

u p o r a b a
INFORMATIKA

1998

ŠTEVILKA 1

JAN/FEB/MAR

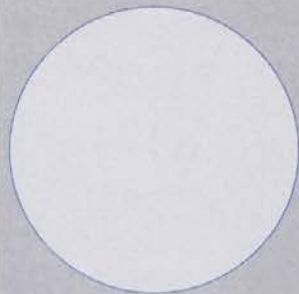
LETNIK VI

ISSN 1318-1882

Aktualno: izobraževanje

Konvergenca digitalnih tehnologij

Vzorci in ponovna uporaba izkušenj



UDELEŽITE SE 5. POSVETOVANJA

DNEVI SLOVENSKE INFORMATIKE

Portorož '98

KI GA ORGANIZIRATA V ČASU
OD 6. – 9. MAJA 1998 V PORTOROŽU

Slovensko društvo Informatika

Združenje za računalništvo in informatiko

strokovno posvetovanje

■
informiranje

■
izobraževanje

■
plenarna razprava in delo v sekcijah

■
stanovsko srečanje

INFORMACIJE IN PRIJAVE:

Slovensko društvo Informatika, 1000 Ljubljana, Vožarski pot 12



Spoštovane bralke in bralci,

Pred kratkim je potekalo na Ekonomski fakulteti v Ljubljani tradicionalno posvetovanje Sekcije za raziskovanje informacijskih sistemov, ki jo že dolgo vrsto let organizira Zveza ekonomistov Slovenije. O posvetovanju samem in o nekaterih njegovih temah in obravnavanih problemih bo govora v tej številki v nadaljevanju, zato o tem na tem mestu ne bomo več govorili. Osredotočili se bomo na nekaj vprašanj, ki so se pojavila v ospredju v okviru razprav na posvetovanju, ki se mi zdijo, za nadaljnji razvoj informatike v Sloveniji, in s tem seveda tudi Slovenije kot države ter njenega družbenega razvoja, posebej pomembna. Gre za vprašanja, ki smo jih v preteklih letih v naši reviji že obravnavali, vendar se vedno znova postavljajo v ospredje kot strateško pomembna, vendar nerešena in neodgovorjena. Gre za stereotipna vprašanja poklicnega profila informatika, njegovega izobraževanja, njegove kariere itd in na drugi strani o tem katera znanja iz informatike so potrebna, da ne rečemo nujna diplomantom drugih tradicionalnih študijskih disciplin, od ekonomije, prava pa morda do medicine ali strojništva. To vprašanje se nam postavlja tudi v zvezi s problematiko vpisa na univerzo, temo, ki ji v tej številki tudi namenimo posebno pozornost. Že nekaj let ugotavljamo, da se vpis študentov na družboslovne študije nenehno povečuje in verjetno že presega vse razumne meje, medtem ko tehniki bijejo plat zvona, saj so nakateri tehnični oddelki skorajda povsem opusteli. Kako pa je z vpisom na programe informatike ter računalništva ter njihove smeri znotraj drugih tradicionalnih disciplin?

V Sloveniji se je na dodiplomski stopnji glede znanj s področja informatike ter študija za to področje na obeh univerzah uveljavil naslednji pristop. V praktično vseh univerzitetnih in strokovnih študijskih programih najdemo določeno število predmetov, ki dajejo študentom temeljna informacijska znanja. Znotraj velikega števila tradicionalnih študijskih programov, tehničnih in družboslovnih imamo vrsto študijskih smeri in modulov, ki študentom teh študijev omogočajo delno specializacijo znotraj izbrane discipline. Poleg tega pa imamo dva do tri študijske programe za katere lahko rečemo, da so specializirani za izobraževanje informatikov - namenoma jih ne bomo poimensko naštevati, da ne bi koga užalili.

Načeloma je tak pristop ustrezen in dovolj fleksibilen. Vsem študentom daje nekaj temeljnega znanja iz obravnavanega področja. Omogoča, da se tisti študentje, ki vidijo na področju informatike svojo dolgoročnejšo usmeritev, vpišejo v specializiran program, tisti, ki pa se recimo odločijo za študij ekonomije in bi radi znotraj tega študija zvedeli nekaj več o informatiki, lahko ta cilj dosežejo tudi v okviru študija ekonomije.



Problem se pokaže šele takrat, ko začnemo analizirati razvojne trende v zadnjih letih. Ugotovimo lahko, da število študentov na specializiranih študijskih programih iz informatike stagnira, da število študentov v specializiranih študijskih smereh znotraj drugih matičnih disciplin izrazito upada in da se v večini študijskih programov na obeh slovenskih univerzah obseg in število ur za informacijske predmete ter vsebine iz leta v leto zmanjšuje. Vse to se dogaja na prelomu tisočletja, ko postaja tako imenovana informacijska družba stvarnost in ko bi radi po bližnjici prišli v Evropsko skupnost. Nekaj je najbrž narobe, toda kaj?

Upadanje števila vpisanih študentov na informacijskih programih in smereh je lahko posledica več dejavnikov. Gotovo je, da je mlada generacija hitro ugotovila, da področje informatike ni tisto, ki bi omogočalo, da bi človek lahko hitro obogatel - razen redkih izjem. Študij informatike je še manj zanimiv, če ga nekdo analizira z vidika razvoja kariere. Kariera informatika se v organizacijah večinoma zaključí na srednjih hierarhičnih ravneh, le redko se informatiki prebijejo v sam vrh vodstvene hierarhije večjih sistemov v gospodarstvu ali v javnem sektorju. V tem pogledu se Slovenija močno razlikuje od večine zahodnih držav, kjer je postalo pravilo, da je eden od članov uprave korporacije praviloma zadolžen za informatiko.

Da upada zanimanje za izbiro informacijskih smeri in modulov znotraj etabliranih disciplin, še posebno študija ekonomije in menedžmenta, pa morda kaže tudi na to, da je nekaj narobe s študijskimi programi in vsebinami informacijskih predmetov, ki se predavajo na teh smereh. Namesto, da bi študentom skušali prikazati, predvsem kako informacijska tehnologija vpliva na razvoj organizacij ter njihovo konkurenčnost, se pravi v kontekstu uporabe na najrazličnejših področjih, se največkrat predavajo izolirani segmenti informacijske ter računalniške teorije od programskih jezikov do podatkovnih struktur in preklonih vezij.

Prav posebno zaskrbljujoče pa je, da se informacijske vsebine in predmeti izrinjajo iz študijskih programov nekaterih disciplin za katere je informatika vitalnega pomena, ekonomsko poslovne vede, menedžment itd. Na eni strani je očitno zelo razširjeno mišljenje, da je informatika to, da vemo kaj je osebni računalnik in kako se ga lahko uporablja pri pisanju besedil ali risanju kakšne prosojnice ali poslovnega grafikona. Ekonomist ali menedžer mora vedeti vse o Taylorju ali Fayolu, da bi pa vedel, kako je potrebno zasnovati informacijski sistem, ki mu bo ponudil vsako jutro na mizo ključne podatke poslovanja njegove organizacije, pa očitno ni pomembno.

Gre za temeljno razvojno vprašanje. Če bodo naši kmetje v bodoče v glavnem mezdni delavci na velikih posestvih v tuji ali fevdalni lasti, potem jim dejansko tehnologije pridelovanja hrane ne bo potrebno poznati, dovolj je da jih naučimo kako se ravna s kosilnico ali traktorjem. Če pričakujemo, da bodo naša podjetja v bodoče v glavnem vodili tujci, potem bo tudi našim bodočim ekonomistom znanje Worda in Excela povsem zadoščalo.

Mirko Vintar



RAZPRAVA

UVODNIK ZEVANJU IN USPOSABLJANJU

NA PODROČJU INFORMATIKE V SLOVENIJI

AKTUALNO

Razprava o izobraževanju in usposabljanju

- 5** ■ ■ ■ ■ na področju informatike v Sloveniji

STROKOVNE RAZPRAVE

CENE BAVEC

- 8** ■ ■ ■ Evropska komisija o zblizevanju digitalnih tehnologij

JURIJ JAKLIČ, JANEZ GRAD

- 15** ■ ■ ■ Kako do bolj fleksibilne predstavitve poslovnih pravil?

TOMAŽ DOMANJKO, IVAN ROZMAN, MARJAN HERIČKO, JÓZSEF GYÖRKÖS

- 24** ■ ■ ■ Vzorci in ponovna uporaba izkušenj

REŠITVE - ORODJA

TOMAŽ MOHORIČ

- 32** ■ ■ ■ Izbirni postopek za vpis na fakultete in samostojne visokošolske zavode v Sloveniji

LILJANA MIHELIČ

- 41** ■ ■ ■ Informixov optimizator poizvedb SQL

POROČILA

FRANC ŽERDIN

- 48** ■ ■ ■ Testiranje programov - kaj pa podatki?

DOGODKI IN ODMEVI

- 50** ■ ■ ■ Konference: takšne in drugačne

- 52** ■ ■ ■ Konferenca Gartner Group

OBVESTILA

- 54** ■ ■ ■ Dnevi slovenske informatike Portorož 98

KOLEDAR PRIREDITEV

- 55** ■ ■ ■ ■

Zahvaljujemo se podjetju Marand d.o.o., Ljubljana, Cesta v mestni log 55,
za sponzoriranje domače strani Slovenskega društva INFORMATIKA



Slovensko društvo INFORMATIKA

se zahvaljuje INFOS-u, Podjetju za sejemske, promocijske in izobraževalne
prireditve, da je omogočilo nastop društva na INFOS '97.

RAZPRAVA

O IZOBRAŽEVANJU IN USPOSABLJANJU NA PODROČJU INFORMATIKE V SLOVENIJI



Sekcija za raziskovanje informacijskih sistemov pri Zvezi ekonomistov Slovenije je v svoje letos že enajsto posvetovanje¹ vključila aktualno temo z naslovom "Izobraževanje in usposabljanje na področju informatike v Sloveniji: spoznanja in predlogi softverskih firm." Vabilu so se odzvali direktorji pomembnejših softverskih firm v Sloveniji. Iz zanimive razprave, v katero so se vključevali tudi mnogi udeleženci posvetovanja, izhajajo številna priporočila in predlogi akcij, kar naj bi izboljšalo možnosti pridobivanja ustreznega znanja na univerzah in drugje. Pričujoče gradivo je bilo dostavljeno sodelujočim v razpravi in udeležencem posvetovanja, rektorjema univerz in dekanom fakultet ter odgovornim ministrstvom. Prepričani smo, da je zanimivo in aktualno tudi za bralce naše revije.



V razpravi so sodelovali panelisti :

- Janez Benčina, direktor, IBM Slovenija
- dr. Jože Gričar, Fakulteta za organizacijske vede Univerze v Mariboru (vodja panela)
- Vasja Herbst, direktor, Oracle Software d.o.o.
- Aaron Marko, direktor, Microsoft Slovenija
- Ivan Zupanc, direktor, Razvojni center Celje - IRC
- Ivan Žerko, direktor, SRC d.o.o.

ter razpravljalci - udeleženci posvetovanja:

- dr. Lea Bregar, Ekonomska fakulteta Univerze v Ljubljani
- dr. Janez Grad, Ekonomska fakulteta Univerze v Ljubljani
- dr. Maja Miličič, AD Consulting d.o.o.
- dr. Gortan Resinovič, Ekonomska fakulteta Univerze v Ljubljani
- Cveto Trampuž, Fakulteta za družbene vede Univerze v Ljubljani
- mag. Tomaž Turk, Ekonomska fakulteta Univerze v Ljubljani
- dr. Mirko Vintar, Visoka upravna šola Univerze v Ljubljani.

UGOTOVITVE

Spreminja se opredelitev nalog informatika; mogoče ga je videti kot arhitekta informacijske podpore poslovnim procesom.

Število odprtih delovnih mest za strokovnjake informatike se povečuje, zlasti v Severni Ameriki in Evropski uniji. Študije Evropske komisije kažejo, da organizacije v državah Evropske unije zaostajajo v uporabi informacijske tehnologije zlasti za ZDA in Japonsko. Sproženi so alarmantni pozivi k ponovnim razmislekom o vlogi informacijske tehnologije v prihajajoči informacijski družbi in njenem vplivu na obseg zaposlenosti. Velja spoznanje, da je univerza najboljši vir kadrov za področje informatike. Predvidena 'proizvodnja' diplomantov na univerzah pa bo najbrž zagotavljala premalo informatikov.

¹ 12. - 13. februar 1998 v Ljubljani.

Zaradi intenzivnejšega povezovanja Slovenije s svetom se bodo povečale potrebe po poznavanju informacijske tehnologije in s tem povezane potrebe po informatikih.

Podatki o odprtih delovnih mestih informatikov v Sloveniji niso na voljo. Obstaja vtis, da je status informatikov v Sloveniji relativno nižji, kot v razvitih državah. To najbrž izhaja iz neosveščenosti o pomenu informacijske tehnologije. Sicer pa ni podatkov o tem, kakšen je položaj informatikov v slovenski družbi.

Vpis na univerzitetni študij informatike ne narašča zadosti, na nekaterih fakultetah, ki izvajajo program informatike, pa celo upada. V praksi se kaže potreba, da bi imeli diplomanti informatike bolj celovito in interdisciplinarno znanje. Ob siceršnji potrebi, da obvladajo posamezne programe (orodja), so pogosto preozko specializirani. Poleg tehnoloških znanj morajo imeti tudi znanja o poslovnih procesih in biti usposobljeni za pridobivanje informacij in razumevanje možnosti njihove uporabe. Diplomanti naj bi poznali več področij.

