poiskali metaforo z bogato strukturo (izogibati se je treba preveč splošnih in abstraktnih metafor), ovrednotili aplikativnost strukture (izbrana metafora mora ponujati relevantne rešitve za problem) ter izbrali tisto metaforo, ki ima široko poznan literarni pomen in je primerna za občinstvo. Metafora mora imeti konceptualno razdaljo med izvorno in ciljno domeno, vendar prav tako najmanj en povezovalni koncept. Med razvojem sistema predlaga avtor uporabo na metafori temelječega ključnega vidika, prestrukturiranje percepcije realnosti, tvorjenje predvidevanj, prepoznavanje neizkoriščenih delov

metafore, generiranje nasprotujočih si delov metafore in pripovedovanje zgodbe metafore.

## 5.1 Vizualizacija metafor

Metafore v programskem inženirstvu v veliki večini obstajajo samo kot jezikovni pripomočki za razlago konceptov, zato se ta razdelek nanaša predvsem na metafore v uporabniških vmesnikih.

Za uspeh metafore je pomembna njena sposobnost predstavitve objekta, ki prenaša informacijo do uporabnika. Pri načrtovanju metafore je treba pretehtati vse možnosti predstavitve določenega koncepta: grafični element, animacijo, zvok, interakcijo itd.

Pomembno je, da uporabnik metaforo prepozna čim hitreje, saj je to prvi korak pri spoznavanju funkcionalnosti programa, ki naj bi ga izražala metafora. Pri vizualizaciji metafor so pomembni predvsem trije dejavniki: realističnost, konsistentnost ter primernost (Alty et al., 2000).

### 5.1.1 Realističnost metafor

Uporabnik naj bi kar najhitreje prepoznal objekt, ki ga izraža metafora, zato bi bilo smiselno metaforo upodobiti čim bolj realistično. Tako bi naj bila uporabnikova obstoječa predstava o objektu iz realnega sveta zlahka prenesena na računalniški koncept. Vendar pa takšno sklepanje ni vedno pravilno. Če se metafora prikaže kot dobesedna stvar iz realnega sveta, morebitne nove funkcije računalniškega koncepta niso takoj vidne. Uporabnik lahko dobi občutek, da sistem deluje v nasprotju z intuicijo.

Primer neprimerne realističnosti metafore se je pojavil v sistemu ROME, v katerem je bila za funkcionalnost shranjevanja dokumentov izbrana metafora kovček. Uporabniki so lahko v kovčku hranili svoje dokumente ali jih javno delili z drugimi uporabniki v sobi. Kovček je bil v sobi (ob prisotnosti drugih uporabnikov) odprt, zaradi česar so uporabniki dvomili v zasebnost svojih dokumentov. Odprt kovček je nakazoval, da so datoteke v njem vidne vsem prisotnim, kar pa se ni skladalo s funkcionalnostjo sistema.

Pri izbiri stopnje realističnosti vizualne metafore je zato treba upoštevati prekrivanje izvirne in ciljne domene ter se zavedati, da bolj realističen videz metafore ni vedno najboljša izbira.

### 5.1.2 Konsistentnost metafor

Poznamo dva tipa konsistentnosti metafor: znotraj metafor in med metaforami.

Konsistentnost znotraj metafore se nanaša na konsistentnost posamezne metafore. Enake metafore v različnih vmesnikih naj vedno težijo k podobnosti, saj imajo uporabniki že izdelano predstavo o delovanju specifične metafore, ki so jo pridobili iz izkušenj delovanja v drugih vmesnikih.

Znotraj posameznega vmesnika je običajno aktivnih več metafor hkrati. Večina vmesnikov, ki upravlja z besedilom, podpira uporabo metafor »izreži« (angl. cut), »kopiraj« (angl. copy) in »prilepi« (angl. past). Ko uporabnik kopira besedilo iz enega dokumenta, pričakuje, da bo drugi vmesnik podpiral funkcijo »prilepi« in da bo tako lahko prenesel besedilo v drugi dokument. Konsistentnost je treba vzdrževati med različnimi metaforami in različnimi vmesniki (Alty et al., 2000).

### 5.1.3 Primernost metafor

Čeprav izbrana metafora pravilno izraža funkcionalnost programa, je lahko neprimerna. Neprimerna metafora vsebuje sporno konotacijo, kar lahko povzroči neustreznost. Najbolj očitne so rasne, spolne ali starostne konotacije, obstajajo pa še druge.

Metafora, ki je stilistično podobna otroški risanki, se lahko na prvi pogled zdi primerna, vendar meče slabo luč na razvijalsko podjetje in na uporabnike programa.

Pri izbiri metafore je treba upoštevati tudi dejstvo, da metafora, ki je primerna danes, v prihodnosti mogoče ne bo več (Alty et al., 2000).

