

U P O R A B N A

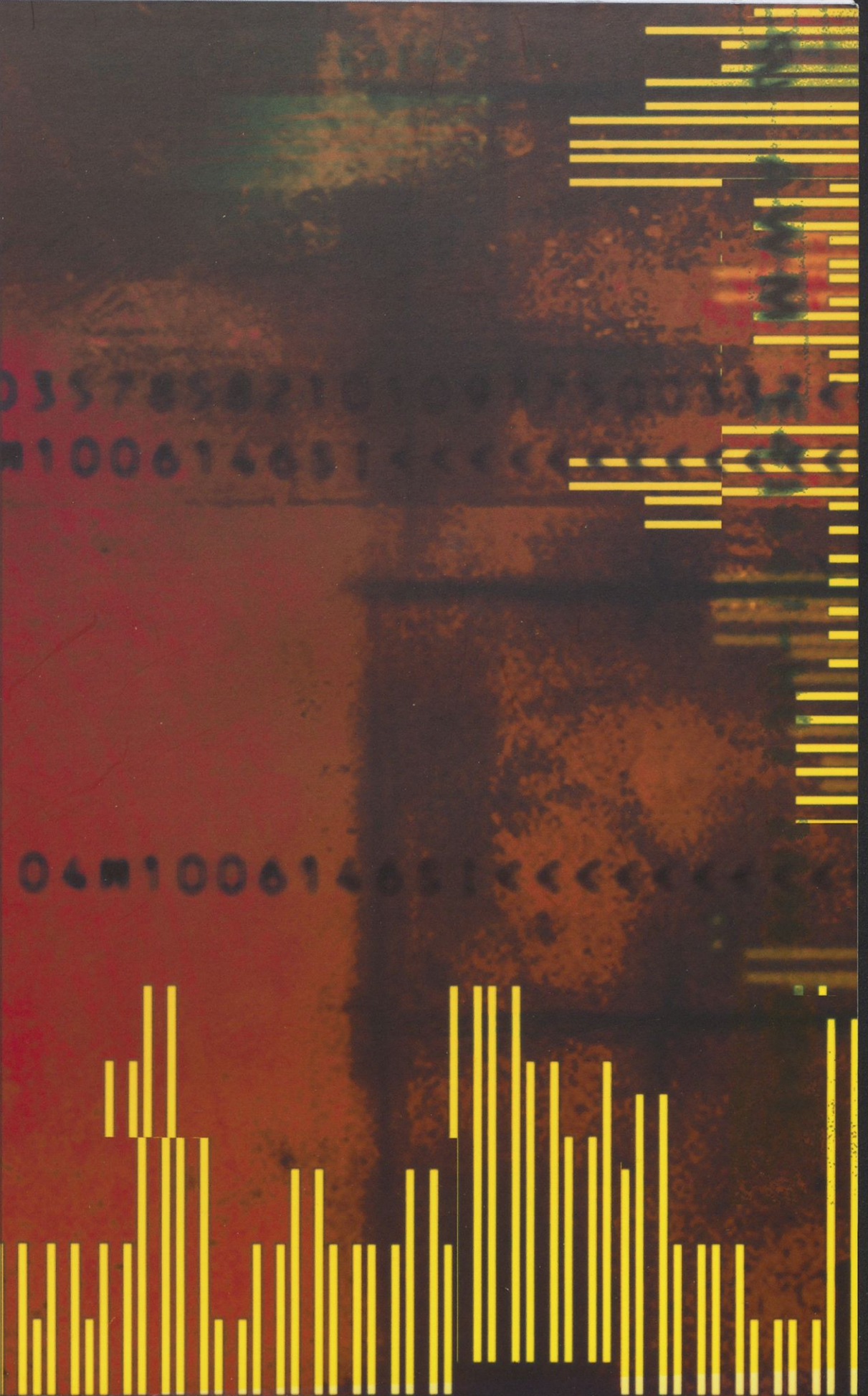
I N F O R M A T I K A

2003

ŠTEVILKA 3

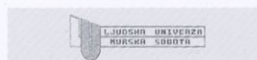
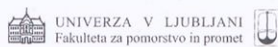
JUL/AVG/SEP

LETNIK XI



Testni 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 Societies) 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. Posebej pomembno je, da velja spričevalo v več kot osemdesetih državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot tri milijone indeksov, v Sloveniji okoli 1700 in podeljenih okoli tisoč spričeval. Za testne centre ECDL so se v Sloveniji usposobile organizacije, katerih logotipi so natisnjeni na tej strani.



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

2003 ŠTEVILKA 3 JUL/AVG/SEP LETNIK XI ISSN 1318-1882

Uvodnik	107
Razprave	
Miroslav Ribič, Andrej Kovačič, Marjan Lončarič: Sodelovanje kot temelj učinkovitega upravljanja oskrbovalne verige	109
Rok Rupnik, Marjan Krisper: Model kontekstno odvisnih mobilnih aplikacij	122
Petra Povalej, Peter Kokol, Jernej Zavašnik: Dve plati uporabe inteligentnih sistemov v medicini	131
Marjan Heričko, Simon Beloglavec, Matjaž B. Jurič, Boštjan Kežmah: Eksplicitna in implicitna uporaba vzorcev pri razvoju informacijskih rešitev	138
Maja Miličič: Človeški dejavnik pri povečevanju kakovosti storitev informacijske dejavnosti	150
Cene Bavec: Informatika med tradicionalnimi in virtualnimi organizacijami	156
Obvestila	
Niko Schlamberger: Generalna skupščina IFIP	164
Razpis za nagrade	165
Najava posvetovanja v Portorožu	166
Koledar prireditev	167

ISSN 1318-1882

Ustanovitelj in izdajatelj:

Slovensko društvo INFORMATIKA
Vožarski pot 12
1000 Ljubljana

Predstavniki

Niko Schlamberger

Odgovorni urednik:

Andrej Kovačič

Uredniški odbor:

Marko Bajec, Vesna Bosilj Vukšič, Dušan Caf, Aljoša Domijan, Janez Grad, Jurij Jaklič, Milton Jenkins, Andrej Kovačič, Tomaž Mohorič, Katarina Puc, Vladislav Rajkovič, Heinrich Reineremann, Ivan Rozman, Niko Schlamberger, John Taylor, Ivan Vezočnik, Mirko Vintar, Tatjana Welzer - Družovec

Recenzenti prispevkov za objavo v reviji Uporabna informatika:

Marko Bajec, Tomaž Banovec, Vladimir Batagelj, Marko Bohanec, Vesna Bosilj Vukšič, Dušan Caf, Srečko Devjak, Aljoša Domijan, Tomaž Erjavec, Matjaž Gams, Tomaž Gornik, Janez Grad, Miro Gradišar, Jože Gričar, Joszef Györkos, Marjan Heričko, Jurij Jaklič, Milton Jenkins, Andrej Kovačič, Iztok Lajovic, Tomaž Mohorič, Katarina Puc, Vladislav Rajkovič, Heinrich Reineremann, Ivan Rozman, Niko Schlamberger, Ivan Vezočnik, Mirko Vintar, Tatjana Welzer - Družovec, Franc Žerdin

Tehnična urednica

Mira Turk Škraba

Oblikovanje

Bons

Prelom

Dušan Weiss, Ada Poklač

Tisk

Prograf

Naklada

700 izvodov

Naslov uredništva

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

Revija izhaja četrtletno. Cena posamezne številke je 3.500 SIT. Letna naročnina za podjetja 13.800 SIT, za vsak nadaljnji izvod 8.900 SIT, za posameznike 4.600 SIT, za študente 2.000 SIT.

Revijo sofinancira Ministrstvo za šolstvo, znanost in šport 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.

© Slovensko društvo INFORMATIKA

Navodila avtorjem

Revija Uporabna informatika objavlja izvirne prispevke domačih in tujih avtorjev na znanstveni, strokovni in informativni ravni. Namenjena je najširši strokovni javnosti, zato je zaželeno, da so tudi znanstveni prispevki napisani čim bolj mogoče poljudno.

Članke objavljamo praviloma v slovenščini, prispevke tujih avtorjev v angleščini.

Prispevki so obojestransko anonimno recenzirani. Vsak članek za rubriko Razprave mora za objavo prejeti dve pozitivni recenziji. O objavi samostojno odloča uredniški odbor.

Prispevki naj bodo lektorirani, v uredništvu opravljamo samo korekturo. Po presoji se bomo posvetovali z avtorjem in članek tudi lektorirali. Prispevki za rubriko Razprave naj imajo dolžino do 40.000, prispevki za rubrike Rešitve, Poročila do 30.000, Obvestila pa do 8.000 znakov.

Naslovu prispevka naj sledi ime in priimek avtorja, ustanova, kjer je zaposlen in elektronski naslov. Članek naj ima v začetku do 10 vrstic dolg izvleček v slovenščini in angleščini, v katerem avtor opiše vsebino prispevka, dosežene rezultate raziskave. Abstract se začne s prevodom naslova v angleščino. Članku dodajte kratek življenjepis avtorja (do 8 vrstic), v katerem poudarite predvsem delovne dosežke.

Pišite v razmaku ene vrstice, brez posebnih ali poudarjenih črk, za ločilom na koncu stavka napravite samo en prazen prostor, ne uporabljajte zamika pri odstavkih.

Revijo tiskamo v črno-beli tehniki s folije, zato barvne slike ali fotografije kot originali niso primerne. Objavljali tudi ne bomo slik zaslonov, razen če niso nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Po možnosti jih pošiljajte posebej, ne v datoteki z besedilom članka. Disketi z besedom priložite izpis na papirju.

Prispevke pošiljajte po elektronski ali navadni pošti na naslov uredništva revije: ui@drustvo-informatika.si, Slovensko društvo INFORMATIKA, Vožarski pot 12, 1000 Ljubljana. Za dodatne informacije se obračajte na tehnično urednico Miro Turk Škraba.

Po odločitvi uredniškega odbora o objavi članka bo avtor prejel pogodbo, s katero bo prenesel vse materialne avtorske pravice na Slovensko društvo INFORMATIKA. Po izidu revije pa bo prejel nakazilo avtorskega honorarja po veljavnem ceniku ali po predlogu odgovornega urednika.

Spoštovane bralke, spoštovani bralci,

v uvodniku prejšnje številke revije Uporabna informatika sem tarnal nad premalo pomembno in odmevno vlogo informatike ter nad kronično neodgovornostjo, da ne rečem ignoranco menedžerjev do uporabe poslovne informatike v slovenskih organizacijah. Pa se je medtem kot po naključju "zgodila informatika" v NLB v vsem svojem "blišču". Ne vem, ali je to, kar se je zgodilo, položaju poslovne informatike in poslovnih rešitev v prid ali ne, vsekakor pa je odmevno in razpoznavno ter vredno nekaj dodatne obravnave.

Projekti, kot je ta v NLB, pri nas žal niso tako redki. Nekateri drugi, manj znani ali zamolčani projekti neuspešne informatizacije poslovanja imajo s poslovnega vidika za naše organizacije še dosti bolj neugodne posledice. Če odmislimo in izključimo koruptivne namene, pristanemo največkrat na neustreznem angažiranju ali nesposobnosti odločevalcev. Zavedam se, da je beseda nesposobnost težka in za tistega, na katerega jo naslavljam, lahko zelo boleča. Brez slabih namenov bom poskusil v nadaljevanju pojasniti specifiko nesposobnosti tistih, ki se pri nas odločajo o informatizaciji na strateški ravni. V večini primerov so tako menedžerji kot vodilni informatiki obremenjeni s predpostavko, da pri nas ni na voljo dovolj tehnoloških in vsebinskih znanj, spretnosti in rešitev, potrebnih za uspešno informacijsko podporo poslovanju. Takšno razmišljanje se nadaljuje s prepričanjem z nič manj pogubnimi posledicami, da bomo zgolj s tehnološko rešitvijo, brez lastnega angažiranja, opravili prenos znanja in najboljše svetovne prakse ter na ta način neboleče izpeljali tudi prenovu poslovanja. Z "grdimi" vsebinskimi in kadrovskimi problemi, kot so ureditev podatkov in postopkov, uvajanje sprememb, izobraževanje in motiviranje kadrov ter organizacijske prilagoditve, se seveda ne bi ukvarjali.

Teza, da tega sami ne znamo oz. da nimamo "nekih specifičnih" znanj je pri nas širše prisotna predvsem pri menedžerjih, žal pa tudi pri informatikih. Združenje za informatiko in telekomunikacije (ZIT) pri Gospodarski zbornici Slovenije kot nekakšen predstavnik stroke je lep primer takšnega obnašanja, nekateri bi rekli slovenske majhnosti. Raziskavo stanja našega trga IT je ZIT zaupalo tuji analitski firmi, ki "ima metodologijo in znanje in v tem delu Evrope edina lahko ponudi tako storitev". Nič nimam proti tujemu znanju in storitvam, me pa žalosti utemeljitev predstavnika ZIT o edinem možnem izvajalcu: "Doslej je bilo veliko tehnoloških študij, modrih knjig, a avtorji niso vključili glavnih akterjev na trgu. Poleg tega so bili slabi analitični procesi, analiza mora biti neodvisna ... " (citirano po: Raziskava: kje sploh je naš IT trg; Delo, 11. avgusta 2003). Presodite, ali te trditve držijo; sam menim, da ne.

Saj gre samo za deset milijonov tolarjev, boste rekli. Zato se vrnimo k razvpitemu in neprimerno "dražjemu" projektu informatike v NLB. Za nameček so najeli še svetovno priznana in primerno drago revizijsko firmo, ki preverja delovanje programske opreme oz. primernost uvedenih rešitev. Upam si staviti, da bo ugotovitev revizije, da gre za uveljavljeno in tehnološko korektno delujočo celovito programsko rešitev. Zajec namreč tiči v drugem grmu. NLB ni edina banka, ki je morala (končno) ob prehodu na transakcijske račune urediti podatke o svojih komitentih. Ni se zavedala obsežnosti problematike urejanja podatkov, ki bi jo morala reševati sproti in dosledno, zanašala se je na tehnološko rešitev, ki pri tem seveda ne more pomagati. Nekatero banke so to "čiščenje in konsolidacijo" podatkov opravile že prej, druge, manjše, so to "šlamastiko" odpravile v roku in brez posebnega vznemirjanja javnosti. Tiste, ki so rešitve razvile in uvedle same, so bile pri tem uspešnejše.

V primeru NLB odločitev o revidiranju informacijske rešitve po mojem prepričanju lastnikom ne prinaša neposredne koristi. Mogoče bo posredno nakazala potrebo po "revidiranju" menedžmenta in vodstva informatike, pa tudi nadzornega sveta, ki so v vseh zadnjih letih tolerirali stanje, da naša njavečja banka ni bila sposobna razpoznati števila in enolično identificirati vseh svojih komitentov. Zgodba bo koristna, če se bodo lastniki, menedžerji in informatiki iz tega primera kaj naučili, predvsem ustrežneje vrednotiti možnosti in pomen poslovne informatike ter potrebo po tesnem in neposrednem sodelovanju na takšnih projektih.

Andrej Kovačič

**Vabilo avtorjem k pripravi prispevkov
za tematsko številko revije Uporabna informatika**

Tema: Informacijska varnost

Obseg podatkov v elektronski obliki nezadržno narašča; za okrog 250 megabytov letno na prebivalca našega planeta. Čeprav je že 95 % vseh dokumentov vsaj v eni fazi obdelanih v elektronski obliki, jih je le 3 % shranjenih v elektronskih arhivih.

Z vse večjim obsegom raste tudi potreba po zagotavljanju zasebnosti in varnosti pri elektronskem poslovanju ter zaščiti podatkov. Na eni strani lahko spremljamo razvoj standardov, na primer znanega BS 7799 za področje informacijske varnosti, ki je leta 2000 dobil tudi mednarodnega naslednika ISO/IEC 17799, na drugi pa razvoj tehnologije, ki to omogoča. Predstavniki Intela na vprašanje, zakaj potrebujemo vse zmogljivejše mikroprocesorje, odgovarjajo, da preprosto zato, ker bo v prihodnje 90 % procesorske moči namenjene varovanju in zaščiti podatkov pri različnih oblikah prenosa.

Kje smo na področju informacijske varnosti v Sloveniji? Na vprašanje bomo skušali odgovoriti v posebni številki revije Uporabna informatika. Prepričani smo, da ni dovolj, da se pomena informacijske varnosti zavedamo in pri tem računamo na to, da bodo komercialne rešitve kmalu naprodaj. Sprašujemo se, kakšne so znanstvene in razvojne rešitve na tem področju? S katerim znanjem razpolagata danes naši univerzi? Ali lahko v prihodnosti pričakujemo domače rešitve, s katerimi bomo lahko osvojili svoj delež svetovnega trga?

Vaše prispevke pričakujemo z željo, da Uporabna informatika ostane eden izmed trdnjih mostov med stroko in prakso.

Področju informacijske varnosti se v bančništvu, zavarovalništvu, zdravstvu in javni upravi že doslej nismo mogli izogniti in zagotovo razpolagamo z ustreznimi izkušnjami in znanjem. Kakšno je stanje v gospodarstvu? Kako ste k zagotavljanju informacijske varnosti pristopili v vaši organizaciji? Vabimo vas, da predstavite konkretne rešitve in koncepte, ki jih vpeljujete v vašem okolju.

Informacijska varnost predstavlja temelj za nadaljnje poslovne korake in je zagotovo aktualna tema v trenutku, ko tudi v Sloveniji vse več uporabljamo svetovni splet, mobilne komunikacije in elektronsko poslovanje.

Vaše prispevke z oznako »Tema2003« pričakujemo na e-naslov: ui@drustvo-informatika.si

do 15. oktobra 2003.

Aljoša Domijan,
gostujoči urednik

▣ Sodelovanje kot temelj učinkovitega upravljanja oskrbovalne verige

Miroslav Ribič, Andrej Kovačič, Marjan Lončarič

Povzetek

Procesni pristop k upravljanju oskrbovalne verige nedvomno pomeni napredek pri izmenjavi informacij in toku proizvodov, vendar pa zaradi slabe povezanosti s kupci, in kar je še pomembneje, s končnimi potrošniki, rezultati pogosto niso optimalni. Tako so nekatera podjetja dosegla občutna izboljšanja poslovnih rezultatov, medtem ko druga ne, in sicer zaradi slabih informacij, nepovezanih sistemov, predvsem pa zaradi slabega medsebojnega sodelovanja.

Abstract

Collaboration as Foundation for Effective Supply Chain Management

There is no doubt that the process orientation of supply chain management has brought much improvement to the flow of information and product. However, it has been disconnected from the customer and, more importantly, from the consumer, resulting in less than optimal result. Some companies have achieved significant results while others have not, primarily because of poor information, disconnected systems and lack of collaboration.

1 Opredelitev sodelovanja

Upravljanje oskrbovalne verige pomeni učinkovito koordinacijo njenih članov, da se zagotovi nemoten materialni tok v obsegu, ki zadovolji tržno povpraševanje (Holten et al., 2002). Dostikrat se zgodi, da zaradi nepopolnih ali nezadostnih informacij materialni tok presega povpraševanje, kar povzroča nepotrebne zaloge in s tem višje stroške. Zato je temeljni cilj upravljanja oskrbovalne verige omejitev učinka naraščajočega povpraševanja (angl. bullwhip effect).

To je pojav, ki je značilen za oskrbovalne verige, za katere je značilno pomanjkljivo sodelovanje. V razmerah, ko so informacijski tokovi znotraj oskrbovalne verige prekinjeni, se lahko povpraševanje oceni le na podlagi opravljenih transakcij. To pripelje do položaja, ko kljub relativno nespremenljivem povpraševanju kupcev stopnja zalog skozi oskrbovalno verigo očitno narašča. Če se, npr. kot posledica promocijskih aktivnosti, povpraševanje končnih potrošnikov malo poveča, se skladno s tem poveča naročilna količina trgovcev pri njihovih proizvajalcih. Ker proizvajalci ne vedo, da je večja naročilna količina le začasna posledica promocijskih aktivnosti trgovcev, ustrezno dopolnijo napoved povpraševanja, s tem pa se poveča obseg njihovih naročil dobaviteljem. Vsak člen v oskrbovalni verigi želi ohraniti kakovost svojih storitev, saj bi v nasprotnem primeru tvegala izgubo tržnega deleža. Zato si mora

zagotoviti višjo stopnjo varnostnih zalog, ki je tako proti koncu oskrbovalne verige večja in večja. Tako se stopnja variabilnosti naročilnih količin povečuje, ko se premikamo od začetka proti koncu oskrbovalne verige, z drugimi besedami, majhen val na začetku oskrbovalne verige, ki je posledica nekoliko povečanega povpraševanja končnih potrošnikov, sproži nesorazmerno velik val, ki se večja in večja proti koncu oskrbovalne verige. Proti temu pojavu se je mogoče boriti le z vzpostavitev medorganizacijskih povezav, s katerimi se povežejo prekinjeni informacijski tokovi. Tako je omogočeno boljše sodelovanje na osnovi izmenjave informacij, s katerim dosežemo zmanjšanje stopnje negotovosti, zmanjšanje spremenljivosti povpraševanja in skrajšanje časa izpolnitve naročila, vse to pa je orožje proti nepotrebni stroškom zalog.

Poznamo več stopenj medorganizacijskih povezav, in sicer od pogajanj na odprtem trgu, povezovanja in koordinacije med partnerji pa vse do sodelovanja. Čeprav se iz slike 1 zdi, da je model medorganizacijskih relacij linearen, ni treba, da se partnerji v oskrbovalni verigi prebijejo skozi vse stopnje, vendar pa se morajo za vsak premik, iz ene stopnje na drugo, prilagoditi novim spremembam (Uchatzy, Karas, 2001).

Sodelovanje predstavlja najvišjo obliko med-organizacijskih odnosov. Pojavi se, ko si dve ali več organizacij delita odgovornosti pri skupnem planiranju in izvedbi ter informacije, ki so posledica merjenja učinkovitosti rezultatov in procesov v oskrbovalni verigi. Sodelovanje je torej priložnost za optimizacijo tako oskrbovalne verige kot tudi odnosov med partnerji. Pri tem igra pomembno vlogo tehnologija, saj omogoča prenos podatkov v realnem času, vendar pa moramo prav tako poudariti človeške dejavnike, saj je le od njih odvisna stopnja pripravljenosti za izmenjavo informacij in znanja.

Tehnologija je torej le pogoj za učinkovito sodelovanje, ključni pa so človeški dejavniki. Strnemo jih lahko v naslednje dejavnike (Uchatzy, Karas, 2001):

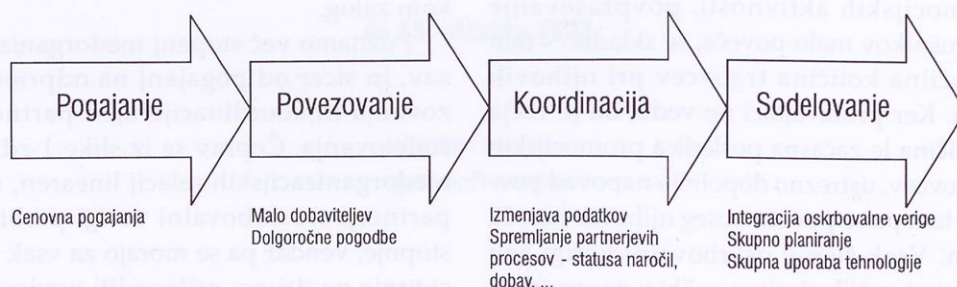
1. *Skupni interesi in jasna pričakovanja* – vsak udeleženc v oskrbovalni verigi mora vedeti, kaj se od njega pričakuje in kaj lahko pričakuje od drugih. Prav tako mora poznati svoj delež pri delitvi dobička oz. izgube, saj lahko le tako pristane na dolgoročno sodelovanje.
2. *Delitev koristi in negativnih posledic* – sodelovanje lahko opredelimo kot delovanje v skupno korist, kar pomeni delitev tveganja, odgovornosti in nagrad, torej »skupno življenje v dobrem in slabem«.
3. *Zaupanje in odprtost* – za učinkovito sodelovanje morajo udeleženci oskrbovalne verige pogosto razpravljati o planiranem in dejanskem poslovanju ter oblikovanju procesov, saj je le tako mogoče nenehno izboljševati oskrbovalno verigo. Pogoj za to je zaupanje in odprtost pri komuniciranju, to pa zahteva miselni preskok, ki ni značilen za tradicionalno poslovanje.
4. *Spoznanje kdo oz. kaj je pomembno* – izredno pomembna je izbira pravih partnerjev in opredelitev pravih aktivnosti v oskrbovalni verigi. Odločitve v

zvezi s tem morajo biti oblikovane tako, da se doseže čim večja dolgoročna korist.

5. *Vodstvo* – brez pravega vodje, ki bi spodbujal celotno oskrbovalno verigo k resničnemu sodelovanju, ni mogoče pričakovati pomembnejših poslovnih rezultatov.
6. *Pomoč in ne kaznovanje* – ko gre vse narobe, kaznovanje praviloma ne prinese izboljšanja stanja. Pravi pristop je skupno reševanje problemov.

Če človeški dejavniki niso upoštevani, lahko tudi najbolj oblikovana oskrbovalna veriga zaide v težave. Resne ovire na poti k sodelovanju predstavljajo (Uchatzy, Karas, 2001):

1. *Omejen vpogled v dogajanje v oskrbovalni verigi* – je ostanek tradicionalnih funkcijsko oblikovanih organizacijskih struktur, kjer je vsakdo razmišljal strogo o svojem delu.
2. *Nenehna pogajanja med člani oskrbovalne verige* – ponavadi zahtevajo veliko časa in energije, hkrati pa dostikrat negativno vplivajo na sodelovanje.
3. *Nepripravljenost vlagati dodatni čas in trud* – sodelovanje zahteva veliko časa in trdega dela, zato morajo vsi dobro poznati pričakovane koristi, da so pripravljeni vložiti dodaten trud.
4. *Tradicionalna računovodska praksa* – temelji le na finančnem vrednotenju ene organizacije in pri tem ne upošteva intelektualnega kapitala. Tako je usmerjena le na merjenje poslovanja organizacije v preteklosti, zanemarija pa znanje ter medorganizacijske povezave in njihovo vrednost.
5. *Neprimerno komuniciranje* – ko postane komuniciranje med partnerji v oskrbovalni verigi nezadostno, pa naj gre za tehnične težave ali človeške dejavnike, se ponavadi problemi povečujejo eksponentno.



Slika 1: Stopnje medorganizacijskih odnosov (Uchatzy, Karas, 2001)

6. *Izdaja* – laganje, zavajanje in napačno prikazovanje informacij je prav gotovo največja ovira, ki predstavlja izločitveni kriterij na poti k resničnemu sodelovanju.

2 Skupno planiranje, napovedovanje in oskrba

Model skupnega planiranja, napovedovanja in oskrbe (angl. Collaborative Planning, Forecasting and Replenishment, v nadaljevanju: CPFR) formalizira proces oskrbe med dvema partnerjema v oskrbovalni verigi, ki se dogovorita o skupnem poslovnem planu, skupni napovedi povpraševanja, skupnem merjenju poslovnih rezultatov in učinkovitosti poslovnih procesov, hkrati pa se odzivata in drug drugemu priznavata kakršnekoli izjeme, ki v procesu nastajajo (VICS, 2002). Temeljne značilnosti modela lahko strnemo v tri točke, in sicer:

1. CPFR predstavlja ogrodje, čigar poslovni procesi so usmerjeni h končnemu potrošniku in oblikovani tako, da je dodana vrednost čim višja. CPFR je strukturiran kot nabor scenarijev, v katerih glavni procesi pripadajo različnim partnerskim vlogam. Partnerji lahko pri sodelovanju izberejo več scenarijev, odvisno od narave proizvodov, promocijskih in sezonskih vplivov ter stopnje pomembnosti njihove vloge v modelu CPFR.
2. Partnerji se sporazumejo o skupni napovedi povpraševanja, ki predstavlja edino osnovo za vsa planiranja znotraj oskrbovalne verige. Pri tem igra ključno vlogo tisti partner, ki je najbližje končnim potrošnikom, praviloma je to trgovina na drobno. Trgovci imajo neposreden stik s potrošniki, zato lahko njihovo obnašanje predvidevajo s pomočjo upravljanja odnosov s potrošniki. Prav tako imajo zbrane podatke o svojih konkurentih, njihovi ponudbi in prodajnih akcijah. Na podlagi tako zbranih različnih pogledov na trg lahko partnerji v oskrbovalni verigi skozi iterativno izmenjavo informacij izboljšajo natančnost napovedi povpraševanja končnih potrošnikov. Končni rezultat je ena sama in skupna napoved povpraševanja končnih potrošnikov, ki je osnova za nadaljnje planiranje v oskrbovalni verigi. Tako se iz napovedi povpraševanja končnih potrošnikov oblikuje ena sama in skupna napoved naročanja, ki predstavlja napoved povpraševanja znotraj oskrbovalne verige. Sodelovanje pri napovedovanju omogoča precej večjo natančnost kot individualna uporaba zapletenih tovrstnih algoritmov. Tako največ

dodane vrednosti izhaja prav iz izmenjave informacij o napovedih povpraševanja.

3. Partnerji se zavežejo k sodelovanju in skupni uporabi informacij s ciljem, da se čimbolj učinkovito odstranijo ovire pri oskrbovalnih procesih. Vrednost skupne napovedi povpraševanja, če vse ostalo ostane nespremenjeno, je boljša koordinacija aktivnosti znotraj oskrbovalne verige. To sicer prinese dobro vidne, vendar ne drastične koristi. Drastične koristi izhajajo iz uporabe skupne napovedi z namenom, da se odstranijo ovire, ki slabijo učinkovitost oskrbovalne verige. Ena od takih ovir je prav gotovo fleksibilnost proizvodnje. Praviloma ima trgovec na zalogi dovolj blaga, da zadovolji potrebe potrošnikov. Kapacitete proizvajalca zato niso učinkovito zasedene, saj je čas naročila potrošnika (proizvod je na zalogi) precej krajši, kot je čas izdelave proizvoda. S podaljšanjem časa naročila tako, da je usklajen s časom proizvodnje, se lahko proizvajalec namesto za proizvodnjo na zalogo odloči za proizvodnjo po naročilu. Tako ni več potrebe po visoki stopnji varnostnih zalog, kar znatno znižuje stroške znotraj celotne oskrbovalne verige. Podoben primer je uporaba skupnega premoženja. Skozi medorganizacijsko terminiranje uporabe skupnega premoženja se lahko proizvodne, transportne in distribucijske kapacitete izkoriščajo bolj optimalno. Lahko se izkaže, da so zaradi tovrstnega sodelovanja kapacitete prevelike. Tako je mogoča razširitev dejavnosti, dajanje kapacitet v najem ali pa njihova prodaja. V vsakem primeru se stroški drastično znižajo.

2.1 Struktura modela CPFR

Originalni dvoslojni model CPFR temelji na sodelovanju dveh partnerjev, in sicer trgovca in proizvajalca. Partnerja med seboj sodelujeta tako, da si izmenjujeta informacije, ki pomembno vplivajo na uspešnost poslovanja celotne oskrbovalne verige. Te informacije so praviloma napovedi prodaje, ki izražajo povpraševanje končnih potrošnikov v prihodnosti. Če je napoved točna, imajo tudi ostali člani oskrbovalne verige veliko možnosti, da prispevajo k visoki stopnji dodane vrednosti h končnemu proizvodu in tako zadovoljijo ali celo presežejo pričakovanja končnih potrošnikov. V nasprotnem primeru pa se lahko zgodi dvoje. Če je napoved prodaje previsoka, bo previsoka tudi proizvodnja znotraj celotne oskrbovalne verige.

To praviloma pomeni visoke stroške zalog ali pa, zaradi prevelike ponudbe, prodajo po nižji ceni. Oboje slabi konkurenčnost celotne oskrbovalne verige. V primeru prenizke napovedi prodaje pa bo povpraševanje večje kot ponudba, kar na monopolnem trgu pomeni prodajo po višjih cenah in posledično nižjo stopnjo zadovoljstva končnih potrošnikov, v konkurenčnem okolju pa izredno visoke stroške v obliki izgube trga.

Scenarij	Napoved povpraševanja	Napoved naročanja	Naročanje
A	Trgovec	Trgovec	Trgovec
B	Trgovec	Proizvajalec	Proizvajalec
C	Trgovec	Trgovec	Proizvajalec
D	Proizvajalec	Proizvajalec	Proizvajalec

Slika 2: Scenariji dvoslojnega modela CPFR (VICS, 2002)

Scenariji modela CPFR, ki so razvidni iz slike 2, predstavljajo različice modela, ki se lahko uporabijo v posamezni oskrbovalni verigi. Uporabi se tista različica, ki najbolj ustreza vsem partnerjem, njihovim možnostim, virom ter drugim poslovnim in sistemskim omejitvam. V vsakem scenariju imata oba partnerja vpliv tako na planiranje, napovedovanje kot tudi na oskrbo, vendar eden izmed njiju prevzame vodilno vlogo. V scenariju A vse primarne procese vodi trgovec. V scenarijih B, C in D nalogo naročanja prevzame proizvajalec. V scenariju B prevzame trgovec pobudo pri napovedi povpraševanja, medtem ko proizvajalec skrbi za oskrbo, v scenariju C pa trgovec skrbi tudi za napoved naročanja. Za scenarij D je značilno, da vse primarne procese vodi proizvajalec. Iz slike 3 je razvidno, da je vsak izmed scenarijev razdeljen v tri faze in devet temeljnih procesov, ki se od scenarija do scenarija nekoliko razlikujejo. Prva faza je planiranje, ki vključuje oblikovanje sporazuma o sodelovanju (1) in skupnega poslovnega plana (2). Druga faza je faza napovedovanja, ki zajema napoved povpraševanja (3), identifikacijo (4) in reševanje izjem (5), ki izhajajo iz same napovedi. Prav tako sodi v fazo napovedovanja tudi napoved naročanja (6) ter identifikacija (7) in reševanje izjem (8), ki so povezane z napovedjo naročanja. V zadnji fazi, to je fazi oskrbe, se izvaja proces naročanja (9). V modelu ločimo skupne aktivnosti (slika 3 – sredina), to so

aktivnosti, ki izvirajo iz sporazuma o sodelovanju, ter aktivnosti trgovca (slika 3 – levo) in aktivnosti proizvajalca (slika 3 – desno).

Model CPFR je bil sprva zamišljen kot dvoslojni model, vključeval je torej le dva poslovna partnerja, to pa je za kompleksne vertikalne oskrbovalne verige premalo. Sicer je mogoče iz parov kupec-prodajalec sestaviti več modelov CPFR in jih med seboj povezati, toda to zahteva ogromno truda za usklajeno delovanje. Rešitev je večslojni model CPFR, ki dopolnjuje model v dveh smereh. Prvič, z zamenjavo izrazov trgovec in proizvajalec z izrazoma kupec in prodajalec postane model CPFR univerzalen ter v taki obliki primeren na vsakem nivoju oskrbovalne verige ne glede na gospodarsko dejavnost.