Poleg poslovnih procesov naj bi diplomirani informatiki bolje razumeli tudi izvajalce procesov in vodje procesov. Informacijsko tehnologijo je namreč potrebno uvesti v delo ljudi v organizacijah. Zato morajo biti diplomanti informatike usposobljeni za reševanje problemov in obvladovanje sprememb. Informacijska tehnologija je zapletena (kompleksna), zato so potrebna ustrezno celovita znanja. Samo znanje o uporabi osebnega računalnika za informatika ni zadostno. Obvladovanje zapletenih sistemov pa zahteva ustrezno kritično maso kadrov, zato bo zgolj slovenski trg najbrž premajhen za zagotavljanje ekip strokovnjakov, ki bodo uvajale sodobne informacijske tehnologije; potrebna bo razširitev dejavnosti na trge v bližnjih državah. Diplomanti imajo premalo praktičnih izkušenj, ki bi si jih morali pridobiti s prakso med študijem. Povezovanje s prakso je pomembno. Sicer je res, da spoznanja iz raziskav na univerzah prihajajo v gospodarstvo, toda smeri raziskovanja v veliki meri vendarle določa industrija. To zlasti velja za razvoj informacijske tehnologije, ki je dejavnost z najintenzivnejšo stopnjo razvoja.

Postavlja se vprašanje, ali se bodo diplomanti fakultet slovenskih univerz lahko potegovali za tako zaposlitev v Evropi, ki bi bila skladna z njihovo siceršnjo formalno diplomo. Mogoče je pričakovati, da bodo velike korporacije iz tujine želele rekrutirati tudi na univerzah v Sloveniji. Povečuje se zahteva po primerljivosti študija in diplom v Sloveniji s tujino. Akademski standardi se morajo približati svetovnim, sicer Slovenija ne bo svetovno konkurenčna. Vzporedno se bodo pojavljala vprašanja vrednotenja izobraževalnih programov in univerz. Kaže se potreba po dokazljivosti usposobljenosti (certificiranju) na področju informatike, saj strokovnjaki, ki se izkažejo s potrdilom (certifikatom), ki je v praksi priznано, dosegajo boljše izhodiščne plače. V praksi se kaže potreba po izobraževanju v zvezi z informacijsko tehnologijo in usposabljanju za njeno uporabo za tri skupine ljudi. Prva, uporabniki nasploh (računalniško in informacijsko opismenjevanje), druga, managerji in tretja, managerji informacijske dejavnosti (chief information officer - CIO). Računalniška in informacijska pismenost je še vedno premajhna, z njo ne moremo biti zadovoljni. Problematika zadeva univerze, srednje šole in podjetja, ki izvajajo tovrstne izobraževalne programe.

V zvezi z drugo skupino kadrov posvečajo managerske šole v Sloveniji informatiki premalo pozornosti. Opazno je pomanjkanje programov za tretjo skupino. V organizaciji, ki tak program izvaja, ugotavljajo, da sicer dobijo kadre, ki jih je mogoče oblikovati, vendar slušatelji nimajo zadostne širine.

Ni sprejemljivo pričakovanje, da se bodo s problemi računalniške in informacijske pismenosti v naslednjih letih ukvarjali samo na ravni srednjih ali celo samo osnovnih šol. Upoštevati je potrebno trenutno stanje študentov - rednih in izrednih, ki vstopajo na univerzo in imajo o uporabi računalnikov premalo znanj in izkušenj.

Posebna skupina vprašanj se nanaša na problematiko spremljanja razvoja informacijske tehnologije po diplomih.

Pri odpiranju in reševanju tovrstne problematike premalo sodelujejo univerze, gospodarske in vladne organizacije ter ponudniki informacijske tehnologije.

PRIPOROČILA

- Pospešiti uvajanje informacijske tehnologije za študij na daljavo in izrabiti priložnosti uporabe interneta za samoučenje. Izrabiti informacijsko tehnologijo za izobraževanje, neodvisno od vrste predmeta in vsebine. Podpreti velja pobudo, da se v Sloveniji vzpostavi center, ki bi spodbujal in usklajeval aktivnosti za uvajanje sodobnih oblik in možnosti izobraževanja na daljavo.
- Iskati nove oblike posredovanja znanj (naprimer: seminar v organizaciji naročnika; uporaba interneta; upoštevanje načel: pravočasnost, potrebe slušatelja in potrebni obseg (just-in-time, just-for-me, just-enough).

- Iskati je potrebno bolj učinkovite in uspešne oblike povezovanja univerze in prakse. Naprimer izvedba prakse študentov v organizacijah, (simbolični) odkup diplome ali magisterija. Iskati je potrebno načine vrednotenja in ocenjevanja uspešnosti prakse.
- Univerze naj presodijo, kakšne so možnosti, da se poveže pridobivanje v praksi priznanega certifikata v zvezi z informacijsko tehnologijo z izpolnjevanjem siceršnjih študijskih obveznosti študenta na fakulteti.
- Univerzitetni profesorji naj sodelujejo pri razvijanju programov na področju informacijske tehnologije v širšem prostoru in se vključujejo v njihovo izvajanje.
- Potrebno je celovito spremljanje povpraševanja po informatikah v Sloveniji (borza delovnih mest).

PREDLOGI AKCIJ

- Ponudniki informacijske tehnologije so pripravljene v še večji meri pomagati univerzam pri zagotavljanju informacijskih tehnologij za raziskovalne in izobraževalne namene.
- Čimpreje zagotoviti večjo interdisciplinarnost izobraževalnih programov, saj za to niso potrebne velike naložbe, ampak predvsem sprememba miselnih okvirov.
- Zagotoviti univerzam več informacijske tehnologije.
- Zagotoviti univerzam več kadrov (asistentov, mladih raziskovalcev) za področje informacijskih sistemov.
- Vključevati raziskovalno in pedagoško usmerjene informatike, ki so se uveljavili v praksi, v izobraževalne programe na univerzah (predstavitve, vaje, seminarske naloge, somentorstvo pri diplomah).
- Povečati delež znanj o informatiki v programih univerz, zlasti poslovnih fakultet.
- Vzpostaviti 'Forum informacijske družbe' s podskupino za izobraževanje. Izvesti študijo o stanju informatike v Sloveniji, vključno s stanjem izobraževanja.
- Ponovno vzpostaviti delovno skupino za spremljanje informatike kot stroke na univerzah.
- Izrabiti pripravljenost direktorjev softverskih firm v Sloveniji - lokalpatriotov, ki tudi v razmerah trdega boja za vsakodnevno preživetje v tržnih razmerah razmišljajo o bodočnosti svojih in drugih otrok - za sodelovanje pri pripravljani Slovenije na informacijsko družbo.
- Pripraviti sestanek vodstev univerz, predstavnikov kateder za informatiko na fakultetah in direktorjev podjetij, ki nudijo informacijsko tehnologijo. Na sestanku naj bi razpravljali o priložnostih za bolj intenzivno sodelovanje vseh prizadetih s ciljem povečati učinkovitost in uspešnost diplomantov in njihovega strokovnega razvoja v praksi.

J. G.

Evropske nagrade na področju informacijskih tehnologij

Evropska komisija je v okviru programa Esprit (informacijske tehnologije) in v sodelovanju z Euro-CASE (European Council of Applied Sciences and Engineering) objavila razpis za tekmovanje za Evropske nagrade na področju informacijskih tehnologij 1998.

Nagrade so namenjene novim, tržno zanimivim izdelkom, ki temeljijo na visokih tehnologijah, namen nagrad pa je spodbuditi odličnost, inovacije in konkurenčnost v industriji.

Tekmovanja se lahko udeležijo podjetja, laboratoriji, univerze, raziskovalne institucije in druge organizacije s področja informacijske tehnologije iz držav članic Evropske zveze, od leta 1996 pa tudi iz vzhodno- in srednjeevropskih držav. Podelili bodo tri glavne nagrade po 200.000 ECU in 25 nagrad po 5000 ECU, od tega jih bo pet namenjenih nagrajencem iz držav Vzhodne in Srednje Evrope.

Prijavni rok: 28. april 1990 (do 17 ure).

Nagrade bodo podeljene ob letošnji Evropski konferenci o informacijskih tehnologijah - EITC 1998, ki bo potekala od 30. novembra do 2. decembra 1998 na Dunaju.

Informacije in prijavitni obrazci:

The European IT Prize/Euro-CASE Secretariat, Ms Helle Bonnet, 16 rue Mazarine, F-75006 Paris, France,
tel. +33 1 44 41 43 94, faks + 33 1 44 41 43 55, E-mail: europcase@institut-de-france.fr, internet: <http://www.it-prize.org>.

EVROPSKA KOMISIJA

O ZBLIŽEVANJU DIGITALNIH TEHNOLOGIJ

Cene Bavec
Ministrstvo za znanost in tehnologijo

Povzetek

Članek obravnava evropske poglede na zблиževanje¹ digitalnih tehnologij in storitev, kot so telekomunikacije, radio in televizija ter informacijske tehnologije. V naslednjih letih naj bi se celotno področje liberaliziralo, storitve pa naj bi bile neodvisne od prenosnih omrežij. To je logičen korak, ki sledi liberalizaciji telekomunikacijskega sektorja v Evropski uniji. Za Slovenijo je Zelena knjiga o zблиževanju, ki jo je decembra 1997 objavila Evropska komisija, izjemno pomemben dokument, ki v veliki meri opredeljuje naš odnos do tega področja.

Abstract

The paper deals with a global European view on the convergence of digital technologies and services like telecommunications, media and information technology. In the following years this sector will be liberalised and digital services will be independent of networks. This is a logical step following liberalisation of telecommunications sector in the EU. The Green Paper on the Convergence, published by the European Commission in December 1997, is an important document for Slovenia, too. It will be the basis for future national regulation.

Ključne besede: telekomunikacije, digitalne tehnologije, informacijske storitve, javni mediji, zakonodaja, Evropska komisija



1. Uvod

Zbliževanje digitalnih tehnologij in storitev odpira vrsto novih problemov in dilem, ki niso vedno v jedru zanimanja poslovnih informatikov in so bližje telekomunikacijam, internetnim storitvam in celo javnim medijem. Po drugi strani pa tehnološki razvoj pogosto omogoča povsem drugačen pogled na reševanje določenih problemov, za katere smo že mislili, da se jih ne moremo lotiti drugače. Vzemimo samo primer interneta in revolucije, ki jo je povzročil v poslovni informatiki. Zato se zdi tudi tema o zблиževanju digitalnih tehnologij in storitev zanimiva za bralce revije *Uporabna informatika*, čeprav spada v "mejno" področje.

Le malo informatikov razmišlja o javnih medijih ali avdio in video storitvah radia in televizije kot o nečem, kar bi lahko vplivalo na njihovo delo in način reševanja strokovnih problemov. To pač ni bila njihova stroka, ki ima jasno opredeljene meje. Že internet je s svojimi možnostmi povzročil dovolj zmede in v temeljih zamajal marsikatero resnico, v katero smo neomejeno verjeli. Informatika je po eksploziji interneta nekaj

povsem drugega, kot je bila prej. Ali informatiki lahko v kratkem času prenesemo še en tak šok? Bojim se, da ga bomo morali! Prepričan sem, da bomo v naslednjih nekaj letih razširili svojo stroko tudi na avdio in video področje in na aplikacije, o katerih še mislili nismo. To je posledica skorajšnje popolne digitalizacije našega pogleda na svet.

V Evropi so se v zadnjem letu zgodili tako pomembni strateški in konceptualni premiki na področju tehnologij, ki nas vodijo v informacijsko družbo, da jim moramo posvetiti vso svojo pozornost, ne glede na to, s katero vejo informatike se ukvarjamo. Tako je Evropska komisija decembra 1997 objavila Zeleno knjigo² o zблиževanju telekomunikacijskega sektorja, javnih medijev in informacijske tehnologije in njenem vplivu na državno regulativo³ (v nadaljevanju uporabljam skrajšani naslov "Zelena knjiga o zблиževanju"). Komisija se je odločila objaviti svoje poglede na zблиževanje digitalnih tehnologij in sedaj pričakuje odmeve v javnosti. Predvideni potek aktivnosti Evropske unije je:

¹ Namesto izraza "zблиževanje" pogosteje srečamo izraz "konvergenca".

² Zelene knjige (angl. Green Paper) so dokumenti, ki jih Evropska komisija daje v javnost zato, da spodbudi javno in strokovno razpravo o določeni temi.

³ Green Paper on the Convergence of the Telecommunications, Media and Information Technology Sectors, and the Implications for Regulation - Towards an Information Society Approach

- Zelena knjiga bo v javni razpravi do konca aprila 1998
- poročilo Evropske komisije o rezultatih javne razprave bo pripravljeno junija 1998
- Svet in Evropski parlament bosta sprejela posebno resolucijo v sredini leta 1998
- na podlagi resolucije Evropskega parlamenta in javne razprave o Zeleni knjigi bo Evropska komisija do konca leta 1998 pripravila akcijski program za področje zблиževanja.

Zelena knjiga o zблиževanju je tretji vidnejši korak Evropske unije v informacijsko družbo. Vse skupaj se je začelo z Bangemannovim poročilom leta 1994 ter nadaljevalo z ministrsko Bonnsko deklaracijo leta 1997. Na moje veliko presenečenje Zelena knjiga o zблиževanju ni doživela nobenega odmeva v slovenski javnosti, čeprav smo z eno nogo že v EU. Kot da nas ne zanima, kaj se tam dogaja in kot da ne vemo, da bomo morali v celoti upoštevati njihove sistemske in zakonske rešitve tudi na tem področju. Upam, da smo ta dokument le spregledali in da se nismo odločili za filozofijo "bo že kako".

Čeprav se nam čas za formalne pripombe na Zeleno knjigo o zблиževanju že izteka, želim s tem prispevkom pritegniti pozornost javnosti ter jo seznaniti z osnovnimi problemi in dilemami, ki jih prinaša zблиževanje digitalnih tehnologij. Hkrati pa upam, da bom s tem spodbudil sprotno spremljanje podobnih dokumentov, ki nastajajo v Evropski komisiji, saj postajajo z našim vstopanjem v Evropsko unijo, hoteli ali ne hoteli, ti dokumenti tudi slovenski. Če k temu dodamo še dejstvo, da se z letošnjim letom začenejo tudi uradna pogajanja z Evropsko unijo o slovenskem polnopravnem članstvu, ki bodo zajela tudi informacijsko tehnologijo v najširšem pomenu besede, potem je zanimanje za dogajanja v Evropski uniji še toliko bolj upravičeno.