## 6 PREPOZNAVANJE FUNKCIONALNOSTI SISTEMA S POMOČJO METAFOR

### 6.1 Opredelitev problema

V članku smo večkrat zapisali, da metafore vsebujejo informacije o sistemskih funkcionalnostih in so zaradi tega predvidoma koristen pripomoček v interakciji človeka z računalnikom. Proti metaforam je bilo podanih tudi nekaj kritik (povzeli smo jih v razdelku 4.2), zato smo s svojo raziskavo želeli ugotoviti, ali metafore dejansko pomagajo pri prepoznavanju sistemskih funkcionalnosti.

### 6.2 Cilji in namen raziskave

Z raziskavo smo skušali ugotoviti, kako dobro uporabniki prepoznavajo funkcionalnost sistema s pomočjo metafor v uporabniških vmesnikih. Zanimalo nas je, ali so metafore same po sebi dovolj dober pripomoček za prepoznavanje funkcionalnosti sistema

oz. ali je za prepoznavanje potrebna dodatna pomoč. Ugotoviti smo tudi želeli, ali uporabniki prepoznajo metafore, ki se v uporabniških vmesnikih ne pojavljajo pogosto oz. s pomočjo katerih metafor uporabniki največkrat prepoznajo funkcionalnost sistema.

### 6.3 Potek raziskave

Za namen raziskave smo izvedli anketo o prepoznavanju sistemskih funkcionalnosti s pomočjo metafor. Uporabili smo vprašalnik s slikami 21 metafor iz različnih aplikacij in operacijskih sistemov. Metafo-

re, izbrane za anketo (slika 5), predstavljajo različne stopnje pogostosti pojavljanja v sistemih.

V prvi fazi so anketiranci navedli funkcionalnost sistema, ki jo po njihovem mnenju izraža določena metafora. V drugi fazi so opredelili, koliko se strinjajo s podanimi odgovori o funkcionalnosti sistema, ki jo izraža določena metafora (po Likertovi lestvici od 1 – se ne strinjam do 5 – zelo se strinjam). Na koncu so sledila še splošna vprašanja o izkušnjah anketirancev v različnih operacijskih sistemih in programih ter z razvojem informacijskih rešitev.



Slika 5: **Metafore, prikazane v anketnem vprašalniku (\* = metafora se pojavi v spletnem brskalniku)**

Anketirali smo 49 uporabnikov, od katerih dva nista v celoti izpolnila ankete, zato smo izključili njune odgovore. Anketiranci so bili stari 19 do 24 let in so predstavniki generacije, ki je dobro seznanjena z različnimi uporabniškimi vmesniki. Od 47 anketirancev se jih 28 ukvarja z razvojem programske opreme (v povprečju 2,7 leta).

Dve metafori (drsnik in zameglitev slike) smo anketirancem najprej pokazali brez sistemske pomoči, nato pa z njo. Metafora drsnik je v uporabniškem vmesniku Microsoft Expression Design upodobljena tako, da na prvi pogled izgleda kot prikazovalnik količine. Šele ko se uporabnik z miško premakne nad polje, se kurzor spremeni in uporabnik ugotovi, da je količina nastavljiva s premikom miške. Pri metafori zameglitev slike je v drugem primeru prikazan namig, ki se pojavi ob postavitvi kurzorja na orodje.

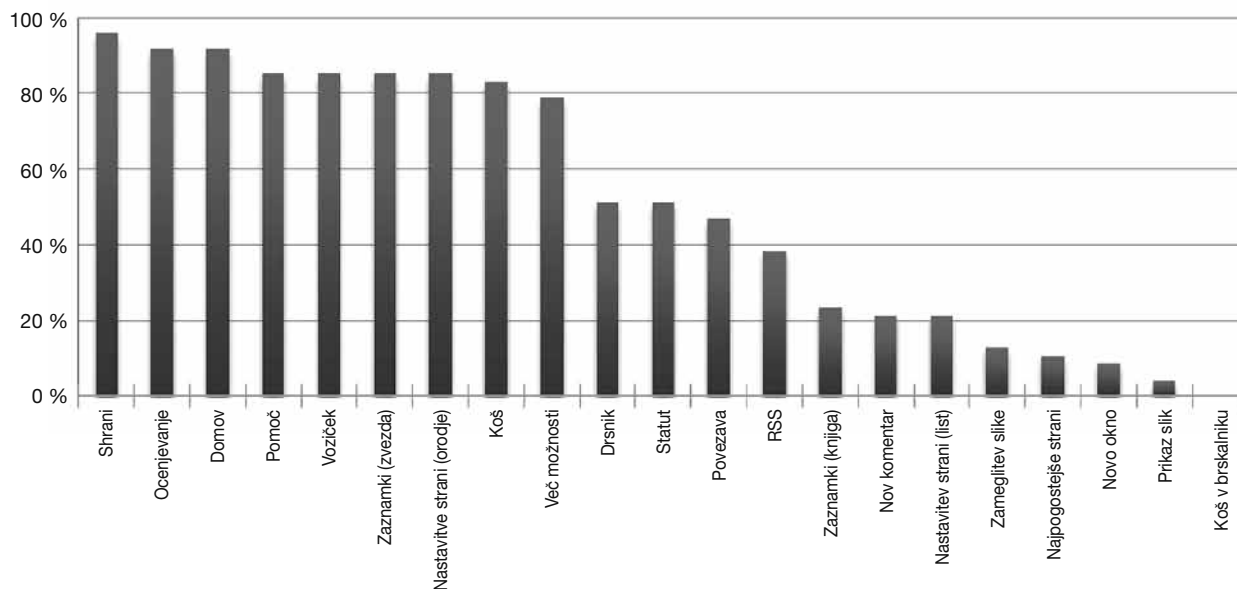
Metafore anketirancem niso bile prikazane znotraj njihovega običajnega okolja, zato pomoč, ki jo