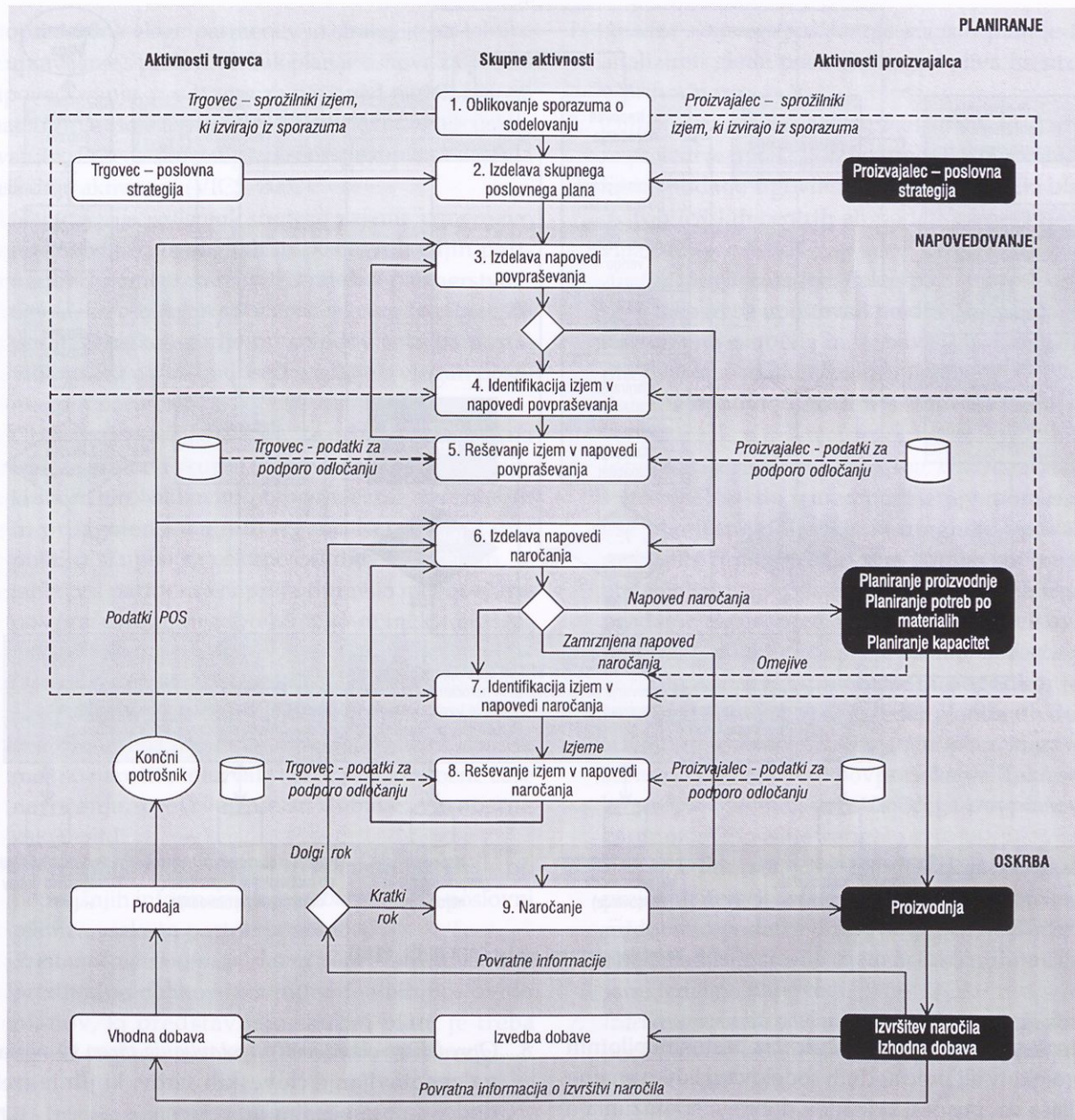
In drugič, razširitev modela z dveh na poljubno število slojev omogoča implementacijo najbolj kompleksnih oskrbovalnih verig. Tovrstno povezovanje podjetij v zunanjo vrednostno verigo, ali bolje rečeno vrednostno mrežo, je prav gotovo največji izziv za številne kompleksne industrije. Implementacija večslojnega modela CPFR omogoča učinkovit boj proti tako imenovanemu učinku naraščajočega povpraševanja, ki se pojavi kot posledica rahlo povečanega povpraševanja končnih potrošnikov.

V nadaljevanju je opisanih devet temeljnih poslovnih procesov modela CPFR, ki so skupni tako za dvoslojni kot tudi za večslojni model.

2.1.1 Oblikovanje sporazuma o sodelovanju

Prvi proces modela CPFR je oblikovanje sporazuma o sodelovanju. V njem kupec in prodajalec vzpostavita okvir skupnega sodelovanja tako, da opredelita svoje cilje in pričakovanja ter aktivnosti in vire, ki so potrebni za učinkovito upravljanje oskrbovalne verige. Oblikovanje sporazuma o sodelovanju, ki je rezultat oz. izhodni dokument procesa, praviloma zahteva izvedbo naslednjih aktivnosti (VICS, 2002):

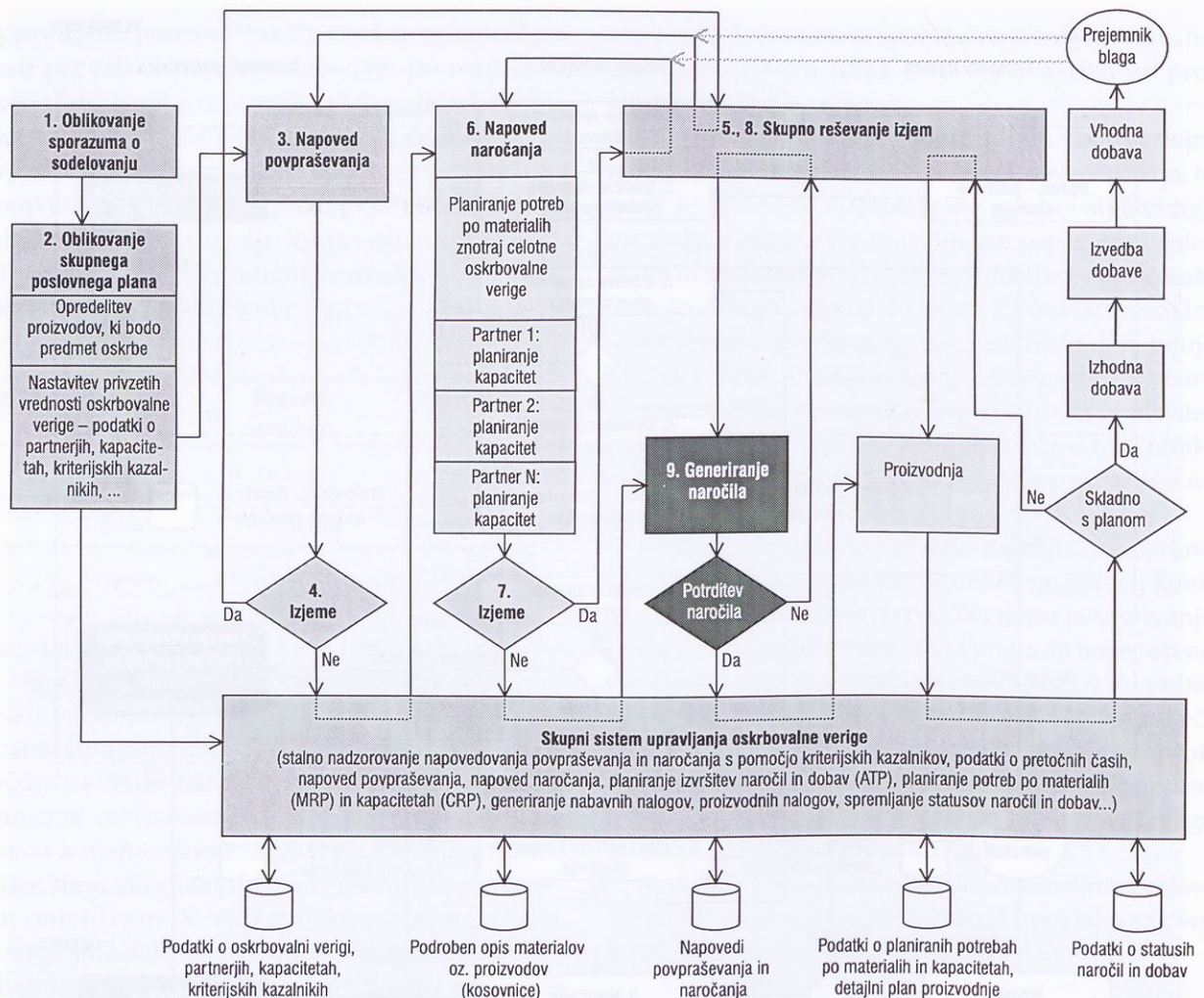
1. *Oblikovanje izjave o sodelovanju.* Izjava je dokument, ki obsega skupno razumevanje ciljev sodelovanja in pripravljenost partnerjev na zagotavljanje vseh potrebnih virov.
2. *Opredelitev ciljev,* ki so merljivi in imajo jasno opredeljene metrike za njihovo merjenje. Prav tako je treba ugotoviti vpliv sodelovanja na poslovanje vsakega izmed partnerjev in temu primerno opredeliti morebitne izjeme, ki bi lahko nastale pri sodelovanju, bodisi pri napovedih povpraševanja, bodisi pri napovedih naročanja.



Slika 3: Dvoslojni procesni model CPFR (VICS, 2002)

- Razprava o zmožnostih, virih in sistemih je potrebna, da ugotovimo, v kolikšni meri lahko vsak izmed partnerjev prispeva k upravljanju oskrbovalne verige.
- Opredeletitev točk sodelovanja in odgovornosti pomeni povezovanje procesov med partnerji. Pri tem je potrebno vzpostaviti oddelke, ki bodo glavni izvajalci teh procesov.
- Določitev skupnih informacijskih potreb je ključnega pomena za uspešno in dolgoročno sodelovanje.

Treba je doseči dogovor o metodologiji napovedovanja, pogostosti osveževanja napovedi, tehnologiji in ostalih podatkovnih elementih. Prav tako je treba določiti kriterijske kazalnike in njihove mejne vrednosti za identifikacijo izjem, tako za proces napovedovanja povpraševanja kot za proces napovedovanja naročanja. Ti kriterijski kazalniki so ključni za kontrolo procesa napovedovanja.



Slika 4: Večslojni procesni model CPFR (VICS, 2002)

6. Upoštevanje predhodnih izkušenj iz preteklih pilotnih projektov ali predhodnih sodelovanj lahko znatno olajša oz. pospeši sklenitev dobrega sporazuma o sodelovanju.
7. Oprelitev obveznosti iz naslova naročanja. To pomeni skleniti dogovore oz. pogodbe na osnovi vrednosti oz. količine za določeno obdobje. V okviru tega obdobja se ponavadi s pomočjo terminskih sporazumov terminira napoved naročanja, in sicer tako, da se izpolnijo pogoji iz vrednostne oz. količinske pogodbe. Prav tako je potrebno določiti časovno obdobje (npr. 1 mesec), ki napoved zamrzne. Zamrznjena napoved pomeni obveznost iz naslova naročanja, saj se količine iz zamrznjene napovedi avtomatično pretvorijo v naročila.
8. Oprelitev obveznosti zagotavljanja virov se nanaša na zagotavljanje človeških virov, ki jih je treba dodeliti posameznim procesom.
9. Oprelitev reševanja nesporazumov pomeni opredelitev temeljnih pravil za ravnanje v primeru nesporazumov.
10. Določitev postopka za revizijo sporazuma je osnova za nenehno izboljševanje procesov. Postopek mora biti tak, da je mogoče model popraviti, če je treba.

2.1.2 Izdelava skupnega poslovnega plana

Proces izdelave skupnega poslovnega plana temelji na že vzpostavljenem sporazumu o sodelovanju in zahteva izmenjavo informacij o poslovnih strategijah partnerjev. Rezultat je vzajemno usklajen poslovni plan, v katerem

so opredeljene vloge partnerjev in strategije ter taktike za upravljanje s proizvodi. Tak plan je osnova za proces napovedovanja, predvsem za napoved naročanja, saj znatno zmanjšuje izjeme in potrebo po dodatnih usklajevanjih. Proces skupnega poslovnega plana zajema naslednje aktivnosti (VICS, 2002):

1. *Identifikacija poslovnih strategij* zajema izmenjavo informacij o strategijah in poslovnih ciljih partnerjev z namenom razviti strateško partnerstvo.
2. *Oprelitev kategorij proizvodov, vlog in ciljev.* Za posamezne kategorije proizvodov je treba postaviti prodajne cilje in doseči soglasje o vlogah posameznih partnerjev.
3. *Oblikovanje prodajne politike.* Za učinkovit nastop na trgu se pripravi skupni plan pospeševanja prodaje, ki mora biti oblikovan po posameznih proizvodih in prilagojen lokalnim trgov. Ker je prodajna politika skupna za celotno oskrbovalno verigo, se lahko vsi partnerji vnaprej pripravijo na povečano povpraševanje in tako izničijo učinek naraščajočega povpraševanja.
4. *Oprelitev profilov posameznih proizvodov* zagotavlja podporo operativnim procesom. Tako je za vsak proizvod treba določiti minimalno količino, ki jo je možno naročiti, intervale naročanja, pretočne čase naročanja, proizvodnje in dobave, varnostne zaloge, itd.
5. *Oblikovanje individualnih poslovnih planov* – po poprejšnjih informacijah je treba uskladiti poslovni plan za vsakega partnerja posebej.
6. *Pristanek na skupni poslovni plan* – s primerjavo predhodno oblikovanih individualnih poslovnih planov, ki predstavljajo skupni plan, je treba doseči soglasje partnerjev v oskrbovalni verigi.

2.1.3 Izdelava napovedi povpraševanja

Napoved povpraševanja oz. napoved prodaje je temeljni plan, iz katerega morajo izhajati vsi drugi, zato se napake oz. nenatančnost pri napovedi povpraševanja prenašajo v ostale plane oskrbovalne verige in individualne plane posameznih partnerjev, vse to pa multiplicira stroške znotraj celotne oskrbovalne verige. Zato je pomembno, da je tovrstno napovedovanje zares natančno. Praviloma se napoved pripravi v vsaj treh dimenzijah, in sicer glede na proizvod, regijo oz. trg in čas. V scenarijih A, B in C modela CPFR za proces izdelave napovedi povpraševanja skrbi kupec, v scenariju D pa prodajalec. Proces izdelave napovedi povpraševanja poteka v naslednjih korakih (VICS, 2002):

1. *Analiza skupnega poslovnega plana* – plan je treba analizirati glede potencialnega vpliva na prodajo v prihodnosti.
2. *Zbiranje in analiza podatkov o porabi.* Pri izdelavi napovedi je treba uporabiti podatke o porabi, to so lahko podatki trgovin na drobno, premiki blaga v distribucijskih centrih ali podatki o porabi proizvajalca.
3. *Analiza drugih podatkov.* Poleg podatkov o porabi je prav tako treba upoštevati predhodne napovedi, še neizvršena naročila in dobave, elastičnost povpraševanja, podatke o potrošnikih in njihovih navadah ter njihovi vrednosti s stališča oskrbovalne verige.
4. *Identifikacija planiranih dogodkov, ki so tesno povezani s prodajo.* Vsakdo izmed partnerjev mora navesti vse planirane dogodke, ki utegnejo vplivati na prodajo v prihodnosti oz. na porabo znotraj oskrbovalne verige, to so lahko odprtje novih trgovin, prodajne akcije, predstavitve novih izdelkov, itd.
5. *Ažuriranje skupnega koledarja planiranih dogodkov.* Z zbranimi podatki o planiranih dogodkih je potrebno ažurirati skupni koledar planiranih dogodkov, kar predstavlja kratkoročni plan, ki ga upoštevamo pri napovedi povpraševanja. Tako se lahko prepreči učinek naraščajočega povpraševanja znotraj oskrbovalne verige.
6. *Zbiranje podatkov o reševanju izjem.* Na podlagi prejšnjih iteracij izdelave napovedi povpraševanja pridobimo podatke o reševanju izjem. To že vnaprej zmanjšuje število izjem in tako prihrani čas pri sami izdelavi napovedi.
7. *Izdelava napovedi povpraševanja.* Na podlagi zbranih podatkov se izdelava napovedi povpraševanja končnih potrošnikov.

2.1.4 Identifikacija izjem v napovedi povpraševanja

Namen identifikacije izjem v napovedi povpraševanja je ugotoviti tista odstopanja, ki padejo izven meja vrednosti kazalnikov, za katere so se partnerji dogovorili v sporazumu o sodelovanju. Ti kazalniki služijo kot varovalke, saj nenehno nadzorujejo proces napovedovanja in kažejo na morebitne težave, ki se lahko pojavijo v prihodnosti. Proces identifikacije izjem v napovedi povpraševanja vključuje naslednje aktivnosti (VICS, 2002):

1. *Identifikacija kazalnikov.* V sporazumu o sodelovanju se identificirajo vsi kazalniki, na podlagi katerih se določijo izjeme. To so kazalniki, ki merijo natančnost

- napovedovanja, stopnjo obračanja zalog, pretočne čase naročil, proizvodnje, dobave, itd.
2. *Identifikacija sprememb skupnega plana.* V skupnem planu je treba identificirati vse spremembe, ki so lahko nastale zaradi novih proizvodnih kapacitet v oskrbovalni verigi, lahko gre za nova skladišča ali pa nove trgovine trgovcev.
 3. *Preverjanje skladnosti napovedi povpraševanja s kazalniki.* Skozi iterativno povratno zanko preverimo vse postavke napovedi povpraševanja s kazalniki in tako izmerimo, ali so kapacitete in ostale zmožnosti partnerjev dovolj velike, da se realizira napoved povpraševanja. Če pri primerjavi posamezne postavke njene vrednosti padejo izven sprejemljivega območja, dogovorjenega s sporazumom o sodelovanju, se postavko označi kot izjemo oz. neskladnost.
 4. *Identifikacija izjem.* Iz postavk, ki jih je mogoče spoznati kot izjeme, sestavimo nabor izjem, ki predstavlja izhodni rezultat procesa identifikacije izjem v napovedi povpraševanja in hkrati vhodni parameter za naslednji proces, to je reševanje izjem.

2.1.5 Reševanje izjem iz napovedi povpraševanja

V primeru, da se v predhodnem procesu odkrijejo izjeme, jih razrešimo s sodelovanjem med partnerji in ustrezno popravimo napoved povpraševanja. Učinkovito reševanje izjem omogoča tudi uporaba tehnologije, ki zagotavlja izmenjavo informacij v realnem času in tako nudi podporo skupnemu odločanju. Proces reševanja izjem v napovedi povpraševanja poteka takole (VICS, 2002):

1. *Pridobitev nabora izjem in podatkov za podporo odločanju.* Nabor izjem je rezultat predhodnega procesa, druge podatke pa pridobimo od partnerjev v oskrbovalni verigi. Praviloma so za učinkovito odločanje potrebni podatki skupnega koledarja poslovnih dogodkov, podatki o časovnih serijah prodaje, nabav, proizvodnje, dobav, podatki o kapacitetah, vrednosti kazalnikov v preteklosti itn., odvisno pač od narave izjeme.
2. *Analiza izjem.* Ponavadi zadostuje analiza vpliva poslovnih dogodkov iz skupnega koledarja, ki v času izdelave napovedi povpraševanja še niso bili upoštevani. Če to ni dovolj, si pomagamo s poprej zbranimi podatki za podporo odločanju, hkrati pa stopnjujemo sodelovanje z elektronskim komuniciranjem, tj. s telefonskimi in video konferencami in tudi z osebnimi sestanki.
3. *Popravek napovedi povpraševanja.* Razrešene izjeme je treba upoštevati tako, da se ustrezno spremeni napoved povpraševanja.

2.1.6 Izdelava napovedi naročanja

Napoved povpraševanja predstavlja predvidevanje povpraševanja končnih potrošnikov. Le-to se preslika v napoved naročanja, pri čemer upoštevamo kapacitete in omejitve oskrbovalne verige. Napoved naročanja na najvišjem nivoju torej pomeni časovno serijo proizvedenih končnih proizvodov, v splošnem oz. na nižjih nivojih pa pomeni povpraševanje po materialih in polizdelkih (skladno s kosovnico končnega proizvoda) znotraj oskrbovalne verige. Napoved naročanja na nižjem nivoju se pripravi z upoštevanjem napovedi na višjem nivoju, za katerega se izvede planiranje potreb po materialih. Pri tem je zelo pomembno, da se upoštevajo pretočni časi naročil, proizvodnje in dobav, ki premikajo čas naročanja po časovni osi nazaj. Proces izdelave naročanja vključuje naslednje aktivnosti (VICS, 2002):

1. *Upoštevanje napovedi povpraševanja in pridobivanje ostalih podatkov.* Napoved naročanja temelji na napovedi povpraševanja končnih potrošnikov, prav tako pa so pomembni podatki o dogodkih, ki vplivajo na napoved naročanja. Upoštevati je treba tudi strategije upravljanja z zalogami, trenutno stanje zalog, statuse neizvršenih naročil in nedobavljenih dobav.
2. *Analiza naročanja in dobav v preteklosti, predvsem pa trenutnih in bodočih omejitev kapacitet.* V povratnih informacijah o že izvršenih naročilih in že dobavljenih dobavah so vgrajene izkušnje, ki so pomemben parameter pri izdelavi napovedi naročanja. Še bolj pa so pomembne omejitve razpoložljivosti kapacitet, saj je od njih odvisna sposobnost oskrbovalne verige, da realizira napovedi naročanja in zadovolji končne potrošnike.
3. *Zbiranje podatkov o reševanju izjem.* Na podlagi prejšnjih iteracij izdelave napovedi naročanja je treba pridobiti podatke o reševanju izjem. To že vnaprej zmanjšuje število izjem in tako prihrani čas pri sami izdelavi napovedi.
4. *Izdelava napovedi naročanja.* Na podlagi napovedi povpraševanja in opravljenih analiz se kreirajo napovedi naročanja za vse materiale, ki sestavljajo končni proizvod. Napoved naročanja je sestavljena glede na material, kupca in čas. Ponavadi se planirajo tedenske, ponekod celo dnevne količine.

Pri tem je potrebno vedeti, da planirane količine v zamrznjenem obdobju ne morejo biti več predmet sprememb, saj predstavljajo vhodni parameter za proces kreiranja naročil.

2.1.7 Identifikacija izjem v napovedi naročanja

Identifikacija izjem v napovedi naročanja se izvaja analogno kot identifikacija izjem v napovedi povpraševanja (VICS, 2002).

2.1.8 Reševanje izjem iz napovedi naročanja

Reševanje izjem iz napovedi naročanja se izvaja analogno kot reševanje izjem iz napovedi povpraševanja (VICS, 2002).

2.1.9 Naročanje

Zadnji proces modela CPFR je proces naročanja. Njegova naloga je transformacija zamrznjenega dela napovedi naročanja v naročila. Proces je sestavljen iz naslednjih aktivnosti (VICS, 2002):

1. *Opredelitev zamrznjene napovedi iz napovedi naročanja.* Zamrznjeno napoved se opredeli na podlagi časovnega odbobja, ki je dogovorjen v sporazumu o sodelovanju in določa obveznosti iz naslova naročanja.
2. *Kreiranje naročila.* Na podlagi zamrznjene napovedi naročanja se kreirajo naročila. Naročilo lahko kreira tako kupec kot prodajalec.
3. *Potrditev naročila.* Ob kreiranju naročila mora tisti, ki je naročilo kreiral, poslati partnerju potrdilo, ki sproži izvršitev naročila.

3 Skupno upravljanje transporta

Model CPFR povzroča revolucionarne učinke pri vseh partnerjih v oskrbovalni verigi. Proces, ki je sestavljen iz devetih aktivnosti, tesno povezuje napoved prodaje z oskrbo znotraj oskrbovalne verige, in sicer tako, da se naročila oskrbe generirajo avtomatično. Tu pa se model CPFR konča, kajti nima opredeljenih korakov za izvedbo naročil v oskrbovalni verigi ter za upravljanje distribucije in transporta. Manjka torej proces, ki bi podpiral pretvorbo naročil v pošiljke, objavlanje ponudb za prevoz pošiljk, izbiro prevoznikov, terminiranje in spremljanje dostav ter plačilo.

To ni slabost modela CPFR, ampak priložnost za njegovo dopolnitev. Naravna razširitev obstoječega modela CPFR je torej preoblikovanje napovedi naročanja v napoved dobav, ki tako zagotavlja natančno izpolnjevanje naročil in s tem visoko stopnjo kako-

vosti storitev. Trendi, kot so elektronsko trgovanje in prilagajanje proizvodov po meri končnim potrošnikom, silijo tako proizvajalce kot tudi trgovce v skrajševanje planskih ciklov, nenehno spreminjanje planov ter njihovo kar se da hitro izvedbo. Hitra odzivnost na tržne spremembe je pogojena s tesnim sodelovanjem vseh partnerjev v oskrbovalni verigi, saj je to edini mehanizem, ki zmore ob spremembi informacij interaktivno vplivati na planiranje in na izvedbo. Tako postaja ločnica med planiranjem in izvedbo vedno bolj nejasna.

S krajšimi planskimi cikli in univerzalnim ciljem »zmanjšati zaloge znotraj celotne oskrbovalne verige«, postaja transport eden izmed ključnih dejavnikov uspeha. Prvič, dandanes je v procesu transporta blaga še ogromno rezerv, saj se ponavadi prevozniki pogajajo s svojimi strankami na ravni posameznih pošiljk. To v končni fazi praviloma povzroča presežke zalog znotraj oskrbovalne verige in neučinkovito uporabo opreme prevoznikov. In drugič, poslovni rezultati partnerjev v oskrbovalni verigi so v veliki meri odvisni od upravljanja zalog. Pomembno je, da so ustrezne količine zalog na pravem mestu ravno takrat, ko je to potrebno. Pogosto je blago že v transportnem procesu, toda njegovega statusa se ne da ugotoviti, ali pa je zaradi nedostopnih kapacitet prevoznika, ki so praviloma neučinkovito izkoriščene, v zamudi. Tako je treba pogosto uporabiti storitve drugih prevoznikov, ki zanje ne ponujajo tako ugodnih pogojev, kar pomeni višje transportne stroške. V izogib tovrstnim negotovostim se podjetja poslužujejo višje stopnje varnostnih zalog, to pa vodi k slabšim poslovnim rezultatom. Vse to so izzivi za skupno upravljanje transporta (angl. Collaborative Transportation Management, v nadaljevanju CTM), kar mora odpraviti takšne težave in tako prispevati k učinkoviti uporabi opreme prevoznikov, posledično nižjim transportnim stroškom, nižjim stroškom zalog, višji stopnji zadovoljstva končnih potrošnikov in nenazadnje boljšim poslovnim rezultatom (White, Browning, 2000).

Model CTM temelji na istih razmerjih med kupci in dobavitelji kot model CPFR. CTM dopolnjuje model CPFR s procesi in informacijami za prevoznike. Nadaljuje tam, kjer se slednji konča, in sicer pri avtomatskem generiranju naročil. Proces razširja s planiranjem, napovedovanjem in izvedbo dobav. Prav tako kot CPFR tudi CTM ne temelji na obstoječih

poslovnih procesih, ampak le-te temeljito prenovi. Tako prevoznik postane partner znotraj oskrbovalne verige. Sodelovanje pri skupnem upravljanju transporta poteka na treh ravneh, in sicer:

- *Strateško planiranje transporta* izhaja iz okvirnih planov prodaje končnih proizvodov in geografske razporeditve transportnih lokacij. Iz tega je moč predvideti obseg pošiljk ter temu primerno planirati prevozne kapacitete.
- *Napovedovanje obsega transporta* temelji na napovedi naročanja znotraj oskrbovalne verige, ki izhaja iz modela CPFPR.
- *Izvedba transporta* pomeni realizacijo napovedi transporta. Le-ta pogosto odstopa od same napovedi, zato so učinkovito izvedbo transporta potrebni dobri komunikacijski procesi in zelo odzivni informacijski sistemi, ki zagotavljajo hitre povratne informacije v proces planiranja. Če ima izbrani prevoznik težave s kapacitetami in zato ni sposoben izpolniti plana, lahko dano izjemo že vnaprej, pred predvidenim rokom dobave, skupno razrešijo vsi partnerji v oskrbovalni verigi.

3.1 Struktura modela CTM

Model CTM je prav tako kot model CPFPR oblikovan tako, da nudi nove možnosti sodelovanja. Kupec in prodajalec skupaj sodelujeta s prevoznikom, da se zagotovi učinkovita izvedba procesov, kot so planiranje dobav, izvedba dobav in plačilo. Model CTM torej vključuje prevoznike v oskrbovalno verigo, in sicer s pomočjo petih poslovnih procesov. Trije od teh so že del modela CPFPR, in sicer skupno planiranje, napoved naročanja in generiranje naročil. Ostala dva procesa sta izvedba prevoza in plačilo prevozniku.

3.1.1 Skupno planiranje transporta

Proces skupnega planiranja transporta je sestavljen iz dveh aktivnosti, in sicer iz oblikovanja sporazuma o sodelovanju med prodajalcem, kupcem in prevoznikom ter izdelavo skupnega poslovnega plana.

Oblikovanje sporazuma o sodelovanju ima enak namen kot pri modelu CPFPR, to je doseči dogovor o sodelovanju. Sporazum določa odnose med kupcem, prodajalcem in prevoznikom. V njem je potrebno opredeliti, kateri proizvodi bodo predmet transporta, transportne poti, točke prevzema in oddaje blaga, oblike transporta, obveznosti iz naslova plačila in zavarovanja tovora (pariteta) ter strategije za upravljanje z izjemami. Na tovrstne odločitve praviloma

vplivajo transportni stroški, povprečni transportni čas, variabilnost transportnega časa in možnost izgub ter nastanka škod.

Izdelava skupnega poslovnega plana izhaja iz plana med prodajalcem in kupcem, kjer so opredeljeni strateški cilji sodelovanja. Na osnovi omenjenih ciljev je mogoče predvideti obseg dobav, to pa prevoznikom omogoča planiranje transportne opreme, vsem partnerjem pa učinkovito upravljanje premoženja (White, Browning, 2000).

3.1.2 Napoved dobav

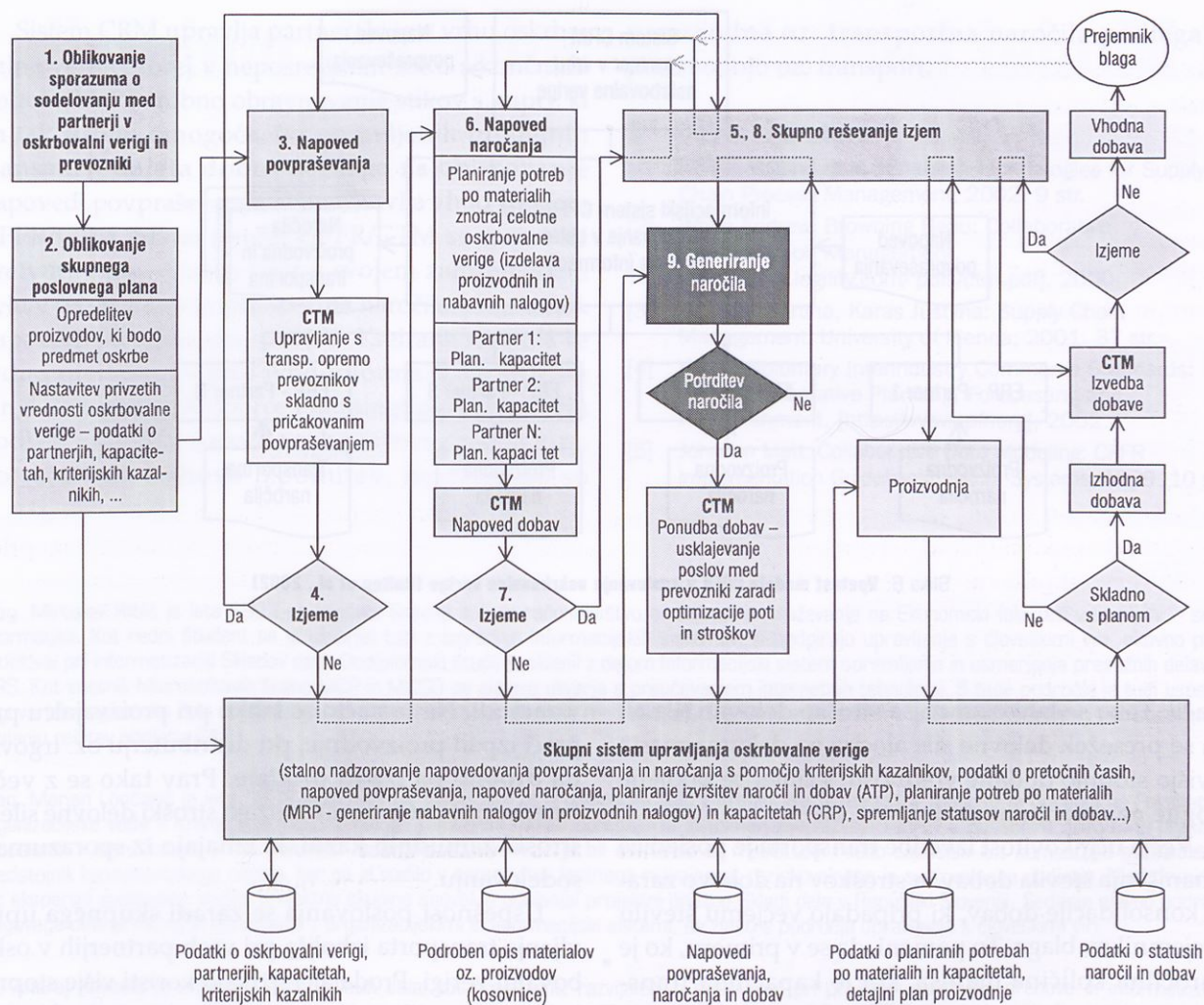
Napoved dobav je v bistvu napoved naročanja, izražena v pošiljkah ali med partnerji v oskrbovalni verigi. Tako lahko prevoznik v skladu z napovedjo dobav upravlja s svojo transportno opremo. Če pri tem pride do kakršnihkoli izjem, ki jih sam prevoznik ne more rešiti, jih razrešijo vsi partnerji skupaj, in sicer tako, da ustrezno prilagodijo roke dobav, količino dobav ali pa zagotovijo nove transportne kapacitete. Tak način poslovanja je za prevoznika nov, saj je ponavadi o potencialnih dobavah obveščen dan ali dva pred nastankom potrebe po dobavi.

3.1.3 Ponudba dobav

Natančno terminiranje dobav izhaja iz zamrznjenega dela napovedi naročanja (iz modela CPFPR), ki je osnova za avtomatsko generiranje naročil. Ko so generirana naročila, so kreirane in terminirane tudi dobave, ki predstavljajo ponudbo za prevoznika oz. prevoznike. Ponudba je izdelana precej pred izvedbo dobav, koliko prej, pa je odvisno od obdobja, ki določa zamrznjeni del napovedi naročanja. Tako ima prevoznik oz. prevozniki dovolj časa za operativno planiranje transportne opreme, ki je potrebna za izvedbo dobav, in sicer tako, da je oprema kar se da učinkovito uporabljena, da se prevozi čim manj »praznih kilometrov« in da so stroški transporta v okviru dogovora, sklenjenega s sporazumom o sodelovanju. Če pri operativnem planiranju transportne opreme prihaja do izjem, npr. nerazpoložljivosti transportne opreme pri določenem prevozniku, se prevozniki s sodelovanjem izogonejo zamudam in stroškom, ki jih te povzročajo znotraj oskrbovalne verige.

3.1.4 Izvedba dobav

Izvedba dobav vključuje končno potrditev naročil za dobavo in izvedbo le-teh. Pri tem je pomembno, da se zagotovi transparentnost statusa dobav. To pomeni,



Slika 5: Integracija modelov CPFR in CTM (White, Browning, 2000)

da se skozi samo izvedbo dobave nenehno posodablja njen status oz. predvideni rok dobave. Če med samo dobavo prihaja do odstopanj od plana (okvara transportne opreme, zamuda), jih morajo partnerji interaktivno razrešiti. Tako partnerji v oskrbovalni verigi lažje predvidijo premike blaga, kar nenazadnje omogoča storitev ATP (angl. Available-To-Promise), ki na podlagi napovedi naročanja in premikov blaga izračuna, kdaj bi lahko bilo realizirano potencialno naročilo oz. kakšen je status naročila.

3.1.5 Plačilo prevozniku

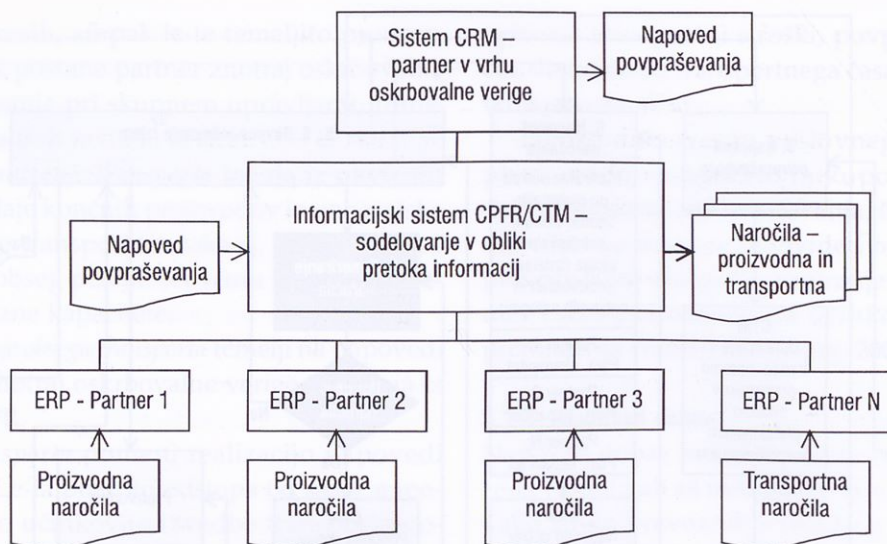
Izstavitve računa in plačilo prevozniku je praviloma avtomatiziran proces, ki ga omogočajo storitve elektronskega bančništva. Znesek je okvirno dogovorjen že v

sporazumu o sodelovanju, in sicer tako, da koristi sodelovanja uživajo vsi partnerji v oskrbovalni verigi. Odstopanja, ki jih partnerji zopet skupaj ovrednotijo, so mogoča v primeru izjem v sami izvedbi dobave.

3.2 Prednosti skupnega upravljanja transporta

Temeljne prednosti skupnega upravljanja transporta blaga se kažejo na naslednjih področjih: učinkovitejši proces upravljanja transporta, učinkovitejša izvedba transporta, transparentnost dobav in nenazadnje uspešnejše poslovanje vseh partnerjev v oskrbovalni verigi (White, Browning, 2000).