Zблиževanje digitalnih tehnologij je strokovno izjemno zanimiv razvoj, ki odpira nove možnosti in poglede na informacijsko družbo, vendar se v tem prispevku ne bom spuščal v tehnološke vidike. Večina bralcev pozna gibanja na področju informacijskih in telekomunikacijskih tehnologij in tudi tehnološke vidike zблиževanja, manj pa jih razmišlja o posledicah tega razvoja. Pri tem ne mislim samo spremenjenega načina delovanja ter družbene in ekonomske vloge posameznih sistemov, kot so radio in televizija, časopisi in publicistika ter telefonija, ampak tudi vpliv zблиževanja na poslovno informatiko, globalno ekonomijo in podobno. Za nas pa je še posebej zanimivo vprašanje: kdaj in kako se bomo tega problema lotili v Sloveniji in kakšne bodo posledice.

V komisiji že deluje vrsta delovnih in posvetovalnih teles, kot sta Forum o informacijski družbi in Orejina strokovna skupina za avdio-vizuelno politiko, ki

bo pripravila posebno poročilo do septembra 1998. Evropska unija se je odločila, da bo ponovno proučila svojo regulativo in strategijo na tem področju, saj se že sedaj vsi strinjajo, da bodo posledice zблиževanja digitalnih tehnologij in na njih zasnovanih storitev veliko bolj vplivale na ekonomijo in družbo nasploh, kot je to veljalo za liberalizacijo klasičnih telekomunikacij. Zato Evropa ne želi zamuditi te priložnosti.

2. Zблиževanje digitalnih tehnologij in storitev

Evropska komisija je definirala zблиževanje digitalnih tehnologij na dva načina, ki sta vsebinsko enaka:

- možnost različnih omrežnih platform da prenašajo enake ali zelo podobne vrste storitev (tehnologija omogoča, da se za prenos tradicionalnih in novih telekomunikacijskih storitev - glas, podatki, zvok ali slike - lahko uporabljajo različna omrežja)
- združevanje različnih proizvodov široke potrošnje, kot so telefoni, televizija in osebni računalniki (različne telekomunikacijske storitve se lahko sprejemajo na različnih napravah).

Zблиževanje digitalnih tehnologij in storitev je na določenih področjih že povsem običajen in sprejet pojav. Prenos glasu po internetu je nadomestilo za telefon, dostop do interneta prek kablanskega sistema postaja realnost tudi v Sloveniji, o omrežni televiziji že čitamo v revijah in podobno. Globalno zблиževanje digitalnih tehnologij, kot jo v svojih dokumentih razume Evropska komisija, pa je precej širše in zajema pravzaprav vse digitalne tehnologije in vse storitve, ki se lahko digitalizirajo. To pa je skoraj neomejeno področje.

V zvezi z radiom, televizijo, telekomunikacijami in javnimi mediji je zanimiva tudi interpretacija pojma monopola. Monopolist je bila pravzaprav država, ki je dodeljevala dovoljenja in koncesije za izvajanje omenjenih storitev. To je v osnovi preprečevalo konkurenco in tržno poslovanje in na široko odpiralo vrata za vpletanje politike. Še posebej je to veljalo za televizijo, saj je bil v veliki večini držav privatni kapital izločen, resnična izjema so bile le ZDA.

Problemi, ki jih je Evropska komisija vzela pod drobnogled in bi jih rada odpravila v državah članicah EU, so naslednji:

- nepredvidljiva zakonodaja
- podeljevanje licenc za vstop na določene trge
- povezovanje omrežij z pravico različnih storitev
- lokacija radijskih frekvenc in drugih virov
- različni pogledi na javni interes
- nezaupanje javnosti v novo okolje
- pomanjkanje standardov, ki podpirajo povezljivost in skladno delovanje različnih omrežij.

3. Nekateri vidiki zblíževanja

Internet je ob splošnem razvoju računalniške tehnologije eden največjih spodbujevalcev globalnega združevanja digitalnih tehnologij in storitev. Prebil je tako tehnološke kot psihološke ograje v svetovnem merilu ter dokazal, da se lahko tudi zapleteni sistemi "samoregulirajo" brez velikega vmešavanja s strani držav. Odprl je nove trge in s tem spodbudil velika vlaganja v različne digitalne tehnologije in predvsem storitve.

Konec leta 1997 je bilo na svetu okoli 200 milijonov uporabnikov interneta (po podatkih Internet Industry Almanacha). Od tega jih je bilo v ZDA 54%, vendar je ta delež padel od 64% leta 1994. Do leta 2000 naj bi ameriški delež padel celo na 40%. Sedanji evropski delež je 22%, japonski pa 8%.

Internet bo v taki ali drugačni obliki v samem jedru informacijske družbe. Vsi morebitni zadržki okoli interneta so le prehodne narave, saj se moramo zavedati, da govorimo o sistemu, ki je star le nekaj let, če pustimo ob strani internet pred uporabo spletnih strani. Proizvajalci informacijske tehnologije, ponudniki storitev in uporabniki še nimajo resnejših izkušenj, kar se vidi že iz tabele 1. Kljub temu pa je že sprožil radikalne spremembe na uporabniški strani kot tudi pri zblíževanju digitalnih tehnologij.

Tabela 1: Čas obstoja spletnih strani v EU

	Mala in srednja podjetja	Univerza, šole, knjižn. in muzeji	Javna uprava
Manj kot 6 mesecev	27,3 %	13,0 %	15,5 %
Med 6 meseci in 1 letom	32,0 %	11,1 %	32,8 %
Med 1 in 3 leti	38,3 %	50,0 %	51,7 %
Več kot 3 leta	2,3 %	25,9 %	0,0 %

(Vir: Databank Consulting, 1997)

S systemskega zornega kota je očitno, da digitalna tehnologija podira meje, ki so jih opredeljevale organizacije, ki so nudile analogne storitve, kot sta že omenjena govorna telefonija ali radio in televizija. Do sedaj so bile njihove storitve med seboj jasno ločene, saj jih je poleg zakonodaje ločevala tudi različna tehnologija. Za televizijo se je vedelo, da se signal prenaša po televizijskih kanalih in sprejema prek anten ali pa po kablenskem sistemu (to je bil prvi znak zblíževanja analognih tehnologij). Podobno je veljalo za radio, telefonijo in tudi ostale komunikacijske storitve, kot so faks in podobno. Za časopis se je vedelo, da ga nekdo tiska in razpošilja. Nihče ni niti v mislih povezoval računalniških in televizijskih omrežij, saj je bilo jasno, da gre za nezdružljive tehnologije in storitve. Vedelo se je, da se osebni računalnik uporablja kot računalnik, televizija za gledanje televizijskega programa, radio za poslušanje radijskega programa, telefon za telefoniranje itd. Z zblíževanjem teh tehnologij pa se je vse to postavilo na glavo.

Dolgoročno gledano bo zblíževanje tehnologij do sedaj ločene storitve združilo v eno samo. Brez dvoma bo to končen rezultat, vprašanje je le to, koliko časa bo trajalo in kakšne družbene in ekonomske posledice bo prineslo. Tisti trenutek, ko bo lahko vsak na svoji spletni strani ponujal tudi video in avdio posnetke, bo v bistvu simuliral televizijski studio. Glede na to, da bo do vsakega doma tekla samo ena fizična povezava, ki bo služila za kablensko televizijo, telefon, dostop do interneta in še kaj, potem je jasno da za uporabnika sploh ne bo razlike med spremljanjem televizijskega programa ali pa med pobiranjem video posnetkov z internetnega strežnika. Ne potrebujemo prav veliko fantazije, da si predstavljamo, kako bo to vplivalo na televizijo kot organizacijo. V tem trenutku je na svetu že 650 omrežnih radijskih postaj in 270 omrežnih video postaj, vendar je to res le začetek. Takega sistema država ne bo mogla več nadzirati in ga usmerjati z podeljevanjem raznih dovoljenj in koncesij, kot dela sedaj.

V letu 1998 bo na svetu prvič prodanih več osebnih računalnikov kot televizorjev. V ZDA pa je že leta 1995 število ur pred osebnim računalnikom preseglo število ur pred televizorjem. Zanimivo je tudi to, da uporabniki interneta 60 % manj gledajo televizijo, kot tisti, ki ga ne uporabljajo.

Zblíževanje je v tem trenutku najbolj opazno na telekomunikacijskem področju, vendar je to le prvi korak v globalno zblíževanje vseh digitalnih tehnologij in storitev. Telekomunikacijski operaterji že pred leti prenesli iz bakrenih kablov na optične kable in satelitske povezave. Vendar se to navzven ni videlo, saj uporabnika telefonov nikoli ni zanimalo, kako se prenaša njihov glas. Vse skupaj je postalo veliko zanimivejše za uporabnike šele, ko so se v igro vpletli drugi akterji. Zmeraj več telekomunikacijskih operaterjev nudi poleg telefonije in prenosa podatkov še dostop do televizijskih programov (zamenjujejo kablenske sisteme), postajajo pa tudi eden glavnih akterjev pri dostopu do interneta, ki bo očitno zagotovil prenos vseh digitalnih storitev. Drugi pomembni akterji so kablenski operaterji, ki že zagotavljajo telekomunikacijske storitve skupaj s telefonijo in z dostopom do interneta.

France Telekom je za 410 milijonov ECU kupil Casemo, največjega nizozemskega TV kablenskega operaterja z 1.1 milijona naročnikov.

Vendar se je v ta dvoboj že vmešala tudi televizija, kot tretji ključni akter. Televizija in telekomunikacijski operaterji imajo v tem trenutku precejšno prednost v tržnem boju, saj razpolagajo z razvejano komunikacijsko infrastrukturo in izjemno finančno močjo. Televizijske organizacije imajo še eno veliko prednost - lahko nudijo tudi vsebino, ne samo prenosnih storitev.

Francoski TV gigant Canal+ bo do konca letošnjega leta začel s prodajo druge generacije digitalnega multi-medijskega vmesnika za domačo uporabo, ki bo omogočal

popolno integracijo vseh komunikacij od TV, osebnih računalnikov in telefonov do mobilne telefonije.

Canal+ je združil svoje internetne aktivnosti z največjim francoskim telekomskim operaterjem Cegetelom in America On-Line. Pokrivali bodo internetno telefonijo, storitve za podjetja, oglaševanje, elektronsko poslovanje in interaktivno TV.

Digitalne satelitske komunikacije postajajo vedno cenejši in globalen način prenosa podatkov in bodo brez dvoma odigrale ključno vlogo pri globalizaciji informacijskih storitev. Vsi tehnični in ekonomski pomisleki proti uporabi satelitskih komunikacij so vedno bolj neupravičeni. Firme z različnih področij, ki so v tem poslu, že iščejo svojo tržno nišo v svetu digitalnih tehnologij. To velja tako za lastnike komunikacijskih satelitov za potrebe telekomov in televizije, kot tudi za firme, kot so IBM, Microsoft in podobno, ki začenejajo lastne satelitske projekte.

CanalSatellite in BskyB bosta v kratkem ponudila neodvisne internetne dostope in elektronsko pošto ter razne interaktivne storitve, kot so dostopi do bank in nakupovanje na daljavo.

V zvezi z evropsko strategijo na področju satelitske tehnologije, ki je povezana z informatiko in digitalnimi storitvami, je posebej zanimiva odločitev, da bo do leta 2005 zgradila lasten sistem satelitske navigacije GNSS (Global Navigation Satellite System) v vrednosti 43 milijard ECU. Razlog za to odločitev so njeni dolgoročni načrti na področju transportne telematike, ki zahtevajo natančno zemeljsko pozicioniranje. Sedanja sistema GPS in Glonass sta namreč pod nadzorom ameriške in ruske vojske in ne nudita tiste odprtosti in varnosti, ki sta potrebni za nevojaško uporabo. Tudi satelitsko pozicioniranje je ena od digitalnih storitev.

Predvsem z zornega kota vsebine digitalnih storitev ne smemo pozabiti celotnega gospodarskega sektorja, ki je zgrajen na publicistični dejavnosti. Ta bo odigral ključno vlogo v zagotavljanju vsebine in načina posredovanja digitalnih storitev (v angleškem jeziku že obstaja za to nov izraz "packing") za širšo javnost.

Nemške, francoske in ameriške publicistične grupacije Bertelsmann, Havas in World Book Publishing so skupaj z IBM objavili, da bodo združili svoje sile v multi-

medijskem konzorciju, ki bo proizvajal in tržil visoko kvalitetne multimedijske proizvode ob uporabi najsoodobnejše tehnologije, predvsem tridimenzionalne grafike in fotografije.

Iz ozadja pa že prihajajo novi akterji, ki si želijo vzeti kos novo nastajajočega sveta tudi zase. To so razne organizacije, ki so si iz različnih razlogov same gradile svoja telekomunikacijska omrežja. V Sloveniji so to na primer elektrogospodarstvo in železnice. V tujini pa gre še za veliko močnejše organizacije, kot sta IBM in Microsoft, ki sta svoj pogled obrnila v vesolje in satelitske komunikacije ter v proizvodnjo "digitalne vsebine". Tabela 2 nekoliko poenostavljeno kaže, kakšna je vloga posameznih sistemov in panog v zblizevanju digitalnih tehnologij in storitev.

Kljub temu, da s tehnološkega zornega kota ni nobenih ovir (kmalu bo to veljalo tudi za zakonodajo in drugo državno regulativo), da bi se vsak ukvarjal z vsem, pa je to le ena plat medalje. Prav gotovo ne bo mogel vsak heker imeti svojega televizijskega studia kar doma, saj ni problem le tehnologija, pravi problem je kvaliteten program, ki ga lahko ponudi. Stroški za pripravo vsebine določene storitve, pa naj bo to televizijski program ali podatkovna baza, se ne bodo hitro zniževali, saj so le malo vezani na tehnologijo. Po drugi strani pa tudi telekomi ne morejo začeti kar naenkrat nuditi storitev s svojimi vsebinami, ker te dejavnosti niso nikoli razvili in opravljali.

4. Posledice zblizevanja

Primer televizije je zanimiv predvsem zaradi tega, ker na razumljiv način kaže, kako radikalne so prihajajoče spremembe v sistemu in načinu nujenja storitev. Zblizevanje digitalnih tehnologij bo imelo izjemno velike posledice tudi na ekonomijo in na vse to, kar danes imenujemo poslovna informatika. Iz izrazito ekonomskega zornega kota je pomembno to, da prinaša zblizevanje nove storitve, na katere prej sploh nismo pomislili in tudi niso bile mogoče.