običajno ponujajo sistemi, ni bila zajeta v raziskavi (razen v drugem prikazu drsnika in orodja za zameglitev).

Odgovorov, v katerih je bilo navedeno ime metafore (npr. koš), nismo šteli za pravilne, saj niso navajali funkcionalnosti (npr. brisanje datotek).

## 6.4 Ugotovitve

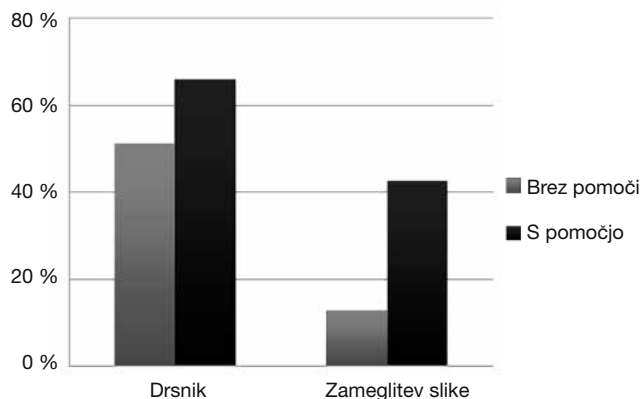
Z anketo smo ugotovili, da je v povprečju 53 odstotkov anketirancev prepoznalo funkcionalnost sistema s pomočjo metafor. Kot lahko opazimo na sliki 6, je večina anketirancev prepoznala funkcionalnost s pomočjo metafor, ki se v uporabniških vmesnikih pojavljajo pogosto (shrani, ocenjevanje, domov, pomoč, voziček itd.). Nizko prepoznavnost lahko opazimo pri metaforah, ki se pojavljajo redko in v specifičnih programih (koš v brskalniku, prikaz slik v brskalniku, novo okno, najpogostejše strani ter zameglitev slike).



Slika 6: **Uspešnost anketirancev pri prepoznavanju funkcionalnosti sistema s pomočjo metafor**

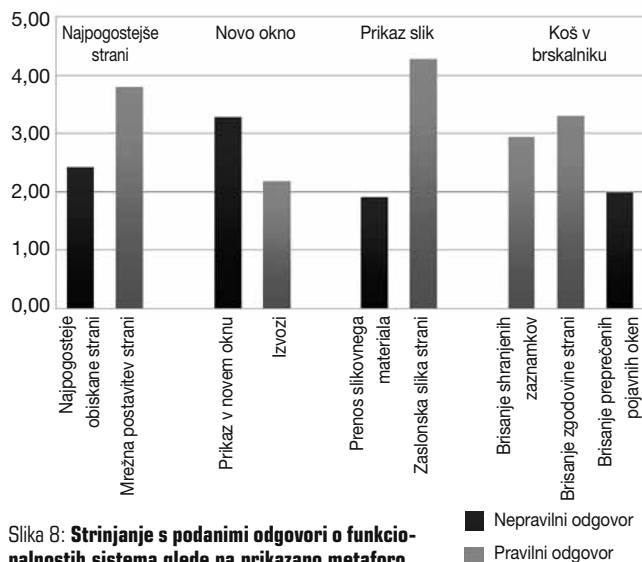
S pomočjo ankete smo ugotovili, da sistemska pomoč izboljša prepoznavanje funkcionalnosti s pomočjo metafor (slika 7). Uporabniki so bili pri pre-

poznavanju drsnika in orodja za zameglitev slike uspešnejši, kadar je bila prikazana vizualna pomoč.