Procesna učinkovitost izhaja iz uporabe informacijske tehnologije, ki omogoča avtomatizirano terminiranje dobav, objavo ponudbe dobav ter avtomatiziran proces



Slika 6: Vpetost modela CPFR v poslovanje oskrbovalne verige (Holten et al., 2002)

plačil. Tako se lahko zmanjša stroške delovne sile ali pa se presežek delovne sile alocira na delovna mesta z višjo stopnjo dodane vrednosti, s čimer se podjetje izogne najemanju nove delovne sile.

Večja učinkovitost izvedbe transporta je posledica zmanjšanja števila dobav in stroškov na dobavo zaradi konsolidacije dobav, ki pripadajo večjemu številu prejemnikov blaga. To pomeni, da se v primeru, ko je naročilna količina manjša, kot je kapaciteta transportnega sredstva (angl. LTL – Less-Than-Truck Load), transportno sredstvo naloži še z naročilom drugega prejemnika blaga, dokler to ni polno naloženo (angl. FTL – Full Truck Load), kajti le tako je transportno sredstvo optimalno izkoriščeno. Optimalna konsolidacija dobav in uporaba transportne opreme je mogoča le ob skupnem planiranju transportnih poti, kajti to je edini način, ki zagotavlja opredelitev ekonomsko najkrajših poti, opredelitev terminiranih postankov, krajše čakalne čase in krajše skupne čase transporta ter uporabo transportne opreme v obe smeri vožnje oz. vožnjo s čim manj »praznimi kilometri«.

Transparentnost dobav omogoča na eni strani sodelovanje partnerjev v oskrbovalni verigi, ki si medsebojno izmenjujejo informacije, na drugi strani pa je uporaba informacijske tehnologije, ki omogoča sodelovnje oz. izmenjavo teh informacij v realnem času. Tako si lahko vsi delijo isto informacijo o statusu dobav in se hitro ustrezno odzovejo na dobave, ki so

v zamudi. Na ta način se lahko pri proizvajalcu prepreči izpad proizvodnje, pri distributerju oz. trgovcu na drobno pa izpad prodaje. Prav tako se z večjo transparentnostjo dobav znižajo stroški delovne sile in stroški zamudnih kazni, ki izhajajo iz sporazuma o sodelovanju.

Uspešnost poslovanja se zaradi skupnega upravljanja transporta izboljša pri vseh partnerjih v oskrbovalni verigi. Prodajalci uživajo koristi višje stopnje kakovosti svojih storitev ob nižjih transportnih stroških. Kupcem se zaradi bolj zanesljivih dobav ni več treba bati izpada zalog, zato si lahko privoščijo nižjo stopnjo varnostnih zalog. Prevozniki pa lahko na eni strani zaradi napovedi dobav učinkovito upravljajo s transportnimi sredstvi, na drugi strani pa z objavljeno ponudbo dobav konsolidirajo pošiljke večjega števila prejemnikov blaga in tako bolj učinkovito izkoriščajo svoje kapacitete.

4 Vpetost sistema CPFR/CTM v informacijski sistem oskrbovalne verige

Informacijski sistem CPFR/CTM je sistem, ki združi planske funkcije posameznih partnerjev v skupno planiranje na nivoju celotne oskrbovalne verige. Kakovost planiranja je odvisna od natančnosti napovedi povpraševanja končnih potrošnikov, zato je smiselno povezati sistem CPFR/CTM s sistemom za upravljanje odnosov s kupci (angl. Customer Relationship Management – v nadaljevanju CRM).

Sistem CRM upravlja partner, ki je v vrhu oskrbovalne verige, torej v neposrednem stiku s končnimi potrošniki. Podrobno obravnavanje stikov s kupci, ki ga tak sistem omogoča, ter opravljenih prodajnih transakcij dajeta dobro podlago za oblikovanje napovedi povpraševanja, ki predstavlja vhod v sistem CPFR/CTM. Skozi sistem CPFR/CTM se napoved pretvori v naročila, in sicer v svojem zamrznjenem delu v proizvodna in transportna naročila, preostanek napovedi pa v planska naročila. Kreirana naročila se redno prenašajo v poslovno-informacijske sisteme (angl. Enterprise Resource Planning) partnerjev. Na podlagi planskih naročil vsak partner planira vire, potrebne za njihovo izpolnitev, medtem ko so

proizvodna oz. transportna naročila podlaga za proizvodnjo oz. transport.

5 Viri in Literatura

- [1] Holten Roland et al.: Enabling Technologies for Supply Chain Process Management, 2002. 9 str.
- [2] White Andrew, Browning Beau: Collaborative Transportation Management. [<http://www.logility.com/pdfs/ctm.pdf>], 2000.
- [3] Uchatzy Martina, Karas Justyna: Supply Chain Management. University of Vienna, 2001. 37 str.
- [4] VICS – Voluntary Interindustry Commerce Standards: CPFR Collaborative Planning, Forecasting and Replenishment. [<http://www.cufr.org>], 2002.
- [5] Johnson Matt: Collaborative Data Modeling: CPFR Implementation Guidelines. Syncra Systems, 1999. 10 str.

Mag. Miroslav Ribič je leta 1993 po končani Srednji šoli za računalništvo nadaljeval izobraževanje na Ekonomski fakulteti v Ljubljani – smer informatika. Kot redni študent se je ukvarjal tudi z izgradnjo informacijskih sistemov, ki podpirajo upravljanje s človeškimi viri, aktivno pa je sodeloval pri informatizaciji Skladov dela. Dodiplomski študij je sklenil z delom Informacijski sistem spremljanja in usmerjanja presežnih delavcev v RS. Kot imetnik Microsoftovih licenc MCP in MCSD se aktivno ukvarja s preučevanjem internetnih tehnologij. S tega področja je tudi uspešno zagovarjal magistrsko delo Implementacija elektronskega poslovanja med podjetji. Trenutno je zaposlen v podjetju IDS Scheer, kjer sodeluje pri uvajanju rešitev podjetja SAP.

Mag. Marjan Lončarič je leta 1981 diplomiral na Visoki šoli za organizacijo dela v Kranju. Naziv magistra znanosti je dosegel na Fakulteti za organizacijske vede v Kranju leta 1991 z nalogo Dinamični model spremljanja in usmerjanja razvoja kadrov s posebnim ozirom na presežke zaposlenih v lesni industriji. Na Fakulteti za organizacijske vede je bil od 1985 do 1990 tudi redno zaposlen kot samostojni svetovalec in predstojnik konzultantskega centra, kar ga je vodilo v ustanovitev lastnega razvojnega, raziskovalnega in svetovalnega podjetja. Med drugim je kot ekspertni svetovalec vladne projektne skupine snoval in udeleževal projektni pristop Skladi dela v Republiki Sloveniji. Sedanje glavno področje njegovega dela je razvoj in upravljanje z organizacijskimi in informacijski sistemi, še zlasti s področja upravljanja s človeškimi viri.

Dr. Andrej Kovačič je v zadnjih desetih letih delal kot projektant, razvijalec in svetovalec pri projektih strateške prenove in informatizacije poslovanja. Je izredni profesor s področja poslovne informatike na Ekonomski fakulteti in Fakulteti za upravo ter predstojnik Inštituta za poslovno informatiko pri EF v Ljubljani. Bil je dolgoletni predsednik programskega odbora Dnevnov slovenske informatike v Portorožu, je član izvršnega odbora Slovenskega društva INFORMATIKA, odgovorni urednik revije Uporabna informatika, svetovalec in veččak s področja vodenja in upravljanja podjetij (PHARE, Zveza ekonomistov) in pooblaščen revizor informacijskih sistemov.

Model kontekstno odvisnih mobilnih aplikacij

Rok Rupnik, Marjan Krisper

Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 LJUBLJANA

rok.rupnik@fri.uni-lj.si, marjan.krisper@fri.uni-lj.si

Povzetek

Informacijska družba je družba stalnega napredka in tehnološkega razvoja. Konvergenca med informacijskimi in telekomunikacijskimi tehnologijami je ena njenih rezreprezentativnih značilnosti. Konvergenca različnih tehnologij omogoča nove storitve in mobilne aplikacije so verjetno najbolj znan primer novih storitev, ki so posledica konvergenca. Prispevek v uvodnem delu predstavlja vlogo mobilnih aplikacij v informacijski družbi. V drugem delu prispevek opredeljuje pojem mobilne aplikacije in njeno namembnost. Preostali del prispevka predstavlja klasičen model mobilnih aplikacij, opredelitev pojma konteksta in model kontekstno odvisnih mobilnih aplikacij.

Abstract

Context-aware mobile applications model

Information Society is the society of ongoing progress and technological development. The convergence between several technology sectors offers an opportunity for the emergence and development of new services. Mobile applications are the consequence of convergence between telecommunications and information technology. They are probably the most known service representing the result of convergence. The first part of the paper introduces the role of mobile applications in the information society. In the second part the mobile application model is presented. The final part introduces the classical mobile application model, the definition of context and the context-aware mobile application model.

1 UVOD

V zadnjih desetih letih smo priča konvergenca¹ področij informacijskih tehnologij in telekomunikacijskih tehnologij² [MobiCom 2001; Muller-Veerse 2000]. Konvergenca je posledica tehnološkega razvoja, njeni dosežki pa omogočajo drugačen pristop ter pogled na reševanje problemov in doseganje ciljev. Že samo internet, eden prvih znanilcev konvergenca, je povzročil velike spremembe in do temeljev spremenil marsikaj, kar je v poslovni informatiki veljalo za ustaljeno in, morda, večno.

Najpomembnejša posledica konvergenca je širjenje doslej jasno opredeljenih mej med področjema. Končni rezultat konvergenca bo zaključek širjenja in s tem izginotje mej, kar bo dejansko pomenilo zlitje obeh področij v eno samo. Poslovna informatika danes je že s pojavitvijo interneta nekaj povsem drugega, kar je bila pred leti. Z gotovostjo pa lahko trdimo, da bo čez nekaj let nekaj povsem drugega, kot je danes. Pojavljale se bodo povsem nove storitve in

aplikacije, dvignila se bo tudi kakovostna raven in način uporabe nekaterih obstoječih aplikacij in storitev.

Konvergenca je znanilec informacijske družbe in neposredno vpliva na hitrost vstopanja vanjo. Premiki v okviru konvergenca in njihove posledice bodo brez dvoma vplivali na področje poslovne informatike in družbo nasploh [Bavec 1998]. Zato je naloga informatikov, da konvergenca in premikom v njenem okviru posvetimo vso pozornost. Mobilne aplikacije, ena glavnih posledic in reprezentativnih rezultatov napredka informacijske družbe, jo vsekakor zahtevajo.

1.1 Informacijska družba

Informacijska družba je pojem, ki ga je težko opredeliti s kratko definicijo, zato tudi v delih različnih avtorjev opredelitev in definicij ni zaslediti [Bavec 1996; Groznik 1999; Vehovar 1999]. Bangemann³ je informacijsko

¹ Bavec je v [Bavec 1998] uporabil tudi izraz zblíževanje. V prispevku se uporablja izraz konvergenca, ker gre za konvergiranje k istemu cilju: enotnim digitalnim storitvam. Ravno tako bi lahko uporabili izraz zlivanje ali pa združevanje [Rupnik 2001a].

² Gre za konvergenca informacijskih tehnologij, telekomunikacijskih tehnologij in medijev. Vendar je za potrebe pričujočega prispevka zanimiva konvergenca med prvima dvema.

³ Bangemannovo poročilo [Bangemann 1994] je morda najpomembnejši dokument Evropske unije za področje informacijske družbe. Cilj poročila je podati priporočila Evropskemu svetu za oblikovanje politike za hiter in neboleč prehod v informacijsko družbo.

družbo opredelil kot revolucijo, ki temelji na informaciji kot odrazu človeškega znanja in v okviru katere dosežki tehnološkega napredka omogočajo procesiranje, shranjevanje, pridobivanje in prenos informacije v različnih oblikah (pogovor, tekst ali vizualne oblike) brez omejitev razdalje, časa in količine [Bangemann 1994]. Informacijsko družbo bi lahko opredelili na naslednji način:

Informacijska družba je izrazito storitveno naravnana družba, kjer uspešnost tako posameznika kot tudi organizacije temelji na sposobnosti pridobiti čimveč različnih podatkov ob pravem času, iz njih hitro izluščiti pravilno informacijo in na njeni podlagi učinkovito ukrepati.

Ugotovimo lahko, da še ni povsem jasno, ali v informacijski družbi že smo, ali se še vedno nahajamo na prehodu v informacijsko družbo. Informacijska družba je nastala oz. nastaja kot posledica tehnološkega razvoja, vendar to še zdaleč ne pomeni, da je informacijska družba zgolj tehnološki pojem [Bavec 1996]. Informacijska družba je kompleksen, izrazito multidisciplinaren pojem, ki presega tehnološke okvire. Informacijska družba je družba znanja in inovacij, zato je neprestani razvoj nujen za njen obstoj [Anttiroiko 2001]. Še več. Rečemo lahko, da so znanje, inovacije in razvoj temelj in gonilna sila informacijske družbe. Ko govorimo o razvoju, nimamo v mislih le tehnološkega razvoja, temveč predvsem razvoj storitev. Prav razvoj storitev, tehnološki razvoj in vzpodbujanje razvojno-raziskovalne dejavnosti ter tehnoloških inovacij so glavne usmeritve Evropske unije za informacijsko družbo [Bavec 1996; eEurope 2002].

Komunikacijski sistemi in omrežja so velikega pomena za delovanje informacijske družbe, saj predstavljajo fizično podlago delovanju storitev [Bangemann 1994; Gams 1999]. Za razvoj informacijske družbe pa je pomembna tudi interoperabilnost med omrežji različnih tehnologij (mobilno omrežje, internet, fiksno omrežje, Bluetooth omrežje), saj omogoča interoperabilnost med storitvami in aplikacijami različnih omrežij. Prav zaradi tega sta GPRS in UMTS tako pomembna za razvoj informacijske družbe [Muller-Veerse 2001].

1.2 Mobilnost: pridobitev in zahteva informacijske družbe

Informacijsko družbo je, kot smo že ugotovili, težko opredeliti in jo natančno definirati. Po drugi strani pa lahko ugotovimo, da je relativno preprosto podati

nekatero njene lastnosti, nekatere zakonitosti in dejstva, ki v njej veljajo. Različni avtorji so v svojih prispevkih skušali ugotoviti nekaj teh lastnosti, zakonitosti in dejstev. Izmed številnih navajamo tiste, ki nakazujejo pomen in vlogo mobilnih aplikacij v informacijski družbi:

- **Metcalfov zakon**, ki pravi, da je koristnost omrežja proporcionalna številu naprav, priključenih v omrežje [Gams 1999]. Število mobilnih naprav je bistveno večje od števila osebnih računalnikov priključenih v internet, kar po Metcalfovem zakonu nakazuje na višjo stopnjo koristnosti, ki jo Internetu prinašajo vanj priključene mobilne naprave.
- **Informacijska družba je družba, v kateri vlada zasičenost z informacijami**, kar vodi v daljši delovni čas, preobremenjenost in manj kakovostno socialno življenje [ibidem]. Uporaba mobilnih naprav omogoča opravljanje nekaterih vrst del tudi v stanju mobilnosti in s tem več prostega časa, manjšo obremenjenost in kakovostnejše socialno življenje.
- **Informacijska družba zahteva dostop do kakovostnih informacij za uspešno vodenje v podjetjih in organizacijah v vsakem trenutku** [ibidem]. Uporaba mobilnih aplikacij za podporo odločanju za vodstvene delavce v podjetjih in organizacijah lahko zato le prispeva h kakovosti vodenja.
- **Zaposleni v podjetjih in organizacijah so izpostavljeni zahtevam po večji produktivnosti**. Zmožnost pridobivanja informacij v stanju mobilnosti lahko le pozitivno vpliva na dvig produktivnosti zaposlenih [UMTS 2000; Agrawal 1999].
- **Pojavlja se vse več oblik in načinov dela, ki bodisi direktno bodisi indirektno zahtevajo mobilni dostop do podatkov in informacij**. Vloga in pomen mobilnih aplikacij pri tovrstnih oblikah in načinih dela sta očitna [UMTS 2000; Tapscott 2000].

Vidimo torej, da so mobilnost ter mobilne aplikacije in storitve tehnološko gledano pridobitev informacijske družbe, način življenja v njej pa jih po drugi strani zahteva [Rupnik 2001c]. Zato o upravičenosti raziskovanja področja mobilnosti, mobilnih aplikacij in storitev ne more biti nobenega dvoma.

1.3 Mobilne naprave in njihove omejitve

V okviru pričujočega prispevka pojem mobilne naprave pokriva vsako napravo, ki [Rupnik 2001a]:

- je v internet povezana prek brezžične povezave in pri tem uporablja enega od standardnih protokolov⁴,
- se v internet poveže sama, ne prek druge naprave.

Zadnja zahteva je potrebna zaradi izločitve notesnika kot mobilne naprave, saj notesnik kot poseben primer osebnega računalnika ne odraža posebnosti mobilnih aplikacij, ki se izvajajo na ostalih mobilnih napravah. Ugotovimo lahko, da predstavlja pri ostalih zvrsteh mobilnih naprav (mobilni telefoni, komunikatorji in dlančniki) njihova velikost omejitev, ki pogojuje vse ostale omejitve mobilnih naprav in posledično tudi omejitve mobilnih aplikacij. Zato je za potrebe pričujočega prispevka notesnik kot mobilna naprava izločen. Enako stališče je v svojem prispevku zavzel tudi Tarasewich [Tarasewich 2002].

Mobilne naprave imajo zaradi svojih posebnih lastnosti, predvsem velikosti oz. majhnosti, določene omejitve. Prva in največja je velikost (majhnost) mobilnih naprav, ki pogojuje vse ostale omejitve. Ostale omejitve mobilnih naprav so predvsem naslednje: relativno malo zmogljivi procesorji, relativno majhen pomnilnik razpoložljiv za aplikacije, velikost zaslona, način povezljivosti v internet, nizka hitrost prenosa podatkov in težaven vnos podatkov [Rupnik 2001a; Agrawal 1999; Sharma 2001; Figge 2001; Spriestersbach 2001].

Mobilne naprave in njihove zmožnosti včasih zato neupravičeno primerjamo z osebnimi računalniki in njihovimi zmožnostmi, na kar je v svojem prispevku še posebej opozoril Spriestersbach [Spriestersbach 2001].

2 MOBILNA APLIKACIJA

Področje mobilnih aplikacij je kot raziskovalno področje razmeroma novo. Znanstvene literature s področja mobilnih aplikacij je relativno malo, opaziti pa je porast števila prispevkov, ki se v večji ali manjši meri dotikajo tega področja. Ugotovimo lahko, da je področje mobilnih aplikacij tipičen (in ne edini) primer, kjer stroka in tehnološki napredek prehitevata znanost [Rupnik 2001b; Hars 2000]. Nekateri premiki v zadnjem času pa kažejo, da se znanost zaostanka zaveda in že ukrepa. Pričakujemo lahko, da bo začela dohitevati stroko.

Mobilna aplikacija je program, ki deluje na mobilni napravi; njegovo izvajanje je porazdeljeno prek več nivojev v okviru večnivojske arhitekture, ki je najprimernejša arhitektura za mobilne aplikacije [Rupnik 2002]. Razlog primernosti večnivojske arhitekture najdemo v poprej omenjenih omejitvah relativno majhnega pomnilnika in relativno slabo zmogljivih procesorjev, saj porazdelitev izvajanja prek več nivojev razbremeni mobilno napravo. Prispevek obravnava mobilne aplikacije poslovnega področja v okviru informacijskih sistemov.

2.1 Nekateri posebnosti mobilnih aplikacij

Prej omenjene omejitve mobilnih naprav narekujejo inovativne pristope na različnih področjih, predvsem na področju vmesnikov. Za področje vnosa in razpoznavanja enostavnih odzivov uporabnika je možno uporabiti na področju aplikacij doslej povsem nove elemente [Rupnik 2002; Bergeron 2001; Sacher 2001]:

- razpoznavanje govora za potrebe govornih ukazov,
- razpoznavanje in detekcija gest ter gibov za potrebe potrjevanja in
- razpoznavanje pisanja kot nadomestilo za vnos prek tipkovnice.

Ugotovimo lahko, da gre pri navedenih elementih po eni strani za odpravljanje že navedenih omejitev in s tem pomanjkljivosti mobilnih naprav, po drugi strani pa gre za pridobitve in povsem nove možnosti, ki se bodo morda razširile tudi na druge zvrsti aplikacij. Posebej zanimivi sta razpoznavanje in detekcija gest in gibov, ki bi tudi pri uporabi aplikacij klasičnih zvrsti uporabnikom omogočila večje udobje pri uporabi.

2.2 Namembnost mobilnih aplikacij

Namen mobilnih aplikacij je uporabniku v stanju mobilnosti omogočiti uporabo aplikacij, ki mu bodo omogočale dostop do zelenih podatkov, prejetje obvestil in uporabo ostalih preprostejših interaktivnih aplikacij [Rupnik 2001a; Bergeron 2001; Rettie 2001]. Osnovni namen mobilnih aplikacij ni omogočiti enak nabor funkcionalnosti kot ga ponujajo druge zvrsti⁵ aplikacij, temveč ob mobilnosti omogočiti zmanjšan in prilagojen nabor funkcionalnosti, ki je pogojen z omejitvami mobilnih naprav. Rekli bi lahko, da je namen mobilnih aplikacij omogočiti mobilnosti prilagojen in

⁴ Mobilna naprava ni direktno povezana v internet, temveč bodisi preko mobilnega omrežja, bodisi preko Bluetooth omrežja.

⁵ V prispevku je izraz zvrst uporabljen za najvišji nivo delitve aplikacij: mobilne, spletne, odjemalec strežnik. Ugotovimo lahko, da gre za tehnološko pogojeno delitev na zvrsti. V okviru vsake od zvrsti pa lahko nadalje govorimo o tipih aplikacij.

mobilnosti primeren nabor funkcionalnosti. Problem tipičnega poslovnega mobilnega uporabnika je namreč potreba po čim boljšem izkoristku časa [Rettie 2001; Buellingen 2001]. Mobilnost in možnost takojšnjega dostopa do podatkov odtehtata manj udobno uporabo mobilnih aplikacij, kar je posledica omejitev mobilnih naprav [Driver 2001; Egan 2000].

Mobilna aplikacija predstavlja drugačen koncept od ostalih zvrsti aplikacij, prikaz česar je eden glavnih namenov pričujočega prispevka. Koncept mobilne aplikacije ne pomeni samo prenos obstoječih aplikacij na manjše odjemalce – mobilne naprave [Egan 2000]. Naj ne bo odveč, če na tem mestu ponovno omenimo Spristersbachov prispevek, v katerem je poleg že navedenega zavzel tudi stališče, da je primerjava mobilnih aplikacij s klasičnimi neupravičena [Spristersbach 2001].

3 MODEL KONTEKSTNO ODVISNIH MOBILNIH APLIKACIJ

Namembnost mobilnih aplikacij in omejitve, ki jih mobilnim aplikacijam narekujejo omejitve mobilnih naprav, predstavljajo zadosten razlog za ločeno obravnavanje mobilnih aplikacij in iskanje posebnosti v mobilnih aplikacijah kot modelu aplikacij [Hampe 2000; Rupnik 2002]. Hampe je v svojem prispevku ugotovil, da je pri razvoju mobilnih aplikacij potrebno imeti mobilnost ves čas pred očmi in razvijati mobilnosti ustrezne mobilne aplikacije [Hampe 2000]. Po njegovem mnenju je to edini način za uveljavitev mobilnih aplikacij, ne glede na njihov tip. Razvoj mobilnih aplikacij, ki bi bile le poenostavitve oz. bi predstavljale le mobilne variante obstoječih aplikacij, ne vodi v uspeh na tem področju [ibidem]. Mobilne aplikacije morajo uporabniku prinesiti novo, inovativno vrednost glede na ostale zvrsti aplikacij [Carlsson 2001].

Izhodišč za razmišljanje o tem, kako se mobilne aplikacije kot nova zvrst aplikacij razlikujejo od ostalih zvrsti aplikacij, je več:

- **Omejitve mobilnih naprav**, ki implicitno določajo tudi omejitve in posebnosti mobilnih aplikacij.
- **Mobilnost sama**. Uporabnik ima v stanju mobilnosti specifične potrebe, omejitve mobilnih naprav ga omejujejo pri uporabi mobilnih aplikacij in zaradi stanja mobilnosti ima uporabnik omejene možnosti uporabe aplikacij.
- **Seznam funkcij**, ki jih opravljajo aplikativni sistemi. Aplikativni sistemi opravljajo naslednje funkcije: zajem, hranjenje, iskanje, transformiranje in

prikazovanje [Alter 1999]. Funkcije nakazujejo tako na potrebo po posebnih, novih funkcijah za mobilne aplikacije, kot tudi na posebno in prilagojeno obravnavanje in implementacijo navedenih funkcij.

3.1 Klasični model mobilnih aplikacij

Prvi in osnovni model mobilnih aplikacij je klasični model. Njegova glavna značilnost je, da uporabnik sam zažene oz. aktivira mobilno aplikacijo v izbranem trenutku. Literatura za tak model uporablja splošen izraz *pull model*. Dejstvo, da je uporabnik zagnal aplikacijo, nakazuje na njegove trenutne informacijske potrebe. V dosedanem poteku prispevka je bilo sicer predstavljeno stališče, da razvoj aplikacij, mobilnih variant klasičnega modela obstoječih aplikacij, ne vodi v uspeh in njihovo uveljavitev [Hampe 2000]. Vendar lahko ugotovimo, da se klasičnemu modelu ne moremo povsem izogniti, kar menijo tudi nekateri drugi avtorji [Carlsson 2001; Tarasewich 2002]. Možnost zagona in uporabe mobilne aplikacije v izbranem trenutku, kar velja za aplikacije klasične zvrsti, še vedno predstavlja dodano vrednost za mobilnega uporabnika.

3.2 Kontekst in kontekstno odvisne mobilne aplikacije

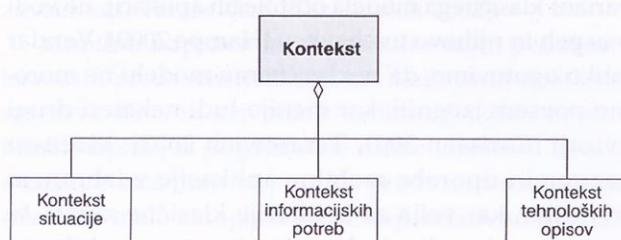
3.2.1 Opredelitev pojma konteksta

Pojem konteksta je nov pojem, ki ga mobilne aplikacije vpeljujejo v informacijski sistem. Schilit je kontekst opredelil kot kombinacijo odvisnosti od konteksta situacije, v kateri se nahaja mobilni uporabnik, ter odvisnosti od opisov objektov, kjer so objekti lahko mobilne naprave ter lokacije, v katerih se nahaja uporabnik [Schilit 1995]. Podobno je kontekst opredelil Dey, ki ga je opredelil kot kakršno koli informacijo, ki jo lahko uporabimo za opis situacije entitete, kjer je entiteta lahko oseba, prostor, lokacija, objekt in naprava [Dey 2000a]. Dey je kontekstno odvisno aplikacijo opredelil kot aplikacijo, ki uporablja kontekst pri oblikovanju vsebine aplikacije [Dey 2000b].

Pomanjkljivost pravkar navedenih opredelitev konteksta je v pomanjkanju upoštevanja informacijskih potreb mobilnega uporabnika. Kontekst lahko po našem mnenju postane koncept na področju informacijskih sistemov le v primeru obravnavanja informacijskih potreb mobilnega uporabnika [Rupnik 2002]. Zato je po našem mnenju treba opredelitev konteksta razširiti s kontekstom informacijskih potreb. Poleg

opisa situacije, v kateri se nahaja uporabnik, mora opredelitev konteksta obsegati tudi uporabnikove informacijske potrebe, ki jih določajo potrebe informacijskega sistema, katerega akter je mobilni uporabnik. Potrebe informacijskega sistema lahko v danem trenutku narekujejo, da je treba prek mobilne aplikacije v poslovni proces v skladu s pristojnostmi, ki jim ima v okviru poslovnih procesov, vključiti določenega mobilnega uporabnika. Ko govorimo o potrebah, imamo torej v mislih potrebe vezane na informacijsko podporo pri izvajanju aktivnosti v okviru poslovnih procesov.

Na podlagi predstavljene problematike lahko podamo naslednjo opredelitev konteksta (slika 1).



Slika 1: Osnovni metamodel strukture konteksta

Kontekst je celovita informacija o:

- situaciji, v kateri se nahaja mobilni uporabnik. To opredelimo kot kontekst situacije.
- njegovih informacijskih potrebah oz. potrebah informacijskega sistema, katerega akter je. To opredelimo kot kontekst informacijskih potreb.
- objektih, lokacijah in mobilnih napravah, ki so del logičnega virtualnega prostora uporabnika. To opredelimo kot kontekst tehnoloških opisov.

3.2.2 Model kontekstno odvisnih mobilnih aplikacij

Kontekstno odvisne mobilne aplikacije delujejo kontekstno odvisno [Rupnik 2002; Rupnik 2003]. Njihovo delovanje je odvisno od konteksta: od informacijskih potreb uporabnika, situacije, v kateri se nahaja, in tehnoloških opisov objektov in naprav v logičnem virtualnem prostoru mobilnega uporabnika.

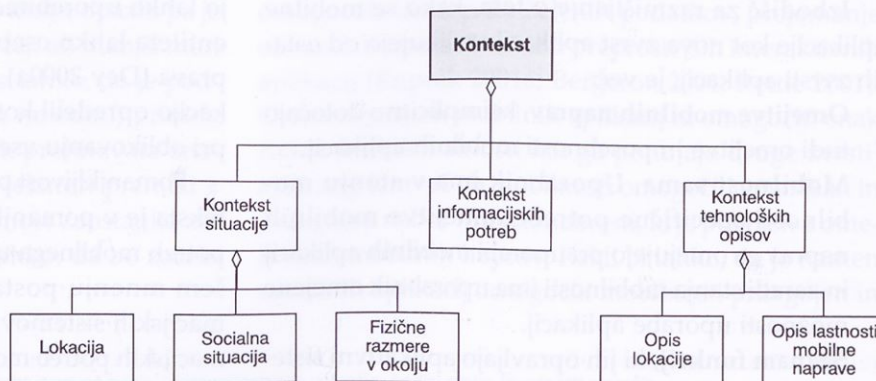
Kontekst določa vsebino aplikacije, določa pa tudi potrebo po sprožitvi oz. trenutek zagona aplikacije. Vsebina aplikacije je bodisi informacija, ki jo aplikacija posreduje, bodisi kontekstu prilagojena (menijska) struktura aplikacije in s tem hierarhično urejen seznam funkcionalnosti, ki jih v danem trenutku potrebuje mobilni uporabnik.

Trenutek proženja aplikacije oz. njenega zagona opredeljujejo pravila za proženje [Rupnik 2002]. Pravila za proženje so poslovna pravila oz. posledica poslovnih pravil, ki veljajo v poslovnem sistemu in jih implementira informacijski sistem [Bajec 2001]. Pravilo za proženje lahko, na primer, opredeljuje, da je potrebno vodstvenemu delavcu posredovati (zagnati) mobilno aplikacijo v primeru, ko eden od kazalnikov poslovanja pade/zraste pod/nad določeno mejo, mobilna aplikacija pa bi vodstvenemu delavcu prikazala vrednost kazalnikov poslovanja in morebitne ostale potrebne podatke ter komentarje.

3.2.3 Model delovanja kontekstno odvisnih mobilnih aplikacij

Za opis delovanja modela kontekstno odvisnih mobilnih aplikacij je bil izbran koncept meta-modela z notacijo razrednih diagramov jezika UML [Booch 1999]. Metamodel je koncept, s pomočjo katerega predstavimo problemsko področje na konceptualni ravni. V njem nastopajo tako koncepti in gradniki iz problemskega področja, kot tudi povezave in razmerja med njimi. Na metamodel lahko gledamo tudi kot na miselni vzorec, ki opisuje problemsko področje, v tem primeru model delovanja kontekstno odvisnih mobilnih aplikacij [Bajec 2001].

Slika 2 nadgrajuje predhodno sliko in prikazuje kontekst kot strukturo v razširjeni obliki. Kontekst situacije predstavlja stanje trenutne situacije mobilnega



Slika 2: Metamodel strukture konteksta

uporabnika, kot jo je opredelil Schilit, in je določena z lokacijo, socialno situacijo in fizičnimi razmerami v okolju. Koncept tehnoloških opisov predstavlja opise in podatke o lokacijah ter opis lastnosti in podatke o mobilnih napravah [Spriestersbach 2001]. Schilit v svoji disertaciji opozarja tudi na pomen opisov različnih objektov in poznavanja podatkov o tem. Za lokacije tipa stavba, na primer, predlaga celo opis celotne strukture: nadstropja, sobe, hodniki... Poudarja pa tudi pomen poznavanja lastnosti mobilne naprave, kar aplikacijskemu strežniku omogoča, da mobilni napravi posreduje njej prilagojen prikaz vsebin [Schilit 1995]. Tudi Tarasewich je opozoril na pomen poznavanja lastnosti in možnosti mobilne naprave ter prilagoditev prikaza vsebine njenim možnostim [Tarasewich 2002]. Kontekst informacijskih potreb pa predstavlja trenutne informacijske potrebe mobilnega uporabnika.

Model na sliki 2 bi lahko označili kot neke vrste hibridni model. Prikazan je v notaciji razrednih diagramov jezika UML in nakazuje strukturo prek razmerij agregacije. Zaradi večje informativnosti pa je oblikovan oz. prikazan kot dekompozicijski diagram, kar zopet prikazuje strukturo konteksta. Tak način prikaza nakazuje na hibridnost predstavitve obravnavanega modela oz. na hibridnost modela samega.

Vidimo, da gre pri kontekstu na eni strani za statične opise na fizičnem nivoju, kar je v strukturi konteksta (slika 2) predstavljeno z opisom lokacije. Statičnost se kaže v majhni dinamiki sprememb, ki nastopa pri opisih objektov. Na drugi strani pa gre za stalno spremljanje konteksta trenutne situacije, v kateri se nahaja mobilni uporabnik in njegovih informacijskih potreb, kar opredeljuje semantični nivo poznavanja konteksta in njegov dinamični nivo. O fizičnem in semantičnem nivoju poznavanja konteksta je govoril tudi Spriestersbach [Spriestersbach 2001]. Lahko bi govorili tudi o statični in dinamični komponenti konteksta.

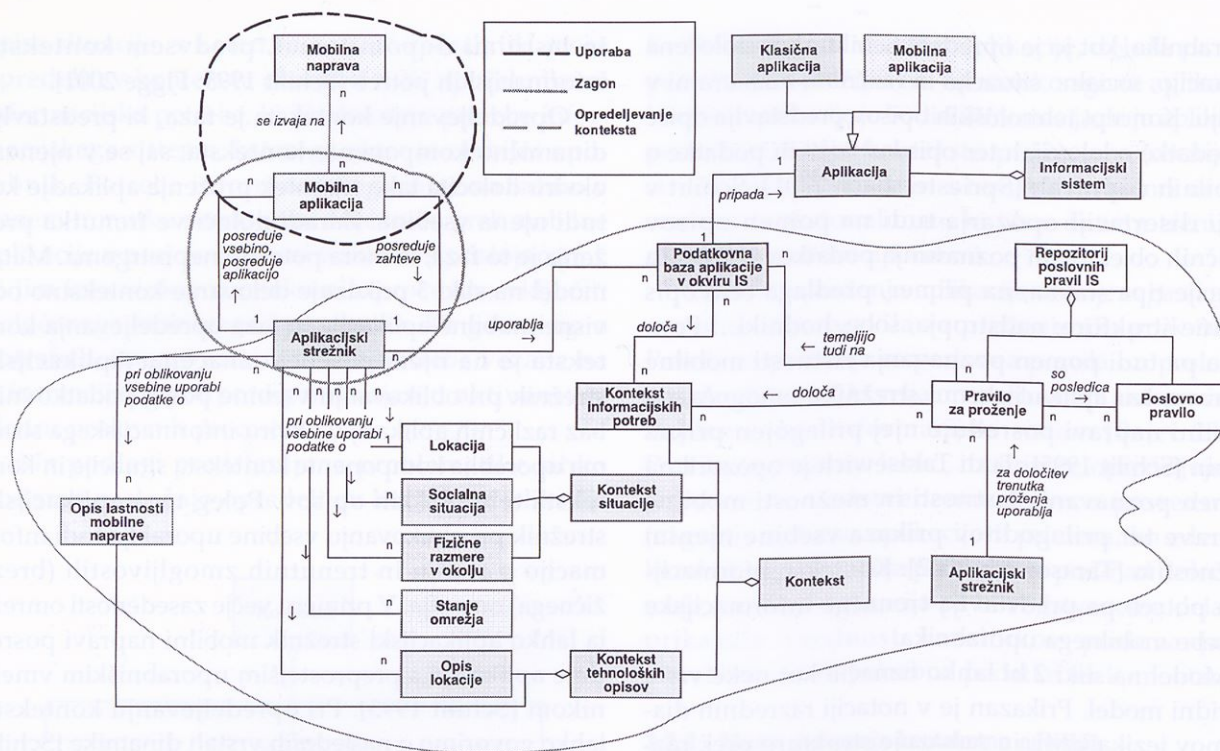
Delovanje kontekstno odvisne mobilne aplikacije je razdeljeno v tri faze. Opredeljevanje konteksta je prva faza in centralni del v delovanju kontekstno odvisnih mobilnih aplikacij, saj kontekst določa tako vsebino kot tudi trenutek sprožitve mobilne aplikacije. Predstavljeno opredeljevanje konteksta temelji na Schilitovem opredeljevanju konteksta, opredeljevanju situacijske odvisnosti iz Fiegejeve raziskave

in lastnimi dopolnitvami, predvsem kontekstu informacijskih potreb [Schilit 1995; Figge 2001].