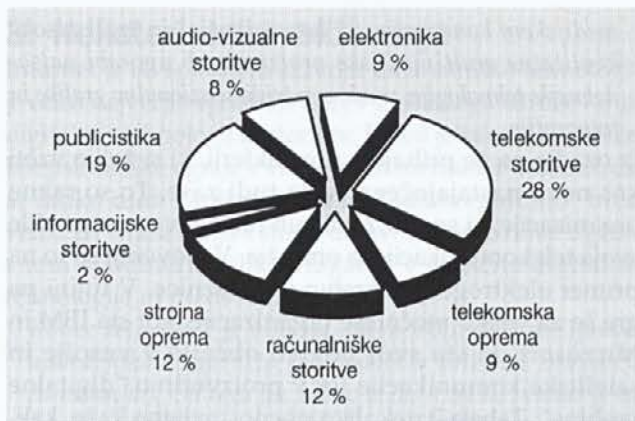
Digitalne tehnologije in storitve so pomembne že zato, ker pomenijo izjemno velik tržni delež, ki se z zblizevanjem celo povečuje, predvsem pa postaja veliko

Tabela 2: Položaj pomembnejših akterjev v zblizevanju

	Telekom	RTV publicisti	Elektronski ponudniki	Internet vsebine	Kreatorji progr. opr.	Proizvajalci
Kreiranje vsebine			■■■		■■■	
Posredovanje vsebine	■	■■■	■■■	■		■
Ponudba storitev	■■■	■■■		■■■		■
Ponudba Infrastrukture	■■■	■				

Legenda: ■■■ zelo močni
■ delno usposobljeni

(Vir: Squires, Sanders Dempsey LLP and Analysis)



Slika 1: Porazdelitev dohodka posameznih sektorjev leta 1996 (Vir: IDATE, 1997)

bolj globalen in konkurenčen. Slika 1 kaže porazdelitev dohodkov po posameznih panogah v letu 1996. Celoten svetovni trg za omenjene panoge je znašal 1.750 milijard ECU, od tega je bil delež EU 508 milijard ECU. To pa je tako velik trg, da sam od sebe spodbuja velika vlaganja v raziskave in razvoj novih produktov.

Vpliv na zaposlenost

Med ekonomisti in informatiki že leta poteka debata o tem, ali informatika zmanjšuje nezaposlenost, ali pa jo celo povečuje. Zagovorniki obeh opcij so postregli z vrsto dokazov za ali proti, vendar ostaja dejstvo, da je bil vpliv informacijske tehnologije in informatike na zaposlenost razmeroma majhen. Zanimiva je bila tudi debata v Evropski uniji. Leta 1993 je Evropska komisija objavila Belo knjigo o rasti, konkurenčnosti in zaposlovanju⁴, imenovano tudi Delorova bela knjiga, ki je prvič omenila, da bi prehod v informacijsko družbo lahko pomenil tudi zmanjšanje nezaposlenosti. Današnja Evropa se še vedno sooča z nezaposlenostjo, zato je očitno, da s to teorijo nekaj ni bilo v redu. Cilj, ki si ga je zadal tedanji predsednik Evropske komisije, je bil brez dvoma pravičen. Problem je le v tem, da je Evropa veliko prehitro pričakovala rezultate. Zmanjševanje brezposelnosti zaradi uvajanja informacijske tehnologije se v obrisih kaže le v ZDA.

Evropo upravičeno skrbi dejstvo, da rast zaposlenosti v sektorju, ki proizvaja informacijsko tehnologijo in nanjo vezane storitve, ni tako hitra kot v ZDA (2,6 milijona v ZDA in 1,8 milijona v EU). Tudi predpostavka, da bo informacijska tehnologija tako izboljšala splošno stanje v ekonomiji, da bo s tem posredno začela odpirati delovna mesta, se dosedaj ni uresničila. Primer Nemčije in še nekaterih drugih držav EU, ki so v izjemnih političnih težavah zaradi nezaposlenosti, samo potrjuje to ugotovitev.

Povečanja števila delovnih mest očitno ne bo prinesla poslovna informatika, saj je že v svoji osnovi usmerjena v racionalizacijo in povečanje učinkovitosti dela, zato raje vidi kakšno delovno mesto manj. Število delovnih mest lahko povečajo samo nove storitve. Vse kaže, da bo ravno zblíževanje digitalnih tehnologij prineslo tiste nove storitve, ki bodo odpirale nova delovna mesta. To je tudi eden od razlogov, da se je EU izjemno hitro odzvala na ta tehnološki razvoj, saj od nastanka Delorove bele knjige ni našla resničnih mehanizmov, ki bi povečali zaposlenost. Zblíževanje digitalnih tehnologij in na njih zasnovanih storitev se je kar naenkrat pojavilo kot novo upanje za Evropo. In Evropska komisija ga je zgrabila z obema rokama.

Vpliv na industrijo

Drugo pomembno vprašanje je, kako bo zblíževanje tehnologij vplivalo na globalno ekonomijo in na kakšen način. Trenutni razvoj novih tržnih produktov sicer ne gre vedno v smeri zblíževanja digitalnih tehnologij, kot jo je opredelila Evropska komisija, gre pa brez dvoma v pravo smer. Digitalni televizor, na primer, še ni osebni računalnik in obratno, vendar je le še korak do popolne združitve na tehnološki ravni. V končni fazi bo to pomenilo, da med posameznimi napravami ne bo bistvene tehnološke razlike, vse bo zgrajeno modularno, jedro vseh naprav bo mikroprocesor, vse bo programirano. To bo omogočilo velike proizvodne serije, specializacijo in izjemno prilagodljivost opreme.

Pomembna lastnost digitalne tehnologije v primerjavi z analogno je njena možnost širitve. Z uporabniške plati pomeni to, da lahko začne z majhnim in s tem cenanim sistemom, ki ga kasneje lahko razširi, ne da bi to vplivalo na njegovo arhitekturo, strukturo in tudi tehnične rešitve. Lahko se le poveča zmogljivost digitalne naprave ali poveča propustnost komunikacijskih kanalov (samo zmogljivejša linija) in podobno. S poslovnega zornega kota ima to izredne prednosti, saj so investicije postopne, poslovno tveganje pa manjše. To je med drugim tudi eden od razlogov, da je internet tako uspešen, saj je izrazito prijazen do širitve.

To se že odraža tudi v industriji, saj se računalniške firme vtikajo v posel firmam, ki so se do sedaj ukvarjale z proizvodnjo avdio in video proizvodov za široko potrošnjo. Velja seveda tudi obratno. Zanimiva in poučna je tudi zgodba Microsofta, ki se je z vsem žarom spustil v programske rešitve za video proizvode široke potrošnje.

Čeprav v svoji zeleni knjigi Evropska komisija ne omenja posebej elektronskega poslovanja, je jasno da se tudi elektronsko poslovanje dotika problema zblíževanja. Poenotena globalna digitalna komunikacijska infrastruktura, dostopnost prenosnega omrežja, kvalitetne digitalne storitve pri prenosu podatkov, možnost zaščite in zavarovanja podatkov ter druge

⁴ Growth, Competitiveness, Employment - The Challenges and Ways Forward into the 21st Century, European Commission

prednosti, ki jih prinaša združevanje digitalnih tehnologij, so brez dvoma velika pridobitev za elektronsko poslovanje. Sestavni del elektronskega poslovanja pa sta proizvodnja in trženje digitalnih proizvodov, to je vsebine v najširšem pomenu besede.

Kulturni in jezikovni vplivi

Zbliževanje digitalnih tehnologij in storitev je izjemno pomembno tudi s sociološkega, kulturnega in jezikovnega vidika. Še posebej za majhen narod, kot smo Slovenci. Globalnost ima tudi svoje pasti! Kdo in zakaj bo, na primer, gledal domačo televizijo, če bo imel prek interneta ali drugače dostop do neomejenega števila drugih televizijskih programov. Kako v taki globalnosti ohraniti svojo nacionalno in osebno identiteto? Kako v taki globalnosti ponuditi drugim svoje kulturne in nacionalne "proizvode"? Takih vprašanj bo na naši poti v informacijsko družbo še zelo veliko.

Tabela 3: Porazdelitev jezikov na spletnih straneh v Evropi

	EU	države izven EU
Samo nacionalni jezik (vključno z VB in Irsko)	56 %	33 %
Samo angleščina (izven VB in Irske)	11 %	12 %
Nacionalni jezik in angleščina	25 %	43 %
Večjezično	8 %	10 %

(Vir: Databank Consulting, 1997)

Problem jezikov je viden na internetu, ki je večinsko angleški, vendar kaže poudariti, da se stanje naglo spreminja. Mimogrede, tu kaže opozoriti tudi na razmišljanja o tem, da bo tehnološki razvoj rešil tudi problem jezikov. Možnosti je več. Idealno bo avtomatično prevajanje pisnega gradiva ali govora. Ne kaže dvomiti, da bo ta problem rešen v neki ne preveč oddaljeni bodočnosti. To bo zopet ena od digitalnih storitev, ki bo interaktivno dostopna. Nekoliko bližje so druge možnosti, ki jih omogoča digitalna tehnologija. Oddajanje po digitalnih kanalih bistveno olajša zvočno sinhronizacijo (virtualni zvočni kanali ob sliki) ali podnapise, tako da bo lahko gledalec izbral med zelo široko paleto različnih jezikov.

Med kulturne vplive lahko vključimo tudi celotno področje izobraževanja, ki ga bodo omenjene tehnologije spremenile v osnovi.

5. Ovire, ki preprečujejo zbliževanje

Zbliževanje digitalnih tehnologij in storitev spodbuja razvoj telekomunikacij, javnih medijev in informacijske tehnologije, vendar se v Evropi in seveda v Sloveniji srečuje z vrsto zakonskih in sistemskih ovir, ki jih postavlja država. Tu ne gre za zavesten odpor države do tehnološkega razvoja, ampak je le odsev dosedanje vloge države pri zagotavljanju pogojev za delovanje

telekomunikacijskih storitev in javnih medijev. Država se zelo težko poslavlja od te vloge in brez ekonomske in še kakšne druge prisile tega nikoli ne bi storila sama od sebe. Vendar se od države pričakuje, da bo nosilec in podpornik sprememb, ki bodo zagotavile globalno zbliževanje digitalnih tehnologij in storitev.

Poglejmo, kje so ovire, na katere je pokazala Evropska komisija:

- **Zakonske omejitve pri izgradnji in uporabi infrastrukture** – V večini evropskih držav je bila v preteklosti to najhujša omejitev, saj je že v osnovi preprečila integracijo digitalnih storitev. Zato je bil prvi cilj evropske strategije popolna liberalizacija celotnega področja telekomunikacij.
- **Cena telekomunikacijskih storitev** – Visoke cene telekomunikacijskih storitev in drugih infrastrukturnih omrežij (npr. televizija, kabelsko omrežje) imajo zelo neugoden vpliv na trg. Eden od razlogov, da je internet tako uspel v ZDA, je pravzaprav njihova »ploščata« struktura cen, ki je omogočala proste lokalne telefonske klice, zelo nizke cene najetih linij itd.
- **Razpoložljivost digitalizirane vsebine** – Povečana zmogljivost tehnologije, kot je povečanje propustnosti komunikacijskih kanalov, povečanje zmogljivosti računalniške opreme in podobno, prenašajo ozko grlo s tehnološke na vsebinsko stran. Vedno bolj prihaja v ospredje vprašanje, kaj lahko ponudiš uporabniku in prodaš.
- **Razdrobljenost evropskega trga** – Evropski trg je zaradi državnih, kulturnih in jezičnih mej zelo razdrobljen, kar še velja posebej za informacije in vsebino, kjer pride do izraza jezik. Ker Evropa želi ohraniti in celo razviti svojo večkulturnost in večjezičnost, je to eden od omejitvenih faktorjev, na katerega bo morala računati.
- **Nezadostna zaščita intelektualne lastnine** – Kdorkoli bo ponujal storitve prek digitalnih omrežij, bo seveda zahteval, da je ta ustrezno zaščiten. To velja seveda tudi za publicistično dejavnost in za ostale ponudnike, ki se odločajo za izrazito inovativne storitve. Da se s tem srečujemo tudi v "off-line" svetu je jasno, v "on-line" okolju pa postane vse skupaj še posebej kritično. Kot primer, prodaja knjig na CD-romih je minimalna, razen knjig, ki nimajo več avtorske zaščite. Podobno velja za ponudbo avtorskih del po internetu. Ta problematika je še vedno teoretično ali načelno nerazčiščena.

Te in take ovire naj bi Evropska unija prednostno odpravila. To je tudi razlog, da je objavila Zeleno knjigo o zbliževanju in s tem začela pripravljati politično podlago za konkretne ukrepe in spremembo zakonodaje.

6. Slovenija

Slovenija se približuje Evropi na izjemno pragmatičen način, ki včasih meji že na fatalizem. Na področju informacijskih tehnologij sicer opazujemo, kaj se dogaja v Evropi, vendar zaenkrat še ne ukrepamo, razen če v to nismo prisiljeni. Stanje pa se bo zelo naglo spremenilo, ko se bomo konec leta začeli pogajati o našem polnopravnem članstvu. Dejstvo namreč je, da se na področju informatike ne bomo imeli kaj pogajati, saj Evropska komisija sploh ne bo odprla teh vprašanj. Postavila nas je pred zelo preprosto izbiro - vzemi ali oddidi. Ker se ne bomo pustili odgnati, bomo do polnopravnega članstva sprejeli vse evropske usmeritve in direktive in s tem rešili vse naše sistemske in zakonske dileme. Vprašanje, ki se ob tem postavlja, je le eno - ali s tem odlašanjem ne bomo izgubili preveč dragocenega časa.