Slika 7: **Uspešnost anketirancev pri prepoznavanju funkcionalnosti sistema s pomočjo metafor z in brez sistemske pomoči**

Štiri metafore, ki so se v prvi fazi najslabše odrezale pri prepoznavanju, so bile slabo prepoznane tudi v drugi fazi, ko so anketiranci navajali stopnjo strinjanja s podanimi funkcionalnostmi (slika 8). Anketiranci so se v večji meri strinjali z napačnimi odgovori predvsem pri metaforah najpogostejše strani (ki jih je spominjala na mrežno postavitve strani) in prenos slik (pri kateri so pomislili na zajemanje zaslonске slike) ter pri metafori koš v brskalniku (za katerega so menili, da izraža brisanje shranjenih zaznamkov ali brisanje zgodovine strani).



Slika 8: **Strinjanje s podanimi odgovori o funkcionalnostih sistema glede na prikazano metaforo**

V tabeli 2 so prikazani rezultati prepoznavanja dodatne funkcionalnosti koša v operacijskem sistemu Mac OS. Samo štirje anketiranci so se strinjali oz. zelo strinjali s funkcionalnostjo izvrzi zunanji pomnilnik, četudi je enajst anketirancev v preteklosti že uporabljalo Mac OS, trije pa ga uporabljajo tudi sedaj. Prav tako ni zaznati povezave med uporabniki, ki so že uporabljali Mac OS, in strinjanjem s prepoznavanjem funkcionalnosti.

Tabela 2: **Prepoznavanje dodatne funkcionalnosti koša v operacijskem sistemu Mac OS**

Strinjanje s funkcionalnostjo: izvrzi zunanji pomnilnik	Uporabniki, ki so že uporabljali Mac OS	Uporabniki, ki še niso uporabljali Mac OS	Uporabniki, ki uporabljajo Mac OS	Uporabniki, ki ne uporabljajo Mac OS
Se ne strinjam – 1	9	30	2	37
Delno se ne strinjam – 2	1	2	1	2
Sem neopredeljen – 3	1	0	0	1
Se strinjam – 4	0	2	0	2
Zelo se strinjam – 5	0	2	0	2

### 6.5 Povzetek raziskave

Anketiranci so večinoma zlahka prepoznali metafore, ki se v vmesnikih pojavljajo pogosto, čeprav jim je bila onemogočena običajna sistemska pomoč. Sklepamo lahko, da poznane metafore uporabnikom olajšajo delo z uporabniškimi vmesniki, saj uporabniki zlahka prepoznajo funkcionalnosti.

Uporabniki so prepoznali manj funkcionalnosti s pomočjo metafor, ki se pojavljajo v specifičnih

uporabniških vmesnikih (Adobe Photoshop), imajo neobičajne oblike (drsnik v orodju Microsoft Expression Design), neobičajne izvorne pomene (knjiga za zaznamke in list za nastavitve brskalnika) ali neobičajne pomene (izvrzi zunanji pomnilnik s premikom ikone na koš v operacijskem sistemu Mac OS).

Ugotovili smo, da nekatere metafore izražajo drugačne funkcionalnosti, kot naj bi jih (najpogostejše strani, prikaz slik ter koš v brskalniku). S tem ko

uporabnike spominjajo na napačno funkcionalnost, jih zavajajo in tako otežujejo interakcijo z računalniki. Zato je nadvse pomembno, da razvijalci uporabniških vmesnikov preudarjeno uporabljajo metafore ter stremijo k izbiri najprimernejše.

Kadar je bila uporabnikom ob novih oz. neobičajnih metaforah prikazana pomoč, je bilo prepoznavanje funkcionalnosti višje. Tako se lahko – predvsem ob vpeljavi nove ali drugačne metafore – razvijalci zanašajo na sistemsko pomoč, ki uporabnikom olajša prepoznavanje.

Iz raziskave lahko povzamemo, da pogoste in poznane metafore uporabnikom olajšajo delo, saj lahko brez systemske pomoči prepoznajo funkcionalnosti. Kadar se uporabniki srečajo z novo metaforo, jih ta ne sme zavajati. Zato morajo razvijalci sistemov dobro premisliti, kaj izraža izbrana metafora in ali se sklada s funkcionalnostjo sistema.

## 7 SKLEP

Čprav se neprestano srečujemo z metaforami, se jih velikokrat ne zavedamo, zato jih v razvoju uporabniških vmesnikov ne bi smeli ignorirati. Nekatere teorije poudarjajo njihov doprinos k interakciji človeka z računalnikom, kar smo delno potrdili tudi z raziskavo. Pogoste metafore namreč omogočajo prepoznavanje funkcionalnosti, kadar systemska pomoč ni na voljo. Metafore, ki so konsistentno uporabljene med različnimi vmesniki, lahko pripomorejo pri interakciji uporabnikov s sistemi.

Prepoznavanje novih metafor je prav tako mogoče, vendar morajo biti te skrbno izbrane. Samo če dobro komunicirajo systemsko funkcionalnost, so lahko v pomoč uporabnikom.