Opredeljevanje konteksta je faza, ki predstavlja dinamično komponento konteksta, saj se v njenem okviru določita tako trenutek proženja aplikacije kot tudi njena vsebina. Zaradi določitve trenutka proženja je to faza, ki mora potekati nepretrgoma. Meta-model na sliki 3 prikazuje delovanje kontekstno odvisne mobilne aplikacije in faza opredeljevanja konteksta je na njem posebej označena. Aplikacijski strežnik pri oblikovanju vsebine poleg podatkovnih baz različnih aplikacij v okviru informacijskega sistema uporablja komponente konteksta situacije in konteksta tehnoloških opisov. Poleg tega aplikacijski strežnik pri oblikovanju vsebine uporablja tudi informacijo o stanju in trenutnih zmožljivostih (brežžičnega) omrežja. V primeru večje zasedenosti omrežja lahko aplikacijski strežnik mobilni napravi posreduje aplikacijo s preprostejšim uporabniškim vmesnikom [Schilit 1995]. Pri opredeljevanju konteksta lahko govorimo o naslednjih vrstah dinamike [Schilit 1995; Rupnik 2002]:

- **dinamika komunikacij:** aplikacijski strežnik pri oblikovanju vsebine uporabi podatke o stanju omrežja,
- **dinamika okolja:** aplikacijski strežnik pri oblikovanju vsebine uporabi informacijo o socialni situaciji in fizičnih razmerah v okolju,
- **dinamika lokacije:** aplikacijski strežnik pri oblikovanju vsebine uporabi informacijo o trenutni lokaciji uporabnika,
- **dinamika informacijskih potreb:** določitev trenutka proženja aplikacije temelji na informacijskih potrebah uporabnika, ki jih določajo informacijske potrebe informacijskega sistema.

V okviru opredeljevanja konteksta določajo pravila za proženje in podatkovne baze različnih informacijskih sistemov kontekst informacijskih potreb (slika 3). Aplikacijski strežnik določi trenutek proženja mobilne aplikacije na podlagi pravil za proženje, ki temeljijo tudi na stanju podatkov v podatkovnih bazah različnih aplikacij v okviru informacijskega sistema. V okviru faze zagona se mobilna aplikacija posreduje mobilnemu uporabniku na mobilno napravo, faza uporabe predstavlja uporabo mobilne aplikacije s strani mobilnega uporabnika.



Slika 3: Metamodel kontekstno odvisnih mobilnih aplikacij*

3.2.4 Tehnološka platforma kontekstno odvisnih mobilnih aplikacij

Iz predstavitve modela kontekstno odvisnih mobilnih aplikacij je razvidno, da je potreben tehnološki pogoj za uvedbo kontekstno odvisnih mobilnih aplikacij stalna, on-line povezljivost mobilnih naprav v mobilno omrežje izpolnjujeta tako GPRS kot tudi UMTS [Agrawal 1999]. Najpomembnejši člen v verigi potrebnih pogojev za možnost njihove uvedbe v informacijske sisteme so operacijski sistemi mobilnih naprav, strežniki in izvajalna okolja, ki omogočajo posredovanje mobilne aplikacije mobilni napravi prek mobilnega omrežja [Microsoft; Sun; Enum].

4 SKLEP

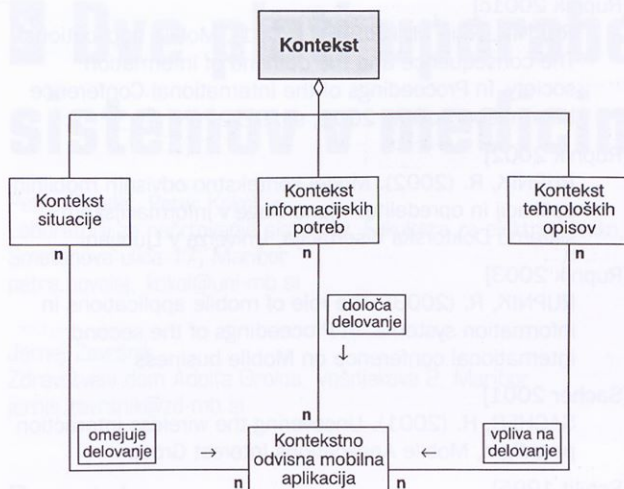
Kontekst je relativno nov koncept, ki ga v informacijski sistem uvajajo mobilne aplikacije. Aplikacije klasične zvrsti koncept konteksta zaobidejo, saj gre zaradi načina njihovega delovanja in uporabe v tradicionalnem delovnem okolju za odvečen koncept. Nekoliko grobo bi lahko rekli, da pride pri aplikacijah klasične zvrsti uporabnik v kontekst aplikacije, med-

tem ko pride pri mobilnih aplikacijah aplikacija v kontekst uporabnika [Rupnik 2002].

V prispevku sta predstavljena klasični model mobilnih aplikacij in model kontekstno odvisnih mobilnih aplikacij; vsak na svoj način predstavlja dodano vrednost za mobilnega uporabnika. Mobilno aplikacijo klasičnega modela mobilni uporabnik zažene v izbranem trenutku, s čimer izrazi svoje trenutne informacijske potrebe, kot jih zazna sam. Kontekstno odvisna mobilna aplikacija pa se zažene sama, na njen zagon vpliva predvsem kontekst informacijskih potreb, kar pomeni, da se kontekstno odvisna mobilna aplikacija zažene v skladu z informacijskimi potrebami informacijskega sistema kot celote v skladu s pristojnostmi mobilnega uporabnika. Kontekst informacijskih potreb določa, kontekst situacije omejuje, kontekst tehnoloških opisov pa vpliva na delovanje kontekstno odvisne mobilne aplikacije (slika 4).

Kontekstno odvisne mobilne aplikacije zahtevajo kontekstno delovanje informacijskega sistema kot celote [Schilit 1995]. Informacijski sistem mora v vsakem trenutku poznati lokacije svojih akterjev,

* Aplikacijski strežnik in mobilna aplikacija se na modelu nahajata dvakrat le zaradi večje preglednosti.



Slika 4: Osnovni metamodel delovanja kontekstno odvisnih mobilnih aplikacij

poznati mora informacijske potrebe sistema kot celote in na podlagi postavljenih pravil in pristojnosti mobilne uporabnike po potrebi vključevati v procese, ki se odvijajo v delovnih okoljih, ki jih informacijski sistem informacijsko podpira.

5 LITERATURA

- [Agrawal 1999]
AGRAWAL, P. in D. FAMOLARI (2000). Mobile Computing in Next Generation Wireless Networks. In Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications, str. 32-39
- [Alter 1999]
ALTER, S. (1999). Information Systems: A Management Perspective, Addison-Wesley
- [Anttiroiko 2001]
ANTTIROIKO, A. V. (2001). Toward the European Information Society. Communications of the ACM, 44(1), str. 31-35
- [Bajec 2001]
BAJEC, M. (2001). Opredelitev izhodišč za celovito obvladovanje poslovnih pravil v organizacijah, Doktorska disertacija, Univerza v Ljubljani
- [Bangemann 1994]
BANGEMANN, M. (1994). Europe and the global information society: recommendations to the European Council, Poročilo - študija Evropske komisije
- [Bavec 1996]
BAVEC, C. (1996). Informacijska družba, Slovenija, Evropski pilotski projekti. Zbornik posvetovanja Dnevi slovenske informatike 1996, str. 536-543
- [Bavec 1998]
BAVEC, C. (1998). Evropska komisija o zblizevanju digitalnih tehnologij. Uporabna informatika, 6(1), str. 8-14
- [Bergeron 2001]
BERGERON, B. (2001). The Wireless Web, McGraw-Hill
- [Booch 1999]
BOOCH, G., J. RUMBAUGH in I. JACOBSON (1999). The Unified Modeling Language User Guide, Addison-Wesley
- [Buellingen 2001]
BUELLINGEN, F. in M. WOERTER (2001). Development perspectives, firm strategies and applications in mobile commerce. In Proceedings of COTIM conference: From e-Commerce to m-Commerce
- [Carlsson 2001]
CARLSSON, C. (2001). Mobile Commerce: Core Issues, products and services. In Proceedings of 14th International Bled Electronic Commerce Conference, str. 45-48
- [Dey 2000a]
DEY, A. K. (2000). Enabling the Use of Context in Interactive Applications. In Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), str. 79-80
- [Dey 2000b]
DEY, A. K. (2000). Providing Architectural Support for Building Context-Aware Applications, Doktorska disertacija, Georgia Institute of Technology
- [Driver 2001]
DRIVER, M. (2000). Portable application development, GartnerGroup Symposium Itxpo 2000
- [eEurope 2002]
EVROPSKA KOMISIJA (2002). eEurope: An information society for all - Progress report, Poročilo - študija Evropske komisije
- [Egan 2000]
EGAN, B. in S. HAYWARD (2000). Wireless Web Architecture and infrastructure, GartnerGroup Symposium Itxpo 2000
- [Enum]
<http://www.enum.org>
- [Figge 2001]
FIGGE, S. (2001). Situation dependent m-commerce applications. In Proceedings of COTIM conference: From e-Commerce to m-Commerce
- [Gams 1999]
GAMS, M. (1999). Information Society Promotes Intelligent Systems. In Proceedings of the International Conference Information Society 1999, str. 1-4
- [Groznik 1999]
GROZNIK, A. in A. KOVAČIČ (1999). Slovenska pot v informacijsko družbo. Uporabna informatika, 6(1), str. 5-8
- [Hampe 2000]
HAMPE, J. F., P. M. C. SWATMAN in P. A. SWATMAN (2000). Mobile Electronic Commerce: Reintermediation in the Payment System. In Proceedings of 13th International Bled Electronic Commerce Conference, str. 693-706

- [Hars 2000]
HARS, A., O. A. E. SAWY, S. GOSAIN, S. HIRT, I. IM, D. KANG, Z. LEE in A. RAVEN (2000). Reengineering IS Research and its Intellectual Infrastructure for the Electronic Economy. *Journal of Organizational Computing and Electronic Commerce*, 10(2), str. 67—83
- [Krisper 2000]
KRISPER, M., R. RUPNIK, M. BAJEC, I. ROZMAN, M. HERIČKO, T. DOMANJKO, M. B. JURIČ, S. BELOGLAVEC, M. KOŽMAN, M. STANTIČ, S. RUBIN, R. TOMAŽIČ, R. JENSTERLE, M. SILIČ, M. COLNAR in A. NOVAKOVIČ (2000). Strateško planiranje, EMRIS — Enotna metodologija razvoja informacijskih sistemov, Center vlade za informatiko
- [Microsoft]
On any device: .NET goes mobile with the .NET compact framework, <http://www.microsoft.com/presspass/features/2001/oct01/10-23netframework.asp>
- [MobiCom 2001]
Projekt MobiCom (2001). Evolution Scenarios for emerging m-commerce services: new policy, market dynamics, methods of work and business models, European Mobile Commerce Survey
- [Muller-Veerse 2000]
MULLER-VEERSE, F. (1000). Mobile Commerce Report, Durlacher Research, Raziskovalna študija
- [Muller-Veerse 2001]
MULLER-VEERSE, F., B. KOHLENBACH, D. BOUT, S. SINGH in G. GOLUB (2001). UMTS Report: An Investment Perspective, Durlacher research, Eqvitec and Helsinki University of Technology, Raziskovalna študija
- [Rupnik 2001a]
RUPNIK, R. (2001). Vloga mobilnih aplikacij v informacijskem sistemu. *Uporabna informatika*, 9(1), str. 24-31
- [Rupnik 2001b]
RUPNIK, R. in M. KRISPER (2001). Mobile commerce: Can it prosper without mobile paradigm? In Proceedings of COTIM conference: From e-Commerce to m-Commerce
- [Rupnik 2001c]
RUPNIK, R. in M. KRISPER (2001). Mobile applications: The consequence and the demand of information society. In Proceedings of the International Conference Information Society 2001, str. 9-12
- [Rupnik 2002]
RUPNIK, R. (2002). Model kontekstno odvisnih mobilnih aplikacij in opredelitev njene vloge v informacijskem sistemu Doktorska disertacija, Univerza v Ljubljani
- [Rupnik 2003]
RUPNIK, R. (2003). The role of mobile applications in information systems. In Proceedings of the second international conference on Mobile business
- [Sacher 2001]
SACHER, H. (2001). Uncovering the wireless interaction paradigm, Mobile Applications Interest Group
- [Schilit 1995]
SCHILIT, W. N. (1995). A System Architecture for Context-Aware Mobile Computing, Doktorska disertacija, Columbia University
- [Priestersbach 2001]
SPRITERSBACH, J. (2001). Integrating Context Information into Enterprise Applications for the Mobile Workforce — A Case Study. In Proceedings of Workshop of Mobile Commerce at the International Conference on Mobile Computing and Networking, str. 55—59
- [Sun]
Over-the-air provisioning with the J2ME wireless toolkit, <http://wireless.java.sun.com/midp/ttips/wtkota/>
- [Tarasewich 2002]
TARASEWICH, P., R. C. NICKERSON, in M. WARKENTIN (2002). Issues in Mobile e-Commerce. Communications of the Association for Information Systems (spletna publikacija)
- [UMTS 2000]
UMTS Forum (2000). Report 8: The Future Mobile Market, Poročilo

Rok Rupnik je zaposlen kot asistent na Fakulteti za računalništvo in informatiko v Ljubljani, kjer je leta 1998 magistriral in leta 2002 doktoriral. Njegovo raziskovalno področje obsega metodologije razvoja informacijskih sistemov, širše področje elektronskega poslovanja, strateško planiranje informacijskih sistemov ter mobilne aplikacije in mobilno poslovanje. V svoji karieri je sodeloval pri več razvojnih projektih informacijskih sistemov, na večini od njih v vlogi koordinatorja in vodje razvoja. Med drugim je tudi soavtor in urednik Enotne metodologije razvoja informacijskih sistemov. Je član Slovenskega društva INFORMATIKA, združenja AIS (Association for Information Systems) in ustanovitveni član slovenske sekcije PMI (Project Management Institute).

Marjan Krisper je zaposlen kot docent na Fakulteti za računalništvo in informatiko v Ljubljani. Njegovo raziskovalno področje obsega metodologije razvoja informacijskih sistemov, področje elektronskega poslovanja, strateško planiranje informacijskih sistemov, agilne metodologije, poslovna pravila in področje prenove poslovnih procesov. V svoji karieri je bil vodja več razvojnih projektih informacijskih sistemov in projektov izdelave strateških planov informacijskih sistemov. Med drugim je tudi nosilec ideje, soavtor in urednik Enotne metodologije razvoja informacijskih sistemov. Je član Slovenskega društva INFORMATIKA, združenja AIS (Association for Information Systems) in ustanovitveni član slovenske sekcije PMI (Project Management Institute).

☒ Dve plati uporabe inteligentnih sistemov v medicini

Petra Povalej, Peter Kokol

Laboratorij za načrtovanje sistemov, Fakulteta za elektrotehniko, računalništvo in informatiko, CIMRŠ, Univerza v Mariboru
Smetanova ulica 17, Maribor
petra.povalej, kokol@uni-mb.si

Jernej Završnik

Zdravstveni dom Adolfa Drolca, Vošnjakova 2, Maribor
jerne.j.zavrsnik@zd-mb.si

Povzetek

Inteligentni sistemi se vse pogosteje uporabljajo na različnih področjih človekovega delovanja (npr. ekonomija, medicina, gospodarstvo ipd.), kjer se pogosto izkažejo kot zelo uporabna orodja. Uporabljamo jih za klasifikacijo, diagnosticiranje, predikcijo, iskanje povezav med različnimi dejavniki, odkrivanje novega znanja ipd. Na učinkovitost uporabe inteligentnih sistemov vpliva množica različnih dejavnikov (npr.: kvaliteta baze podatkov, relacije med vhodnimi in izhodnimi atributi, izbira metodologije inteligentnega sistema, vrsta in oblika podatkov ipd.), na katere imamo lahko v celotnem procesu inteligentne analize večji oz. manjši vpliv. V pričujočem prispevku smo se osredotočili na zanesljivost inteligentnih sistemov na realnih primerih s področja odločanja in diagnosticiranja, ki pokažeta prednosti in slabosti takšne analize. V procesu iskanja najkvalitetnejše rešitve smo preizkusili mnogo različnih pristopov gradnje klasifikacijskega modela (klasični, hibridni in multimetodni). Kljub vložnemu trudu se je iskanje kvalitetnega klasifikacijskega modela za neinvazivno določanje ravni holesterola v krvi izkazalo za neuspešno, saj zbrani parametri pri zbiranju podatkov (vhodni atributi) očitno niso dajali zadostne informacije o ravni holesterola v krvi. Nasprotno pa se je uporaba inteligentnih sistemov za pomoč pri diagnosticiranju prolapsa mitralne valvule (PMV) izkazala kot zelo učinkovita in zanesljiva.

Abstract

Two Sights of Using Intelligent Systems in Medicine

Intelligent systems have been often successfully applied on various fields such as business, economy, medicine, etc. They have been used for classification, diagnosing, prediction, knowledge discovery, etc. The efficiency of intelligent systems analysis very much depends on a type, format and quality of data gathered. However, the existence of appropriate relationships among attributes and outcomes and of course the selection of the right methodology is also crucial. In this paper we focused on the reliability of intelligent systems applied on a real-world problems where the benefits and drawbacks of such an analysis can be seen. With the aim of finding the most reliable solutions, different approaches for classification model construction were used. Nonetheless, the construction of reliable classification model for non-invasive determination of a blood cholesterol level proved to be unsuccessful mostly because of the lack of relationships among the attributes and the decisions. On the other hand, the use of intelligent systems analysis was very successful and reliable for diagnosing mitral valvule prolapse.

1 Uvod

Inteligentni sistemi so se izkazali kot zelo uporabno orodje na različnih področjih človekovega udejstvovanja, še posebno pa se širi njihova uporaba v medicini. Pogosto se uporabljajo za pomoč pri odločanju, klasifikaciji, predikciji, iskanju vzorcev, odkrivanju novih znanj, inteligentni analizi podatkov ter optimizaciji.

Zavedati se moramo, da na učinkovitost uporabe inteligentnih sistemov na realnih bazah podatkov močno vpliva vrsta različnih dejavnikov, kot na primer: kvaliteta baze podatkov, relacije med vhod-

nimi in izhodnimi atributi, oblika in vrsta podatkov ter izbira ustrezne metodologije gradnje inteligentnih sistemov. Kvalitetna priprava podatkov (izbira ustreznih parametrov) ter sam postopek zbiranja podatkov (natančnost pri merjenju, natančnost pri vnašanju podatkov ipd.) sta pogosto že odločilnega pomena za nadaljnji potek in uspešnost inteligentne analize. Prav ti dejavniki so ponavadi "krivi" za uspeh oz. neuspeh uporabe inteligentnih sistemov na realnih bazah podatkov.

V pričujočem prispevku se bomo osredotočili na oblikovanje uporabniku prijaznega klasifikacijskega modela, ki temelji na odločitvenih drevesih. Z izbiro predstavitve izluščenega znanja se bomo le delno omejili, saj bomo za gradnjo odločitvenih dreves uporabili več različnih pristopov (klasični, hibridni, multimetodni).

Predstavili bomo primer uporabe inteligentnih sistemov za pomoč pri določanju ravni holesterola v krvi, ki se je izkazal kot neuspešen predvsem zaradi slabe korelacije med vhodnimi atributi in odločitvenimi razredi ter relativno slabe kvalitete baze podatkov. V drugem primeru bomo prikazali zelo učinkovito uporabo različnih metodologij inteligentnih sistemov za oblikovanje klasifikacijskega modela za pomoč pri diagnosticiranju prolapsa mitralne valvule (PMV).

2 Inteligentni sistemi v medicini

Najpomembnejša lastnost inteligentnih sistemov je njihova zmožnost učenja ter sprejemanja odločitev na podlagi informacij iz preteklosti. Učenje vedno poteka na že zbranih podatkih, ki jih razdelimo na učno in testno množico. Inteligentni sistem mora poiskati pravila s procesom strojnega učenja na učni množici ter nato pravila preveriti s klasifikacijsko natančnostjo na neznanih podatkih iz testne množice. To je bistvenega pomena, saj poskušamo na takšen način doseči splošno uporabnost dobljenega modela na novih, še neznanih realnih primerkih.

Obstaja več metodologij inteligentnih sistemov, od katerih so najbolj znane nevronske mreže, odločitvena drevesa, genetski algoritmi, grobe množice ter hibridni sistemi, ki predstavljajo več (ponavadi dva) pristopov združenih v enega. S hibridnim pristopom poskušamo združiti prednosti, ki so značilne za posamezne pristope ter na takšen način doseči bolj kvalitetne rezultate. V literaturi [1] pogosto zasledimo hibride genetskih algoritmov in odločitvenih dreves [4], nevronskih mrež in odločitvenih dreves, grobih množic in nevronskih mrež [13] ipd.

Vsak posamezen pristop izgradnje ima določene prednosti in slabosti in prav zaradi tega moramo izbrati metodologije prilagoditi problemu ter zahtevam, ki jih moramo zadovoljiti pri reševanju le-tega. Nekatere metodologije, na primer odločitvena drevesa, nam dajejo dobro sliko glede odločitve, medtem ko druge metodologije, na primer nevronske mreže, ne dajejo pojasnil o sprejeti odločitvi.

Pogosto je zelo pomembno, da so odločitve, ki jih predlaga inteligentni sistem, razumljive ljudem, ki stojijo za procesom. Razumljivost inteligentnih sistemov je še posebno pomembna v medicini. Vsak zdravnik namreč odgovarja za svoje odločitve (npr. diagnozo), zato je s tega vidika zelo pomembno, da razume, kateri dejavniki so pripeljali do dane odločitve. Inteligentni sistem v tem primeru zdravniku predstavlja orodje za pomoč pri odločanju.

Razumljiva in enostavna predstavitev klasifikacijskega modela je pomembna tudi za nadzor delovanja modela. V primeru, da začne klasifikacijski model v nekem trenutku producirati napačne odločitve, je možno odkriti in analizirati vzroke za napačno delovanje le, če uporabljamo dobro razumljiv klasifikacijski model. V nasprotnem primeru, če klasifikacijski model deluje kot »črna skrinjica« (npr. nevronske mreže), je zelo težko odkriti, kaj je privedlo do napačnega obnašanja.

V medicini se inteligentni sistemi pogosto uporabljajo tudi za potrjevanje znanih dejstev in odkrivanje novega znanja (oz. novih – še ne znanih povezav med danimi atributi in diagnozo). Glavni problem, ki se pojavlja v tem primeru, je ocena kvalitete novega znanja. Novo znanje se lahko izkaže le kot posledica prekomerne prilagoditve klasifikacijskega modela učnim objektom in tako predstavlja nesmiselne povezave. S tega vidika je zelo pomembno sodelovanje s strokovnjakom z danega področja (npr. kardiolog, pediater ipd.), ki oceni kvaliteto novo odkritega znanja.

2.1 Izbira metodologije in učenje

Izbira metodologije inteligentnih sistemov je v veliki meri odvisna od zastavljenega problema. Razumljivost zgrajenega klasifikacijskega modela je v našem primeru ključnega pomena pri izbiri metodologije. Glavna prednost odločitvenih dreves [7] pred ostalimi metodologijami je njihova enostavnost in razumljivost, kar pojasnjuje pogosto uporabo odločitvenih dreves tudi v medicini. Njihova prednost je tudi, da jih je prav tako možno uporabiti brez računalnika, saj preprosto sledimo hierarhični strukturi znanja.

Zavedati se moramo, da vsestransko uporabna metodologija inteligentnih sistemov, ki daje v vseh primerih najkvalitetnejše rezultate, ne obstaja. Zato je zelo pomembno, da se pri reševanju problema ne omejimo na eno samo metodologijo, temveč z eksperimentiranjem najdemo najboljšo. V nadaljevanju

bomo na kratko predstavili osnovne razlike med metodologijami, ki smo jih uporabili v raziskavi.

a) Klasična metodologija gradnje odločitvenih dreves

V procesu klasične gradnje odločitvenih dreves je izbira najprimernejšega atributa v posameznem vozlišču odločitvenega drevesa odvisna od informacijske vsebnosti atributa, ki se oceni na osnovi hevristične funkcije. Najpogosteje uporabljene hevristične funkcije za oceno informacijske vsebnosti atributa, ki smo jih uporabili tudi v naši raziskavi, so naslednje:

- stopnja informacijskega prirastka (*Information gain ratio*) [8], ki izhaja iz entropije (*ID3*),
- χ^2 (*Chi-square*) [1],
- Gini [10],
- J-measure [9].

Poleg zgoraj naštetih osnovnih hevrističnih funkcij smo generirali tudi različne linearne kombinacije [6] le-teh ter njihovo učinkovitost preizkusili na obeh bazah podatkov. Kvaliteto izgrajenih odločitvenih dreves smo poskušali izboljšati z uporabo boostinga – metode za izboljšavo natančnosti poljubnega algoritma za učenje [2].

b) Gradnja odločitvenih dreves z genetskimi algoritmi – hibridni pristop

Gradnja odločitvenih dreves z genetskimi algoritmi je hibridni pristop, ki združuje prednosti genetskih algoritmov (npr.: neobčutljivost na šum, večja verjetnost za pridobitev optimalne rešitve) in prednosti odločitvenih dreves (enostavna in razumljiva predstavitev rezultatov). Gradnja odločitvenega drevesa poteka podobno kot na klasičen način, z razliko, da se v procesu iskanja optimalnega odločitvenega drevesa uporabi evolucijski pristop.

c) Multimetodni pristop

V primerjavi s klasičnimi hibridnimi pristopi multimetodni pristop [3] omogoča uporabo različnih metodologij v enem samem modelu znanja, kjer vsaka izmed metod odraža lastne prednosti in tudi omejitve. Posledično torej pričakujemo, da takšna kombinacija različnih metod v večini primerov pripelje do boljših rezultatov.

2.2 Ocena kvalitete zgrajenega modela

Univerzalen kriterij za oceno kakovosti zgrajenega klasifikacijskega modela na žalost ne obstaja. Pogosto se uporabljajo indikatorji, kot so skupna natančnost,

senzitivnost, specifičnost. V predstavljeni raziskavi smo se odločili za uporabo naslednjih indikatorjev:

- skupna natančnost = $\frac{\text{število pravilno klasificiranih objektov}}{\text{število vseh objektov}}$;
(*average total accuracy*)
- povprečna natančnost odločitvenih razredov
(*average class accuracy*).

Povprečna skupna natančnost odločitvenega drevesa nam pogosto ne daje realne informacije o dejanski kvaliteti odločitvenega drevesa, saj se v večji meri srečujemo s problemi, pri katerih je klasifikacija objektov določenega odločitvenega razreda zelo težka. Tako lahko naučeno odločitveno drevo klasificira testne objekte z visoko skupno natančnostjo, vendar pa je natančnost klasifikacije določenega odločitvenega razreda zelo nizka. Takšno odločitveno drevo je v praksi neuporabno, saj se je specializiralo za klasifikacijo samo določenih odločitev (ne pa vseh). Skupna natančnost odločitvenega drevesa torej ni zadosten indikator kvalitete naučenega drevesa. Mnogo več nam o kvaliteti odločitvenega drevesa pove povprečna natančnost odločitvenih razredov.

Pri reševanju medicinskih problemov za evalvacijo kvalitete odločitvenih dreves pogosto uporabljamo izraza senzitivnost in specifičnost. Senzitivnost označujemo kot odstotek pravilno klasificiranih zdravih, specifičnost pa kot odstotek pravilno klasificiranih bolnih subjektov:

- senzitivnost = $\frac{tp}{tp + fn}$; (*sensitivity*)
- specifičnost = $\frac{tn}{tn + fp}$; (*specificity*)

Oznake, uporabljene v zgornjih enačbah, imajo naslednji pomen:

- tp = True Positive = število pravilno klasificiranih pozitivnih objektov
- fp = False Positive = število napačno klasificiranih pozitivnih objektov
- fn = False Negative = število napačno klasificiranih negativnih objektov
- tn = True Negative = število pravilno klasificiranih negativnih objektov

3 Dve plati uporabe inteligentnih sistemov v medicini

Uspešnost aplikacije inteligentnih sistemov za reševanje realnih problemov je odvisna od mnogih dejavnikov. V prvi fazi lahko trdimo, da je učinkovitost

inteligentnega sistema v veliki meri odvisna od kvalitete in vrste zbranih podatkov. Pri uporabi realnih baz podatkov se pogosto soočimo z naslednjimi težavami:

- šum v podatkih – pojavlja se zaradi nenatančnosti meritev, napak pri vnašanju podatkov, napačnih podatkov ter manjkajočih podatkov,
- majhne baze podatkov – za slovenske razmere je že baza s podatki o nekaj tisoč pacientih redka, vendar pa včasih tudi nekaj tisoč primerkov v bazi podatkov ni dovolj za izgradnjo učinkovitega modela,
- neenakomerna zastopanost posameznih odločitvenih razredov – posledično lahko pride do prekomernega prilagajanja klasifikacijskega modela bolj zastopani odločitvi,
- varnost podatkov – podatki o bolnikih morajo biti varovani, zato pogosto medicinske ustanove niso pripravljene teh podatkov posredovati.

Poleg kvalitete dane baze podatkov je vsekakor zelo pomembna tudi ustrezna relacija med vhodnimi atributi in odločitvami. V primeru, da te relacije ni ali pa je zelo šibka, seveda ne moremo pričakovati, da bo inteligentni sistem to relacijo našel.

Seveda na učinkovitost inteligentnega sistema ne vpliva samo baza podatkov, temveč je le-ta v veliki meri odvisna tudi od izbire ustrezne metodologije. Že osnovna praksa nam pokaže, da so določene metodologije bolj oz. manj uspešne na različnih bazah podatkov. Prav zaradi tega je potrebno v procesu iskanja najbolj kvalitetne rešitve danega problema preizkusiti čim več različnih metodologij.

V nadaljevanju bomo predstavili primera aplikacije inteligentnih sistemov na dveh realnih medicinskih bazah podatkov.

PRIMER I:

Gradnja modela za pomoč pri določanju ravni holesterola v krvi

V današnjem času se vedno pogosteje srečujemo s problemom previsoke vrednosti maščob v krvi, predvsem pri odraslih. Zadnje študije kažejo, da tudi otroci zaužijejo preveliko količino maščob v vsakodnevni prehrani, kar jim daje slabo predispozicijo za prihodnost. S tega vidika je še posebej pomembno, da že v zgodnjih letih odkrijemo povečane vrednosti maščob v krvi otrok ter tako zmanjšamo verjetnost pojava s tem povezanih težav v odrasli dobi. Ker se otroci bojijo invazivnih raziskav (npr. klasičnega odvzema krvi), smo želeli v raziskavi najti enostavno

izmerljive kazalce nepravilnih vrednosti holesterolov. S pomočjo inteligentnih orodij smo poskušali poiskati korelacije med danimi atributi in vrednostjo maščob v krvi ter možnost neinvazivnega napovedovanja previsokih/prenizkih vrednosti. V ta namen smo skupaj z zdravstvenim domom Adolfa Drolca izvedli raziskavo, ki jo podajamo v pričujočem prispevku.

Opis baze podatkov

Podatki, ki so bili uporabljeni za gradnjo klasifikacijskega modela za določanje ravni holesterola v krvi, so podatki o 712 petletnih otrocih, ki so bili zbrani v Zdravstvenem domu Adolfa Drolca v Mariboru. 539 otrok je imelo pri pregledu normalno raven holesterola v krvi, 173 otrok pa nenormalno raven. Atributi, ki so služili kot osnova za izgradnjo klasifikacijskega modela, so dejansko podatki, zbrani z enostavnimi meritvami (npr: obseg glave, obseg nadlahtnice, kožna guba, krvni tlak ipd.), ki jih izvedejo pri standardnem sistematskem pregledu otrok.

Uspešnost gradnje klasifikacijskega modela je precej odvisna tudi od razdelitve osnovne baze podatkov na učno in testno množico. Zaradi primerljivosti rezultatov, pridobljenih z različnimi metodologijami, smo v vseh primerih uporabljali isto učno in testno množico. Pri generiranju učne množice smo bili pozorni na čim bolj enakomerno porazdelitev učnih objektov glede na odločitvena razreda (normalen/nenormalen holesterol).

Analiza rezultatov

Podatki, ki so bili uporabljeni za gradnjo klasifikacijskega modela za določanje ravni holesterola v krvi, so zgodovinski podatki o 712 petletnih otrocih, od katerih je 539 otrok imelo normalno raven holesterola v krvi in 173 nenormalno raven. Baza podatkov je torej razmeroma majhna, poleg tega pa je tudi porazdelitev objektov glede na odločitveni atribut zelo neenakomerna.

Rezultati gradnje odločitvenih dreves z vsemi zgoraj opisanimi metodologijami so prikazani v tabeli 1. Metode so razvrščene glede na povprečno natančnost odločitvenih razredov zgrajenega klasifikacijskega modela, ki se giblje v mejah od 48.57% (Boost Greedy J measure + Chi square) do 64.84% (Boost Greedy Gini * Chi square). Povprečna skupna natančnost pa se giblje med 41.91% (Greedy J measure * Gini, Greedy J measure * ID3) in 94.31% (Multimethod). Opazimo lahko precej veliko razliko v kvaliteti posameznih metodologij v povprečni

natančnosti odločitvenih razredov in še mnogo večjo razliko pri upoštevanju povprečne skupne natančnosti zgrajenih klasifikacijskih modelov.

Opazili smo, da se pogosto pojavlja problem klasifikacije objektov z nenormalno ravniho holesterola (nizka senzitivnost). Takšna drevesa so v praksi neuporabna, saj je natančnost klasificiranja z vsemi metodami precej nizka. Zaradi zelo slabega osnovnega klasifikatorja tudi boosting v večini primerov ni bil uspešen pri izboljšavi natančnosti letega. Kljub uporabi različnih metodologij torej nismo uspeli generirati klasifikacijskega modela, ki bi bil dovolj zanesljiv za uporabo v praksi. Vzroke za neuspešnost lahko delno iščemo v sami bazi podatkov (premajhno število vnosov, premalo otrok z nenormalno ravniho holesterola, šum v podatkih), poleg tega pa je povezava med vhodnimi atributi in ravniho holesterola verjetno prešibka, da bi lahko dokaj zanesljivo skleпали o odločitvi.

Tabela 1:
Določanje ravni holesterola v krvi; skupna natančnost (\square) in povprečna natančnost odločitvenih razredov (Δ) na testni množici ter rang metode glede na povprečno natančnost odločitvenih razredov

Method	\square (%)	Δ (%)	Rank
Greedy ID3	53.29	57.92	17
Boost Greedy ID3	54.79	51.30	34
Greedy Gini	61.97	57.58	19
Boost Greedy Gini	51.79	52.18	33
Greedy Chi square	57.48	55.20	24
Boost Greedy Chi square	64.97	54.22	27
Greedy J measure	48.20	55.22	23
Boost Greedy J measure	54.49	61.03	9
Greedy Gini + ID3	55.98	56.88	21
Boost Greedy Gini + ID3	60.18	64.05	2
Greedy Gini * ID3	53.59	60.55	10
Boost Greedy Gini * ID3	60.18	54.16	28
Greedy Chi square + ID3	61.97	60.05	13
Boost Greedy Chi square + ID3	64.37	53.90	31
Greedy Chi square * ID3	60.18	54.16	28
Boost Greedy Chi square * ID3	61.67	59.89	14
Greedy Gini + Chi square	60.47	56.79	22
Boost Greedy Gini + Chi square	62.57	57.90	18
Greedy Gini * Chi square	60.18	54.16	28
Boost Greedy Gini * Chi square	61.67	64.84	1
Greedy J measure + ID3	51.19	54.34	26

Boost Greedy J measure + ID3	66.16	62.28	4
Greedy J measure * ID3	41.91	61.78	5
Boost Greedy J measure * ID3	60.18	64.05	2
Greedy J measure + Chi square	59.28	61.10	7
Boost Greedy J measure + Chi square	58.98	48.57	38
Greedy J measure * Chi square	51.49	54.50	25
Boost Greedy J measure * Chi square	59.28	61.10	8
Greedy J measure + Gini	52.39	50.03	35
Boost Greedy J measure + Gini	61.37	59.74	15
Greedy J measure * Gini	41.91	61.78	5
Boost Greedy J measure * Gini	59.28	53.68	32
Greedy linear	47.60	57.38	20
Greedy Boost linear	58.08	60.46	11
Greedy Voting	52.09	49.87	37
Genetic	79.69	59.40	16
Multimethod	94.31	50.00	36
Nested Boost	62.27	60.21	12

PRIMER II:

Gradnja modela za pomoč pri diagnosticiranju prolapsa

Prolaps mitralne valvule (PMV) je v svoji definiciji opisan kot odmik srčne zaklopke iz normalne pozicije. Posledice tega odmika so motnje v pretoku krvi (vrtnici), neobičajne bolečine v prsih, možnost nastajanja krvnih strdkov in celo smrt. Simptome PMV je mogoče najti pri petih do desetih odstotkih prebivalstva. Pri milejših oblikah PMV zdravljenje ni potrebno, medtem ko pri težjih oblikah PMV pride v poštev tudi kirurški poseg. Ker je pri postavljanju diagnoze poleg naprave za pregled z ultrazvokom potrebno še precej znanja in izkušenj, so se pojavile težnje po postavitvi novih kriterijev za odkrivanje PMV. Ideja je najti bolezn, simptome in sindrome, povezane s PMV, ki bi jih lahko razpoznal splošni zdravnik in tako odkril potencialne paciente s PMV. Cilj podane raziskave je bil odkriti sindrome PMV-ja pri osebah med 1. in 18. letom.