Svetovna trgovinska organizacija (WTO) je objavila, da je Sporazum WTO o bazičnih telekomunikacijah, ki je stopil v veljavo 5. februarja 1998, podpisalo 72 držav, ki pokrivajo 93% globalnega telekomunikacijskega trga, vrednega 500 milijard ECU. Sporazum je v osnovi enak, kot je sistem liberalizacije, ki ga je s 1. januarjem 1998 uveljavila Evropska komisija za svoje članice. Slovenije zaenkrat ni med podpisniki. Kot vemo, bo v Sloveniji postopek liberalizacije trajal formalno do leta 2001, v praksi pa najbrž do polnopravnega članstva, če si v pogajanjih z Evropsko komisijo ne bomo izborili še kakšnega odloga. Vendar je to malo verjetno.

Vlada RS je obravnavala Ministrsko Bonnsko deklaracijo o globalnih komunikacijskih omrežjih in jo tudi sprejela. Politična volja za reševanje teh problemov na evropski način je torej tukaj. Zakonodajna na področju intelektualne lastnine je zelo blizu evropski. Podobno velja za varovanje podatkov. Na področju predpisovanja standardov imamo zaradi izjemno obsežnega dela več problemov. V glavnem pa jih, sicer parcialno in nepovezano, še kar uspešno pripravljamo. Velik problem je še javni sektor zaradi izvajanja javnih naročil, vendar nam tudi tukaj Evropska komisija že gleda pod prste. Vse kaže, da kljub trenutnim problemom in zastojem lahko upamo na najboljše tudi pri našem prehodu v informacijsko družbo.

Mislim, da bodo v Sloveniji v določenih krogih zelo veliki odpori proti sistemskim ukrepom, ki bodo spodbujali zblíževanje digitalnih tehnologij in še posebej storitev. Zelo težko si, na primer, predstavljam tako radikalno liberalizacijo radia in televizije, kot jo v

delovnih gradivih predvideva Evropska komisija. Veliko lažje nam bo šlo na tehnološkem področju.

7. Zaključek

Za zaključek pogledjmo samo to, kako si Evropska komisija predstavlja vlogo države in njenih institucij pri spodbujanju novih tehnologij in storitev. Izhaja iz načelnega stališča, da je potrebno državno regulacijo omejiti samo na tista področja, kjer je to nujno potrebno da se dosežejo jasno opredeljeni cilji. Vključevanje v konvergentno okolje mora omogočiti vsem, ki so za to sposobni. In končno, državno urejanje mora biti predvsem v korist uporabnikov.

Zblíževanje digitalnih tehnologij se bo odrazilo tudi v poslovni informatiki. V tem članku se nisem imel namena spuščati v to področje, saj si zasluži posebno obravnavo. Že na prvi pogled je očitno, da dobivajo tudi poslovni informatiki dodatna orodja in tudi dodatne naloge. Organizacije bodo vedno bolj digitalizirane in bodo potrebovale vedno nove storitve. Informatiki smo pred leti začeli svojo kariero v podjetjih kot nadomestni računovodje s finančnimi obdelavami, potem smo se z večjim ali manjšim uspehom spustili na področje sistemov za odločanje. Trenutno smo z internetom v prvi vrsti tistih, ki skrbijo za stike z javnostjo in sestankovanje na daljavo. Za naslednji korak pa se bomo morali posebej pripraviti in razmisliti, kako se lotiti digitalnih organizacij in njihove digitalne okolice.

Literatura

- 1 Green Paper on the Convergence of the Telecommunications, Media and Information Technology Sectors, and the Implications for Regulation - Towards an Information Society Approach. European Commission, COM(97)623, 3 December 1997, Brussels
- 2 Cohesion and the Information Society. European Commission, COM(96), 22 January 1996, Brussels
- 3 Europe and the Global Information Society - Recommendations to European Council, European Commission, 1994, Brussels
- 4 Information Society Trends, Issues 74-77, European Commission, Brussels
- 5 Growth, Competitiveness, Employment - The Challenges and Ways Forward into the 21st Century, White Paper, European Commission, 1993, Brussels
- 6 BAVEC, Cene (ur.), SCHRAG, Anton (ur.), WOJCIESZKO, Grazyna (ur.). Report. Implementation of the Action Plan. Ljubljana: Ministry of Science and Technology of the Republic of Slovenia, 1997. Ljubljana. 179 str. ISBN 961-6101-03-X. ŠCOBISS-ID 306062Č
- 7 BAVEC, Cene, BÜSCHER, Reinhard. The path to the information society: Options and strategies for Slovenia. Quark. [English ed.], 1996, str. 30-33. [COBISS-ID 226124]

Dr. Cene Bavec je državni podsekretar v Ministrstvu za znanost in tehnologijo in pokriva znanstveno in tehnološko multilateralno sodelovanje in sodelovanje z Evropsko komisijo. Zadnja tri leta je aktivno sodeloval v delu EU-CEEC foruma o informacijski družbi, ki ga je vodil evropski komisar M. Bangemann. Vodil je tudi dva enoletna mednarodna panela predstavnikov vseh pridruženih članic in Evropske komisije za pripravo strokovnih gradiv za delo Forumu. Je član Evropskega foruma o informacijski družbi, član posebne strokovne skupine Evropske komisije za področje informacijske družbe ter predstavnik pridruženih članic v G-7 pilotskem projektu elektronskega poslovanja. V tem času je imel preko deset vabljenih predavanj na konferencah v organizaciji Evropske komisije in članic EU.

KAKO DO BOLJ FLEKSIBILNE PREDSTAVITVE POSLOVNIH PRAVIL?

Jurij Jaklič, Janez Grad
Ekonomska fakulteta, Univerza v Ljubljani
e-pošta: jurij.jaklic@uni-lj.si

Povzetek

V prispevku predlagamo način predstavitve poslovnih pravil v informacijskih sistemih, ki omogoča večjo fleksibilnost poslovnih pravil, njihovo večkratno uporabljivost, hkrati pa približa delo z njimi tudi uporabniku. Predlagani način predstavitve temelji na trinivojskem pogledu na odločitvena pravila, s katerimi oblikujemo poslovna pravila, objektnem modelu ter logični odločitveni tabeli. Rešitev smo tudi praktično preverili na dva načina: z izdelavo prototipa, ki v celoti implementira trinivojski pogled na pravila, ter z analizo možne uporabe na konkretnem primeru.

Summary

In the paper we propose and describe a way of business rules representation within information systems that enables their greater flexibility and reusability. The representation is based on the three level view of decision rules, object oriented models and decision tables. The proposed approach of representation was carried through and tested in two ways: (i) by designing a prototype which fully implemented the three level view of decision rules, and (ii) by analyzing its possible application within a selected case problem.



1 PROBLEMI PREDSTAVITVE ODLOČITVENIH PRAVIL V INFORMACIJSKIH SISTEMIH

1.1 Problematika predstavitve poslovnih pravil

Vsak poslovni sistem se mora odzivati na hitre in neprestane spremembe v okolju, kar povzroča probleme pri vzdrževanju in upravljanju informacijske tehnologije v podjetjih [19]. Ker informatiki pogosto ne morejo pravočasno ugoditi zahtevam uporabnikov po novih rešitvah in spremembah obstoječih, se vedno pogosteje dogaja, da uporabniki (resda preprostejše) rešitve razvijejo samostojno s pomočjo sodobne, prijazne tehnologije, pri tem pa jim z nasveti lahko pomagajo tudi informatiki. Hkrati pridejo do rešitev tudi hitreje, saj poznajo področje, za katerega oblikujejo rešitev, in se s tem izognejo pogosti in intenzivni komunikaciji z analitiki in načrtovalci [13]. Seveda morajo imeti uporabniki na voljo ustrezno temeljno znanje informatike in do uporabnika prijazno in za uporabo enostavno informacijsko tehnologijo.

1.2 Dosedanji razvoj

V dosedanjem razvoju informacijske tehnologije je bil storjen velik napredek na področju predstavitve po-

datkov. Preprost relacijski podatkovni model, neodvisnost podatkov od programskih rešitev, ki jo zagotavljajo sistemi za upravljanje baz podatkov, logična neodvisnost podatkov, ki je dosežena s trinivojskim modelom, ter prijazni grafični uporabniški vmesniki – vse to omogoča, da lahko uporabnik s temeljnim znanjem informatike danes *samostojno* modelira podatke, zgradi preprosto bazo podatkov in jo uporablja. Zato je tudi večina poslovnih rešitev, ki jih izdelajo uporabniki samostojno, s področja manipulacije (poizvedovanje po bazi podatkov) ali analize podatkov (preglednice).

Objektni pristop je omogočil velik napredek tudi na področju izgradnje in vzdrževanja uporabniških vmesnikov (zaslonskih mask in poročil). Tako je mogoče danes hitro narediti grafične uporabniške vmesnike s postavljanjem objektov na delovno površino ter preprostim povezovanjem dogodkov in aktivnosti.

Tudi za poslovna pravila velja, da jih moramo, če želimo slediti in se prilagajati spremembam v okolju, neprestano spreminjati, saj predstavljajo del modela realnega sveta. Za uporabnika to pomeni težavo, saj v največ primerih tega ne zna in ne zmore storiti sam. Zato bi bilo zaželeno, da bi lahko uporabnik samostojno manipuliral (definiral, spreminjal, dodajal, brisal) tudi poslovna pravila.

1.2.1 Vizualizacija poslovnih pravil

V fazi analize sistema in načrtovanja informacijskega sistema moramo uporabiti za predstavitev odločitvenih pravil tehniko, ki je hkrati dovolj blizu naravnemu načinu razmišljanja (opisni pripovedi) in tudi dovolj formalna, da lahko pravila čim lažje kodiramo. Za predstavitev odločitvenih pravil poznamo danes več formalnih tehnik, med katerimi so najbolj znane: pravila "če - potem", odločitvena drevesa, logične odločitvene tabele in logični diagrami poteka. Njihova skupna značilnost je predvsem usmerjenost v dokumentacijo, torej zapis - prikaz poslovnih pravil, kar je posledica namena njihove uporabe. Zaradi težnje po približevanju uporabniku je poudarjena komponenta vseh tehnik predvsem vizualizacija poslovnih pravil, hkrati pa so te tehnike oddaljene od predstavitve istih pravil v informacijskih sistemih. To pomeni, da obstaja prepad med zunanjimi predstavitevami poslovnih pravil in njihovim kodiranjem.

Na področju primernosti različnih predstavitev odločitvenih pravil (logične odločitvene tabele, odločitvena drevesa, pravila "če - potem", strukturirana angleščina ...) je bilo opravljenih več empiričnih raziskav, na primer [15], [26] in [30], ki te predstavitve primerjajo med seboj z vidika uporabnika. Večina teh raziskav ni pokazala prednosti pri uporabi ene tehnike pred ostalimi predstavitevami ali pa si izsledki raziskav med seboj celo nasprotujejo. Zato lahko sklepamo, da je izbor primerne načina predstavitve odločitvenih pravil pogojen predvsem z osebnimi preferencami in z vrsto (kompleksnostjo) problema.

1.2.2 Logične odločitvene tabele

Logična odločitvena tabela je način prikaza odločitvenih pravil z uporabo tabelarične strukture. Zaradi zgoščenega in preglednega načina prikazovanja pogojev in aktivnosti [12], ki so vključene v odločitvah, so logične odločitvene tabele zelo primerne za prikazovanje, razumevanje in spremljanje logike dogajanja kompleksnih odločitvenih problemov. S povezovanjem več odločitvenih tabel lahko modeliramo poljubno kompleksno odločitveno situacijo.

Tako opredeljene logične odločitvene tabele so, kot že omenjeno, namenjene predvsem predstavitvi - vizualizaciji odločitvenih pravil. V nadaljnjem razvoju logičnih odločitvenih tabel pa se mnogo uporablja tudi konceptualna definicija logične odločitvene tabele, ki ne implicira načina predstavitve odločitvenih pravil, saj logično odločitveno tabelo definiramo [29] kot funkcijo, ki preslika kartezični produkt stanj pogojev v kartezični produkt stanj aktivnosti:

$$DT: CT_1 \times CT_2 \times \dots \times CT_{num} \rightarrow V_1 \times AV_2 \times \dots \times AV_{anum}$$

kjer sta CT_i urejena množica n_i stanj pogojev subjekta pogoja CS_i , ter AV_j množica vrednosti aktivnosti.

Pri pridobivanju in vzdrževanju odločitvenih pravil se srečujemo s protislovji in pomanjkljivostmi v teh pravilih. Zato je izrednega pomena preverjanje in potrjevanje pravilnosti opisa odločitvene logike. V tem pogledu imajo logične odločitvene tabele pred ostalimi predstavitvenimi tehnikami pomembno prednost, saj je z njihovo uporabo mogoče na enostaven način preverjati želene lastnosti odločitvenih pravil: popolnost, odsotnost protislovnosti ter odsotnost odvečnosti.

Popolnost. Pravila so popolna, če lahko odgovorijo na vsa možna stanja v domeni. Z drugimi besedami, za vsako kombinacijo vrednosti parametrov mora biti s pravili določeno, kaj naj se zgodi. Popolnost torej ustreza robustnosti [8]. Če so pravila predstavljena v odločitveni tabeli, je popolnost mogoče enostavno preverjati s časovno zahtevnostjo $O(n)$, kjer je n število pogojev. Časovno bolj zahtevno pa je iskanje manjkajočih pravil, to je $O(2^n)$ [8].

Dvoumnost, protislovnost in odvečnost. Kadar obstaja kombinacija vrednosti parametrov, ki zadošča pogojem dveh ali več pravil z različnimi množicami aktivnosti, pravimo, da so pravila dvoumna. Kadar se množice aktivnosti dvoumnih pravil med seboj izključujejo (ne morejo biti izvršene hkrati), so pravila protislovnostna. Če pa so množice aktivnosti ekvivalentne, pa so nekatera odločitvena pravila odvečna. Časovna zahtevnost znanih algoritmov za odkrivanje protislovnosti in odvečnosti je $O(n^2)$ [8].

V [22] avtorja razdelita uporabo logičnih odločitvenih tabel v dve fazi: *aktivno*, v kateri modeliramo odločitveni proces, in *pasivno*, v kateri le uporabljamo logične odločitvene tabele za odločanje. Očitno je, da so pri pasivni uporabi nepopolnost, protislovnost in odvečnost nezaželene lastnosti poslovnih pravil. Avtorja pa zagovarjata pozitivno vlogo omenjenih lastnosti v fazi modeliranja.