Običajno se ob metaforah nahajajo poimenovanja funkcionalnosti, večina aplikacij pa vsebuje tudi pomoč, s katero se lahko uporabnik kadar koli posvetuje. Dokazali smo, da je pomoč primerna še posebno pri neznanih ali preoblikovanih metaforah, saj olajša prepoznavanje. Ob redni uporabi pa dobi metafora polno moč in takrat jo lahko uporabniki s pridom izkoriščajo.

Nadaljnje raziskave na področju metafor bi se med drugim morale osrediniti na vpeljavo novih metafor – od ideje in vizualne podobe do same implementacije – ter raziskati, koliko in kdaj naj bodo metafore realistične, kaj naj izražajo ter kako testirati njihovo primernost.

## 8 VIRI, LITERATURA

- [1] Alty, J. L., Knott, R. P., Anderson, B. & Smyth, M. (2000). A framework for engineering metaphor at the user interface. *Interacting with Computers*, 13(2), 301–322. Elsevier.
- [2] Barr, P. (2003). *User-Interface Metaphors in Theory and Practice*. Wellington.
- [3] Beck, K. (2005). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley Professional.
- [4] Blackwell, A. F. (1995). Metaphors we Program By: Space, Action and Society in Java. In R. M. Baecker (Ed.), *Proceedings of 18th PPIG, Brighton*. Brighton.
- [5] Blackwell, A. F. (2006). The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction*, 13(4), 490–530.
- [6] Boyd, N. S. (2004). *Software Metaphors*. Dostopno na: <http://www.educery.com/papers/rhetoric/metaphors>.
- [7] Carroll, J. M., Mack, R. L. (1984). Learning to use a word processor: by doing, by thinking, and by knowing. In Thomas J., Schneider M. L. (Eds.), *Human factors in computer systems*. 13–51. Norwood: Ablex Publishing Corp.
- [8] Cooper, A. (1995). The Myth of Metaphor. *Originally Published in Visual Basic Programmer's Journal*. Dostopno na: <http://web.cs.wpi.edu/~dcb/courses/CS3041/Myth-of-Metaphor.pdf>.
- [9] Erickson, T. D. (1995). Working with interface metaphors. In R. M. Baecker, J. Grudin, W. A. S. Buxton & S. Greenberg (Eds.), *The art of Human-Computer Interface Design*. 147–151. San Francisco: Morgan Kaufmann Publishers.
- [10] Honderich, T. (1995). *The Oxford Companion to Philosophy*. Oxford: Oxford University Press.
- [11] Honeycutt, L. (2004). *Aristotle's Rhetoric*. Dostopno na: <http://www.public.iastate.edu/~honeyl/Rhetoric/rhet3-10.html>.
- [12] Khaled, R. & Noble, J. (2005). Extreme programming system metaphor: A semiotic approach. In Billinghamurst M (Eds.), *Proceedings of the 7th International Workshop on Organisational Semiotics*. 109–117. Newcastle.
- [13] Koren, S. (2011). *Slovar informatike*. Dostopno na: [http://www.islovar.org/iskanje\\_enostavno.asp](http://www.islovar.org/iskanje_enostavno.asp).
- [14] Lakoff, G. & Johnson, M. (2003). *Metaphors We Live By*. University Of Chicago Press.
- [15] Madsen, K. H. (1994). A guide to metaphorical design. *Communications of the ACM*, 37(12), 57–62.
- [16] Neale, D. C. & Carroll, J. M. (1997). The Role of Metaphors in User Interface Design. In Helander Marting G. (Eds.), *Handbook of Human-Computer Interaction*. 441–462. New York: Elsevier.
- [17] Pirhonen, A. (2005). To simulate or to stimulate? In search of the power of metaphor in design. In Pirhonen A et al (Eds.), *Future Interaction Design*. 105–124. New York: Springer.
- [18] Reddy, M. (1993). The conduit metaphor: A case of frame conflict in our language about language. In A. Ortony (Ed.), *Metaphor and Thought*. 164–201. Cambridge: Cambridge University Press.
- [19] Saffer, D. (2005). *Saffer, D. "The Role of Metaphor in Interaction Design."* Thesis. Carnegie Mellon University.
- [20] Züllighoven, H. (2005). *Object-oriented construction handbook: developing application-oriented software with the tools & materials approach*. Heidelberg: Morgan Kaufman.

■

Saša Kuhar je asistentka za področje informatike na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, kjer je leta 2009 diplomirala s področja metafor. Trenutno je študentka bolonjskega doktorskega študija Medijske komunikacije, v okviru katerega se ukvarja s temi raziskovalnimi področji: vizualizacija podatkov, pregledne plošče, uporabniški vmesniki, interakcija človek–računalnik. Aktivno se udeležuje domačih in tujih konferenc s področja informatike.