Opis baze podatkov

Z uporabo Monte Carlo metode vzorčenja je bilo izbranih 900 otrok in mladostnikov starih od 1 do 18 let. Na poziv k rutinskemu pregledu se jih je odzvalo 631, katerih zdravstveno stanje so ocenili na osnovi posebej pripravljenega protokola za sindrom PMV-ja. Protokol je obravnaval 103 parametre, za katere je verjetno, da označujejo prisotnost PMV-ja. Porazdelitev

glede na odločitvene razrede je bila naslednja: 5% "prolaps", 6% "tihi prolaps" in 89% "ni prolapsa". Baza podatkov ni vsebovala manjkajočih vrednosti.

Analiza rezultatov

Učno množico je sestavljalo 501 oseb, v testni množici pa je bilo 130 subjektov. Vsak subjekt je bil opisan s 102 parametroma, ki so opisovali pacientove bolezni, bolezni njegovih staršev, parametre preiskav z ultrazvokom itd. Baze nismo razdelili sami v razmerju 2/3 za učno množico in 1/3 za testno množico, temveč smo zaradi primerljivosti uporabili že obstoječo učno in testno množico. Relativne frekvence posameznih odločitev so bile približno enake v učni in testni množici.

Rezultati klasifikacije so predstavljeni v tabeli 2. Metode v tabeli so razvrščene glede na povprečno natančnost odločitvenih razredov inducirane klasifikacijskega modela. Opazimo lahko, da se povprečna natančnost odločitvenih razredov giblje od 40.55% (Boost Greedy J measure * Gini, Boost Greedy J measure + Gini) do 84.31% (Multimethod). Povprečna skupna natančnost pa se giblje v mejah od 80.77% (Greedy J measure * Gini) do 93.84% (Multimethod). Oba indikatorja kvalitete klasifikacijskega modela torej kažeta na to, da je multimetodni pristop na tej bazi prekosil vse ostale preizkušene metodologije.

Tabela 2:
PMV; skupna natančnost (□) in povprečna natančnost odločitvenih razredov (Δ) na testni množici ter rang metode glede na povprečno natančnost odločitvenih razredov

Method	□ (%)	Δ (%)	Rank
Greedy ID3	88.46	73.39	12
Boost Greedy ID3	91.54	70.94	13
Greedy Gini	88.46	77.87	9
Boost Greedy Gini	87.69	64.15	20
Greedy Chi square	90.00	82.91	5
Boost Greedy Chi square	90.77	57.21	24
Greedy J measure	83.08	50.84	32
Boost Greedy J measure	89.23	53.08	29
Greedy Gini + ID3	92.31	79.27	6
Boost Greedy Gini + ID3	90.77	69.75	16
Greedy Gini * ID3	90.00	78.43	7
Boost Greedy Gini * ID3	92.31	57.77	23
Greedy Chi square + ID3	91.54	83.47	2
Boost Greedy Chi square + ID3	89.23	56.65	25

Greedy Chi square * ID3	90.77	70.66	14
Boost Greedy Chi square * ID3	93.08	70.59	15
Greedy Gini + Chi square	91.54	83.47	2
Boost Greedy Gini + Chi square	89.23	56.65	25
Greedy Gini * Chi square	87.69	69.54	17
Boost Greedy Gini * Chi square	90.00	61.42	22
Greedy J measure + ID3	86.15	51.96	30
Boost Greedy J measure + ID3	89.23	48.60	35
Greedy J measure * ID3	81.54	50.28	33
Boost Greedy J measure * ID3	90.77	53.64	28
Greedy J measure + Chi square	88.46	74.30	11
Boost Greedy J measure + Chi square	91.54	66.46	18
Greedy J measure * Chi square	83.85	55.60	27
Boost Greedy J measure * Chi square	87.69	48.95	34
Greedy J measure + Gini	85.39	51.68	31
Boost Greedy J measure + Gini	89.23	40.55	37
Greedy J measure * Gini	80.77	41.95	36
Boost Greedy J measure * Gini	89.23	40.55	37
Greedy linear	91.54	83.47	2
Greedy Boost linear	91.54	74.51	10
Greedy Voting	90.00	78.43	7
Genetic	92.31	65.83	19
Multimethod	93.84	84.31	1
Nested Boost	93.08	62.54	21

4 SKLEP

Učinkovitost uporabe inteligentnih sistemov pri reševanju realnih problemov je v veliki meri odvisna od same kvalitete baze podatkov ter izbire ustrezne metodologije inteligentnega sistema.

V pričujočem prispevku smo predstavili primer učinkovite in primer neučinkovite aplikacije inteligentnih sistemov na realnih medicinskih bazah podatkov. Uporabili smo več metodologij inteligentnih sistemov, ki temeljijo na različnih pristopih gradnje odločitvenih dreves (klasični, hibridni in multimetodni pristop). Pokazali smo, da je pri iskanju najkvalitetnejše rešitve potrebno preizkusiti več različnih metodologij, saj ena sama univerzalna metodologija ne obstaja. Na danih realnih problemih smo prikazali uspešnost oz. neuspešnost inteligentnih sistemov.

Uporaba inteligentnih sistemov za določanje ravni holesterola v krvi pri petletnih otrocih na osnovi neinvazivnih indikatorjev se je izkazala za neuspešno v

vseh primerih, saj na osnovi dane baze podatkov kljub uporabi različnih metodologij ni bilo mogoče zgraditi kvalitetnega klasifikacijskega modela. Vzroke lahko iščemo predvsem v nezadostni korelaciji med vhodnimi atributi in odločitvami.

V nasprotju z neuspešnostjo prvega primera pa je bila kvaliteta klasifikacijskega modela za pomoč pri diagnosticiranju prolapsa (PMV) zelo visoka. Med vsemi preizkušenimi metodologijami se je kot najuspešnejši izkazal multimetodni pristop, s katerim smo uspeli generirati klasifikacijski model s 84,31% povprečno natančnostjo odločitvenih razredov na testni množici. Visoka natančnost klasifikacije omogoča aplikacijo klasifikacijskega modela tudi v praksi, kot eno izmed metod za pomoč pri diagnosticiranju PMV.

Razlike v kvaliteti, na osnovi različnih metodologij zgrajenih klasifikacijskih modelov, so bile v obeh primerih zelo visoke, kar ponovno kaže v prid sicer časovno potratne uporabe različnih metodologij v procesu iskanja kvalitetnih rešitev.

VIRI IN LITERATURA

- [1] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984.
- [2] Freund, Y., Schapire, R. E., Experiments with a new boosting algorithm, V: Proceedings Thirteenth International Conference on Machine Learning, Morgan Kaufman, San Francisco, 1996, 148–156.
- [3] Lenič, M., Kokol, P., Combining Classifiers with Multimethod Approach, V: Soft computing systems: design, management and applications, IOS Press, 2002, 374–383.
- [4] Podgorelec, V., Kokol, P., Induction of medical decision trees with genetic algorithms. V: Proceedings of the international ICSC congress on Computational intelligence methods and applications, 1999.
- [5] Podgorelec, V., Kokol, P., Yamamoto, R., Masuda, G., Sakamoto, N., Knowledge discovery with genetically induced decision trees, V: International ICSC congress on Computational intelligence: methods and applications (CIMA'2001), 2001.
- [6] Povalej P., Lenič M., Kokol P., Reliability of decision trees, V: Journal of Medical Systems, in press.
- [7] Quinlan, J. R., C4.5: Programs for machine learning. Morgan Kaufmann publishers, San Mateo, CA, 1993.
- [8] Quinlan, J. R., Induction of decision trees, Machine Learning vol. 1, Kluwer Academic Publishers, 1986.
- [9] Smyth, P., Goodman, R. M. Rule induction using information theory, V: Piatetsky-Schapiro, G., Frawley, W. J. (Eds.), Knowledge Discovery in Databases, AAAI Press, 1991, 159–176.
- [10] White, A. P., Liu W. Z., Bias in information-based measures in decision tree induction, Machine Learning, 1994, 15:321–329.
- [11] Wolpert, D., Macready, W., No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1):67–82.
- [12] Zorman, M., Hibridni pristopi strojnega učenja. Doktorska disertacija, Maribor, 2001.
- [13] Zorman, M., Kokol, P., Redukcija iskalnega hiperprostora rešitev s hibridom grobih množic in nevronske mreže. V: Proceedings of the tenth ERK Conference Portorož, Slovenia, 2001, 205–208.

Petra Povalej je diplomirala na smeri računalništvo z matematiko na Pedagoški fakulteti Univerze v Mariboru. Je mlada raziskovalka in študentka tretjega letnika doktorskega študija iz računalništva in informatike na tej ustanovi. Njeno glavno raziskovalno področje so inteligentni sistemi. Je dobitnica prve nagrade na tekmovanju študentskih člankov na konferenci Computer Based Medical Systems 2002 in dobitnica nagrade za najboljši poster na konferenci 4th Asia-Pacific Conference on Simulated Evolution and Learning.

Dr. Peter Kokol je diplomiral s področja elektrotehnike in doktoriral s področja računalništva na Univerzi v Mariboru. Njegova raziskovalna področja so inteligentni sistemi, teorija sistemov, teorija kaosa in kvaliteta programske opreme. Je vodja nacionalnih in mednarodnih projektov, med njimi tudi projektov 5QP. Njegova bibliografija obsega več kot 500 enot, od tega več kot 60 izvornih znanstvenih člankov. Bil je predsednik organizacijskih in programskih odborov več svetovnih konferenc.

Mag. Jernej Završnik, dr. med., je diplomiral in magistriral na Medicinski fakulteti v Zagrebu in specializiral iz pediatrije. Trenutno je predstojnik Oddelka za varstvo žena, otrok in mladine v Zdravstvenem domu Maribor. Njegova raziskovana področja so echocardiografija, pediatrija in medicinska informatika.

█ Eksplicitna in implicitna uporaba vzorcev pri razvoju informacijskih rešitev

Marjan Heričko, Simon Beloglavec, Matjaž B. Jurič, Boštjan Kežmah

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko, Smetanova 17, 2000 Maribor
e-pošta: marjan.hericko@uni-mb.si

Povzetek

Uporaba komponent in ogrodij ter z njimi povezanih vzorcev je temelj sodobnega razvoja informacijskih rešitev. Prispevek pokaže uporabo nekaterih osnovnih načrtovalskih vzorcev v sklopu sodobnih programskih jezikov ter nakaže, da lahko ločimo med implicitno in eksplicitno uporabo vzorcev. Implicitna uporaba pomeni, da se razvijalec ne zaveda celovite interne strukture vzorca, a lahko kljub temu uspešno izkorišča pridobitve, ki izhajajo iz uporabe vzorca. Eksplicitna uporaba zahteva celovito razumevanje in poznavanje vzorca. Raziskava je pokazala, da aplicirani vzorci interno – po svoji strukturi – večinoma niso preprosti, tudi zato, ker jih lahko, podobno kot komponente in ostale gradnike, združujemo ter tako oblikujemo sestavljene vzorce. Ugotovitve kažejo, da lahko tudi nad vzorci uporabimo vse tri temeljne tipe abstrakcij: klasifikacijo, agregacijo in generalizacijo. Prispevek pokaže, da je za uspešen razvoj in uporabo komponent, predvsem strežniških, nujno poznavanje in razumevanje množice načrtovalskih vzorcev. Vloga in način uporabe vzorca pa določata potrebni nivo poznavanja in razumevanja vzorca.

Abstract

Explicit and Implicit Use of Software Patterns

The application of components and frameworks with corresponding design patterns is basic for contemporary software development. This article demonstrates the usage of fundamental design patterns in modern programming languages and points out the difference between implicit and explicit usage of design patterns. Implicit usage means that the developer is not aware of the whole structure of the design pattern, but he can nevertheless successfully exploit its benefits. Explicit usage requires in-depth understanding of the design pattern. Our research has shown that the internal structure of used patterns is not simple. The fact is that design patterns can be combined into compound patterns in a similar way as other software components. Our findings have shown that it is possible to use all three basic types of abstractions with design patterns: classification, aggregation and generalization. The article shows that awareness and understanding of appropriate design patterns are crucial for successful development and application of software components, especially server-side components. The role and manner of the design pattern usage define the required level and depth of understanding of it.

1 Uvod

Naša zmožnost zagotavljanja ustreznih programskih in informacijskih rešitev že od samih začetkov računalništva temelji na abstrakciji – bodisi na nivoju abstrahiranja sklopov in elementov strojne opreme, abstrahiranja na nivoju sistemske ali aplikativne programske opreme ali v smislu abstrakcij na logičnem nivoju internega ustroja programskih rešitev, ki temeljijo na modularnih, objektnih in/ali komponentnih arhitekturah. Abstrakcija je tista, ki omogoča, da obvladujemo kompleksnost, saj z njeno pomočjo skrijemo oz. zanemarimo za določen kontekst nepomembne vidike. Zgodovinski razvoj – od programske logike, zapečene v krmilnih vezjih, do aktualnega stanja - od strojne in programske platforme neodvisne vmesne kode, kot jo npr. zasledimo pri tehnologijah J2EE in .NET, dokazuje, da se nivo abstrakcije nenehno viša. To pomeni, da se skozi profiliranje in specializacijo vlog večji del informatikov vse manj posveča tehničnim in vse bolj poslovnim

izzivom. Nenazadnje je to pomemben dosežek in podpira naše poslanstvo – informacijske rešitve morajo biti namenjene podpori doseganja zastavljenih poslovnih ciljev.

Žal pa abstrakcijo v sklopu razvoja informacijskih rešitev večinoma apliciramo le nad konkretnimi komponentami – strojnimi in programskimi. Pri tem uporabljamo tri pomembne vrste abstrakcije, ki so:

- *klasifikacija* – primerke razvrščamo v skupine na osnovi skupnih lastnosti,
- *agregacija* – osredotočimo se na dejstvo, da deli skupaj tvorijo neko celoto, zanemarimo razlike med deli (z agregacijo je tesno povezano ograjevanje oz. skrivanje implementacije) in
- *generalizacija/specializacija* – skupine podobnih množic organiziramo v hierarhije glede na splošnost/specifičnost.

Z uporabo omenjenih abstrakcij lahko oblikujemo tri različne poglede na obravnavan subjekt (npr. problemsko področje, programsko rešitev, komponento), in sicer:

- *konceptualni pogled* - osredotočamo se na identifikacijo tistih gradnikov, ki so relevantni za identifikacijo rešitve v neki domeni, ne ukvarjamo se z nobenim izmed vidikov implementacije rešitve,
- *specifikacijski pogled* - obravnavamo vmesnike gradnike, ne pa interne realizacije rešitev in
- *implementacijski pogled* - obravnavamo vse podrobnosti implementacije oz. realizacije rešitve.

Temelj vsake inženirske discipline je vsekakor ponovna uporaba – vendar ne zgolj na nivoju izgradnje novih sistemov na osnovi obstoječih, že preizkušenih komponent, temveč tudi na osnovi ponovne uporabe na višjih nivojih abstrakcije – na nivoju ponovne uporabe izkušenj in spoznanj. Takšne abstraktne, ponovno uporabne koncepte, ki opisujejo in podajajo izkušnje oz. idejne rešitve, imenujemo vzorci (»patterns«). V sklopu tega prispevka bomo raziskali, kako lahko tudi za vzorce, uporabne v sklopu razvoja informacijskih rešitev, uporabimo in opredelimo prej omenjene vrste abstrakcije – podobno kot to počnemo za konkretne programske in računalniške komponente. Predvsem pa nas zanima širši pomen in vloga vzorcev pri sodobnem razvoju, temelječem na komponentah in ogrodjih. Raziskali smo uporabo vzorcev v osnovnih knjižnicah dveh najsodobnejših programskih jezikov (Java, C#) in ugotavljali, ali obstaja možnost, da vzorce uporabimo implicitno, kar pomeni, da se v bistvu ne zavedamo celovite interne strukture vzorca, a lahko kljub temu uspešno izkoriščamo prednosti, ki jih prinaša uporaba vzorca. Eksplícitna uporaba vzorcev prav nasprotno zahteva dobro poznavanje posameznega vzorca. Zanimalo nas je, ali se morda zahtevnost in nivo eksplícitne uporabe vzorcev veča, če razvijamo rešitve na osnovi komponentnih modelov. Prispevek zaključimo s povzetkom ugotovitev ter smernicami nadaljnjega dela.

2 Stopnjevanje abstrakcije ponovne uporabe

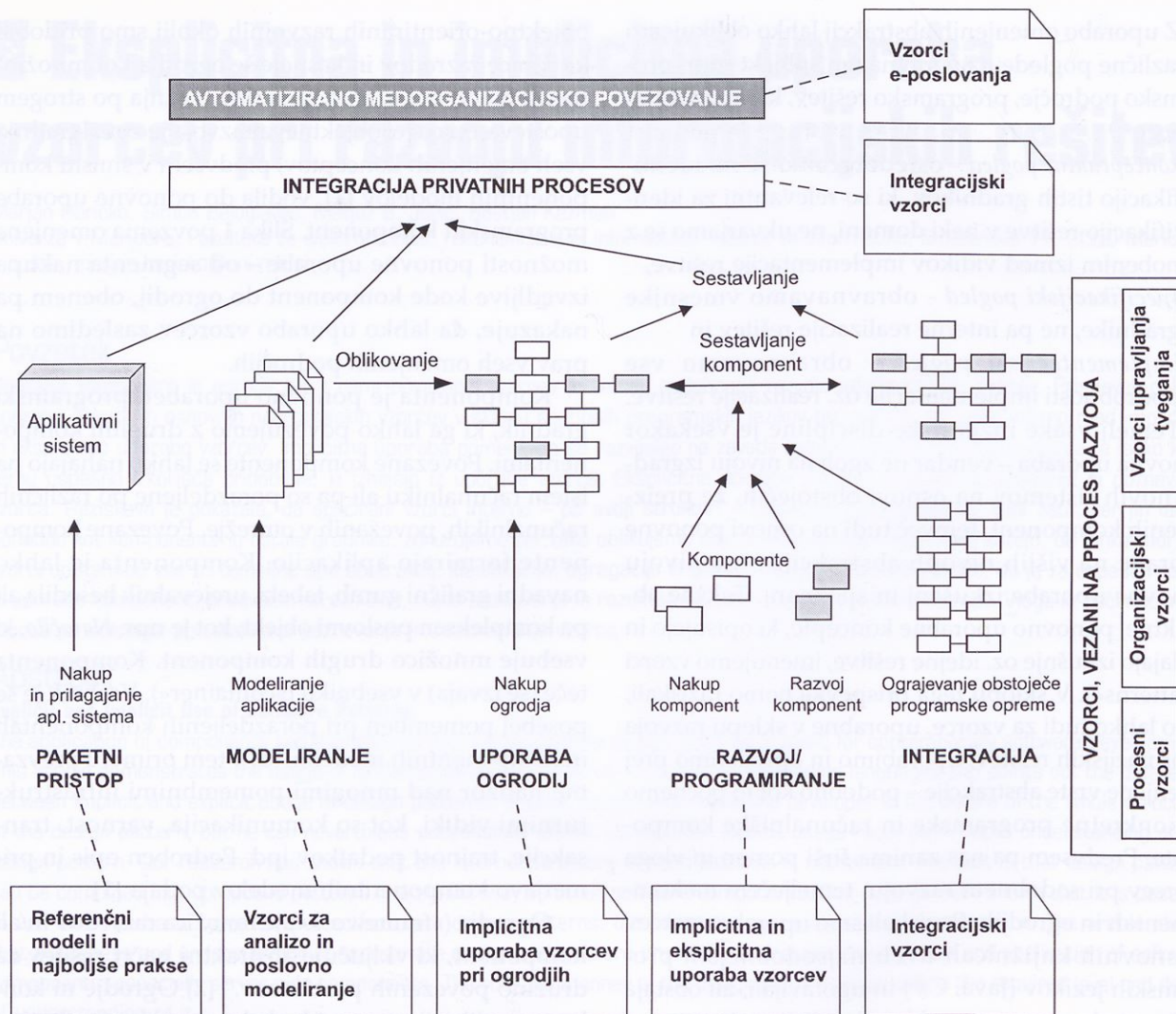
Pri razvoju informacijskih rešitev že od samih začetkov ponovno uporabljamo celotne pakete izvedljive kode ali pa zgolj dele programske kode – skupine stavkov, ki smo jih vključevali v svoje aplikacije. Konstrukti modularnosti so olajšali delo in zvišali stopnjo ponovne uporabe na nivo metod in funkcij – uveljavile so se knjižnice funkcij in makrojev. Z nastopom

objektno-orientiranih razvojnih okolij smo pridobili knjižnice razredov in kasneje še ogrodja kot množice medsebojno povezanih razredov. Težnja po strogem upoštevanju načel objektnega razvoja je z nadgradnjo vseh omenjenih konceptov, predvsem v smislu komponentnih modelov [1], vodila do ponovne uporabe programskih komponent. Slika 1 povzema omenjene možnosti ponovne uporabe – od segmenta nakupa izvedljive kode komponent do ogrodij, obenem pa nakazuje, da lahko uporabo vzorcev zasledimo na prav vseh omenjenih področjih.

Komponenta je ponovno uporaben programski gradnik, ki ga lahko povezujemo z drugimi komponentami. Povezane komponente se lahko nahajajo na istem računalniku ali pa so porazdeljene po različnih računalnikih, povezanih v omrežje. Povezane komponente formirajo aplikacijo. Komponenta je lahko navadni grafični gumb, tabela, urejevalnik besedila ali pa kompleksen poslovni objekt, kot je npr. *Naročilo*, ki vsebuje množico drugih komponent. Komponenta teče (se izvaja) v vsebniku (»container«). Vsebnik je še posebej pomemben pri porazdeljenih komponentah in komponentnih modelih, saj v tem primeru prevzame nadzor nad mnogimi pomembnimi infrastrukturnimi vidiki, kot so komunikacija, varnost, transakcije, trajnost podatkov ipd. Podroben opis in primerjavo komponentnih modelov podaja [1].

Ogrodje (»framework«) je množica razredov in/ali komponent, ki vključuje abstraktni načrt rešitev za družino povezanih problemov.“ [2] Ogrodje ni konkretna aplikacija, temveč le skelet za aplikacije. Ta skelet oz. ogrodje zaživi, ko razvijalci zagotovijo specifične komponente, nujne za delovanje rešitve. Sicer pa ogrodja vsebujejo več komponent in imajo bolj kompleksen vmesnik kot npr. sestavljene komponente. V vsakem primeru ogrodja zagotavljajo množico storitev, skupnih več aplikacijam. Storitve ogrodja so običajno vezane na določeno področje, npr. izgradnja grafičnih uporabniških vmesnikov, ogrodja poslovnih objektov za posamezne domene (bančništvo, zavarovalništvo, telekomunikacije) ali kot infrastruktura elektronskega poslovanja (npr. ebXML). Sistematično primerjavo in opis ogrodij podaja [3].

Vse bolj razširjena uporaba ogrodij v povezavi z definicijo, ki omenja abstraktni načrt, kaže na to, da lahko pri razvoju uporabimo ideje rešitev iz preteklosti. Predvsem tiste ideje, ki so se že večkrat izkazale in dokazale kot dobre in učinkovite. O tem, kako reševati ponavljajoče se podobne probleme na



Slika 1: Gradniki ponovne uporabe in uporaba vzorcev pri razvoju informacijskih rešitev

nekaterih drugih področjih, lahko razberemo iz raznih priročnikov. Nekatere med njimi smo mnogi tudi sami uporabljali, npr. matematični priročnik, fizikalni, kemijski ipd. V bistvu smo v njih poiskali idejno rešitev nekega problema, ki smo jo nato uporabili v svojem kontekstu. Tako oblikovane izkušnje in spoznanja imenujemo vzorec. Ena izmed definicij vzorca je: *Vzorec opisuje problem, ki se večkrat pojavlja v našem okolju, podaja jedro njegove rešitve na tak način, da lahko idejo rešitve uporabimo v več različnih primerih, ne da bi pot od ideje do rešitve prehodili na enak način* [4]. Vzorec torej povezuje problem z idejno rešitvijo v določenem kontekstu.

Čeprav ni razlogov, da se ne bi določene strategije uveljavile in uporabile v poljubni disciplini ali aktivnosti razvoja, pa se v praksi izkaže, da se strategije in tehnike običajno uveljavljajo od spodaj navzgor (najprej na nivoju kodiranja in načrtovanja ter šele kasneje pri poslovnem modeliranju). Podobno velja tudi za vzorce, katerih katalogi so se uveljavili najprej na področju načrtovanja in implementacije in šele kasneje še na vseh ostalih področjih, tudi na področju procesa razvoja. Še vedno pa ostajajo kot najbolj pogosto omenjani načrtovalski vzorci, ki jih poznamo pod imenom vzorci četverice oz. vzorci GoF [5]. Te vzorce lahko delimo po namenu na ustvarjalne,

strukturne in vedenjske, glede na področje uporabnosti pa na objektne in razredne. Podrobna analiza vzorcev, ki smo jo izvedli, je pokazala, da je med navedenimi triindvajsetimi vzorci v bistvu le petnajst osnovnih, dva sta variacija oz. izpeljanka ostalih, štirje so jezikovno odvisni, medtem ko sta dva v bistvu le idioma objektnega razmišljanja [6].

Vzorci lahko torej klasificiramo na različne načine, npr. po namenu, uporabnosti in tudi po tipu aktivnosti, v sklopu katerih so uporabni. Za področje razvoja programske opreme obstajajo vzorci tako za področje implementacije kot načrtovanja, analize pa tudi projektnega vodenja, organiziranja, oblikovanja procesnega modela in modeliranja poslovnih procesov (Tabela I). Klasifikacija vzorcev je torej nujno potrebna, saj bi sicer le s težavo identificirali ustrezni katalog vzorcev in se znotraj kataloga omejili na množico tistih vzorcev, ki so relevantni za naš problem in sprejemljivi v našem kontekstu. Potrebo po klasifikaciji pogojuje tudi rast števila vzorcev, zato ne zadostuje zgolj klasičen opis vzorca, ki običajno vsebuje [5]: ime, namen, motivacijo, uporabnost, vpliv, strukturo, učinek, sorodne vzorce in znane primere uporabe. Sicer pa lahko na spletu zasledimo množico katalogov vzorcev in diskusijskih baz. Ena boljših izhodiščnih točk za področje vzorcev je npr. http://www.cetus-links.org/oo_patterns.html

3 Implícitna in eksplícitna uporaba vzorcev

Če se osredotočimo na temeljne aktivnosti razvoja programskih rešitev – zajemanje zahtev, načrtovanje sistema in implementacijo, lahko ugotovimo, da se je uporaba vzorcev uveljavila predvsem, če že ne izključno, v povezavi z objektnim razvojem. Tudi zato se bomo v nadaljevanju osredotočili na raziskavo uporabe načrtovalskih vzorcev (»design patterns«), implementiranih v sklopu Jave in C#, kot predstavnikov najbolj razširjenih sodobnih jezikov. Nenazadnje je Java s svojim nastankom sredi prejšnjega desetletja bistveno pripomogla k pospešeni vpeljavi in uveljavitvi

objektne tehnologije ter komponentnega razvoja kot temeljev razvoja sodobnih informacijskih rešitev. C# te smernice le še dopolnjuje.

Raziskave o pridobitvah uporabe vzorcev večinoma temeljijo na formalizaciji opisa vzorcev, predvsem njihove interne strukture. Na ta način lahko sicer olajšamo ugotavljanje odvisnosti in povezanosti vzorcev [7], žal pa formalni opis ne rešuje temeljnega izziva – kako izboljšati ponovno uporabo izkušenj, opredeljenih kot vzorcev. Ker izkušnje iz prakse kažejo, da vzorci večinoma niso intuitivni in preprosti za razumevanje, nas je zanimalo, ali lahko morda vzorce pri razvoju v Javi in C# uporabljamo spotoma in nehote – torej implícitno, ali pa je za njihovo apliciranje potreben temeljit razmislek in zahtevano poglobljeno razumevanje oz. eksplícitna uporaba vzorcev. Preliminarna ocena stanja glede razumevanja in kompleksnosti posameznih vzorcev načrtovanja GoF, v kateri je sodelovalo dvajset razvijalcev, v povprečju več kot s tremi leti izkušenj z objektnim programiranjem, je namreč pokazala, da so vzorci različno razumljeni in uporabljani. Izmed triindvajsetih vzorcev jih anketiranci namreč kar polovico ne poznajo v podrobnosti [8] – se pa kot razvijalci v Javi oz. C# s temi vzorci nenehno srečujejo.

3.1 Implícitna uporaba vzorcev

Mnogi, ki pri razvoju uporabljajo Javo in C#, se ne zavedajo, da v teh okoljih praktično ni razvoja brez uporabe vzorcev [9]. Razlog za to je dejstvo, da razvijalci te vzorce uporabljajo implícitno, saj so ti ograjeni znotraj razredov bogatega nabora knjižnic oz. paketov, ki jih zagotavlja Java oz. ogrodje .NET. V nadaljevanju bomo omenili le nekatere vzorce, več jih zasledimo v [9,10] za Javo in v [11] za C#.

Vzorec *Edinec* zagotavlja, da lahko za nek razred kreiramo samo en primer ek oz. le enega predstavnika. Da to zagotovimo, je potrebno onemogočiti splošen dostop do konstruktorja razreda, dostop do edinega primerka pa nadzorujemo v javni razredni (statični)

Tabela I: **Avtorji oz. skrbniki katalogov znanih vzorcev za posamezne aktivnosti**

Poslovno modeliranje	Analiza	Načrtovanje	Implementacija	Arhitektura/ Integracija	Upravljanje tveganja	Procesni vzorci
Eriksson&Penker	Coad Fowler	GoF Bus Coplien, Schmidt	Beck, Alpert (Smalltalk) Cooper (C#) Grand (Java)	Schmidt Jurič	Cockburn	Ambler Coplien

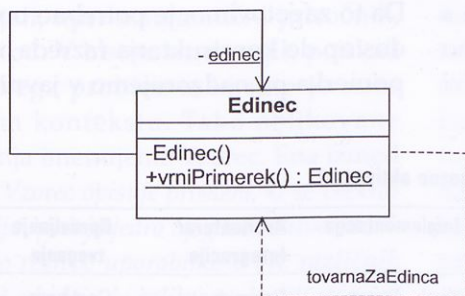
metodi, ki po potrebi poskrbi tudi za kreiranje edinega primerka – če ta seveda še ne obstaja. Na sliki 2 je podan primer kode, ki kaže idejo te rešitve. Konkretna uporaba tega vzorca pomeni, da bo razvijalec, razen preimenovanja razreda, temu dodal še specifične attribute ter ostale metode, ki zagotavljajo storitve objekta kot edinega predstavnika tega razreda. V tem primeru govorimo o eksplícitni uporabi vzorca. Na takšen način so ga tudi uporabili snovalci in razvijalci razredov, kot je npr. *RunTime* v Javi oz. *Application* v .NET. V sklopu implicitne uporabe se razvijalci razen tega, da ustrezno referenco na edini primerek pridobijo preko razredne (statične) metode omenjenih razredov, niti ne zavedajo uporabljenega vzorca.

Naslednje, ne najbolj znano dejstvo je, da so v Javi in C# vsi objekti razreda *String* v bistvu nespremenljivi, kar pomeni, da niza znakov ne moremo spremeniti. Pri implementaciji razreda *String* namreč srečamo aplikacijo vzorca *Nespremenljiv*. Ta vzorec je uporaben v različnih kontekstih, vsem pa je skupno, da obstaja primerek vzorca, ki je uporabljen v več razredih, hkrati ima objekt statično naravo – praviloma se ne spreminja. Tako C# kot Java namreč za potrebe dela z objekti tipa *String* vzdržujeta fond objektov nizov, ki so bili oblikovani in so v določenem trenutku še aktualni za sistem. Vzorec povečuje robustnost objektov, ki si delijo skupno referenco na en sam objekt, in zmanjšuje kompleksnost hkratnega dostopa do objekta. To dosega s prepovedjo spremembe kate-regakoli podatka, ki določa stanje objekta, potem ko je bil le-ta kreiran. Podobno idejo zasledimo tudi pri vzorcu *Zrno*. S tem ko uporabljamo objekte razreda *String*, se torej soočamo z implicitno uporabo vzorcev *Zrno* in *Nespremenljiv*. Aplikacijo vzorca *Nespremenljiv*

pa npr. v jeziku C# srečamo tudi pri objektih – delegatih.

Pri razvoju v Javi in C# pogosto srečamo uporabo vzorca *Iterator*, še najbolj je uporaba iteratorjev vidna pri delu z množicami objektov pri uporabi t. i. ogrodja kolekcij (*Collection Framework*). Vmesnik *Collection* igra vlogo vmesnika do implementacije kolekcije oz. množice objektov, ki zna oblikovati iteratorje za prehod nad objekti, ki jih vsebuje. Osnovni cilj uporabe tega vzorca pa je, da se sprehodimo prek vseh primerkov oz. objektov v kolekciji, ne oziraje se na njeno implementacijo, ki je lahko npr. primerek razreda *Vector*, *HashTable*, *Set* ali poljuben drug objekt, ki zagotavlja implementacijo vmesnika. S proženjem metode *iterator()* namreč pridobimo referenco na ustrezni vmesnik, ki omogoča, da se z metodo *next()* pomaknemo na naslednji objekt, objekt zberemo z *remove()* in/ali prožimo metodo *hasNext()*, da preverimo, ali smo obdelali vse objekte v kolekciji. Uporaba tega vzorca zagotavlja neodvisnost razreda odjemalca od same implementacije kolekcije. Nenazadnje pa lahko prek različnih implementacij vmesnika *Iterator* zagotovimo različne poti in zaporedja obdelave/prečkanja objektov ter tako vplivamo tudi na zmogljivosti sistema.