Logične odločitvene tabele so bile zaradi omenjenih možnosti preverjanja popolnosti in nedvoumnosti v ta namen že uporabljene v ekspertnih sistemih [16] ter za testiranje programov [23]. Njihova uspešnost se je pokazala tudi v raziskavi [31], v kateri so ugotovili precej nepopolnosti, protislovnosti in odvečnosti na primeru kliničnih navodil.

1.2.3 Kodiranje poslovnih pravil

Čeprav so poslovna pravila jedro kompleksnih poslovnih uporabniških rešitev, je bilo prav na tem področju narejenega najmanj, saj je potrebno danes veliko večino programske logike kodirati v programskih jezikih tretje generacije. Čeprav so bili tudi na tem področju v zadnjih letih narejeni določeni koraki, ki jih kratko predstavljamo v nadaljevanju, pa ostaja problem večje avtomatizacije kodiranja poslovne logike še vedno odprt [2].

Omejitve različnih vrst: referenčne, omejitve območja ali eksplicitne omejitve, ki jih je mogoče definirati v razširjenem relacijskem modelu (t.i. RM/T model [27]), predstavljajo določene vrste poslovnih pravil. Mnogo več poslovnih pravil je mogoče v okviru baze podatkov kodirati s sprožilci. Robustnost informacijskih sistemov se s tem poveča, saj se poveča konsistentnost poslovnih pravil v rešitvah, ki uporabljajo isto bazo podatkov.

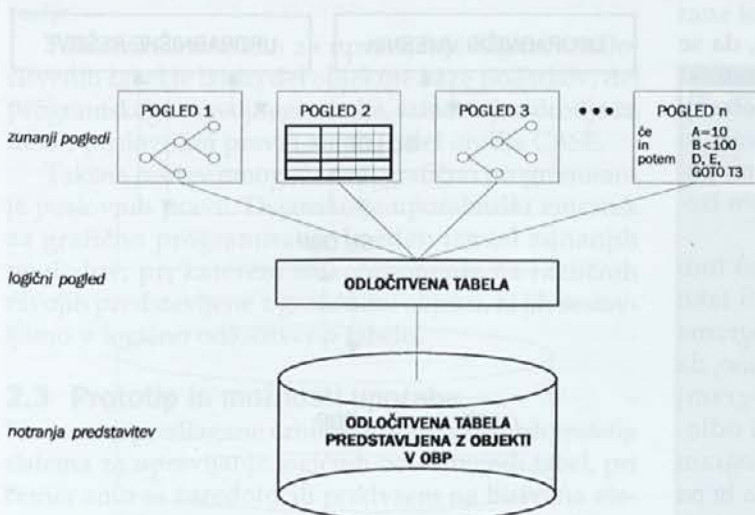
Za objektni pristop je značilno, da so poslovna pravila kodirana v metodah objektov, z drugimi besedami na strani strežnikov. Tudi to ima za posledico lažje vzdrževanje, pravila so bolj konsistentna, razviti sistemi so bolj prožni ...

V obeh primerih: pri sprožilcih in pri metodah objektov pa so poslovna pravila še vedno kodirana v programskem jeziku tretje ali četrte generacije. Prednosti, ki jih prinaša objektni pristop, so predvsem posledica spremenjene topologije programa, manj pa posledica spremembe načina kodiranja poslovnih pravil.

1.3 Želene lastnosti predstavitev poslovnih pravil

Zdaj lahko opredelimo, kakšne zahteve bi moral izpolnjevati način predstavitve poslovnih pravil:

- Isto tehniko prikaza poslovnih pravil naj bi bilo mogoče uporabljati v vseh fazah razvoja informacijskega sistema, tudi pri kodiranju in vzdrževanju sistema. Z drugimi besedami, ne glede na način notranje predstavitve poslovnih pravil (kodiranja) naj si bo mogoče poslovna pravila ogledati in jih spreminjati z uporabo ene izmed tehnik, ki so namenjene predvsem vizualizaciji in so prijazne do uporabnika hkrati pa dovolj formalne za zapis v računalniško podprtem informacijskem sistemu.
- Orodje za kodiranje in ažuriranje poslovnih pravil naj bo vizualno.



Slika 1: Trinivojski pogled na poslovna pravila

- Vizualizacija naj bo fleksibilna, kar pomeni, da lahko uporabnik izbere tako tehniko za predstavitev poslovnih pravil, ki najbolj ustreza njemu in obravnavanemu problemu.
- Kodirana pravila naj bodo fleksibilna, kar pomeni, da naj jih bo mogoče spreminjati na enostaven način, spremembe pa naj se takoj odražajo v informacijskem sistemu.
- Želene lastnosti poslovnih pravil (popolnost, odsotnost odvečnosti in protislovnosti) naj se preverjajo avtomatično.
- Poslovna pravila in njihove komponente naj bo mogoče uporabiti večkrat, ne samo znotraj ene programske rešitve.

2 PREDLAGANA REŠITEV

Ugotovimo lahko, da že obstajajo tehnike in pristopi s posameznimi značilnostmi, ki smo jih opredelili kot želene lastnosti predstavitve poslovnih pravil. Na primer: mogoče je izbirati različne tehnike za vizualizacijo glede na uporabnikove preference in značilnosti problema, logične odločitvene tabele omogočajo avtomatično preverjanje lastnosti poslovnih pravil, objektni pristop zagotavlja fleksibilnost in ponovno uporabljivost ... Menimo, da je mogoče te tehnike in pristope ustrezno kombinirati tako, da hkrati dosežemo večino zastavljenih ciljev. V ta namen lahko uporabimo trinivojski pogled na poslovna pravila, ki ga predstavljamo v nadaljevanju.

2.1 Trinivojski pogled na poslovna pravila

Kombiniranje tehnik in pristopov omogoča trinivojski pogled na poslovna pravila, ki ločuje zunanje poglede na poslovna pravila, to je poglede uporabnikov, conceptualni pogled, na katerem so pravila predstavljena z logičnimi odločitvenimi tabelami, ter notranjo predstavitev, kjer so logične odločitvene tabele in vse njihove komponente shranjene kot objekti v objektni bazi podatkov¹ (slika 1).

V tem primeru uporabimo logične odločitvene tabele le kot *logično predstavitev* poslovnih pravil, kar omogoča avtomatsko in sprotno preverjanje popolnosti, protislovnosti in odvečnosti.

Uporabnik pa si sam izbere *zunanji pogled* na poslovna pravila, oziroma predstavitev le-teh. Ta pogled je lahko spet logična odločitvena tabela, lahko so

¹ Seveda bi lahko uporabili tudi objektno relacijske (univerzalne) baze podatkov. Ker je mogoče objektni pristop uporabiti tudi v okolju relacijskih baz podatkov, je predlagani pristop primeren tudi za relacijska okolja.

odločitvena drevesa, pravila "če - potem" in podobno. Z neodvisnostjo med zunanji predstavitevami in logično predstavitevjo in s primernimi grafičnimi vmesniki za posamezne vrste zunanjih pogledov omogočimo uporabniku, da na enostaven in sebi prilagojen način kodira in uporablja kompleksna poslovna pravila.

Pri modeliranju odločitvenih pravil in odločanju je vizualizacija, kot ugotavljajo avtorji v [26], izrednega pomena. Poglavitni namen različnih tehnik predstavitve odločitvenih pravil je prav vizualizacija. Če (kadar) namen programske rešitve ni vizualizacija (npr. pri izvajanju), oblika predstavitve ni pomembna.

Če analiziramo predlagani trinivojski pogled na odločitvena pravila, ugotovimo, da ta popolnoma ustreza tem ugotovitvam, saj ločuje zunanji pogled (predstavitve, vizualizacijo) in logični ter notranji nivo, pri katerih vizualizacija ni pomembna, saj uporabnik teh nivojev "ne vidi". Na logičnem nivoju smo uporabili konceptualno definicijo logičnih odločitvenih tabel (glej podpoglavje 1.2.2), ki je neodvisna od predstavitve – vizualizacije.

Notranja predstavitev oziroma implementacija logičnih odločitvenih tabel je načeloma lahko poljubna, vendar predlagamo, da jih implementiramo z objekti v *objektni bazi podatkov*. Z analizo konceptualne definicije logične odločitvene tabele lahko ugotovimo, da je ta sestavljena iz podkomponent in te naprej iz podkomponent na nižjih nivojih: pravil, subjektov pogojev, vrednosti subjektov pogojev, subjektov aktivnosti ... Z objekti predstavimo tako logično odločitveno tabelo kot tudi vse njene komponente. S postavitvijo logičnih odločitvenih tabel v okolje objektno usmerjenih baz podatkov podedujemo pridobitve objektno usmerjenega pristopa: ponovno uporabljivost vseh elementov (od odločitvenih tabel do pravil in pogojev), možnost vzporednega izvajanja pravil brez dodatnega programiranja ...

Preslikave med različnimi nivoji omogočajo, da se vsaka uporabnikova sprememba poslovnih pravil, ki se zgodi na zunanjem nivoju, takoj odraža v notranji predstavitvi in s tem v vseh programskih rešitvah, ki ta poslovna pravila uporabljajo, vključno z vsemi zunanjimi pogledi. Posledično je mogoče v vsakem trenutku spremeniti pogled na pravila.

Na logične odločitvene tabele lahko gledamo tudi kot na način predstavitve programske kode. S tako implementacijo odločitvenih tabel torej del programa predstavimo kot objekte v bazi podatkov. Če vemo, da je mogoče vsak logični diagram poteka (program) emulirati z eno ali več povezanimi logičnimi odločitvenimi tabelami [20], potem lahko vsak program predstavimo v objektni bazi podatkov. Ideja, da bi na program gledali kot na objekte, ni nova, je pa obetajoča in še dokaj neraziskana [1].

2.2 Implementacija rešitve

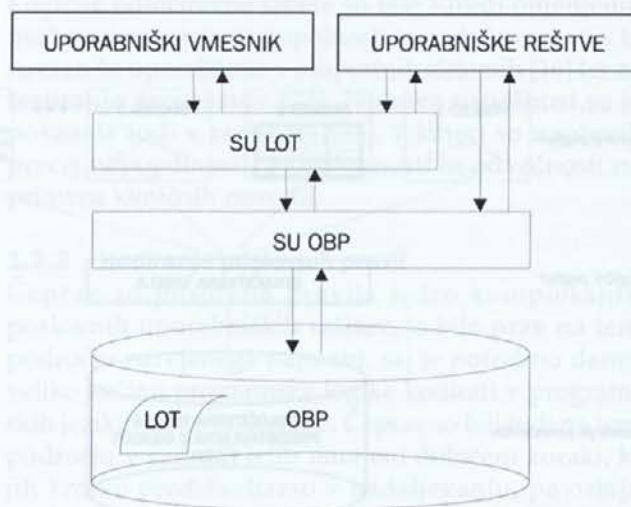
Predlagano rešitev implementiramo s sistemom za upravljanje logičnih odločitvenih tabel, s pomočjo katerega lahko:

- tvorimo nove komponente logične odločitvene tabele,
- iz obstoječih komponent tvorimo nove na višjih nivojih,
- tvorimo logično odločitveno tabelo iz obstoječih komponent,
- spreminjamo obstoječe komponente in logične odločitvene tabele ter
- izvajamo logične odločitvene tabele.

Del sistema za upravljanje logičnih odločitvenih tabel pa so tudi uporabniški vmesniki, ki ustrezajo zunanjim pogledom na poslovna pravila (slika 2).

Sistem za upravljanje logičnih odločitvenih tabel sestavljajo torej *predefimirani razredi objektov*, ki predstavljajo komponente logične odločitvene tabele, skupaj z implementacijo pripadajočih operacij ter *uporabniški vmesniki* za posamezne vrste zunanjih pogledov. Najpomembnejše metode posameznega razreda objektov so tiste, ki omogočajo delo z logičnimi odločitvenimi tabelami: povezovanje komponent v komponente na višjem nivoju, odstranjevanje komponent, spreminjanje, izvajanje – ovrednotenje komponent, preverjanje zelenih lastnosti poslovnih pravil ... Poleg teh metod pripadajo posameznim razredom objektov še metode, ki omogočajo delo na nivoju zunanjih pogledov (izpis objekta, na primer pogoja, za različne poglede na logične odločitvene tabele).

Predlagamo sedem razredov objektov, ki omogočajo implementacijo logičnih odločitvenih tabel v objektni bazi podatkov: razred *logičnih odločitvenih tabel*,



Slika 2: Arhitektura sistema

razred *parametrov*, razred *pravil*, razred *pogovr*, razred *subjektov pogojev*, razred *operandov* ter razred *subjektov aktivnosti*.

Subjekti pogojev, ki nastopajo v logičnih odločitvenih tabelah, so v večini primerov neki relacijski izrazi, v katerih primerjamo med seboj dve ali več vrednosti. V objektnem okolju bodo te vrednosti svojstva (atributi) nekaterih objektov. Tudi kadar nastopajo v pogojih neke fiksne vrednosti, ki nastopajo le v logičnih odločitvenih tabelah, jih lahko obravnavamo kot attribute objektov. Objekte, katerih svojstva nastopajo v pogojih, bomo imenovali *parametri* logične odločitvene tabele. Sprememba operanda v subjektu pogoja je torej mogoča na dva načina: spremeni se vrednost atributa parametra ali pa zamenjamo parameter z drugim objektom istega razreda.

Objekti razreda *pogovr* predstavljajo povezavo med subjekti pogojev in pripadajočo (zahtevano) vrednostjo v nekem pravilu.

V objektnem okolju je logična odločitvena tabela povezana z drugimi objekti na dva načina:

- V subjektih pogojev uporablja vrednosti atributov objektov – parametrov logične odločitvene tabele.
- Z aktivnostmi spreminja stanje sistema. Subjekti aktivnosti torej predstavljajo zahtevo objektu naj izvede svojo metodo. Tudi ti objekti so parametri logične odločitvene tabele.