■

Marjan Heričko je redni profesor za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, kjer je nosilec več predmetov, ki so v pristojnosti Inštituta za informatiko. Je namestnik predstojnika Inštituta ter vodja Laboratorija za informacijske sisteme. Doktoriral je leta 1998 na Univerzi v Mariboru na področju zagotavljanja kakovosti objektno orientiranega razvoja programske opreme. Njegovo raziskovalno delo zajema vsa področja razvoja sodobnih informacijskih rešitev in storitev s poudarkom na naprednih pristopih k modeliranju in načrtovanju informacijskih sistemov, načrtovalskih vzorcih in metrikah. V zadnjem obdobju je aktiven tudi na področju storitvene znanosti in storitvenega inženirstva kot tudi uporabniško usmerjenega razvoja.



## Sklepi konference Informatika v javni upravi 2011

V sklepih konference IJU 2010 so bila predstavljena pričakovanja udeležencev glede nadaljnega razvoja informatike v javni upravi. Sklepi so bili poslani resornima ministroma za javno upravo ter za visoko šolstvo, znanost in tehnologijo, posredovani so bili medijem in objavljeni na spletni strani dogodka IJU 2010.

Udeleženci konference IJU 2011 z zadovoljstvom ugotavljamo, da so bila od sprejetja lanskih sklepov IJU 2010 izpolnjena pričakovanja, ki jih navajamo v nadaljevanju.

Projekti centralnih in horizontalnih funkcij postajajo vodilo in prioriteta izvajanja akcijskega načrta e-poslovanja javne uprave do leta 2015 in pomenijo temelj in pogoj za nove sektorske projekte. Vzpostavitev nekaterih novih horizontalnih funkcij je bila izvedena v okviru sektorskih projektov (e-Sociala, e-Pravosodje idr.).

Upoštevanje smernic nacionalnega interoperabilnostnega okvira, nacionalne prostorske infrastrukture in drugih sektorjev vzpostavlja možnosti za učinkovitejšo uporabo horizontalnih funkcij v sektorskih – vsebinskih projektih. To pa pomeni tudi podlago za doseganje čezmejne interoperabilnosti in s tem izpolnjevanje zahtev delovanja notranjega trga Evropske unije.

Kljub vzpostavljenim formalnim oblikam sodelovanja se počasi krepi sodelovanje tako na področju horizontalnih funkcij kot za potrebe vzpostavitve posameznih vsebinskih projektov na področju informatizacije postopkov in odprave administrativnih ovir.

Poleg nekaterih uresničenih pričakovanj so pred nami ostala še mnoga odprta vprašanja, povezana z vlogo in pomenom informatike pri reformi javne uprave.

Če je bila rdeča nit prve konference kako povezati informacijske otoke, in se je ta v drugi konferenci izrazila v zahtevo, naj postane gonilo razvoja javne uprave, smo sedaj soočeni s konkretnimi vprašanji, ki zadevajo položaj informatike v luči reforme javne uprave. Pri tem še posebej izpostavljamo umeščenost pristojnosti razvoja elektronskega poslovanja na področju lokalne samouprave.

Čeprav presega okvir in pristojnosti informatike, je skoraj nujno omeniti, da bi informatika kot stroka lahko odigrala pomembno vlogo pri sistematični funkcionalni analizi vseh resorjev za odgovore na vprašanja, katere funkcije, pristojnosti in službe se podvajajo, katere so odveč in katere bi morda lahko prenesli na zunanje izvajalce.

### Sporočilo in sklepi konference Informatika v javni upravi 2011

1. Potrebna je celovita analiza na področju informatizacije storitev javne uprave (nalog, projektov, kadrov, financ) in nato načrtovanje institucije informatike na državni ravni.
2. Programu odprave administrativnih ovir za 25 odstotkov, ki se nahaja v fazi implementacije sprememb posamezne zakonodaje, dati vso podporo na najvišji ravni izvršne oblasti. Vzajemna prenova procesov in informatizacija sta ključna za racionalno in sodobno javno upravo.
3. Pri projektih informatizacije je treba natančno presoditi njihove učinke in slediti potrebam in pričakovanjem njihovih uporabnikov.
4. Dosežene rezultate je treba ustrezno promovirati.
5. Narediti fleksibilen sistem zaposlovanja, da bo omogočal pridobitve novih, predvsem mladih strokovnih kadrov. Za uspešnejši razvoj infor-

matike in s tem javne uprave bi morali vzpostaviti stimulativen način nagrajevanja.

6. Zagotoviti je treba učinkovitejši prenos idej in dobrih praks med javnim, zasebnim sektorjem in univerzami ter s tem omogočiti pospešen vstop novih tehnoloških trendov in sodobnih pristopov pri realizaciji novih projektov.
7. Pripraviti je treba modele za uspešno izvedbo projektov javno-zasebnega partnerstva na področju informacijskih in komunikacijskih tehnologij.