Kot prikaz uporabnosti vzorca *Iterator* služi v Javi razred *Collections*, ki zagotavlja splošne pripomočke za urejanje in kopiranje kolekcij objektov. Še najpogostejše pa razvijalci uporabljajo izpeljanko vzorca *Iterator*, ki se imenuje *Enumeration*. V jeziku C# zasledimo apliciranje vzorca *Iterator* pri poljih ter pri vmesnikih kot so *ICollection*, *IList*, torej za razrede, ki podpirajo vmesnik *IEnumerator*. C# gre pri tem celo korak dlje – rezervirana beseda *foreach* omogoča, da na enostaven način obdelamo vse objekte v neki kolekciji (primer 1).



```

public class Edinec {
    private static Edinec edinec;
    private Edinec() {}

    public static Edinec vrniPrimerек()
    {
        if (edinec == null)
            edinec = new Edinec();
        return edinec;
    }
}
  
```

Slika 2: **Struktura vzorca *Edinec* in skelet implementacije v C# oz. Javi**

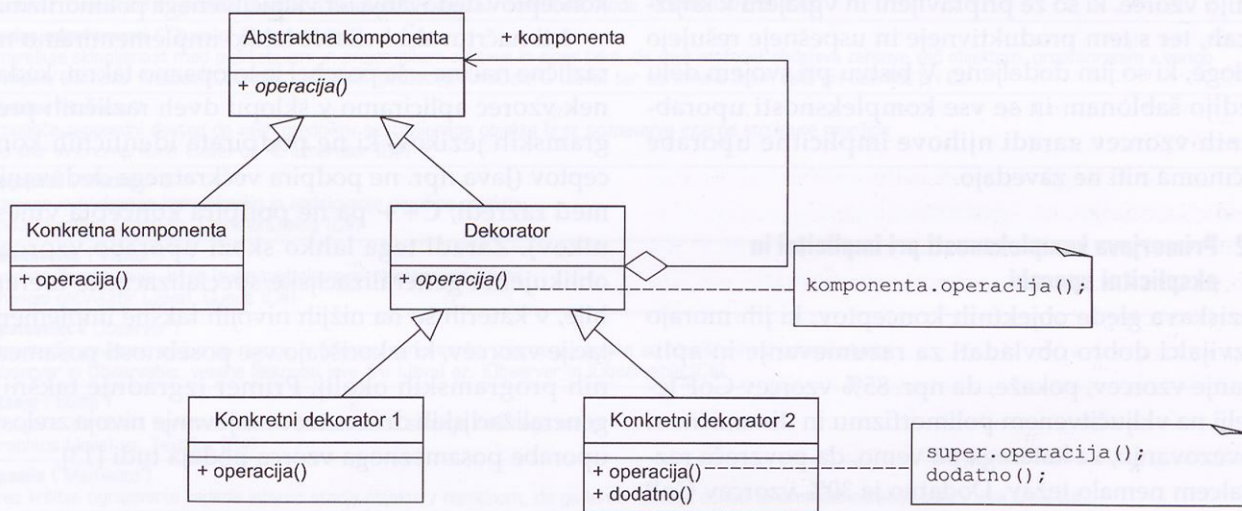
Java	C#
<pre>Iterator i = kolekcija.iterator(); while (i.hasNext()) System.out.println(i.next());</pre>	<pre>foreach (i in kolekcija) Console.WriteLine(i);</pre>

Primer 1: **Primer koriščenja vzorca *Iterator* v Javi in C#**

Pri delu z vhodno-izhodnimi tokovi preprosto ne moremo zaobiti uporabe vzorca *Dekorator* (slika 3). Ta omogoča, da objektu dinamično, v času izvajanja, dodajamo funkcionalnosti. Na osnovi hierarhije dedovanja namreč oblikujemo strukturo, ki omogoča, da pri oblikovanju z novimi storitvami obogatenih objektov v konstruktorju navedemo objekt, ki mu dodajmo funkcionalnost, veriga objektov pa prek delegiranja in uporabe vključitvenega polimorfizma rezultira v širši funkcionalnosti. Druga možnost bi bila, da kreiramo vse možne kombinacije podrazredov, za kar pa potrebujemo še podporo večkratnemu dedovanju med razredi, ki pa je Java in C# ne omogočata (večkratno dedovanje je podprto le med vmesniki).

Pri vhodno izhodnih operacijah večkrat naletimo na zahtevo, da potrebujemo funkcionalnost dveh razredov, npr. razreda za delo z datotekami in razreda, ki omogoča branje podatkov osnovnih tipov. Sama koda (primer 2) je precej enostavna, a le kot posledica dejstva, da je v ozadju oblikovana ustrezna struktura, ideja katere je prikazana na sliki 3. Omenjeni vzorec je običajno osnova tudi za ustrezno serializacijo objektov, brez katere ni mogoče zgraditi sodobnih sistemov, ki temeljijo na komunikaciji porazdeljenih objektov.

Pomemben vidik vsakega objektnega sistema in okolja je komunikacija med objekti oz. dogodkovni model, ki to komunikacijo omogoča. Dogodkovni model, ki je temeljil na dedovanju, je z verzijo 1.1 v Javi nadomestil dogodkovni model, ki temelji na delegiranju. Komponenta, ki je izvor dogodkov, mora o dogodku obvestiti vse objekte, ki so se registrirali kot poslušalci za to vrsto dogodka (pogoj je, da zagotavljajo ustrezno funkcionalnost oz. implementirajo ustrezní vmesnik). V bistvu lahko nek objekt nastopa v vlogi poslušalca različnih tipov dogodkov. Vzorec *Delegat* omogoča, da namesto oblikovanja novega



Slika 3: **Struktura vzorca *Dekorator***

Java	C#
<pre>FileReader fr = new FileReader ("a.txt"); BufferedReader br = new BufferedReader(fr); int i = br.read();</pre>	<pre>FileStream fr = new FileStream("a.txt", FileMode.Open); BufferedStream br = new BufferedStream(fr); int i = br.ReadByte();</pre>

Primer 2: **Implicitna uporaba vzorca *Dekorator* v Javi**

podrazreda, razred zgolj delegira odgovornost za izvedbo neke operacije drugemu objektu. Delegiranje je pogosto s konceptualnega vidika primernejše kot uporaba dedovanja, še posebej v primerih, ko imamo opravka z asociacijami tipa »je-vloga«.

Podrobnejši razmislek pokaže, da smo vzorec *Delegat* srečali že v sklopu vzorca *Dekorator*, saj je abstraktni razred s pomočjo delegiranja odgovornost za izvedbo operacije prenesel drugemu objektu. To dokazuje, da lahko vzorce obravnavamo na podoben način kot komponente – iz manjših lahko sestavljamo večje, bolj zapletene, ki rešujejo oz. podajajo ideje rešitev kompleksnejših problemov. Uporabo vzorca *Delegat* v jeziku C# oz. okolju .NET v splošnem olajšata že vgrajena razreda *Delegat* in *Multicast-Delegat*. Jezik C# pa uporabo tega vzorca še dodatno poenostavi, saj lahko uporabimo rezervirano besedo *delegat* ter se tako izognemo neposrednemu delu z razredi.

Celovitejši seznam ter obsežnejša razlaga vzorcev, ki jih zasledimo že v osnovni Javi in ogrodju .NET, je na voljo v [9,10,11]. Vsekakor pa že nekaj preprostih primerov, ki smo jih podali, dokazuje, da lahko tudi manj izkušeni razvijalci v Javi in okolju .NET izkoristijo vzorce, ki so že pripravljene in vgrajene v knjižnicah, ter s tem produktivneje in uspešneje rešujejo naloge, ki so jim dodeljene. V bistvu pri svojem delu sledijo šablonam in se vse kompleksnosti uporabljenih vzorcev zaradi njihove implícitne uporabe večinoma niti ne zavedajo.

3.2 Primerjava kompleksnosti pri implícitni in eksplicitni uporabi

Raziskava glede objektnih konceptov, ki jih morajo razvijalci dobro obvladati za razumevanje in apliciranje vzorcev, pokaže, da npr. 85% vzorcev GoF temelji na vključitvenem polimorfizmu in dinamičnem povezovanju, za katerega pa vemo, da povzročata razvijalcem nemalo težav. Dodatno je 30% vzorcev GoF takšnih, da zahtevajo tudi sledenje in upoštevanje ustreznega zaporedja proženja metod. Analiza med razvijalci [8] je pokazala, da razvijalci vzorce GoF pogosto medsebojno zamenjujejo in da jih več kot polovico ne razumejo v tolikšni meri, da bi ta omogočala njihovo eksplicitno uporabo. Kot kažejo predstavljeni primeri v tem poglavju, pa se s prav istimi vzorci implícitno srečujejo pri uporabi standardnih knjižnic sodobnih jezikov. Ugotovitev dodatno potrjuje spoznanje, da eksperti v določenih domenah

pri modeliranju in razvoju sistemov vzorce uporabljajo v nesistematični obliki oz. se njihove uporabe ne zavedajo, dokler jim ta ni eksplicitno predstavljena oz. dokler vzorci niso ustrezno dokumentirani [12].

Strukturo vzorcev GoF smo podrobneje analizirali tudi s pomočjo uveljavljenih objektnih metrik. Podrobni rezultati so predstavljeni v [8]. Za potrebe prispevka se osredotočimo le na že predstavljen vzorec *Dekorator*. Metrična analiza apliciranja – torej eksplicitne uporabe vzorca *Dekorator* pokaže, da imamo opravka z najmanj 4 razredi, povprečna globina dedovanja je 2, delež predefiniranih metod je 33%. Ugotovimo lahko, da pri implícitni uporabi – glej primer 2, sploh ni potrebno poznavati dedovanja in polimorfizma – zadostuje že, da sledimo ustreznima korakoma – oblikujemo osnovni objekt, ki ga nato posredujemo kot argument v konstruktor, s pomočjo katerega oblikujemo objekt dekorator. Ugotovimo lahko, da implícitna uporaba vzorcev, npr. pri tokovih, omogoča veliko stopnjo abstrakcije, razvijalec pa se sooča z bistveno manjšo kompleksnostjo, kot če bi sam oblikoval ustrezno strukturo dedovanja. V kolikor bi želeli že apliciran vzorec *Dekorator* še razširiti, pa bi bilo potrebno delno poznavanje strukture in s tem tudi konceptov dedovanja ter vključitvenega polimorfizma.

Isti načrtovalski vzorec lahko implementiramo na različne načine – še posebej je to opazno takrat, kadar nek vzorec apliciramo v sklopu dveh različnih programskih jezikov, ki ne podpirata identičnih konceptov (Java npr. ne podpira večkratnega dedovanja med razredi, C++ pa ne podpira koncepta vmesnikov). Zaradi tega lahko skozi uporabo vzorcev oblikujemo generalizacijske/specializacijske hierarhije, v katerih so na nižjih nivojih takšne implementacije vzorcev, ki izkoriščajo vse posebnosti posameznih programskih okolij. Primer izgradnje takšnih generalizacijskih dreves ter ocenjevanje nivoja zrelosti uporabe posameznega vzorca podaja tudi [13].

4.1 Komponente in implícitna uporaba z zahtevo po poznavanju vzorca

Prehod na komponentni razvoj pogojuje dodatna znanja, saj že npr. uporaba javanskega komponentnega modela (Java Beans) zahteva poznavanje in sledenje določenim vzorcem. Omenili smo, da aktualni dogodkovni modeli (tako v Javi kot v .NET) temeljijo na delegiranju. Če npr. pri oblikovanju ustreznega odziva pri uporabi grafičnega uporabniškega vmesnika, razen načina za implementacijo

Edinec ("Singleton") zagotavlja, da ima razred en sam primerek RunTime (Java) oz. Application (C#)
Abstraktna tovarna ("Abstract Factory") zagotavlja vmesnik za kreiranje družin povezanih ali odvisnih objektov brez določitve njihovih konkretnih razredov Toolkit in URLStreamHandlerFactory (Java) oz. XmlDocument (C#)
Graditelj ("Builder") loči konstrukcijo kompleksnega objekta od njegove predstavitve – isti konstrukcijski proces lahko oblikuje različne predstavitve ModelTreeBuilder (Java) oz. EventArgs (C#)
Prototip ("Prototype") določa vrsto objektov z uporabo prototipnega primerka, na osnovi katerega se oblikujejo kopije Cloneable (Java) oz. ICloneable (C#)
Tovarniška metoda ("Factory Method") definira vmesnik za kreiranje objekta, podrazredom pa prepusti odgovornost za določitev konkretnega razreda, za katerega oblikuje primerek Component.getGraphics(), Toolkit.getImage(), Collator.getInstance() (Java) oz. PageHandlerFactory.getHandler() (C#)
Adapter ("Adapter") preoblikuje vmesnik razreda v vmesnik, ki ga pričakuje odjemalec JDBCDriver, Adapter-ji poslušalcev (Java) oz. RCW (Run-Time Callable Wrapper) in CCW (COM callable Wrapper) (C#)
Dekorator ("Decorator") omogoča dinamično dodajanje odgovornosti FilterReader ipd. pri delu z vhodno/izhodnimi tokovi v Javi in C#
Namestnik ("Proxy") zagotavlja nadomestni objekt, ki kontrolira dostop do originalnega objekta RMIProxy (Javi) oz. Remoting, OleDbConnection (C#)
Most ("Bridge") omogoča, da se vmesnik in implementacija ločeno spreminjata Component in ComponentPeer (Java) oz. IComponent in ISite (C#)
Fasada ("Façade") zagotavlja enoten vmesnik za množico vmesnikov v podsistemu URL (Java) oz. DataSet (C#)
Zrno ("Flyweight") uporaba skupnega primerka za učinkovito podporo večjega števila manjših objektov String, fond povezav, sejnih zrn pri EJB (Java) oz. FontFamily (C#)
Kompozicija ("Composite") združuje objekte v drevesne strukture, ki prikazujejo hierarhije celota-deli Vsebniki (Container) in grafični gradniki (Component) v Javi oz. TreeNode, Node, Frame, Control (C#)
Veriga odgovornosti ("Chain of Responsibility") zmanjšuje sklopljenost med pošiljateljem in prejemnikom zahteve in sicer tako, da daje možnost obdelave zahteve več objektom, organiziranim v verigo omejene lastnosti (VetoableChange), getFont(), getBackgroundColor() (Java) ter obravnava izjem tako pri Javi kot pri C#
Iterator ("Iterator") omogoča zaporedni dostop do vseh gradnikov sestavljenega objekta brez poznavanja interne strukture množice Iterator in Enumeration (Java) oz. IEnumerator (C#)
Pogajalec ("Mediator") ograjuje in nadzoruje komunikacijo in sodelovanje množice objektov FocusManager (Java) oz. CommonDialog (C#)
Obiskovalec ("Visitor") predstavlja operacijo, ki se izvaja nad elementi objektne strukture ChangeFontVisitor (Java), Queue (C#)
Opazovalec ("Observer") vzpostavlja povezavo ena-proti-mnogo, v primeru spremembe izvornega objekta so obveščeni vsi odvisni objekti Observer in Observable, vezane lastnosti java zrn (Java) oz. IObservable in IObservable (C#)
Stanje ("State") omogoča, da objekt spremeni odziv v odvisnosti od svojega internega stanja Graphics (Java) oz. TextBox (C#)
Spomin ("Memento") brez kršitve ograževanja zajame interno stanje objekta z namenom, da ga lahko kasneje obnovi binarna serializacija tako v Javi kot v C#
Interpreter ("Interpreter") za dani jezik definira predstavitev slovnice in intrerpreterja stavkov tega jezika Matcher in Patterns v javax.regex (Java) oz. XMLValidatingReader (C#)
Komanda ("Command") ogradi zahtevo oz. ukaz kot objekt UndoableEdit, AccessibleAction (Java) oz. SQLCommand, UndoContext (C#)
Strategija ("Strategy") definira in ogradi družino algoritmov in tako zagotovi njihovo zamenljivost LayoutManager, CheckInputStream (Java) oz. IRemotingFormatter, CryptoAPITransform (C#)
Metoda predloge ("Template Method") definira skelet algoritma operacije, pri čemer odgovornost za določene korake prepusti podrazredom Metode setter in getter pri entitetnih zrnih tipa CMP (EJB), večina uporabe virtualnih metod, tako v Javi kot v C#

Tabela II: **Pregled vzorcev Gof s primeri apliciranja v Javi in C# oz. .NET**

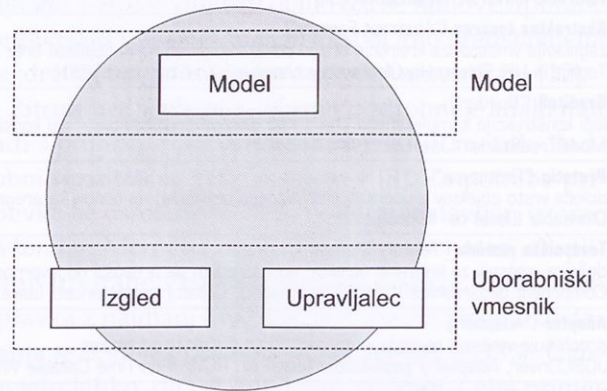
vmesnika (ali uporabe adapterja) ter registriranja objekta – poslušalca, ni potrebno poznati podrobne interne zgradbe in uporabe dogodkovnega modela, pa v primeru, ko sami razvijamo komponente – zrna, moramo poznati in aplicirati tudi vse podrobnosti omenjene zasnove. Komponente namreč preko ustreznih metod za registracijo poslušalcev določajo dogodke, ki se lahko pojavijo na komponenti. Prek tega tudi razvojna okolja identificirajo dogodke posamezne komponente in omogočajo dogodkovno vodeno povezovanje komponent. Pri izvajanju komponent pa zasledimo uporabo ustvarjalnega vzorca, imenovanega *Prototip*.

Kot primer navedimo knjižnico *Swing*, ki je del JFC (Java Foundation Classes), ponuja pa nabor gradnikov (javanskih komponent oz. zrn) za izgradnjo grafičnih uporabniških vmesnikov. Razen tega, da pri izkoriščanju te knjižnice uporaba nekaterih vzorcev izhaja že iz dejstva, da gre za javanske komponente, uspešnega dela z gradniki knjižnice *Swing* ni mogoče pričakovati brez poznavanja arhitekture vzorca MVC (Model-View-Controller). V praksi ga zelo pogosto, ko je implementiran v določenem okolju, imenujejo kar ogrodje. Razvoj, uporaba in izvajanje *Swing* komponent temelji na dejstvu, da je potrebno zagotoviti tri segmente: *model*, *pogled* in *upravljavca*, kjer sta izgled in upravljevec zaradi močne soodvisnosti združena v enem objektu (delegat). Osnovni cilj je ločitev podatkov od njihovega prikaza, kar je ena izmed splošnih smernic pri razvoju informacijskih rešitev.

Poenostavljeno to pomeni, da je manipulacija in obravnava podatkov, ki se prikazujejo s pomočjo neke grafične komponente, ločena od samega prikaza. Implementacija razreda, ki služi kot osnova za oblikovanje objekta z modelom, pa je kljub vsemu precej odvisna od kompleksnosti komponente za prikaz. V primeru uporabe komponente *JTable* je tako npr. potrebno poznati in implementirati šest vmesnikov in devet razredov (slika 4).

Uporabnost in prednosti uporabe arhitekturnega vzorca MVC se pokažejo v primerih, ko je potrebno modelu zamenjati izgled ali izgledu zamenjati model, zagotoviti različne poglede na iste podatke, porazdeliti

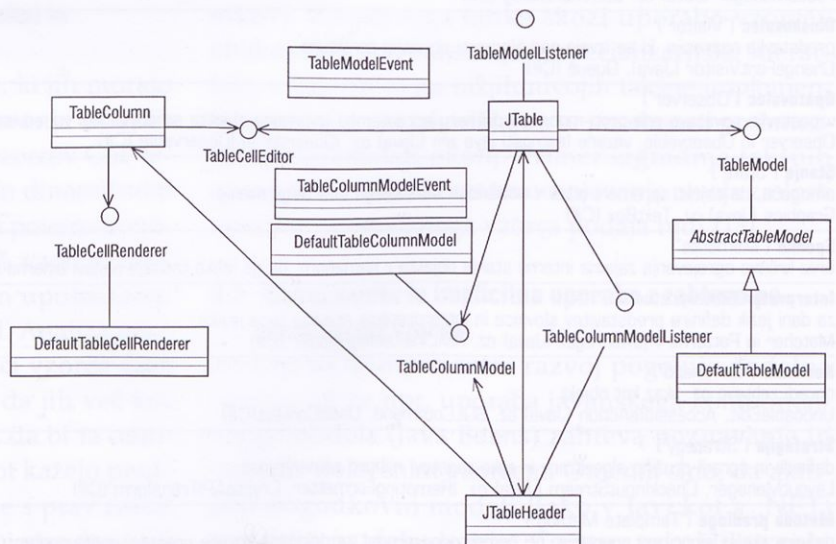
JFC UI komponenta



Slika 4. Ideja vzorca MVC, uporabljena v JFC

aplikacijo ali zaradi povečanja zmogljivosti zagotoviti dodatne objekte s poslovno logiko oz. modelom.

Podobno kot pri vzorcu *Dekorator* je tudi vzorec MVC v bistvu sestavljen iz manjših vzorcev, imenovanih *Opazovalec*, *Kompozicija* in *Strategija*, kar intuitivno potrjuje potrebo po agregaciji vzorcev – formalne dokaze, da lahko vzorce združujemo v kompleksnejše vzorce podajata [6,7]. Tudi glede vzorcev je eden izmed pomembnih vidikov uspešne uporabe ustrezno dokumentiranje ponovno uporabnih gradnikov. Večinoma se raziskovalci osredotočajo na obliko dokumentiranja vzorcev za potrebe njihove eksplicitne uporabe. V [6] smo tudi sami definirali formalno



Primer 4: Razredi in vmesniki, povezani z *JTable* v *Javi*

osnovo za opis vzorcev. Pri tem smo se osredotočili predvsem na formalno predstavitev vzorcev in njihovih medsebojnih relacij, tudi z namenom definiranja mehanizmov povezovanja in integracije, torej za eksplícitno uporabo. Izkazalo se je, da lahko principe kombiniranja in sestavljanja novih komponent iz manjših komponent, v precejšnji meri izkoristimo tudi na področju vzorcev in le-te povezujemo ter tako oblikujemo sestavljene vzorce, ki predstavljajo idejo rešitev kompleksnejših nalog in problemov.

4.2 Vzorcji pri razvoju strežniških poslovnih komponent

Glede na dejstvo, da že uporaba »navadnih« komponent zahteva dobro poznavanje mnogih vzorcev, je pričakovati, da bodo potrebe po razumevanju in uporabi vzorcev na nivoju strežniških komponent, npr. javanskih zrn EJB (Enterprise Java Beans) ali .NET komponent, še večje. Izkaže pa se, da strežniške tehnologije razvijalca v precejšnji meri razbremenijo nekaterih skrbi in obveznosti, kot so npr. zagotavljanje trajnosti, transakcijskega obnašanja, varnosti. Te vidike sistema določamo deklarativno, kar pa pomeni, da te mehanizme zagotovijo strežniška okolja (aplikacijski strežniki s pripadajočimi vsebniki), in sicer z apliciranjem ustreznih vzorcev, kot so npr. *Abstraktna tovarna*, *Prototip*, *Zrno* (za fonde povezav do virov), *Namestnik* (za potrebe proženja oddaljenih metod) ipd. Več tovrstnih vzorcev smo opisali v [14,15]. Na drugi strani pa lahko že z apliciranjem preprostih vzorcev rešimo nekatere izzive, ki jih prinaša razvoj porazdeljenih sistemov, npr. zagotavljanje ustrezne zmogljivosti in odzivnih časov. Kot primer navedimo le nekatere izmed »preprostih« vzorcev, ki jih lahko apliciramo v ta namen [15]:

- *Sejna fasada* (Session facade) – vzorec združuje aplikacijske funkcije v en poenostavljen vmesnik, ki je namenjen izključno odjemalcu ter s tem skriva podrobnosti implementacije poslovnega opravila;
- *Poslovni delegat* (Business Delegate) je v bistvu poseben primer vzorca *Delegat*, saj med oddaljeni poslovni objekt in odjemalca vstavi nov vmesnik, katerega namen je skriti kompleksnost oddaljene komunikacije;
- *Vrednostni objekt* (Value object) izboljšuje izmenjavo podatkov med različnimi nivoji porazdeljenega sistema – osnovna ideja je, da namesto posameznega podatka prenesemo večji nabor podatkov;

- *Dostopni podatkovni objekt* (Data Access Object) narekuje realizacijo posebnih virtualnih komponent, ki ogradijo specifično programsko kodo za dostop do posameznih zunanjih virov.

Ko ob besedi *vzorec* uporabljamo atribut »preprost« je treba upoštevati, da se pogosto izkaže, da je ideja rešitve nekega problema v bistvu enostavna – seveda takrat, ko jo poznamo. Izkaže se tudi, da zahtevni problemi pogojujejo kombiniranje in združevanje delnih rešitev, tudi idej rešitev. Ugotovili smo že, da lahko vzorce združujemo v večje gradnike. Razvijalci v vlogi implicitnega uporabnika vzorcev morajo zaradi načel komponentnega razvoja poznati in razumeti le del celotne komponentne zgradbe vzorca. Komponentni modeli spodbujajo tudi specializacijo vlog pri razvoju. Tako so tipične vloge pri razvoju strežniških komponent [16]: razvijalec komponent, sestavljalca komponent, nameščevalec, administrator zrn in ponudnik vsebnika/strežnika.

Razvijalec zgradi komponente, jih prevede in jih posreduje v obliki posebne arhivske datoteke EAR-JAR (Enterprise ARchive-Java ARchive) pri EJB oz. zbirne datoteke (»assembly«) pri ogrodju .NET. Razvijalec je odgovoren za definicijo vmesnikov komponent in implementacijo razredov s poslovno logiko. Prav tako mora priskrbeti posebne opisne datoteke – deskriptorje, ki vsebujejo parametre komponent (nanašajo se na vire, ki jih uporablja komponenta). Rezultate dela razvijalca komponent prevzame sestavljalca, ki poveže posamezne dele v eno namestitveno enoto, ki je pripravljena za namestitvev na aplikacijski strežnik. V tako namestitveno enoto običajno združimo še spletne komponente (npr. JSP strani in servlete oz. ASP.NET strani). Nameščevalec nastavi pripravljeno enoto na specifičen aplikacijski strežnik, pri tem uporablja orodja, ki so predpisana s strani specifičnega aplikacijskega strežnika. Administrator komponent je odgovoren za nastavitvev parametrov samega aplikacijskega strežnika, kot tudi parametrov vsebnika zrna. Specifikacija navaja ločeno še ponudnika strežnika in vsebnika, vendar sta v praksi ti dve vlogi združeni. Pri samem načrtovanju je specializiran načrtovalec nujno potreben, saj se razred s poslovno logiko iz konceptualnega modela preslika v več razredov, ko razgrajujemo model v implementacijske podrobnosti. V bistvu smo že omenili, da mora ponudnik vsebnika aplicirati mnoge vzorce (od *Abstraktna tovarna* do *Namestnika*), podobno velja za načrtovalca in razvijalca

komponent, ki vzorce eksplicitno uporabljata. Sestavljavae komponent v večini primerov shaja z razumevanjem vzorcev, potrebnih za implícitno uporabo, ob sestavljanju kompleksnejših komponent pa tudi on sledi ustreznim vzorcem, ki jih uporabi eksplicitno ali z zadostnim razumevanjem vsaj implícitno.

6 Sklep in smernice raziskav

Uporaba ogrodij komponent in z njimi povezanih vzorcev je ključ sodobnega razvoja tako na nivoju zasnove informacijskih rešitev kot tudi na nivoju definiranja in informatizacije medorganizacijskih povezav in poslovanja. Komponente, ogrodja in vzorci naj bi olajšali razvoj informacijskih rešitev. Podrobna raziskava pokaže, da vzorci interno, po svoji strukturi, niso preprosti, tudi zato, ker jih lahko podobno kot komponente in ostale gradnike združujemo in kombiniramo ter tako oblikujemo sestavljene vzorce. Z uporabo metrik kompleksnosti (upoštevanje hierarhije dedovanja, števila vpletenih razredov in asociacij med razredi) smo ugotovili, da že večina temeljnih vzorcev v principu ni preprosta. Na srečo pa sama implementacija in uporaba agregacije ter vmesnika skrije kompleksnost apliciranega vzorca.

Zato je pomembno, da tudi vzorce, podobno kot objekte in komponente, obravnavamo v luči treh, uvodoma opredeljenih abstrakcij (tabela III). Klasifikacija omogoča, da lahko določen vzorec umestimo v ustrežno kategorijo, določimo njegov namen in učinke in tako lažje sprejmemo odločitev v postopku izbire ustreznega vzorca za podani problem. Uporaba agregacije podpira idejo o sestavljenih vzorcih ter delno tudi implícitno uporabo, saj nas pri tej zanima zgolj specifikacijski nivo, ne pa tudi implementacijski. Generalizacija in specializacija je pri vzorcih potrebna zaradi različnih možnosti implementacijskega nivoja vzorcev.

Na osnovi pridobljenih spoznanj in izhodišč, ki jih podaja pričujoči prispevek, ugotavljamo, da je vzorce

zagotovo treba obravnavati vsaj z dveh vidikov, ki sta pogojena predvsem z vlogami razvijalcev, ki se z vzorci srečujejo in jih uporabljajo. Apliciranje vzorcev pri razvoju informacijskih rešitev pogojuje kompleksnejša znanja in razumevanje, medtem ko uporaba že integriranih vzorcev zahteva nekoliko drugačna, občasno tudi delno poenostavljena znanja. Menimo, da bomo lahko na osnovi že definirane formalnega zapisa interne zgradbe in relacij med vzorci zasnovali in izpeljali tudi t. i. zunanji pogled na vzorce, ki ga je potrebno formalizirati, če želimo, da bodo razvojna orodja, razen eksplicitne uporabe temeljnih vzorcev, podprla tudi sestavljanje vzorcev in razvijalcem olajšala iz tega izhajajočo implícitno uporabo vzorcev. Slednja sicer dopušča določen nivo abstrakcije, predvsem zaradi komponentnega pristopa k integraciji vzorcev, na drugi strani pa kljub vsemu prinaša med razvijalce precejšnjo kompleksnost in zahtevnost razumevanja, ki jo lahko olajša le primerna oblika ograjevanja in višji nivo njene avtomatizacije.

V prispevku smo prikazali, da predstavlja uporaba vzorcev enega od osnovnih načinov uspešnega razvoja sodobnih informacijskih rešitev. Pravzaprav edini način. Spoznali smo, da so že v osnovni Javi in ogrodju .NET uporabljeni različni tipi vzorcev, npr. ustvarjalni vzorci (npr. *Edinec*, *Prototip*, *Abstraktna tovarna*), strukturni vzorci (npr. *Dekorator*) kot tudi vzorci obnašanja (npr. *Iterator*, *Opazovalec*). Uporaba teh, že vgrajenih vzorcev, predstavlja neke vrste implícitno uporabo vzorcev, saj razvijalcu ni potrebno poznati vseh podrobnosti uporabljenega vzorca, temveč le osnove apliciranja. Izkaže pa se, da je uporaba in podrobno poznavanje vzorcev pri zasnovi kompleksnejših, komponentno zasnovanih rešitev nujna. Zaradi delitve vlog pri komponentnem razvoju, še posebej v sklopu strežniških tehnologij, kot sta J2EE (Java 2 Enterprise Edition) in .NET, pa lahko ugotovimo, da je delitev na eksplicitno in implícitno

Tabela III: **Uporaba treh osnovnih abstrakcij pri sodobnem razvoju**

	objektni pristop	komponentni razvoj	uporaba vzorcev
Klasifikacija	objektov v razrede	nabor storitev v vmesnikih	klasifikacije vzorcev po področjih, aktivnost
Agregacija	sestavljene (kompleksni) objekti kot agregati in kompoziti	sestavljene komponente (nivo granularnosti)	sestavljanje vzorcev
Generalizacija	dedovanje med razredi	vmesnikov komponent	družine implementacij vzorcev

uporabo vzorcev smiselna in koristna. Temu je potrebno prilagoditi tudi program izobraževanja posameznih razvijalcev in pri tem najprej opredeliti stopnjo uporabe vzorcev, ki je lahko:

- implicitna brez zavedanja o prisotnosti vzorcev,
- implicitna z zavedanjem prisotnosti vzorcev in potrebo po njihovem poznavanju ter
- eksplicitna s poznavanjem strukture vzorca.

Literatura

1. Jurič M. B., "Nova generacija komponentnih modelov CORBA 3, COM+, EJB", zbornik pete konference OTS'2000 *Objektna tehnologija v Sloveniji*, str. 18–29.
2. Johnson R., Foote B., "Designing Reusable Classes", *Journal of Object-Oriented Programming*, let. 1, št. 2, 1988, str. 22–35.
3. Krajnc A., Štok B., "Uporaba ogrodij v objektno-orientiranih aplikacijah", zbornik sedme konference OTS'2001 *Objektna tehnologija v Sloveniji*, str. 63–75.
4. Alexander C. et al., *A Pattern Language*, New York, Oxford University Press, 1977.
5. Gamma E., Helm R., Johnson R., Vlissides R., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995.
6. Domajnko T., Rozman I., Heričko M., "Analiza vzorcev načrtovanja", *Elektrotehniški vestnik*, let. 67, št. 5, str. 261–267.
7. Domajnko T., *Formalizacija opisa vzorcev*, FERI Maribor, doktorska disertacija, 2001.
8. Heričko et al., *Analiza razumevanja, kompleksnosti in uporabnosti vzorcev načrtovanja GoF*, FERI Maribor, Inštitut za informatiko, poročilo, februar 2003, 25 strani.
9. Domajnko T., Heričko M., "Vzorci in Java", zbornik četrte konference OTS'99 *Objektna tehnologija v Sloveniji*, FERI Maribor, str. 185–198.
10. Grand M., *Patterns in Java – volume 1*, Wiley, 1998.
11. Cooper J.W., *C# Design Patterns – A Tutorial*, Addison-Wesley, 2002.
12. J. Eckstein, "Architekturen für das E-business", *JavaSpektrum*, let. 8, št.1, januar 2003, str. 37–39.
13. Zhao Y., "Developing Pattern Implementation Knowledge for Reinforcing Software Design Patterns", *Proceedings of the 21st IASTED Int. Conf. Applied Informatics*, str. 967–972.
14. Jurič M. B. et al., *J2EE Design Patterns Applied*, Wrox Press Inc., 2002.
15. Rozman I. et al., *Integracijska arhitektura: strategije, postopki in metode integracije*, FERI Maribor, Inštitut za informatiko, tehnično poročilo, julij 2002, 133 strani.
16. Beloglavec S., Rozman I., "Vloge pri razvoju strežniških komponent", zbornik konference IS'2002 *Sodelovanje in informacijska družba*, str. 53–56.

Marjan Heričko je docent na Inštitutu za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Njegovo raziskovalno-razvojno delo obsega vse vidike objektno tehnologije in komponentnega razvoja, s poudarkom na metodologijah razvoja, metrikah in razvojnih okoljih. Svoje izkušnje je predstavil v številnih prispevkih na domačih in tujih konferencah ter revijah. Je tehnični koordinator aktivnosti Centra za objektno tehnologijo in predsednik konference OTS *Objektna tehnologija v Sloveniji*. Diplomiral, magistriral in doktoriral je na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru.

Simon Beloglavec je asistent na Inštitutu za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru in aktivni član Centra za objektno tehnologijo, v okviru katerega je sodeloval pri pripravi in izvedbi številnih seminarov delavnic iz področja jave, objektnega načrtovanja in vzorcev. Njegovo raziskovalno delo zajema javo tehnologijo, komponentni razvoj, aplikacijske strežnike in vzorce. Diplomiral je na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru, kjer pripravlja tudi svojo doktorsko disertacijo.

Matjaž B. Jurič je docent na Inštitutu za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Njegovo raziskovalno-razvojno delo obsega porazdeljene objekte, komponentne modele, vzorce, metode integracije, spletne storitve in tehnologije medorganizacijskega povezovanja. Je avtor več knjig, izdanih pri založbi Wrox Press. Diplomiral, magistriral in doktoriral je na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru.

Boštjan Kežmah je asistent na Inštitutu za informatiko na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Njegovo raziskovalno-razvojno delo obsega celovite informacijske rešitve, sodobna razvojna okolja, metode integracije in spletne storitve. Diplomiral je na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru, magistriral pa na Ekonomsko poslovni fakulteti v Mariboru.

Človeški dejavnik pri povečevanju kakovosti storitev informacijske dejavnosti

Maja Miličić

Povzetek

Prispevek obravnava dva pomembna vidika vplivov človeškega dejavnika na kakovost storitev informacijske dejavnosti: zadovoljstvo kupcev ali uporabnikov s kakovostjo storitve informacijske tehnologije glede na njihove potrebe in pričakovanja in vpliv zadovoljstva zaposlenih glede na vsebino in pogoje dela ter osebni razvoj.

Abstract

Aspects of Human Factor Impact upon Information Services

The article considers two important aspects of human factor impact upon information services: the impact of customer satisfaction with the quality of information technology services corresponding to their needs and expectations and the impact of employees' satisfaction considering the content, and conditions of work and personal growth.

1 Področje raziskovanja

Večdesetletno obdobje razvoja in uporabe informacijske tehnologije (IT) v poslovnih sistemih zaznamujejo različne faze: od začetnih korakov »preslikave« poslovnih procesov v ustrezno kodirane računalniške rešitve, prek uporabe, ki jo je pogojevala vrsta in značilnosti razpoložljive računalniške opreme, do t. i. zrele dobe, ko ni več mogoče ločevati IT od kakovosti izdelkov ali storitev, ki temeljijo na njeni uporabi.

Da bi lahko pravilno dojeli vso kompleksnost problematike, ne smemo spregledati širšega družbenega konteksta, ki determinira razvoj tehnologije. Nove tehnologije zrcalijo družbo, reproducirajo in utelešajo kompleksno kombinacijo profesionalnih, tehničnih, ekonomskih in političnih dejavnikov [6]. Invencija in uporaba novih tehnologij sta kot družbena procesa odvisna tako od zahtev in interesov družbe kot tudi od znanj in veščin posameznikov.

Informacijska dejavnost je nosilka razvoja in uporabe IT v nekem okolju. Temelji na uresničevanju kakovostnih izdelkov in storitev, ki so usklajeni s pričakovanji kupcev in uporabnikov in dosegljivi v pravem času in na pravi način. Tega ni mogoče doseči brez učinkovitega notranjega komuniciranja med zaposlenimi in managementom. Zato se v zadnjem desetletju pozornost znanstvene in strokovne javnosti vse bolj usmerja k raziskovanju socioloških in psiholoških vidikov procesov informatizacije, torej človeškega dejavnika. To hkrati

kaže na nujnost povezovanja humanističnih dognanj in izkušenj s specialističnimi, v našem primeru informacijskimi znanji. V širšem pomenu lahko govorimo o informacijski kulturi, ki se razvija pod vplivom sodobnih IT.