Izvajanje logične odločitvene tabele, predstavljene z objekti, temelji na klicih metod za ovrednotenje posameznih pravil, ki pripadajo tabeli. Postopek se nadaljuje po hierarhiji komponent vse do podkomponent na najnižjih nivojih. Izvajanje torej poteka prav tako modularno, kot je modularna struktura logične odločitvene tabele. Modularno izvajanje seveda pomeni, da je izvajanje tako predstavljenih poslovnih pravil časovno bolj zahtevno od enakovrednega programa, kodiranega v programskem jeziku tretje generacije.

Tako izdelan sistem za upravljanje logičnih odločitvenih tabel je lahko del objektne baze podatkov, del programskega razvojnega okolja, samostojno orodje za delo s poslovnimi pravili ali pa je del orodja CASE.

Takšna rešitev omogoča tudi grafično programiranje poslovnih pravil. Dejansko je uporabniški vmesnik za grafično programiranje le eden izmed zunanjih pogledov, pri katerem so komponente na različnih nivojih predstavljene z grafičnimi objekti, ki jih sestavljamo v logično odločitveno tabelo.

2.3 Prototip in možnosti uporabe

Na osnovi predlagane arhitekture smo izdelali *prototip* sistema za upravljanje logičnih odločitvenih tabel, pri čemer smo se osredotočili predvsem na bistvena elementa predlaganega sistema, to je trinivojski model ter predstavitev komponent z objekti, kar ima za posledico

modularno sestavljivost odločitvenih pravil ter ponovno uporabljivost. Poglavitne cilje pri izdelavi prototipa lahko strnemo v naslednje:

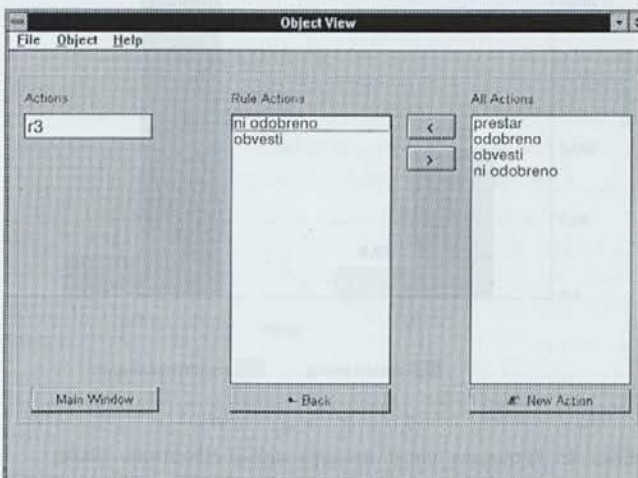
- Proučevati smo želeli predvsem funkcionalnost sistema, zato nismo posvečali posebne pozornosti uporabniškemu vmesniku, čeprav je ta izjemnega pomena, če želimo sistem nameniti uporabniku.
- Iz istega razloga tudi nismo implementirali optimizacijskih metod (npr. relativne frekvence pravil).
- Prototip naj upošteva trinivojski model, s katerim uporabniku ponujamo različne poglede na poslovna pravila, ki so na logičnem nivoju predstavljena z logičnimi odločitvenimi tabelami, le-te pa dejansko predstavljene v objektnem modelu z objekti, ki so komponente odločitvene tabele. Poseben poudarek smo dali spodnjemu nivoju, torej notranji predstavitvi z objekti.

V okviru analize izdelave in uporabe prototipa rešitve so nas posebej zanimali naslednji vidiki:

- težave pri implementaciji predlaganega modela,
- možnost uporabe objektnega pristopa, ki ga predvideva rešitev, v okoljih, kjer informacijski sistem temelji na relacijskem podatkovnem modelu, ter časovna zahtevnost izvajanja z objekti predstavljene logične odločitvene tabele.

V prototipu smo v celoti implementirali trinivojsko arhitekturo sistema. Preslikava med logičnim in notranjim nivojem je implementirana s povezavami med objekti, ki komponente povežejo v logično odločitveno tabelo, ter metodami objektov, ki omogočajo delo z logično odločitveno tabelo (tvorba, izvajanje, ažuriranje, preverjanje pravilnosti).

Na zunanjem – predstavitevsem nivoju lahko uporabnik izbere pogled na odločitvena pravila, ki mu najbolj ustreza. Izbere v pogovornem oknu tabelo in nato način prikaza pravil. V prototipu sta implementirana le dva pogleda: tabelarični in objektni.



Slika 3: Objektni pogled na pravila – "sestavljanje aktivnosti"

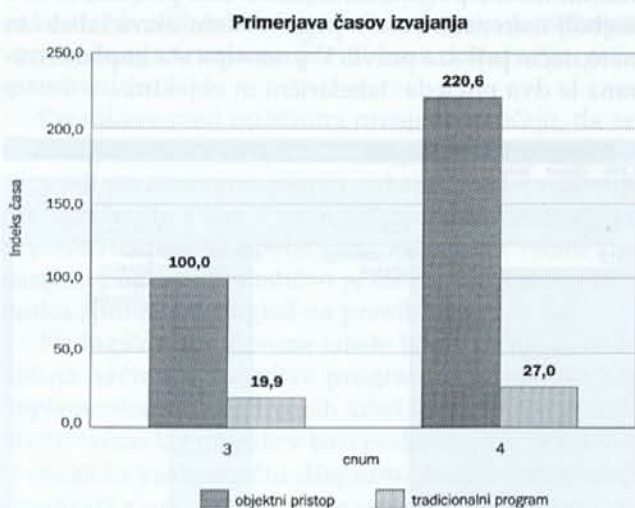
Objektni pogled je blizu notranji predstavitvi odločitvenih pravil, saj uporabnik vidi komponente in podkomponente, ki jih sestavljajo. Uporabnik lahko logično odločitveno tabelo sestavlja z že obstoječimi komponentami (ponovna uporabljivost) ali pa tvori nove. Nadaljnji razvoj tega pogleda bi nas pripeljal do vmesnika za grafično programiranje poslovnih pravil.

2.3.1 Hitrost izvajanja logičnih odločitvenih tabel, predstavljenih z objekti

Kot že povedano v predstavitvi predlagane rešitve, je pričakovana časovna zahtevnost izvajanja tako predstavljenih poslovnih pravil večja od enakovrednega programa, kodiranega v programskem jeziku tretje generacije.

Primerjavo učinkovitosti (hitrosti izvajanja) smo preverili z eksperimentom, v katerem smo merili hitrost izvajanja logične odločitvene tabele z omejenimi pogoji, predstavljene z objekti, in ekvivalentnega programa, kodiranega v programskem jeziku tretje generacije.

Na sliki 4 so prikazani dobljeni rezultati merjenja velikega števila ponovitev izvajanja logične odločitvene tabele z elementarnimi pravili in enostavnimi subjekti pogojev, predstavljene z objekti in s tradicionalnim programom, za število subjektov pogojev $cnum=3$ in $cnum=4$. Ker nas zanima primerjava časov, na grafikonu niso prikazane absolutne vrednosti, temveč le indeksi. Po pričakovanjih so bili časi izvajanja logične odločitvene tabele, predstavljene z objekti, daljši.



Slika 4: Primerjava hitrosti izvajanja logične odločitvene tabele, predstavljene z objekti, in enakovrednega "tradicionalnega" programa

Velik delež presežnega časa izvajanja gre na račun modularnosti in sestavljivosti logične odločitvene tabele, če je predstavljena z objekti v objektni bazi podatkov. Kljub temu lahko za pospešitev izvajanja uporabimo nekatere tehnike, ki so že znane za tradicionalne logične odločitvene tabele, npr. *relativno frekvenco pravil* [21], pri čemer naj bo frekvenca pravila eden od privatnih atributov objektov, ki predstavljajo pravila.

Eden izmed virov nepotrebnega podaljševanja časa izvajanja je večkratno vrednotenje istih komponent logične odločitvene tabele. Tako se npr. za vsako pravilo, v katerem nastopa neki subjekt pogoja (ne nastopa indiferentnost zanj), ta subjekt pogoja ovrednoti. To podvajanje lahko odpravimo tako, da se ob ovrednotenju komponente njena vrednost shrani kot vrednost privatnega atributa. Povprečni čas izvajanja logične odločitvene tabele se lahko na ta način bistveno zmanjša, predvsem v bolj statičnih okoljih, hkrati pa se poveča čas ob ažuriranju.

Posebej zanimiva se zdi možnost vzporednega izvajanja pravil logične odločitvene tabele, saj so sama objektna okolja izredno primerna za tako izvajanje [24]. Po drugi strani so tudi logične odločitvene tabele po svoji naravi vzporedne, saj je vrstni red pravil nepomemben². Za večino drugih predstavitev odločitvenih pravil (logični diagram poteka, odločitvena drevesa) to ne velja.

2.3.2 Analiza možne uporabe rešitve

V nadaljevanju smo ugotavljali smiselnost praktične uporabe rešitve in težave, na katere bi utegnili naleteti pri njenem uvajanju v realno okolje. V ta namen smo analizirali možnost uporabe predlagane rešitve na realnem problemu izdelave informacijskega sistema referata za študijske zadeve Ekonomske fakultete v Ljubljani. Pri tem smo se naslonili na načrt [9], ki ga je že prej izdelala skupina študentov pod mentorstvom avtorja metode TAD, ki je bila pri izdelavi načrta uporabljena.

V prvi fazi smo skušali identificirati podprobleme znotraj celotnega načrta, ki bi bili zanimivi za realizacijo z odločitvenimi tabelami, predstavljenimi z objekti v objektni bazi podatkov. Pri identifikaciji podproblemov smo izhajali iz dejstva, da je uporaba predlagane rešitve smiselna predvsem pri kompleksnih odločitvenih problemih, to je takih, pri katerih je število subjektov pogojev in aktivnosti razmeroma veliko. Analiza načrta je pokazala, da sta primerni dve (približno enako kompleksni) nalogi:

- vpis študenta, kjer mora referent preveriti vse pogoje za vpis v višji letnik, ter

² Razen pri samem (zaporednem) izvajanju, ko je vrstni red pomemben pri minimizaciji časa, potrebnega za odgovor.

- *prijava na izpit*, kjer mora referent preveriti vse pogoje za pristop k izpitu.

Za nadaljnjo obdelavo smo izbrali drugi problem.

Natančnejšo analizo naloge *prijava na izpit* smo opravili pri referentkah v referatu za dodiplomski študij Ekonomske fakultete. Rezultat te analize je logična odločitvena tabela z enajstimi subjekti pogojev, štirinajstimi aktivnostmi in dvanajstimi (ne-elementarnimi) pravili, ki je prikazana na sliki 5.

Nato smo identificirali objekte, ki so povezani s tem odločitvenim problemom, in se lotili ugotavljanja problemov, na katere bi utegnili naleteti pri kodiranju obravnavanega odločitvenega problema.

Implementacija omenjene logične odločitvene tabele zahteva razmislek, saj smo hitro ugotovili, da oblika le enega subjekta pogoja ustreza klasični obliki subjekta pogoja, kjer z relacijskimi operatorji primerjamo dve numerični vrednosti (aritmetična izraza). Večina drugih subjektov pogojev zahteva preverjanje, ali v nekem razredu objektov obstaja objekt z določenimi vrednostmi atributov. Npr. subjekt pogoja "Ali je samo ena prijava vnaprej?" ustreza vprašanju "Ali v razredu objektov PRIJAVA obstaja objekt, ki

je povezan s študentom (objektom razreda ŠTUDENT), ki se želi prijaviti, in z izpitnim rokom (objektom razreda ROK), za katerega se želi prijaviti?"

Realizacija takega subjekta pogoja je mogoča na dva načina:

- V definiciji razreda objektov POGOJ lahko dodamo definicijo operatorja *je v* in implementiramo ustrezno metodo, ki ovrednoti ta pogoj.
- V sistemih, kjer tak operator ni definiran, lahko preverjanje realiziramo tako, da subjekte pogojev nadomestimo s klici logičnih odločitvenih tabel. Preverjanje prisotnosti objekta v razredu torej realiziramo z logično tabelo. V ta namen mora obstajati mehanizem posredovanja parametrov ene logične odločitvene tabele drugi.

Oglejmo si še aktivnosti. Večina aktivnosti je enostavnih (izpis ali zapis podatkov), le aktivnost *Sprejem prijave* je nekoliko zahtevnejša, saj skriva: tvorbo objekta razreda PRIJAVA, povezavo tega objekta z ustreznima objektoma razredov ŠTUDENT in ROK ter izpis ustreznega sporočila (npr. "Prijava sprejeta").

Ugotovitve analize možne uporabe rešitve v okviru obravnavanega problema lahko strnemo v nekaj točk.

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂
Datum prijave < datum roka – čas za prijavo ?	ne										da	da
Ali že obstaja prijava?		da									ne	ne
Ali je že opravil izpit?			da								ne	ne
Ali obstaja rok?				da							ne	ne
Ali ima študent predmet v predmetniku?					da						ne	ne
Ali so izpolnjeni predpogoji?						da					ne	ne
Ali je samo ena prijava vnaprej?							da				ne	ne
Ali je vpisana ocena za zadnjo prijavo?								da			ne	ne
Ali je št. prijav v študijskem letu preseženo?									da		ne	ne
Ali je skupno št. prijav preseženo?										da	ne	ne
Ali plača izpit?											da	ne
Izpis "Prepozna prijava"	X											
Izpis "Prijava že obstaja"		X										
Izpis "Izpit že opravljen"			X									
Izpis "Rok ne obstaja"				X								
Izpis "Predmeta ni v predmetniku"					X							
Izpis "Niso opravljeni predpogoji"						X						
Izpis "Prijava za ta rok že obstaja"							X					
Izpis "Ocena za prejšnji rok še ni vpisana"								X				
Izpis "Št. prijav v tem študijskem letu preseženo"									X			
Izpis "Skupno št. prijav preseženo"										X		
Izpis "Izpit je potrebno plačati"											X	
Izpis potrdila o plačilu												X
Zabeleži podatke o neuspeli prijavi	X	X	X	X	X	X	X	X	X	X		
Sprejem prijave											X	X

Slika 5: Logična odločitvena tabela naloge prijava na izpit

- Pri implementaciji rešitve je potrebno nadaljnje razvojno delo na nekaterih delih rešitve. Tako je npr. potrebno implementirati možnost povezovanja logičnih odločitvenih tabel, različne operatorje, ki naj bi jih bilo mogoče uporabljati v subjektih pogojev, ter možnost uporabe klicev metod objektov v subjektih pogojev.
- Izkazalo se je, da bi bila rešitev za odločitveni problem preverjanja pogojev za prijavo na izpit izredno primerna, saj omogoča enostavno spreminjanje pogojev ob morebitnih (relativno pogostih) spremembah le-teh. To pomeni enostavnejše in s tem tudi cenejše vzdrževanje programske opreme.
- Hkrati bi bilo mogoče na enostaven način dodajati in spreminjati pogoje za pristop k izpitu, ki so specifični za posamezni predmet.