8. Slovenija naj še intenzivneje sodeluje pri izmenjavi dobrih praks in pri mednarodnih projektih z drugimi najbolj razvitimi državami na tem področju. Enako naj bo še intenzivnejše tudi sodelovanje gospodarstva, akademske sfere in javne uprave ter še večja promocija slovenskega znanja in dobrih praks v drugih državah.

*Brdo pri Kranju, 29. novembra 2011*

*Bralcem in sodelavcem revije  
Uporabna informatika  
želimo uspešno in ustvarjalno  
novo leto 2012*

*Uredništvo*

## Iz Islovarja

Islovar je spletni terminološki slovar informatike, ki ga ureja jezikovna sekcija Slovenskega društva Informatika na naslovu <http://www.islovar.org>. V tej številki revije objavljamo nekaj izrazov s področja zgradbe informacijskega sistema. Izraze lahko komentirate, tako da se prijavite v poglavju Nov uporabnik, poiščete izraz, ki ga želite komentirati, in zapišete svoj komentar.

**aktivnost** -i ž (*angl. activity*)

skupina logično povezanih in v celoto zaokroženih zaporednih opravil

**arhitektúra** -e ž (*angl. architecture*)

struktura in način povezovanja sestavnih delov sistema; *sin.* zgradba; *prim.* ciljna arhitektura

**arhitektúra posrednikov zahtév skúpnih**

**objéktov** -e -- -- -- -- ž (*angl. common object request broker architecture, krat. CORBA*)  
način povezovanja objektov, izvedenih v različnih računalniških okoljih

**cíljna arhitektúra** -e -e ž (*angl. target architecture*)

struktura in način povezovanja sestavnih delov, predvidena za dokončan sistem, napravo; *prim.* arhitektura

**CORBA CORBE** [kórba] ž *krat. (angl. common object request broker architecture)*

*gl.* arhitektura posrednikov zahtev skupnih objektov

**fízična arhitektúra** -e -e ž (*angl. physical architecture*)

struktura in način povezovanja programov, programskih komponent, podatkovnih baz v informacijskem sistemu ali programski rešitvi; *prim.* logična arhitektura

**funkcionalna arhitektúra** -e -e ž (*angl. functional architecture*)

hierarhična členitev funkcionalnosti informacijskega sistema

**informacijska arhitektúra** -e -e ž (*angl. information architecture*)

struktura in način povezovanja podatkov, postopkov, ljudi, strojne in programske opreme v informacijskem sistemu

**ISA ISE** [ísa] ž *krat. (angl. Industry Standard Architecture)*

specifikacija vodila, ki se je uporabljalo v osebnih računalnikih

**komponênta** -e ž (*angl. component*)

*gl.* programska komponenta

**lógična arhitektúra** -e -e ž (*angl. logical architecture*)

struktura in način povezovanja funkcij in aktivnosti informacijskega sistema ali programske rešitve; *prim.* fizična arhitektura

**OMA OME** [óma] ž *krat. (angl. object management architecture, krat. OMA)*

referenčni model za razvoj objektno usmerjenih porazdeljenih uporabniških programov

**online storítev** -- -tve [ónlajn] ž (*angl. online service*)

informacijska storitev, ki deluje na priključni način; *sin.* priključna storitev

**ponôvna uporabnost** -e -i ž (*angl. reusability*)

lastnost podprograma, programske komponente, da se lahko uporabi kot del programskih rešitev, za katere prvotno ni bil namenjen

**priključna storitev** -e -tve ž (*angl.* online service)  
informacijska storitev, ki deluje na priključni način; *sin.* online storitev

**programska komponenta** -e -e ž (*angl.* software component)  
samostojen del računalniškega programa, programske rešitve, ki opravlja funkcijo (1), je ponovno uporaben in navadno nameščen na aplikacijskem strežniku; *sin.* komponenta; *prim.* modul, podprogram

**programska rešitev** -e -tve ž (*angl.* software solution)  
računalniški program ali več računalniških programov, prirejen zahtevam posameznega uporabnika in usklajen s tehnološko arhitekturo; *prim.* aplikacija

**SOA SOE** [sóa] ž *krat.* (*angl.* service oriented architecture)  
*gl.* storitveno usmerjena arhitektura

**splétna storitev** -e -tve ž (*angl.* web service)  
storitev (1), ki je dostopna na spletu

**storitev** -tve ž (*angl.* service)  
1. naloga, ki jo izvede naprava, programska komponenta na zahtevo druge naprave, programske komponente, uporabnika  
2. delo, ki se opravi za naročnika