V pričujočem prispevku bomo z več vidikov analizirali vlogo človeškega dejavnika pri doseganju večje kakovosti storitev informacijske dejavnosti (storitev IT). Utemeljili bomo pomembnost spremljanja zadovoljstva uporabnikov s kakovostjo storitev na konkretnem primeru in tudi zadovoljstva zaposlenih v informacijski dejavnosti kot enega izmed pomembnih notranjih dejavnikov pri uresničevanju zelene kakovosti storitev.

2 Človeški dejavnik in storitve IT

Informacijska dejavnost temelji praviloma na medsebojnem sodelovanju naslednjih profilov udeležencev različnih strokovnih usmeritev:

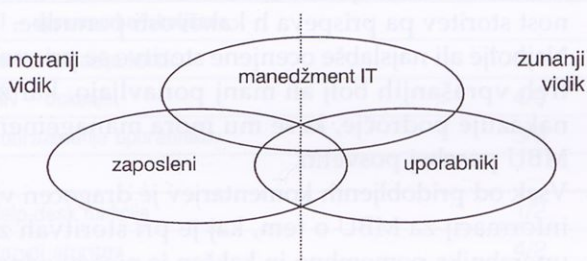
- *uporabniki*: zahteve uporabnikov opredelijo *zakaj* IT, in sicer glede na značilnosti in kakovost storitev, ki jih potrebujejo in pričakujejo
- *strokovnjaki IT*: kot informatiki iščejo odgovore na vprašanja *kaj* lahko dosežejo z IT in kot računalničarji *kako* optimalno uporabljati IT pri uresničevanju konkretne rešitve

- *menadžment*: glede na jasno poslovno vizijo in poslovne cilje *upravlja* razpoložljive človeške in druge vire za doseganje neposrednega in uspešnega učinka informacijske dejavnosti v prid konkurenčnosti, za doseganje zadovoljstva uporabnikov s kakovostjo izdelkov in storitev ter, ne nazadnje, v prid osebnostnega razvoja in zadovoljstva zaposlenih.

Glede na storitve lahko obravnavamo človeški dejavnik tudi na naslednji način:

- *zunANJI vidik*: težišče je na uporabniku, v središču pozornosti so namreč značilnosti storitev, ki jih uporabnik potrebuje in pričakuje ter hkrati ocenjuje stopnjo odstopanj od želenega, in
- *notranji vidik*: težišče je na zaposlenih, ker gre za zmožnosti informacijske dejavnosti za uresničevanje pričakovane kakovosti storitve.

V obeh primerih je management ključni povezovalni dejavnik.



Slika 1: Model človeških dejavnikov pri storitvah IT

Glede na jasna in sprejemljiva razmerja med različnimi profili udeležencev na področju informacijske dejavnosti bi bilo pričakovati, da so zadeve tudi v praksi koordinirane in optimalno usklajene, vendar temu ni tako. Ustrezno razvoju in širjenju uporabe IT na različnih področjih človeške dejavnosti se je tudi pozornost na t.i. človeške vire ali človeški dejavnik spreminjala. Tako je bila v začetnih razvojnih fazah uvajanja novih tehnologij pozornost usmerjena predvsem na strokovnjake IT, potem na managerje, sedanjost fazo razvoja pa zaznamuje usmeritev k uporabnikom in zaposlenim.

Nova verzija standarda za upravljanje kakovosti ISO 9001:2000, ki jo lahko apliciramo tudi na področje informacijske dejavnosti, posveča naslednjim vidikom obravnave človeških virov posebno pozornost [2]: (1) zadovoljstvu kupcev in uporabnikov s kakovostjo

izdelkov in storitev glede na potrebe in pričakovanja, (2) kakovosti upravljanja človeških virov s poudarkom na individualnih in socioloških vidikih upravljanja, ter (3) zadovoljstvu zaposlenih s kakovostjo dela in pogoji, v katerih delujejo.

3 Spremljanje zadovoljstva uporabnikov storitev – primer iz prakse

Spremljanje zadovoljstva uporabnikov storitev na področju bančnega kartičnega poslovanja uresničuje podjetje MBU, d. o. o.¹ (v nadaljevanju MBU) v skladu z usmeritvami in navodili mednarodnega standarda upravljanja kakovosti ISO 9001:2000. Ob krajši razlagi ciljev, koncepta in rezultatov najnovejše raziskave bomo povzeli nekatere pomembnejše izsledke, ki lahko potrdijo aktivno in pomembno vlogo uporabnikov storitev.

3.1 Dejavnost in usmeritve MBU

Dejavnost MBU zajema koordinirano projektiranje, zgradbo in uporabo skupne informacijske infrastrukture za sodobno bančno tehnologijo kartičnega poslovanja, ki omogoča različne vrste transakcij z domačimi in mednarodnimi karticami. Temeljne storitve, ki jih MBU ponuja bankam uporabnicam, so: zgradba in vzdrževanje komunikacijskega omrežja za bankomate in EFTPOS terminale, MBModul za oddajanje, prejem in obdelavo podatkov med banko in MBU, medsebojni obračuni stroškov storitev med različnimi uporabnicami ipd. Med dopolnilnimi storitvami lahko kot bolj pomembne omenimo: help-desk ali pomoč uporabnikom, izobraževanje uporabnikov, pripravo priročnikov za usposabljanje uporabnikov, razvoj ali uvajanje novih storitev in povečevanje števila bank uporabnic.

Že leta 1999 je bil med strateškimi cilji MBU definiran in tudi uspešno uresničen projekt inovacije obstoječega sistema kakovosti [5], katerega temeljni cilj je bil ugotoviti, kako banke kot uporabnice storitev MBU ocenjujejo doseženo stopnjo kakovosti storitev in kakšne so strateške razvojne usmeritve. Projekt je temeljil na izhodiščih mednarodnega standarda kakovosti ISO 9000:2000 in standarda upravljanja kakovosti ISO 9001:2000.

Izsledki navedene raziskave so spodbudili MBU k uresnitvi pomembnih ukrepov za izboljšanje kakovosti, med drugim uvajanje funkcije help-deska,

¹ MBU, d. o. o. za informatički inženjering i međubankarske usluge, Zagreb, Hrvatska.

kakovosten razvoj izobraževanja različnih profilov uporabnikov, še posebej managementa bank, hitrejši razvoj novih storitev, npr. prehoda na čip kartice ipd.

3.2 Koncept raziskave

Glede na dolgoročno strategijo povečevanja kakovosti storitev in v skladu z zahtevami standardov ISO 90001:2000, je management MBU v letu 2002 z novim projektom merjenja zadovoljstva uporabnikov želel ugotoviti, kakšna so odstopanja med ponudbo in pričakovanji uporabnikov glede na obstoječe storitve in kaj le-ti pričakujejo v prihodnje [1].

V anketo kot metodo raziskovanja je bilo vključeno 18 bank oz. 50 udeležencev, managerjev različnih profilov. Aktivno se je odzvalo 31 managerjev (62,00 %) iz 15 bank (83,33 %), in sicer:

Poslovna funkcija menedžerja	Število (%)
Poslovanje z občani	10 (32,25)
Kartično poslovanje	8 (25,81)
Informatika /IT	8 (25,81)
Druga poslovna področja	5 (16,13)
Skupaj:	31 (100,00)

Vprašalnik je zajemal 3 temeljna vprašanja o kakovosti 12 izbranih storitev: oceno obstoječe kakovosti, oceno stopnje zadovoljstva pri reševanju reklamacij ali problemov in oceno kakovosti pri uvajanju sprememb ali izboljšav. V zaključnem delu vprašalnika je bilo mogoče splošno komentirati tako samo raziskovanje kot tudi medsebojno poslovno sodelovanje banke uporabnice storitev in MBU.

Samo ocenjevanje se je izvajalo na dva načina, in sicer: (1) z obkroževanjem ene izmed ponujenih ocen (1: nezadovoljivo, 2: zadovoljivo, 3: dobro, 4: prav dobro, 5: odlično) in (2) z navajanjem komentarjev glede na zastavljeno vprašanje in konkretno storitev.

3.3 Pomembnejši izsledki raziskave

Povzeli bomo le najpomembnejše ugotovitve, ki temeljijo na ocenah in komentarjih iz izpolnjenih vprašalnikov. V tabeli 1 so prikazane po 3 storitve v okviru posameznega vprašanja z najboljšimi in najslabšimi povprečnimi ocenami. Le-te so bile izračunane na podlagi vseh odgovorov na nivoju konkretne storitve, temu ustrezno je potem opredeljen tudi rang kakovosti. Navaja se tudi število različnih komentarjev vezanih na

posamezno storitev, in sicer pozitivnih ali negativnih oz. predlogov. 'Pozitivni' komentarji podpirajo sedanjo kakovost storitve in usmeritve MBU, 'negativni' in predlogi kažejo na konkretne pomanjkljivosti ali pa nakazujejo aktivnosti za zmanjševanje le-teh.

Na podlagi analize pridobljenih povprečnih ocen in podanih komentarjev lahko kot pomembnejše povzamemo naslednje ugotovitve:

1. Ne glede na to, da je tudi najnižja povprečna ocena višja od 3,00 - dobro, bi bilo napačno sklepati o splošnem zadovoljstvu uporabnikov s kakovostjo storitev. Vsaka od pridobljenih ocen znotraj intervala ocen od 3,00 do 5,00 namreč zahteva skrbno analizo in obravnavo.
2. Udeleženci niso delali bistvenih razlik glede na vsebino in pomen storitev, npr. med temeljnimi: personalizacija kartice, EFTPOS terminali, ali dopolnilnimi storitvami: izobraževanje uporabnikov, priročniki ipd. Lahko sklepamo, da so uporabnikom enako pomembne vse storitve, različnost storitev pa prispeva h kakovosti ponudbe.
3. Najbolje ali najslabše ocenjene storitve se pri vseh treh vprašanih bolj ali manj ponavljajo, kar že nakazuje področje, ki se mu mora management MBU posebej posvetiti.
4. Vsak od pridobljenih komentarjev je dragocen vir informacij za MBU o tem, kaj je pri storitvah za uporabnika pomembno in kakšen je njegov odnos do posamezne storitve. Za MBU je to hkrati opozorilo, kateri problemi so najbolj izpostavljeni in na katere uporabnik pričakuje pozitiven odziv. Zagotovitev povratne informacije je eden izmed ključnih dejavnikov uspešnosti tovrstnih raziskovanj [3].
5. Med različnimi komentarji je bilo zaslediti zanimljivo majhno število tistih, ki se nanašajo neposredno na IT, kar pomeni, da se uporabniki opredeljujejo predvsem glede na svoja pričakovanja in se ne obremenjujejo z lastnostmi informacijsko-tehnološke podpore storitvam.

Na splošno se je večina komentarjev nanašala na časovne, stroškovne, razvojne vidike storitev in še posebej na problematiko medsebojnega sodelovanja med bankami in MBU pri izboljševanju kakovosti in razvoju storitev nasploh. Navedeno ugotovitev potrjujejo nekateri komentarji iz zaključnega dela raziskave:

- najvišjo oceno lahko damo poslovnemu sodelovanju - odlično

Naziv storitve	Število komentarjev	Povprečna ocena	Standardni odklon	Rang kakovosti
Ocena: 1-nezadovoljivo;...; 5-odlično	pozit./negat., predlogi			
I – Obstoječa kakovost storitev				
Personalizacija kartic	8/1	4,45	0.51	1
Izobraževanje uporabnikov	5/1	4,26	0.66	2
PIN – tiskanje	7/3	4,17	0.66	3
...
Priročniki za uporabnike	3/5	3,61	0.88	10
Razvoj storitev	1/10	3,30	0.88	11
Delovanje EFTPOS terminalov	6/6	3,26	1.10	12
II – reševanje reklamacij/problemov				
Personalizacija kartic	5/0	4,50	0.51	1
PIN – tiskanje	6/0	4,43	0.50	2
Medsebojni obračuni bank	4/2	4,33	0.76	3
...
Priročniki za uporabnike	2/2	3,71	0.85	10
Razvoj storitev	3/2	3,48	0.94	11
Delovanje EFTPOS terminalov	3/3	3,39	2.13	12
III – spremembe/izboljšave				
Personalizacija kartic	5/1	4,33	0.57	1
PIN – tiskanje	4/0	4,33	0.48	1
Izobraževanje uporabnikov	5/0	4,12	0.43	3
...
Help-desk funkcija	1/2	3,63	1.21	10
Razvoj storitev	4/2	3,58	0.62	11
Delovanje EFTPOS terminalov	2/5	3,25	0.79	12

Tabela 1: Pregled števila komentarjev in povprečne ocene kvalitete storitev

- pri razvoju je treba več pozornosti posvetiti željam bank
- na letni ravni naj bi imel MBU 30% samostojnih projektov, banke naj bi odločale o 70% razvojnih projektov
- MBU mora spremljati razvoj na svetovnem tržišču in temu ustrezno razvijati storitve za potrebe svojih bank uporabnic
- MBU naj bi prevzel več razvojnih usmeritev, izboljšal organizacijo in se usmeril k prodaji storitev s konkurenčno ceno
- pogrešajo bolj aktiven in odločen pristop glede skupnih akcij pri usklajevanju in sprejemanju pomembnejših sklepov o obstoječih ali novih storitvah ipd.

Navedene ugotovitve potrjujejo, da uporabniki aktivno spremljajo kakovost storitev, se zavedajo svoje

vloge pri tem in kažejo pripravljenost sodelovati z različnih vidikov.

4 Zadovoljstvo zaposlenih kot notranji dejavnik kakovosti storitev

Pričakovano kakovost storitve zagotavljajo nedvomno vsi zaposleni v informacijski dejavnosti, z različnim deležem in vlogo, kar pa se v praksi velikokrat spregleda. Na splošno pa velja, da ni zadovoljnega uporabnika, če ni zadovoljnega zaposlenega.

Narava dela, potreba po nenehnem izobraževanju in usposabljanju, zahtevnost in kompleksnost nalog, stresno počutje, izrazita individualnost profesionalcev v IT so le nekateri tipični spremljevalci informacijske dejavnosti. Med moteče dejavnike pri zaposlenih v IT lahko štejemo tudi težave pri sodelovanju, nezadostno poznavanje poslovnih ciljev in procesov, nizko

raven motiviranosti ali nezadovoljstvo z vrsto in načinom dela, osebnim razvojem in priznanji za opravljeno delo ipd.

V prid navedenim ugotovitvam lahko navedemo nekatere izsledke internega merjenja in analize zadovoljstva zaposlenih MBU, ki je sledilo že navedeni raziskavi zadovoljstva uporabnikov s storitvami.

Tako so bili kot bolj *kritični* dejavniki navedeni naslednji:

- motivacija in nagrajevanje na individualnem nivoju (povprečna ocena: 2,24),
- pretok informacij o poslovnih rezultatih organizacije (povprečna ocena: 2,45),
- učinkovitost in organiziranost poslovanja (povprečna ocena: 2,89).

Med *pozitivnimi* dejavniki so bili najboljše ocenjeni naslednji:

- pogoji dela in delovni prostor (povprečna ocena: 3,83),
- medsebojni odnosi (povprečna ocena: 3,47),
- upoštevanje kompetentnosti s strani vodilnih in sodelavcev (povprečna ocena: 3,00).

Med komentarji in predlogi glede na različne vidike poslovnega delovanja MBU je prevladoval pozitiven skupni interes v smeri različnih izboljšav na ravni organizacije in šele potem individualni motivacijski dejavniki in pričakovanja. Naj povzamemo nekatere zanimivejše komentarje:

- narediti poslovno politiko in cilje jasne vsakemu zaposlenemu,
- podpirati in spoštovati uveljavljeno hierarhijo,
- ocenjevati, vrednotiti in nagrajevati opravljeno delo,
- izobraževati in usposabljanje zaposlene na področju komuniciranja in medsebojnega sodelovanja.

Vsako ustvarjalno delo in tudi delo strokovnjakov IT spremlja visoka zavest o potrebi po nenehnem izpopolnjevanju lastnih znanj in spretnosti. Zadeva je bolj kompleksna, če je treba izstopiti iz temeljnega strokovnega okvira, kar se v praksi pogosto dogaja, kot na primer, če visoko kvalificiran strokovnjak prevzame funkcijo managerja in s tem tudi naloge, ki zahtevajo dopolnina znanja iz ekonomije, prava, sociologije itd.

Tukaj med moteče dejavnike lahko prištejemo čisto človeško naravnost, da tista znanja, ki jih

posameznik ne pozna podrobno, praviloma podcenjuje. Le-teh se praviloma loteva ohlapno na nivoju splošnega informiranja in zato ne analitično. Posledica tega je tudi kopičenje problemov in postopoma vse bolj formalno komuniciranje, ki zmanjšuje možnosti za rešitve.

V navodilih, ki jih navaja ISO standard z vidika upravljanja človeških virov, so prioritetni: izobraževanje, usposabljanje, uporaba orodij in pridobivanje izkušenj. Vsekakor gre za znanja in spretnosti za zagotovitev visoke stopnje profesionalnega opravljanja nalog vsakega posameznika. Z vidika osebnih potreb in motivacije zaposlenega je treba spremljati vsaj še nekatere psihosociološke dejavnike, med katerimi so sposobnosti timskega dela, ciljna usmerjenost in motiviranost za nalogo, upravičeno pričakovanje priznanj in nagrade za dobro opravljeno delo, občutek enakovredne obravnave ter spoštljivega obravnavanja v poslovnem okolju itd. Zato je predpogoj za zagotavljanje boljših storitev poleg stalnega izobraževanja zaposlenih, uporaba sodobnih metod komuniciranja, koordinacije in sodelovanja [4]. Gre pravzaprav za celovit pristop k vsakemu zaposlenemu in ustvarjanje splošnega pozitivnega vzdušja znotraj informacijske dejavnosti.

5 Sklep

Spremljanje človeškega dejavnika z vidika kakovosti storitev IT je kontinuiran proces. Povzetek rezultatov konkretne raziskave potrjuje pomen spremljanja zadovoljstva uporabnikov z vidika kakovosti storitev. Kultura poslovnega okolja, pozitivna stališča in občutek pripadnosti, konstruktivno obnašanje so nekateri dejavniki, ki vplivajo na uspešnost in učinkovitost informacijske dejavnosti glede na zadovoljstvo zaposlenih.

Naloge managementa z vidika upravljanja človeških dejavnikov so zahtevne in kompleksne. Uresničevanje le-teh zahteva večdimenzionalno združevanje managerskih znanj z nekaterimi znanji s področja psiholoških in socioloških ved. Med njimi so ključnega pomena tista, ki prispevajo h kakovosti medsebojnega komuniciranja, povečevanju motiviranosti na individualnem nivoju ter ustvarjalnem sodelovanju vseh udeležencev informacijske dejavnosti.

6 Literatura

- [1] Bauer, S.:
Projekt 'Mjerenje zadovoljstva korisnika usluga MBNET-a',
povratna informacija za sudionike i banke korisnice
usluga, MBU d.o.o. za informatički inženjering i
međubankarske usluge, Zagreb, veljača 2003.
- [2] ISO 9001:2000:
Quality Management Systems - Requirements, ISO/DIS
9004:2000.
- [3] Hill, N., Brierley, J., McDougall, R.:
How to Measure Customer Satisfaction, Gower
Publishing Limited, UK, 1999.
- [4] Križman, I.:
Vloga uporabnikov podatkov pri celostnem obvladovanju
kakovosti storitev državne statistike, magistrsko delo,
Fakulteta za družbene vede, Univerza v Ljubljani,
Ljubljana, 2002.
- [5] Miličič, M.:
Zadovoljstvo uporabnikov s kakovostjo storitev v
bančnem kartičnem poslovanju, Uporabna informatika,
IX (2001), št. 2, str. 60–66.
- [6] Sorčan, S.:
Tehnologija kot materialna kultura, časopis za kritiko
znanosti – tematska številka Rizična družba, Študentska
založba, let. XXV, 1997, št. 183, Ljubljana, str. 41–66.

Dr. Maja Miličič je doktorirala leta 1995 na Ekonomsko-poslovni fakulteti Univerze v Mariboru na področju informacijskih znanosti. Leta 2000 je bila izvoljena v naziv docentke za področje poslovne informatike na Ekonomski fakulteti Univerze v Ljubljani. Deset let je tudi sodelovala kot asistentka pri predmetu metode komuniciranja s takratno Fakulteto za elektrotehniko in računalništvo Univerze v Ljubljani. Njene najpomembnejše izkušnje, znanstveno in raziskovalno delo so s področja informatizacije v bančništvu, pa tudi izvirne raziskave o psihološko-socioloških vidikih procesov informatizacije.

23. septembra 2003 je umrla Maja Miličič,
ne da bi videla izid članka,
ki ga je z zadnjimi močmi napisala za to revijo.

Premagala jo je težka, neozdravljiva bolezen,
ki ji je dolga leta pogumno kljubovala.

Maja je izredno cenila človeške odnose in jih tudi znala gojiti;
ni slučajno,
da se je posvetila raziskovanju komuniciranja v informatiki.

Imela je številne prijatelje.

Bila je dejavna članica društva,
s strokovnimi prispevki v reviji Uporabna informatika,
z referati na posvetovanjih Dnevi slovenske informatike,
v zadnjem času pa tudi v jezikovni sekciji.

Pogrešali jo bomo.

■ Informatika med tradicionalnimi in virtualnimi organizacijami

Dr. Gene Bavec, Univerza na Primorskem, Fakulteta za management, Cankarjeva 4, Koper
cene.bavec@guest.arnes.si

Povzetek

Razumevanje sodobnih organizacijskih paradigem postaja izjemno pomembno tudi za informatike, ki z inovativnimi tehnološkimi rešitvami pogosto prehitujejo svoje kolege, ki naj bi skrbeli za organizacijo in vsebinske vidike novih storitev. V prispevku so povzeti pomebnejši deli sodobne teorije organizacije, posebej Mowshowitzovo preklopno načelo, ki je slovenskim informatikom manj znano. Prikazan je tudi eden od načinov za ocenjevanje stopnje virtualizacije, ki je bil uporabljen na primeru slovenske organizacije. V razpravi je podano razmišljanje o odnosu med informatiko in organizacijo v luči novih organizacijskih teorij, ki med drugim napovedujejo zблиževanje informacijskih in organizacijskih ved.

Abstract

Informatics between Traditional and Virtual Organizations

With innovative technological solutions, IT professionals are often ahead of those responsible for organization. For that reason, they have to understand new organizational paradigms and even participate in their development. The paper presents basics of theory of virtual organization, particularly less known Mowshowitz's Switching Principle. The discussion is focused on relations between IT and organization in the light of new organizational theories. One of the most intriguing questions is how to define and measure organizational virtuality. An example of a Slovenian organization is presented and discussed.

1 Uvod

Uvajanje informacijske in komunikacijske tehnologije je praviloma povezano s spremembo organizacije in uvajanjem novih poslovnih modelov. To je seveda ugotovitev, ki ji v sodobnem poslovnem svetu nihče ne oporeka, vendar je kljub temu predmet neprestanih razprav med informatiki in menedžmentom, ki je zadolžen za organizacijo podjetja. Zato je zanimivo pogledati sodobne trende na področju organizacijskih ved tudi z zornega kota informatikov in računalniških strokovnjakov, ki z inovativnimi tehnološkimi rešitvami pogosto prehitujejo svoje kolege, ki naj bi skrbeli za organizacijo in vsebinske vidike novih storitev.

Z menedžerskega zornega kota je lažje obvladovati zahtevne računalniške projekte kot pa organizacijske spremembe z vsemi posrednimi in neposrednimi problemi, ki jih prinašajo. Razlogov je več. Eden je prav gotovo, da je vodenje računalniških projektov metodološko izjemno dobro opredeljeno in objektivizirano za razliko od menedžmenta organizacijskih sprememb, kjer prevladujejo veliko manj objektivni kriteriji, povezani z ekonomskimi in sociološkimi vidiki organizacij. Drugi razlog je menedžment, ki si intuitivno izbira lažje in manj tvegane poti – tehnološke projekte lahko naročijo pri zunanjih izvajalcih, organizacijske spremembe pa morajo

izpeljati sami. To zahteva veliko znanja in pogosto celo osebnega poguma. Izkušnje namreč kažejo, da so organizacijske spremembe za menedžment precej bolj tvegane kot pa večina tehnoloških projektov.

Čeprav je to nekoliko vprašljiva teza, je vsaj v Sloveniji očitno, da imamo več dobrih računalniških strokovnjakov kot strokovnjakov za organizacijo. Zato smo pogosto priča paradoksalnemu stanju, da informatiki prevzemajo nase odgovornost za organizacijske spremembe, za katere niso niti pristojni niti usposobljeni. Najbrž je to eden od razlogov, da se veliko informacijskih projektov konča na pol poti z dobro tehnološko in precej slabšo organizacijsko rešitvijo. Obratni primeri so precej bolj redki.

Seveda so sčasoma zaradi okoliščin informatiki pridobili precejšnje organizacijske izkušnje in so skoraj povsem nadomestili strokovnjake za »organizacijo dela« v večini podjetij. Vendar nove organizacijske paradigme, posebej iz anglosaškega sveta, zelo počasi prodirajo v naše okolje, ki je pod močnim vplivom tradicionalne srednjeevropske organizacijske šole. Informatiki teh problemov večinoma nimajo, saj je njihova usposobljenost veliko bolj globalna, zaradi naglih tehnoloških sprememb pa so postali prilagodljivi

in odprti za novosti. Kompleksnost novih organizacijskih oblik, kot so na primer virtualne organizacije, pa zahteva veliko globlje in specializirano znanje, ki ga od informatikov ne moremo kar zahtevati. Če parafraziramo znan izrek o vojakih in politikih, bi lahko rekli, da postajajo sodobne organizacije presna zadeva, da bi jih lahko prepuščali informatikom. Zato je za menedžerje in informatike pomembno, da razvijejo enoten pogled na sodobne organizacije in si ustrezno razdelijo naloge in pristojnosti.

Ena od inovacij, ki jo lahko v veliki meri pripišemo informatikom, so brez dvoma mrežne in geografsko porazdeljene organizacijske oblike, ki temeljijo na intenzivni uporabi interneta. Vedno več organizacijskih komunikacij poteka prek interneta, s čimer se v organizacijo vnašajo elementi virtualnosti, saj je »realnih« medosebnih stikov vedno manj. Proces se je začel intuitivno, saj so bile organizacijske spremembe posledica inovativnih tehnoloških rešitev, ki so pripeljale do nastanka »pika kom« podjetij. Sprožil se je plaz organizacijskih inovacij, ki mu še daleč ni videti konca. Raziskovalci in menedžerji se večinoma strinjajo, da bosta informacijska družba in digitalna ekonomija sloneli na novi organizacijski paradigmi, ki jo danes imenujemo virtualna organizacija. Čeprav je izraz dvoumen in zahteva podrobnejšo opredelitev, smo v zadnjem času priča nastanku pravih virtualnih podjetij, ki se izrazito razlikujejo od hierarhičnih in centraliziranih organizacij, ki so bile prevladujoča organizacijska oblika industrijske družbe.

Nova organizacijska paradigma je postala privlačno področje tudi za raziskovalce, ki so začeli povezovati informacijske in organizacijske vede v izrazito multidisciplinarni pogled na organizacijo. Rezultati so že vidni, saj se kažejo obrisi nove teorije organizacije, ki se v nekaterih pogledih radikalno razlikuje od tradicionalnih. Sodobna teorija virtualnih organizacij teži za tem, da bi postala splošna teorija organizacije, ki bi zajela tako tradicionalne (hierarhične, matrične in mrežne) oblike kot tudi nove virtualne organizacije. Trenutno jo tvorita dva tokova, ki pa se približujeta in stekata v isto strugo. Prvi tok tvorijo pretežno tehnološki pogledi, kjer se virtualnost odraža v geografski porazdeljenosti in sodobnih komunikacijah. Drugi tok so bolj menedžersko in poslovno usmerjeni pogledi, ki vidijo virtualnost predvsem v izjemni organizacijski prilagodljivosti.

Koncept virtualnih organizacij se razvija izjemno hitro in se naglo oddaljuje od prvotnega koncepta

»pika kom« podjetij ter postaja splošen organizacijski koncept. Virtualne organizacije, kot jih razumemo danes, so vrh ledene gore, saj vidimo le manjši del gmote, ki jo imenujemo sodobna organizacija. Temeljni izzivi sodobnih organizacij so odmaknjeni od vsakdanjega blišča sodobnih tehnologij in interneta in so osredotočeni na posameznika, ki je v industrijski družbi izgubil svojo identiteto. Čeprav je sodobna tehnologija na videz odrinila človeka v ozadje, pa razvoj virtualnih organizacij kaže, da mu v resnici vrača veljavo, saj gradi nove organizacijske oblike predvsem na njegovi kreativnosti.

V tem prispevku bomo povzeli nekatere najnovejše izsledke teorije virtualnih organizacij in poskušali oceniti, v katero smer se bo razvijala nova organizacijska paradigma. Virtualne organizacije so veliko bolj zapleteni sistemi, kot se je zdelo še pred nekaj leti, zato so intuitivni pristopi k njihovi izgradnji vedno manj uspešni. Poznavanje temeljnih teoretičnih izhodišč, na katerih slonijo sodobne organizacije, postaja enako pomembno znanje tako za menedžerje kot informatike. Za informatike pa je vpogled v teorijo virtualnih organizacij pomemben predvsem zato, ker jim odstira nove, njim neznane poglede na organizacijo. To velja še posebej za tiste vidike sodobnih organizacij, ki so tehnološko nevtralni.

2 Elementi teorije virtualnih organizacij

Ko govorimo o teoriji virtualnih organizacij, seveda ne govorimo o dokončno razviti teoriji, temveč o drobcih mozaika, ki se postopoma oblikuje v globalno sliko nove organizacijske paradigme. Še vedno je veliko terminoloških nejasnosti in dvoumnosti, določeni koncepti so si povsem nasprotni, še vedno je malo objektivnih meril, ki govorijo o kvaliteti in učinkovitosti novih organizacijskih oblik in podobno (Skyrme, 1998). Kljub temu so virtualne organizacije izjemno pomemben razvojni korak, saj se kljub različnim pogledom skoraj vsi raziskovalci in menedžerji strinjajo, da bodo prej ali slej postale prevladujoča organizacijska oblika bodoče družbe.

Izraz »virtualna organizacija« brez dvoma odraža nekatere najpomembnejše lastnosti nove organizacijske oblike, vendar je zavajajoč, saj nekako zlorablja modno besedo »virtualnost«. Organizacije so vedno človeške združbe in jih ne moremo nadomestiti z nečim, česar ni, kar naj bi pomenila virtualnost (Drucker, 1999). To pomeni, da moramo natančno vedeti, kaj je v organizaciji lahko virtualno in kaj ne more biti.

Zato se celotna teorija virtualnih organizacij začneja pri opredeljevanju novih izrazov in redefiniciji tradicionalnih pojmov, ki jih srečujemo v organizacijskih in informacijskih vedah.

Seveda se tudi pri obravnavi virtualnih podjetij srečujemo s podobnimi terminološkimi dvomnostmi kot v tradicionalni teoriji organizacije, kjer se prepletata pojma organizacije kot strukturne urejenosti ter podjetja kot združbe posameznikov (Bavec, 1995, Sandhoff, 1999). Tako kot pri tradicionalni teoriji organizacije sta tudi v novi teoriji izraza "organizacija" in "podjetje" pogosto sinonima, saj je virtualna organizacija le sinonim za virtualno podjetje ali korporacijo. V nekaterih primerih moramo zaradi javnosti natančno določiti, kdaj mislimo na strukturo urejenosti in kdaj na podjetje ali združbo ljudi (Saabeel et al., 2002).

Virtualno podjetje

Ena od najpogosteje citiranih definicij virtualnega podjetja je, da je to začasna mreža neodvisnih institucij, podjetij ali specializiranih posameznikov, ki se ob intenzivni uporabi informacijske in komunikacijske tehnologije združujejo z namenom, da bi na trgu dosegli določeno prednost pred konkurenti. Navzven deluje mreža kot samostojno ali enovito podjetje, ki je zelo blizu temu, kar si v praksi predstavljamo pod »pika kom« podjetji. V tej definiciji se skriva nekaj pomembnih novosti. Virtualna organizacija je predvsem začasna tvorba, ki je ustanovljena zato, da lahko partnerji dosežejo točno določene poslovne cilje in deluje samo toliko časa, dokler taki cilji niso doseženi. Prave virtualne organizacije praviloma ne propadajo, saj je njihova temeljna značilnost, da ugasnejo, ko niso več v funkciji. Zato je tudi vprašljiva teza o množičnem propadanju »pika kom« podjetij v letih 2000 in 2001. Veliko verjetneje je, da je šlo za množično prestrukturiranje, saj je bi bil njihov propad le redko povezan z velikimi poslovnimi izgubami. Pomembna značilnost virtualnih organizacij je tudi neodvisnost partnerjev, ki v virtualni organizaciji opravljajo samo tisto funkcijo, za katero so najbolj usposobljeni (angl. core competence) in z njo ne konkurirajo ostalim partnerjem, temveč jih dopolnjujejo.

Virtualne organizacije so v osnovi mrežne organizacije, ki jih obravnava tudi tradicionalna teorija (Lipnack in Stamps, 1994). Čeprav so virtualne organizacije v zasnovi zelo podobne mrežnim, so nekatere značilnostih virtualnih organizacij veliko bolj izrazite:

- prestopanje organizacijskih meja (pristojnosti posameznih menedžerjev se raztezajo prek meja njihove matične organizacije, podobno velja tudi za druga delovna mesta);
- medsebojno dopolnjevanje partnerjev (vsak partner vstopa v virtualno organizacijo z dejavnostjo, v kateri je boljši in konkurenčnejši od ostalih);
- geografska porazdelitev (virtualne organizacije so praviloma geografsko razpršene na večjih razdaljah);
- menjanje udeležencev ali partnerjev (do zamenjave partnerjev pride, ko se pokaže, da je na nekem področju drug partner učinkovitejši in da so stroški zamenjave manjši od ekonomske koristi, ki jo zamenjava prinaša);
- enakopravnost udeležencev (za razliko od klasičnih poslovnih povezav, kjer skoraj vedno prevladuje najmočnejši partner, se v virtualnih organizacijah teži po skoraj popolni enakopravnosti);
- elektronska komunikacija (brez informacijske in telekomunikacijske tehnologije virtualne organizacije niso možne).

S poslovnega zornega kota sta osnovna cilja virtualne organiziranosti optimalna izraba finančnih, tehnoloških, človeških in drugih virov, s katerimi razpolagajo partnerji ter največja možna odzivnost na zunanje okoliščine. Virtualne organizacije praviloma izrabljajo več virov, kot jih realno imajo, s tem da se navidezno združujejo z drugimi organizacijami in izrabljajo še njihove vire. Oba pogoja, optimizacija virov in prilagodljivost, sta nujna za poslovanje v globalni ekonomiji, zato delujejo najuspešnejše virtualne organizacije globalno in so najpogostejše v okoljih, ki slonijo na globalni ekonomiji, kot so ZDA in razviti del Evrope (Papows, 1999).

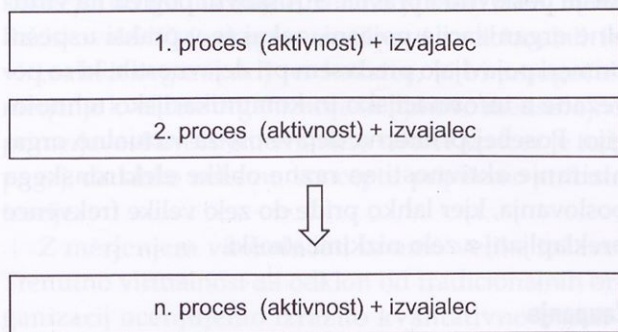
Preklopno načelo

Zgornja definicija virtualnega podjetja je zelo blizu informatikom in tudi menedžmentu, ki vidi v informacijskih in komunikacijskih tehnologijah predvsem možnost za nove proizvode in storitve. Vendar je preveč intuitivna in ne nudi dovolj prostora za objektivno primerjavo med virtualnimi in tradicionalnimi organizacijami ter za oceno njihove učinkovitosti. Šele v zadnjih nekaj letih so se začele pojavljati tudi druge definicije in pogledi na virtualnost, ki napovedujejo rojstvo razmeroma visoko strukturirane teorije (Ude, 2001). Ali bo znana pod izrazom teorija virtualnih organizacij ali kakšnim drugim imenom, pa bo pokazal čas.

Najbolj odmeven pogled na virtualne organizacije, ki je tehnološko povsem nevtralen, je razvil Mowshowitz (1999), ki je uvedel koncept preklopnega načela (ang. switching principle). Preklopno načelo je teoretična novost, saj je v tradicionalni teoriji organizacije povsem neznano, pomeni pa temeljno lastnost virtualnih organizacij ter jedro nove organizacijske paradigme. Preklopno načelo omogoča posplošitev tradicionalnega koncepta organizacije, saj lahko postavimo tezo, da je virtualna organizacija vsaka organizacija, katere delovanje temelji na preklopnem načelu. Ker je koncept nov tudi informatikom, ga bomo v nadaljevanju nekoliko podrobneje pojasnili.