**storitveno usmérjena arhitektúra** -- -e -e ž  
(*angl.* service-oriented architecture, *krat.* SOA)  
struktura in način povezovanja šibko sklopljenih programskih komponent, programov v informacijskem sistemu ali programski rešitvi za večjo povezljivost in lažje vzdrževanje

**strôjna arhitektúra** -e -e ž (*angl.* hardware architecture)  
struktura in način povezovanja delov računalniškega sistema

**šíbka sklópljenost** -e -i ž (*angl.* loose coupling)  
značilnost sistema, katerega elementi niso odvisni od notranje strukture in izvedbe drugih elementov v sistemu

**tehnolóška arhitektúra** -e -e ž (*angl.* technical architecture)  
struktura in način povezovanja računalnikov, računalniških omrežij, vmesne programske opreme, ki omogoča praktično izvedbo informacijske arhitekture

**vmésna programška opréma** -e -e -e ž  
(*angl.* middleware)  
programska oprema, ki povezuje posamične programe, programske komponente

*Izbor pripravlja in ureja  
Katarina Puc s sodelavci Islovarja.*

## Pomembni spletni naslovi

- IFIP News: <http://www.ifip.org/images/stories/ifip/public/Newsletter/news> ali [www.ifip.org](http://www.ifip.org) → Newsletter
- IT Star Newsletter: [www.itstar.eu](http://www.itstar.eu)
- ECDL: [www.ecdl.com](http://www.ecdl.com)
- CEPIS: [www.cepis.com](http://www.cepis.com)

## Dostop do dveh tujih strokovnih revij

- Revija **Upgrade** (CEPIS) v angleščini (ISSN 1684-5285) je dostopna na spletnem naslovu: <http://www.upgrade-cepis.org/issues/2008/4/upgrade-vol-IX-4.html>.
- Revija **Novática** (CEPIS) v španščini (ISSN 0211-2124) je dostopna na spletnem naslovu: <http://www.ati.es/novatica/>.

# Pristopna izjava

## za članstvo v Slovenskem društvu INFORMATIKA

### Pravne osebe izpolnijo samo drugi del razpredelnice

Ime in priimek	
Datum rojstva	
Stopnja izobrazbe	srednja, višja, visoka
Naziv	prof., doc., spec., mag., dr.
<b>Domači naslov</b>	
Poštna št. in kraj	
Ulica in hišna številka	
Telefon (stacionarni/mobilni)	

### Zaposlitev člana oz. člana - pravna oseba

Podjetje, organizacija	
Kontaktna oseba	
Davčna številka	
Poštna št. in kraj	
Ulica in hišna številka**	
Telefon	
Faks	
E-pošta	

### Zanimajo me naslednja področja/sekcije\*

- jezik
- informacijski sistemi
- operacijske raziskave
- seniorji
- zgodovina informatike
- poslovna informatika
- poslovne storitve
- informacijske storitve
- komunikacije in omrežja
- softver
- hardver
- upravná informatika
- geoinformatika
- izobraževanje

podpis

kraj, datum

Pošto društva želim prejemati na domači naslov / v službo.

Članarina znaša: 18,00 € - redna

7,20 € - za dodiplomske študente in seniorje (ob predložitvi dokazila o statusu)

120,00 € - za pravne osebe

Članarino, ki vključuje glasilo društva – revijo **Uporabna informatika**, bom poravnal sam / jo bo poravnal delodajalec.

DDV je vključen v članarino.



## Naročilnica na revijo UPORABNA INFORMATIKA

Naročnina znaša: 35,00 € za fizične osebe

85,00 € za pravne osebe – prvi izvod

60,00 € za pravne osebe – vsak naslednji izvod

15,00 € za študente in seniorje (ob predložitvi dokazila o statusu)

DDV je vključen v naročnino.

ime in priimek ali naziv pravne osebe in ime kontaktne osebe

davčna številka, transakcijski račun

naslov plačnika

naslov, na katerega želite prejemati revijo (če je drugačen od naslova plačnika)

telefon/telefaks

elektronska pošta

Podpis

Datum

# Izpitni centri ECDL



## ➤ **Znanstveni prispevki**

Matej Rebec, Gene Bavec

**Nekateri varnostni vidiki informacijskih tehnologij na področju turizma v Sloveniji**

Boris Ovčjak, Gregor Jošt, Gregor Polančič, Marjan Heričko

**Primerjava učinkovitosti delovanja aplikacij v oblaku in v lokalnem okolju**

Ivana Dvorski Lacković

**Human Resource Information Systems in Croatian Banks: current practice and trends**

## ➤ **Pregledni znanstveni prispevki**

Saša Kuhar, Marjan Heričko

**Uporabnost metafor v informatiki**

## ➤ **Informacije**

**Sklepi konference Informatika v javni upravi 2011**

**Iz Islovarja**

ISSN 1318-1882



9 771318 188001