Pri tradicionalnem pogledu na organizacijo se procesi (aktivnosti) in izvajalci procesov obravnavajo kot nerazdružljivo povezana celota (slika 1). Že v postopku načrtovanja se za vsak proces določi (načrtovanje), kdo ga bo izvajal. Kasnejše odstopanje od načrtovanih izvajalcev je znak slabe organizacije. Menedžment je vedel, da lahko s finančnimi, razvojnimi, tehnološkimi in kadrovskimi viri izdeluje le določen tip proizvodov ali storitev določene kvalitete in določene tehnološke zahtevnosti. Zato je proizvodnjo in izbor proizvodov prilagajal zmožnostim svojega podjetja. Lahko se je sicer zatekel k iskanju zunanjih izvajalcev ali strateškemu partnerstvu, vendar je bil to le izhod v sili za manjši del dejavnosti. V glavnem je vedno veljalo načelo – čim več je potrebno narediti s svojimi viri v svojem podjetju.

Koncept virtualne organiziranosti pa je po Mowshowitzu (1997) radikalno drugačen. Ta radikalnost na prvi pogled niti ni tako dramatična, vendar podrobnejši razmislek pokaže, da prekinjamo s konceptom organizacije, ki je prevladovala v industrijski družbi. Pri načrtovanju virtualnih organizacij so

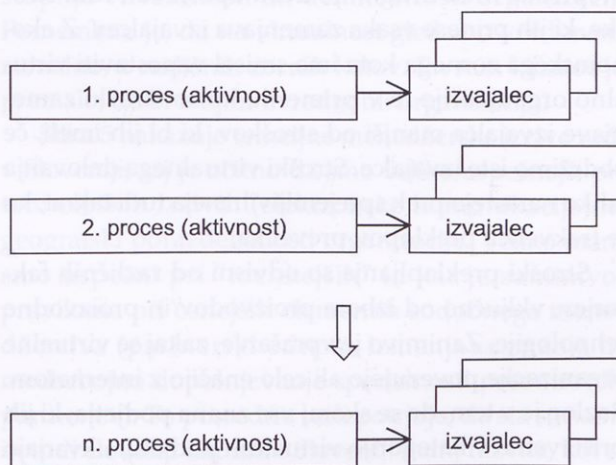


Slika 1: Pri menedžmentu tradicionalnih organizacij se procesi (aktivnosti) in izvajalci načrtujejo kot povezana in neločljiva celota

procesu in možni izvajalci v fazi načrtovanja popolnoma ločeni (slika 2). To pomeni, da pri načrtovanju določenih proizvodov ali storitev ne razmišljamo o tem, kaj lahko naredi naše podjetje, kaj naj bi naredili naši poslovni partnerji ali kaj bi lahko kupili na trgu, ampak samo to, kateri procesi in aktivnosti so potrebni za določen proizvod ali storitev. Šele v fazi izvajanja virtualno organizirane aktivnosti (na primer izdelava določene komponente ali določena storitev v vrednostni verigi) določimo, kateri izvajalec bo opravljal določeno aktivnost. Kot bomo videli kasneje, je osnovni problem, ki ga je treba rešiti, kako se organizirati, da se lahko izvajalci dinamično izbirajo in tudi zamenjujejo.

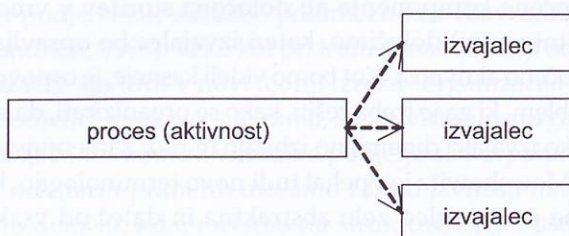
Mowshowitz je vpeljal tudi novo terminologijo, ki je na prvi pogled zelo abstraktna in daleč od vsakdanjega poslovnega jezika. Procese in aktivnosti je v fazi načrtovanja imenoval abstraktne zahteve, saj niso povezane s konkretnimi izvajalci. *Abstraktne zahteve* so logično definirane potrebe po določenih aktivnostih, ki so opredeljene ne glede na to, kdo jih bo izvajal. *Konkretni izvajalci*, ki bi lahko izvedli določene aktivnosti, so z zornega kota virtualne organizacije le viri, ki se lahko uporabijo za izvajanje abstraktnih zahtev.

Popolno ločevanje abstraktnih zahtev od konkretnih izvajalcev je bistvo virtualnih organizacij, ki zagotavlja izjemno prilagodljivost in je neprimerljivo večja kot pri tradicionalnih organizacijah. Spet nekoliko poenostavljeno rečeno, v virtualno organiziranih podjetjih lahko skoraj trenutno zamenjamo izvajalca določene aktivnosti, saj so bile že v osnovi načrtovane popolnoma neodvisno od konkretnega izvajalca.



Slika 2: Pri menedžmentu virtualnih organizacij se procesi (aktivnosti) načrtujejo ločeno od izvajalcev

Povzemimo še nekaj izrazov, ki jih vpeljuje teorija virtualnih organizacij. Metoda vključevanja (alokacije) ali izključevanja (dealokacije) izvajalcev se imenuje preklapljanje (slika 3). Matematično rečeno vsebujejo virtualno organizirane aktivnosti množico konkretnih izvajalcev, množico abstraktnih zahtev in postopke, ki preslikajo izvajalce na zahteve.



Slika 3: Preklopno načelo (dinamična alokacija možnih izvajalcev)

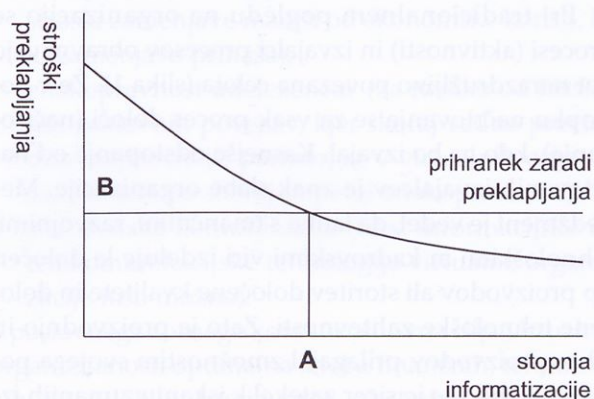
Celoten koncept preklapljanja je Mowshowitz imenoval preklopno načelo in je po njegovem mnenju osnovna teoretična novost, ki jo prinaša nova paradigma. V primeru, ko v organizaciji ni dinamične zamenjave izvajalcev, se virtualna organizacija pretvori v tradicionalno. Zanimivo je, da so bile enostavnejše oblike preklopnega načela znane že pred internetnim obdobjem še posebej v logistiki. Pri kompleksnih transportnih sistemih se je pogosto sproti odločalo, kdo izmed možnih prevoznikov bo opravil določen prevoz. Seveda je bila uporaba tega načela omejena predvsem z obstoječo tehnologijo.

Praktična uporaba preklopnega načela seveda ni preprosta. Med drugim je treba upoštevati tudi stroške, ki jih prinese vsaka zamenjava izvajalcev. Z ekonomskega zornega kota ima smisel vzpostaviti virtualno organizacijo le v primerih, ko so stroški zamenjave izvajalca manjši od stroškov, ki bi jih imeli, če obdržimo iste izvajalce. Stroški virtualnega delovanja lahko narastejo prek sprejemljivih meja tudi takrat, ko je frekvenca prekopov prevelika.

Stroški preklapljanja so odvisni od različnih faktorjev, vključno od izbora proizvodov in proizvodne tehnologije. Zanimivo je vprašanje, zakaj se virtualne organizacije povezujejo ali celo enačijo z internetom. Razlog je v tem, da se skoraj vsa znana podjetja, ki jih prištevamo v kategorijo virtualnih podjetij, ukvarjajo s takimi dejavnostmi, pri katerih se stroški morebitne zamenjave partnerjev zmanjšujejo, če se stopnja informatizacije povečuje. Če Amazon.com v svoj sistem

vključi ali pa izključi določeno založbo, se to praktično ne odraža na stroških njihovega poslovanja. Celoten sistem je namreč tako visoko informatiziran, da je to le preprosto tehnično vprašanje.

Slika 4 na zelo shematiziran in poenostavljen način prikazuje odnos med informatizacijo, stroški preklapljanja in pragom, od katerega je preklapljanje ekonomsko upravičeno. Dokler stopnja informatizacije ne doseže ravni A, so stroški preklapljanja večji od prihranka zaradi izbora ugodnejšega izvajalca. Izbor novega izvajalca postane upravičen šele, ko je stopnja informatizacije višja od ravni A.



Slika 4: Stroški preklapljanja (B) in stopnja informatizacije (A) določajo, od kje naprej je preklopno načelo ekonomsko upravičeno

Če pogledamo, katere so tiste dejavnosti, ki se najlažje virtualizirajo, vidimo, da so to predvsem informacijske storitve, ki so zasnovane na uporabi interneta in informacijskih tehnologij. Tu pride do izraza učinkovita uporaba informacijske in komunikacijske tehnologije, ki lahko v določenih primerih radikalno zniža stroške, kar preklapljanje in s tem virtualne organizacije poslovno upraviči. Stroškovni pogled na virtualne organizacije pojasni, zakaj se v praksi uspešni primeri pojavljajo predvsem pri dejavnostih, ki so povezane z informacijsko in komunikacijsko tehnologijo. Posebej primerna dejavnost za virtualno organiziranje aktivnosti so razne oblike elektronskega poslovanja, kjer lahko pride do zelo velike frekvence preklapljanj z zelo nizkimi stroški.

Zaupanje

Posebno mesto v teoriji virtualnih organizacijah ima zaupanje, ki je postalo samostojna ekonomska kategorija. Sistemi, kjer je stopnja zaupanja visoka, so

cenejši in učinkovitejši od sistemov, kjer je ta nizka. Zaupanje pomeni manj nadzornih in obrambnih mehanizmov, večjo poslovno varnost in podobno. Vsi vemo, da je v primeru internetnih organizacij zaupanje zelo pomembno, saj svojega poslovnega partnerja ne vidimo in ga ne moremo preverjati in nadzorovati s klasičnimi metodami. Skleniti posel in poslati številko kreditne kartice prek interneta pomeni, da zaupamo partnerju, ki za nas obstaja samo v virtualnem svetu interneta. Tako zaupanje ni slepo, saj uporaba elektronskih podpisov in druge varovalne metode zmanjšuje možnost zlorab.

Zaupanje je del Mowshowitzove teorije in preklonnega načela. Kot smo videli, lahko zamenjavo partnerjev finančno vrednotimo, vendar vedno ostane določeno tveganje, povezano z vprašanjem, ali je novi partner dovolj zanesljiv. Če v virtualni organizaciji en partner zataji ali prekine vrednostno verigo, ogrozi vse ostale, saj ga praviloma ne morejo nadomestiti s svojimi viri. Edino, kar jim preostane, je, da »preklopijo« na drugega partnerja, kar pa je povezano s stroški in izgubo časa. Zato se za izbor novega partnerja odločimo šele, ko sta izpolnjena oba pogoja: zamenjava je ekonomsko upravičena in novemu partnerju dovolj zaupamo. Očitno je zaupanje eden od razlogov, da virtualne organizacije nastajajo predvsem v ekonomsko razvitih okoljih, kjer je poslovna kultura in z njo povezano zaupanje dovolj visoko.

Stopnja virtualnosti

Organizacij ne moremo razdeliti na tradicionalne in virtualne. Pravih virtualnih organizacij je izjemno malo, veliko več je takih, ki kažejo večje ali manjše znake virtualnosti. To pomeni, da so nekje med tradicionalnimi in virtualnimi. Seveda bi nas zanimalo, kako virtualna je določena organizacija in koliko je učinkovitejša od tradicionalne, vendar je takih meril malo. Raziskovalci iščejo skupne značilnosti tradicionalnih in virtualnih organizacij ter kazalce, ki bi omogočali merjenje kvalitete organizacije in njihovo medsebojno primerjavo. Konsistenten sistem takih kazalcev je tudi pogoj, da lahko nekemu konceptu pripišemo attribute teorije.

Z merjenjem virtualnosti imamo velike težave. Trenutno virtualnost ali odklon od tradicionalnih organizacij ocenjujemo izrazito kvalitativno in subjektivno. Pristopi k ocenjevanju virtualnosti so zelo parcialni in omejeni na različne interpretacije virtualnosti (Bauer, Köszegi, 2003). Na sliki 5 je prikazana

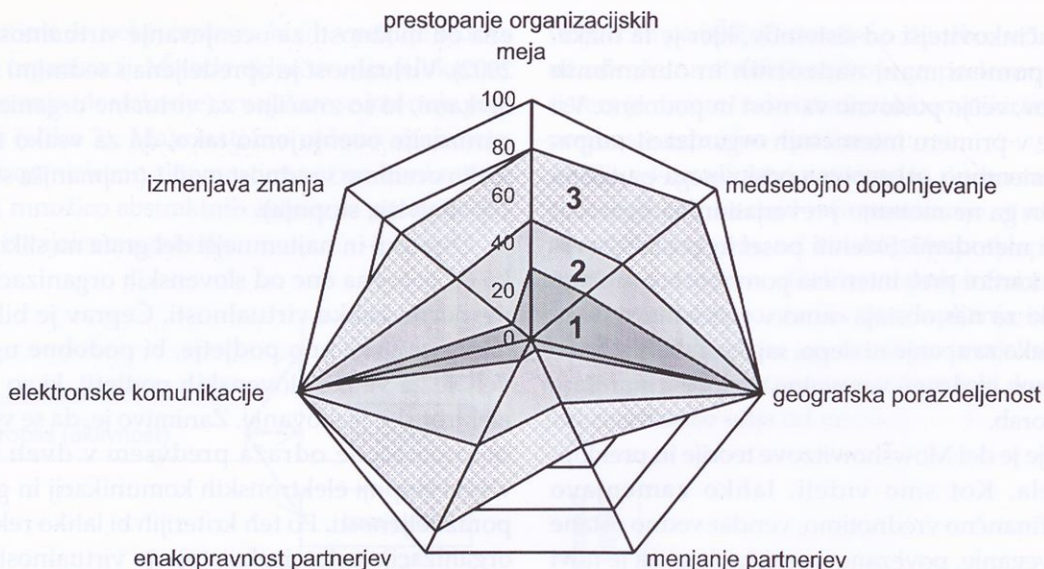
ena od možnosti za ocenjevanje virtualnosti (Bavec, 2002). Virtualnost je opredeljena s sedmimi spremenljivkami, ki so značilne za virtualne organizacije. Organizacijo ocenjujemo tako, da za vsako spremenljivko ocenimo vrednost med 1 (najmanjša stopnja) in 100 (najvišja stopnja).

Osrednji in najtemnejši del grafa na sliki 5 z oznako (1) je ocena ene od slovenskih organizacij, ki kaže nesporne znake virtualnosti. Čeprav je bila analiza narejena le za eno podjetje, bi podobne ugotovitve veljale za veliko slovenskih podjetij, ki so prešle na elektronsko poslovanje. Zanimivo je, da se virtualnost te organizacije odraža predvsem v dveh kazalcih: visoki stopnji elektronskih komunikacij in geografski porazdeljenosti. Po teh kriterijih bi lahko rekli, da ima organizacija zelo visoko stopnjo virtualnosti. Vendar ostali parametri kažejo povsem drugo sliko - zelo majhen odmik od tradicionalne organiziranosti. Vprašanje je, ali je taka organizacija virtualna ali ne. Lahko sklenem, da je njena stopnja virtualnosti še vedno majhna kljub dvema »uspešnima« kazalcema.

Nekoliko svetlejši del grafa z oznako (2) je le hipoteza, saj se s tako organizacijo v Sloveniji še nismo srečali (kar seveda ne pomeni, da jih ni). Kaže pa na značilnosti organizacije, ki bi bila nekje na pol poti med virtualno in tradicionalno. Najsvetlejši del grafa z oznako (3) pa predstavlja izrazito virtualno organizacijo, čeprav ne moremo govoriti o 100% virtualnosti. Vendar je vprašanje ali je popolna virtualnost sploh mogoča.

Omenjeni primer je prikazan predvsem zato, da osvetli vsaj nekatere probleme povezane z merjenjem stopnje virtualnosti ali učinkovitosti organizacij. Pomembno je, da na virtualnost gledamo večdimenzionalno, z različnih zornih kotov. En sam kazalec je premalo, da bi z njim ovrednotili organizacijo.

Slika 5 nakazuje temeljne menedžerske izzive tudi v Sloveniji, saj je očitno da smo veliko uspešnejši pri tehnoloških izzivih (elektronske komunikacije) in geografski porazdeljenosti organizacije. Veliko manj smo uspešni pri "mehkejših" in kar je zanimivo, praviloma pri cenejših elementih sodobnega menedžerstva (partnerski odnosi, izmenjava znanja in medsebojno dopolnjevanje specialnosti posameznih partnerjev). To posredno tudi potrjuje, da sta socialni in tudi človeški kapital v Sloveniji manj razvita, kot si želimo priznati, saj podobne analize v nekaterih drugih evropskih državah kažejo precej drugačno sliko (Paldam in Svendsen 1998).



Slika 5: Stopnja virtualnosti

4 Razprava in sklep

Zgodovinski pogled na informatiko kaže, da v sedemdesetih letih informacijska tehnologija ni bistveno vplivala na organiziranost, saj je praviloma avtomatizirala obstoječe procese. Informacijska tehnologija je bila orodje za povečanje učinkovitosti operativnega dela. Za devetdeseta leta pa je značilno, da ni bilo uspešnega projekta brez ustrezne reorganizacije poslovanja in re-inženiringa procesov. Menedžerji se danes zavedajo pomena informacijskih in komunikacijskih tehnologij in izrabljajo tehnologijo za povečanje učinkovitosti organizacij. V bistvu pa še vedno gledajo ločeno na tehnologijo in na organizacijo. V bližnji bodočnosti bo prišlo do popolne simbioze tehnologije in organizacije ali celo več – organizacija se bo popolnoma prilagodila možnostim, ki jih bo nudila tehnologija (Papows, 1999). Informatikom se ta misel niti ne zdi tako tuja, za večino menedžerjev pa še vedno pomeni hujši primer herezije. Vendar prihaja, vsaj na teoretični ravni, do velikih miselnih premikov, ki že nakazujejo bodočnost menedžmenta v izrazito virtualnih okoljih

Virtualne organizacije, kot jih poznamo danes, so le prvi korak v to smer, vendar že kažejo na nadaljnje zblíževanje informatike in organizacije. Temeljni koncepti virtualne organiziranosti so informatikom blizu, čeprav jih na osnovi dolgoletnih izkušenj dojemajo izrazito intuitivno. Zahteva po prilagodljivosti, ki se

odraža v preklonem načelu, je izvedljiva samo ob popolni izrabi informacijske in predvsem komunikacijske tehnologije. Elektronsko poslovanje med drugim rešuje tudi problem izbora in zamenjave poslovnih partnerjev. S tem, ko olajšuje povezavo s poslovnimi partnerji, posredno olajšuje tudi njihovo morebitno preklapljanje. Zato so visoko informatizirane organizacije zelo prilagodljive, kar je osnovna lastnost virtualnih organizacij. Podobno velja za uporabo informacijskih tehnologij v proizvodnji, od digitalno krmiljenih strojev do robotov, ki prinašajo organizacijsko prilagodljivost neposredno v proizvodnjo. Popolna digitalizacija in avtomatizacija proizvodnje bo še bolj spodbudila nastanek virtualnih organizacij, kot pa poslovna informatika, saj bo preklonno načelo razširila s storitvenih tudi na proizvodne aktivnosti.

Posebej kaže opozoriti na sam izraz »virtualna organizacija«, ki še zdaleč ne pomeni popolne organizacijske virtualnosti. Kot smo videli iz primera, je največ, kar lahko danes dosežemo v praksi, to, da določena oblika organizacije kaže večje ali manjše znake virtualnosti. To pomeni, da smo še daleč od pravih virtualnih organizacij, v katerih bo preklonno načelo temeljno menedžersko načelo. Tudi metamenedžment, ki ga je Mowshowitz definiriral kot menedžment virtualno organiziranih aktivnosti, je še vedno bolj

akademska kot poslovna kategorija. Vendar je razvoj novih organizacijskih oblik tako hiter, da se stanje lahko spremeni že v nekaj letih.

Za zaključek omenimo še Slovenijo, kjer so pogoji za razvoj virtualnih organizacij brez dvoma ugodni. Informatika je na dovolj visoki stopnji, da lahko neposredno spodbuja radikalnejše organizacijske spremembe, kar potrjujejo primeri nekaterih izjemno dobro organiziranih slovenskih podjetij. Naša majhnost in geografska oddaljenost od pomembnejših poslovnih središč je dodatna spodbuda za uporabo najsodobnejših komunikacijskih tehnologij, ki so tehnološka osnova virtualnih organizacij. Posebna poslovna priložnost je vključevanje naših podjetij v virtualne organizacije, ki nastajajo v tujini. V takih primerih velikost podjetja ni pomembna, pomembno je predvsem znanje in konkurenčnost na ozkem segmentu vrednostne verige (angl. core competence).

Viri in literatura

- [1] Bauer R., Köszegi S.T. (2003): Measuring the Degree of Virtualization, *Electronic Journal of Organizational Virtualness*, Vol. 5, No. 2.
- [2] Bavec C. (2001): On the Modelling of Management Decision-making Processes in Organized Anarchy, *Informatica*, 25 (2001).
- [3] Bavec C. (2002): An assessment of the Organization Virtuality with three different reference models, *Informatica*, 26 (2002).
- [4] Bavec C. (2002): Na poti k teoriji virtualnih organizacij, *Organizacija*, 35, 4 (2002), Kranj.
- [5] Bavec C. (2003): Management in e-izzivi, Zbornik 3. strokovnega posveta VŠM v Kopru, Portorož, 2002.
- [6] Drucker P. F. (1999), *Management Challenges for the 21st Century*, Butterworth-Heineman.
- [7] Mowshowitz A. (1997): *Virtual Organization: A vision of Management in the Information Age*, The Information Society, Vol. 10.
- [8] Mowshowitz A. (1999): The Switching Principle in Virtual Organization, *Proceedings of the 2nd International VoNet Workshop*, September 23–24, 1999, Simowa Verlag Bern.
- [9] Paldam M., Svendsen G. T. (1998), *Missing social capital and the transition in Eastern Europe*, Aarhus School of Business, Department of Economics, Prismet, Denmark.
- [10] Papows, J. (1999), *Enterprise.com, Market Leadership in the Information Age*, Nicolas Brealey Publishing, London.
- [11] Saabeel W., Verduijn T. M., Hagdorn L., Kumar K. (2002): A Model of Virtual Organisation: A Structure and Process Perspective, *Electronic Journal of Organizational Virtualness*, Vol. 4, No. 1.
- [12] Skyrme D. (1998): The Reality of Virtuality, *VoNet Workshop*, April 27–28, 1998, Simowa Verlag Bern.
- [13] Ule A. (1992): *Sodobna teorija znanosti*, ZPS, Ljubljana.

Dr. Cene Bavec je zaposlen kot docent na Fakulteti za management v Kopru. Diplomiral je na Fakulteti za naravoslovje in tehnologijo v Ljubljani, magistriral in doktoriral pa na Ekonomski fakulteti v Ljubljani. Zadnjih petnajst let je bil zaposlen v državni upravi, kjer je opravljal različne funkcije povezane z informacijsko tehnologijo od vodenja projekta vzpostavitve Centra vlade za informatiko do državnega sekretarja za tehnologijo. Več let je sodeloval v projektih Evropske komisije in drugih mednarodnih projektih. Bil je član upravnega odbora evropskega foruma o informacijski družbi. Po letu 2000 je postal dekan Visoke šole za management v Kopru. Zadnje leto kot zunanji sodelavec IBM koordinira sodelovanje IBM z univerzami v Srednji Evropi in Rusiji.

Generalna skupščina IFIP

Vilna, Litva, 31. avgust– 2. september 2003

Generalna skupščina International Federation for Information Processing (IFIP), katere član je od leta 1998 tudi Slovensko društvo INFORMATIKA, je bila letos v Litvi in po končani konferenci WITFOR (World Information Technology Forum). Generalna skupščina je najvišji organ IFIP, katere člani z glasovalno pravico (če so njihova društva poravnala letno članarino) so predstavniki društev članic IFIP, predsedniki tehničnih odborov ter člani izvršnega in upravnega odbora. Dnevni red skupščine obsega precej formalnosti, ki so nujne za urejeno poslovanje organizacije in ki jih poznamo tudi iz delovanja našega društva, strokovni del skupščine pa se odvija na sestankih odborov, imenovanih za različna področja delovanja IFIP, in na tehnični skupščini, katere člani so predsedniki tehničnih odborov. Na generalni skupščini sprejemajo v članstvo tudi nove članice; letos sta bili sprejeti društvi informatikov Egipta in Sirije. IFIP tako prerašča v svetovno organizacijo in nekaj diskusije je bilo tudi o tem, ali bi bilo treba spremeniti ime in logotip. Slednji ponazarja računalniško vezje iz začetka druge polovice prejšnjega stoletja in predlagano je bilo, naj bi ga posodobili tako, da bi odražal vsebino delovanja organizacije.

Formalni del generalne skupščine je obsegal poročila sekretarja, predsednika in blagajnika, poročila odborov za publikacije, marketing, države v razvoju, finančnega odbora, odbora za aktivnosti in tehnične skupščine. Vsi odbori se sestanejo pred zasedanjem skupščine, da lahko njihovi predsedniki uskladijo poročila in pripravijo predstavitev. Več diskusije je bilo o finančnem poročilu, nekaj pa tudi o nepričakovani razrešitvi izvršnega direktorja Plamena Nedkova, ki se je sicer zgodila pred skupščino in je bila predmet živahne elektronske korespondence udeležencev generalne skupščine. Zanimiva so bila poročila o konferenci WITFOR in svetovnem kongresu IFIP 2002 v Montrealu. Predvsem slednji je bil predmet kritične razprave na sestanku izvršnega odbora letos spomladi, kjer je bilo prvo poročilo zavrnjeno in zahtevano novo, ki bi pojasnilo finančne okoliščine kongresa. Ocena je bila namreč, da je bil kongres medijski uspeh, vendar pa finančna katastrofa. Novo poročilo je bilo podano in sprejeto, pri čemer je bilo bistveno, da bo začetni vložek IFIP v kongres povrnjen. WITFOR je bil ocenjen kot pomembna svetovna konferenca predvsem za države v razvoju in je bila ocenjena kot izjemno uspešna. Udeležencev je bilo prek 1.500, kar pa je verjetno posledica dejstva, da jo je v glavnem financirala Litva in da je bila kotizacija 14 dni pred konferenco zaradi skromnega števila udeležencev ukinjena. Naslednja svetovna kongresa IFIP bosta leta 2004 v Toulousu in 2006 v Jeruzalemu. Posebej glede slednjega je bilo

izraženih kar nekaj dvomov in sprememba mesta ali celo države ni izključena.

V delu skupščine, ki zadeva delovanje IFIP, je bilo precej razprave o vlogi in zadolžitvah nacionalnih predstavnikov v tehničnih odborih in o članih izvršnega odbora IFIP. Po statutu IFIP traja mandat predstavnikov tri leta in imenovana je bila delovna skupina, ki bo ponovno opredelila njihove zadolžitve in pristojnosti. To lahko pozdravimo, saj bo tudi v pomoč članicam IFIP pri delegiranju predstavnikov v odbore. Dolžnost članov izvršnega odbora je spremljati delo teles IFIP, da bi bilo korektno in legalno. Način njihovega dela doslej ni bil podrobneje določen in druga delovna skupina bo pripravila smernice za njihovo delo.

Generalna skupščina tudi voli predsednika, podpredsednike in člane izvršnega odbora, ki niso njeni člani po položaju. Za predsednika je bil ponovno izvoljen sedanji predsednik Klaus Brunstein iz Nemčije. Enemu od treh podpredsednikov je mandat potekel in tako je bilo izpraznjeno podpredsedniško mesto. Kandidata sta bila dva, dosedanji podpredsednik Avstralec Prins Ralston in Niko Schlamberger, ki je bil lani s triletnim mandatom izvoljen za člana izvršnega odbora. Za novega podpredsednika IFIP je bil izvoljen Niko Schlamberger, kar razumemo kot priznanje Sloveniji, društvu in nje-mu.

Niko Schlamberger

Slovensko društvo INFORMATIKA

zbira na podlagi 53. člena statuta in pravilnika o priznanjih
predloge za
priznanja Slovenskega društva INFORMATIKA

1. Priznanje se lahko podeli posamezniku ali pravni osebi za:

- dosežke na področju uporabne in znanstvene informatike ter vidne prispevke na področju razvoja informacijske družbe in razvoja novih načinov in tehnologij dela na področju informatike,
- dolgoletno uspešno delo v društvu ali v drugih društvih, ki so sodelovala z društvom pri programskih vprašanjih,
- razvoj mednarodnega sodelovanja in izmenjavo dosežkov na tem področju,
- izjemne dosežke na področju razvoja konceptov, programskih orodij, naprav in tehnologij v zvezi z informatiko,
- uspešno sodelovanje z društvom,
- publicistično delo na področju informatike in informacijske družbe in
- izjemne dosežke na področjih, ki zadevajo vprašanja informatike.

2. Predlog mora vsebovati:

- podatke o prejemniku priznanja,
- opis dosežka,
- predlagano priznanje,
- dokazila o dosežku,
- podatke o predlagatelju.

Podrobni pogoji so navedeni v pravilniku na naslovu <http://www.drustvo-informatika.si>

Predloge pošljite do vključno 30. januarja 2004 na naslov:

Slovensko društvo INFORMATIKA
1000 Ljubljana, Vožarski pot 12
z oznako "PRIZNANJA 2004"

Predloge bo v skladu s pravilnikom obravnavala komisija za priznanja in jih s svojim mnenjem posredovala izvršnemu odboru društva. Priznanja bodo javno podeljena na otvoritvi posvetovanja Dnevi slovenske informatike aprila 2004.

Ponovno vas vabimo, da si rezervirate čas za udeležbo na posvetovanju

XI. DNEVI SLOVENSKE INFORMATIKE

14.–16. aprila 2004

Kongresni center Grand hotela Emona, Portorož

predavanja domačih in tujih strokovnjakov

▪
okrogle mize

▪
delavnice

▪
razstave

▪
družabni dogodki

Program posvetovanja:

poslovna informatika in elektronsko poslovanje

▪
informacijske tehnologije in internet

▪
informacijska kultura in družba

Informacije: www.dsi2004.org

Stičišče znanja, tehnologij in poslovnih rešitev
The meeting point of knowledge, technologies and solutions

telekomunikacije
telecommunications



www.telekomunikacije.org

4. mednarodna konferenca o telekomunikacijah
4th International Telecommunications Conference

3. - 4. november 2003
Grand Hotel Emona, Portorož, Slovenija

3 rd Conference on Information Security	22.-25. okt. 2003	Nikozija, Ciper	CEPIS	cepis@vde.com http://www.cepis.org
2 nd Workshop on Application Specific Processors (WASP '03)	2. dec. 2003	San Diego, ZDA	ACM - SIGMICRO, IEEE	http://dha.ucsd.edu/wasp2/
36 th International Symposium on Microarchitecture (MICRO-36)	3.-5. dec. 2003	San Diego, ZDA	ACM - SIGMICRO, IEEE	http://www.microarch.org/micro36/
2 nd International Symposium on Code generation and Optimization (CGO 2004)	mar. 2004	San Jose, California, ZDA	ACM - SIGMICRO, IEEE	http://www.cgo.org
International Symposium on Performance Analysis Systems and Software (ISPASS-2004)	10.-12. mar. 2004	Austin, Texas, ZDA	IEEE	http://ispass.org
ACM International Conference on Computing Frontiers (CF '04)	14.-16. apr. 2004	Ischia, Italija	ACM - SIGMICRO	http://www.computingfrontiers.org beaty@emess.msod.edu
5 th Working Conference on Knowledge Management in Electronic Government (KMGov2004)	17.-19. maj 2004	Krems, Avstrija	Krems Danube University Krems, IFIP WG 8.3 & WG 8.5	http://falcon.ifs.uni-linz.ac.at/kmgov2004/ http://falcon.ifs.uni-linz.ac.at wimmer@ifs.uni-linz.ac.at
IFIP World Computer Congress (WCC 2004)	22.-27. avg. 2004	Toulouse, Francija	IFIP	http://www.wcc2004.org dervillers@wcc2004.org



MARAND
Napredna računalniška hiša

SRCESI
sistemske integracije

Pristopna izjava

Želim postati član Slovenskega društva INFORMATIKA

Prosim, da mi pošljete položnico za plačilo članarine SIT 5.200 (kot študentu SIT 2.400) in me sproti obveščate o aktivnostih v društvu.

(ime in priimek, s tiskanimi črkami)

(poklic)

(domači naslov in telefon)

(službeni naslov in telefon)

(elektronska pošta)

Datum:

Podpis:

Članarina SIT 5.200,- (plačljiva v dveh obrokih) vključuje tudi naročnino za revijo Uporabna informatika. Študenti imajo posebno ugodnost: plačujejo članarino SIT 2.400,- in za to prejema tudi revijo. Izpolnjeno naročilnico ali pristopno izjavo pošljite na naslov:

Slovensko društvo INFORMATIKA, Vožarski pot 12, 1000 Ljubljana.

Lahko pa izpolnite obrazec na domači strani društva: <http://www.drustvo-informatika.si>

Naročilnica na revijo UPORABNA INFORMATIKA

Revijo naročam(o) s plačilom letne naročnine SIT 4.600

izvodov po pogojih za podjetja SIT 13.800 za eno letno naročnino in SIT 8.900 za vsako nadaljnjo naročnino

po pogojih za študente letno SIT 2.000

(ime in priimek, s tiskanimi črkami)

(podjetje)

(davčna številka)

(ulica, hišna številka)

(pošta)

Datum:

Podpis:

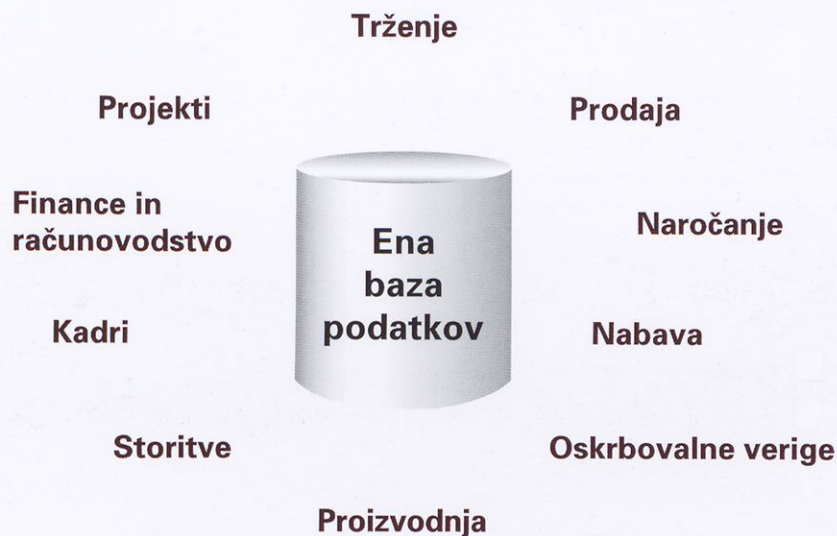
Naročnino bomo poravnali najkasneje v roku 8 dni po prejemu računa

INTERNET

Vse bralce revije obveščamo, da lahko najdete domačo stran društva na naslovu: <http://www.drustvo-informatika.si>

Obiščite tudi spletne strani mednarodnih organizacij, v katere je včlanjeno naše društvo: IFIP: www.ifip.or.at, ECDL: www.ecdl.com, CEPIS: www.cepis.com

Popoln E-Business Suite



**Vse aplikacije zasnovane enotno.
Vse informacije na enem mestu.**

ORACLE®

www.oracle.si

► **Razprave**

Miroslav Ribič, Andrej Kovačič, Marjan Lončarič
Sodelovanje kot temelj za učinkovito upravljanje oskrbovalne verige

Rok Rupnik, Marjan Krisper
Model kontekstno odvisnih mobilnih aplikacij

Petra Povalej, Peter Kokol, Jernej Zavašnik
Dve plati inteligentnih sistemov v medicini

Marjan Heričko, Simon Beloglavec, Matjaž B. Jurič, Boštjan Kežmah
Eksplisitna in implicitna uporaba vzorcev pri razvoju informacijskih rešitev

Maja Miličič
Človeški dejavnik pri povečevanju kakovosti storitev informacijske dejavnosti

Cene Bavec
Informatika med tradicionalnimi in virtualnimi organizacijami

► **Obvestila**

Niko Schlamberger
Generalna skupščina IFIP

ISSN 1318-1882



9 771318 188001