3 PREDNOSTI IN SLABOSTI PREDLAGANE REŠITVE

Predlagana rešitev ima nekaj pomembnih prednosti pri modeliranju, kodiranju in uporabi poslovnih pravil v informacijskih sistemih, med katerimi so najpomembnejše naslednje:

- Način dela s poslovnimi pravili je enak v vseh fazah razvoja informacijskega sistema, saj zunanji pogled, ki ga uporabljamo tudi pri kodiranju, ažuriranju in uporabi pravil, ustreza eni izmed tehnik za vizualizacijo poslovnih pravil.
- Na ta način predstavljena poslovna pravila zagotavljajo večjo fleksibilnost in robustnost sistema, saj so spremembe poslovnih pravil enostavne in ne vplivajo na ostale dele sistema.
- Ker so metode definirane pri razredih objektov, ki so predefinirani, ni potrebno programa ob tvorjenju ali spreminjanju logične odločitvene tabele prevajati. S tem je zagotovljena večja fleksibilnost poslovnih pravil.
- Predstavitev poslovnih pravil z objekti, ki predstavljajo komponente logičnih odločitvenih tabel poveča ponovno uporabljivost programske kode na dva načina:
 - Komponente lahko poljubno združujemo v pravila in tabele s tvorbo povezav med objekti. Velja pa seveda tudi obratno, saj lahko povezave tudi odstranimo.
 - Ker je med objekti mogoče tvoriti povezave tipa M:N, to pomeni, da lahko komponente na nižjem nivoju sodelujejo v več komponentah na višjem nivoju. Na primer, en subjekt aktivnosti lahko nastopa v več pravilih, eno pravilo pa v več logičnih odločitvenih tabelah. Na ta način je povečana ponovna uporabljivost znotraj uporabniške rešitve.
- Z razvojem ustreznega zunanjega pogleda je

mogoče grafično programiranje poslovnih pravil. Pri tem gre za neposredno delo s poslovnimi pravili in ne za generiranje programske kode.

- Zelene lastnosti poslovnih pravil se preverjajo avtomatično. Preverjanje mora biti opsijsko, saj ni vedno zaželeno, da poteka sprotno.

Poleg prednosti pa ima trinivojski pogled na poslovna pravila in notranja predstavitev le-teh z objekti tudi nekatere slabosti:

- Zunanji pogledi morajo biti predvideni vnaprej, kar pomeni, da si uporabnik ne more tvoriti nekega svojega zunanjega pogleda. Seveda mora sistem za upravljanje logičnih odločitvenih tabel omogočati, da si uporabniki čimbolj prilagodijo vgrajene zunanje poglede svojim potrebam.
- Prav tako je potrebno vnaprej definirati operatorje (npr. za vektorske operande ali za preverjanje obstoja objekta z določeno lastnostjo v množici objektov), ki jih smemo uporabljati v subjektih pogojev.
- Časovna zahtevnost izvajanja tako predstavljenih poslovnih pravil se poveča.

4 SKLEP

V prispevku smo predstavili način predstavitve poslovnih pravil, ki temelji na trinivojskem pogledu na poslovna pravila, pri čemer so pravila na notranjem nivoju predstavljena z objekti v bazi podatkov, kar omogoča predvsem večjo fleksibilnost poslovnih pravil in enostavnejše delo tudi za uporabnika.

Z logičnimi odločitvenimi tabelami in posledično na predlagani način je mogoče predstaviti vsak logični diagram poteka. Vendar predlagani model ne rešuje vseh problemov predstavitev poslovnih pravil v informacijskih sistemih. Iz analize predlagane rešitve, prototipa in primera uporabe lahko sklepamo, da je uporaba smiselna predvsem v primerih, ko so prednosti bolj izrazite, to pa je predvsem v primeru kompleksnih poslovnih pravil, za katera sta značilna pogosto spreminjanje in potreba po preverjanju popolnosti in nedvoumnosti. Pri odločitvi za uporabo predlaganega načina predstavitve poslovnih pravil so torej odločilni faktorji predvsem:

- pogostost sprememb pravil,
- odzivni čas vzdrževalcev sistema,
- potrebe po avtomatičnem preverjanju lastnosti pravil,
- kompleksnost problema,
- možnosti za ponovno uporabo komponent poslovnih pravil ter
- pomembnost hitrosti izvajanja.

Za uporabo v realnih okoljih je potrebno nadaljnje raziskovalno in razvojno delo predvsem na naslednjih področjih:

- razvoj zunanjih pogledov,
- povezava z univerzalnimi (objektno relacijskimi) bazami podatkov,
- optimizacija izvajanja pravil,
- možnost dela z večdimenzionalnimi tabelami,
- razvoj za porazdeljena okolja ...

Sklenemo lahko, da bi bilo v mnogih kompleksnih odločitvenih problemih, kjer se odločitvena pravila pogosto spreminjajo, smiselno in smotrno uporabiti predlagano rešitev, kar bi omogočilo večjo fleksibilnost programskih rešitev in hkrati pocenilo vzdrževanje le-teh.

5 LITERATURA IN VIRI

- [1] BANCILHON, François, DELOBEL, Claude, KANELAKIS, Paris: Building an Object-Oriented Database System: The Story of O2, Morgan Kaufmann Publishers, San Mateo, California, 1992, ISBN 1-55860-169-4
- [2] BAUM, David: The Right TOOLS for Coding BUSINESS Rules, Datamation, 1995 št. 4 (str. 36-38)
- [3] CRAGUN, Brian J., STEUDEL, Harold J.: A decision-table-based processor for checking completeness and consistency in rule-based expert systems, International Journal of Man-Machine Studies, 1987 št. 5 (str. 633-648), ISSN 0020-7373.
- [4] DAMIJ Talib: Načrt informacijskega sistema za referat Ekonomske fakultete za dodiplomski študij izdelan po metodi TAD (Tabular Application Development). Ljubljana : Ekonomska fakulteta, 1997. 11 str.
- [5] ELMASRI, Ramez, NAVATHE, Shamkant B.: Fundamentals of Database Systems, Addison-Wesley, Redwood City, 1989, ISBN 0-201-5309-2
- [6] GRADIŠAR Miro, RESINOVIČ Gortan: Informatika v poslovnem okolju. Ljubljana: Ekonomska fakulteta v Ljubljani, 1996. 479 str.
- [7] HALVERSON Richard Jr.: Programming the Hawaii Parallel Computer. Proceedings of the 2nd ACM International Workshop on FPGAs, Berkley, CA, 1994. 7 str.
- [8] HICKS, Sam A., WASSERMAN William E.: Computers in Taxation, The Tax Adviser, 1991 št. 7 (str. 471-473)
- [9] HOFFER J. A., GEORGE J. F., VALACHIC J. S.: *Modern System Analysis and Design*. Reading : The Benjamin/Cummings Publishing Company, 1996.
- [10] JENKINS A. Milton: Evolution of Information Technology and the Problems and Opportunities it Presents to Business. 50th Anniversary of the Faculty of Economics, Ljubljana, Slovenia, International Conference, Summary of Abstracts, Ljubljana : Faculty of Economics, 1996, str. 572-575.
- [11] LEW, Art: On the Emulation of Flowcharts by Decision Tables, Communications of the ACM, 1982 št. 12 (str. 895-905).
- [12] LONDON, Keith R.: Decision Tables, Auerbach Publishers Inc., 1972, ISBN 0-87769-122-3.
- [13] MAES, R., Van DIJK J. E. M.: On the Role of Ambiguity and Incompleteness in the Design of Decision Tables and Rule-Based Systems, The Computer Journal, 1988 št. 6 (str. 481-489)
- [14] MAES, R.: On the representation of program structures by decision tables: a critical assessment, The Computer Journal, 1978 št. 4 (str. 290-295), ISSN 0010-4620.
- [15] MARTIN, James.: Principles of Object-Oriented Analysis and Design, Prentice-Hall, Englewood Cliffs, 1993, ISBN 0-13-720871-5.
- [16] McDANIEL, Herman, ed.: Applications of Decision Tables, Brandon/System Press, Inc., Princeton, 1970
- [17] TSICHRITZIS Dionysios C., LOCHOVSKY Frederick H.: Data Models. Englewood Cliffs : Prentice-Hall, Inc., 1982. 343 str.
- [18] VAN THIENEN Jan, DRIES E.: Illustration of a Decision Table Tool for Specifying and Implementing Knowledge Based System. International Journal of Artificial Intelligence Tools, 3 (1994), 2, str. 267-288.
- [19] VAN THIENEN Jan, WETS Geert: Integration of the Decision Table Formalism With a Relational Database Environment. Information Systems, 20 (1995), 7, str. 595-616.
- [20] VESSEY Iris, WEBER Ron: Structured Tools and Conditional Logic: An Empirical Investigation. Communications of the ACM, 29 (1986), 1, str. 48-57.
- [21] WEARS, Robert L. et al.: Using Decision Tables to Verify the Logical Consistency and Completeness of Clinical Guidelines: Fevers Without Sources in Children Under Age Three Years, <http://solaris.ckm.ucsf.edu/Originals/SAEMabs/z.abs103.html>, University of Florida, 1996

Jurij Jaklič je diplomiral iz uporabne matematike na Oddelku za matematiko in mehaniko Fakultete za naravoslovje in tehnologijo v Ljubljani. Magistriral je na Oddelku za računalništvo Univerze v Houstonu, ZDA. Doktorsko disertacijo je obranil leta 1997 na Ekonomski fakulteti v Ljubljani, kjer je tudi zaposlen kot asistent pri predmetih informatike. Ukvarja se predvsem s problemi s področja baz podatkov.

Janez Grad je doktoriral iz matematičnih znanosti na Vseučilišču v Zagrebu. Od l. 1973 sodeluje kot učitelj za informatiko na Ekonomski fakulteti v Ljubljani, od leta 1985 dalje kot redni profesor. Ukvarjal se je s programiranjem na računalniku in z reševanjem problema lastnih vrednosti in vektorjev matrik, v zadnjih letih pa se ukvarja z reševanjem problemov s področja operacijskega raziskovanja in baz podatkov.

VZORCI IN PONOVNA UPORABA IZKUŠENJ

Tomaž Domajnko, Ivan Rozman, Marjan Heričko, József Györkös
Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

Povzetek

V prispevku so podani osnovni koncepti vzorcev. Vzorci opisujejo uspešne rešitve težav programskega inženirstva. Predstavljena je vloga uporabe vzorcev v razvojnem procesu. Vzorci ne pomenijo le novega pristopa k razvoju, temveč tudi izboljšujejo komunikacijo v razvojni skupini. Končni cilj uporabe vzorcev je razvoj priložnikov za programsko inženirstvo, ki bodo vsebovali preizkušene rešitve problemov v določenem kontekstu.

Abstract

In the article, basic concepts of software patterns are introduced. A software pattern describes a successful solution to a set of similar software development problems. Patterns do not only describe useful techniques, they also improve communication between members of development teams. The final goal of patterns usage is to provide handbooks for software engineering, containing tested solutions of problems in certain context.



PONOVNA UPORABA

Objektna tehnologija je vodilna tehnologija razvoja informacijskih sistemov. Uveljavitev objektne tehnologije je v razvoj programskih sistemov prinesla nove pristope, tehnike in zahtevo po novih znanjih. Ena od ključnih prednosti objektne tehnologije je, da gradi kompleksne sisteme iz razmeroma preprostih gradnikov – objektov. Pri tem se poraja ideja, da je preproste gradnike možno uporabiti na več načinov in s tem doseči boljšo produktivnost. Govorimo o ponovni uporabi. V okviru objektne tehnologije obstaja več pristopov k ponovni uporabi, na katero gledamo kot na sposobnost razvoja novih sistemov na podlagi že obstoječih komponent. Pri tem se moramo zavedati, da se ponovna uporaba ne bo zgodila sama od sebe. Potrebno je skrbno načrtovati aktivnosti, ki bodo privedle do ponovne uporabe in s tem krajših razvojnih časov, nižjih stroškov razvoja in vzdrževanja ter povečane kakovosti programski proizvodov [9].

Uvajanje aktivnosti v razvojni proces pomeni spremembe. Spremembe lahko grobo razdelimo na dve skupini. Prvo skupino tvorijo spremembe, ki prilagajajo razvojni proces za ponovno uporabo, medtem ko drugo skupino tvorijo spremembe, namenjene dvigu stopnje ponovne uporabe programskih proizvodov. Ponovno lahko uporabimo vse programske proizvode, pod pogojem da so prilagojeni ponovni uporabi. To pomeni, da so

primerno dokumentirani, dostopni in uporabni v različnih kontekstih.

V času razvoja nastajajo programski proizvodi, ki vplivajo na način ponovne uporabe. Tako lahko ponovno uporabo programskih proizvodov razdelimo na:

- *ponovno uporabo delov programske kode*, kar pomeni ponovno uporabo na najnižjem nivoju abstrakcije, saj ponovno uporabljamo konkretno implementacijo dela problema,
- *ponovno uporabo razredov*, pri kateri ponovno uporabljamo implementacijo abstrakcije določenega dela realnega sveta,
- *ponovno uporabo idej, ki vodijo do rešitve*, to je ponovno uporabo na najvišjem nivoju abstrakcije, saj ponovno uporabimo idejo, ki je v preteklosti vodila do uspešne rešitve v določenem kontekstu.

Ponovna uporaba prvih dveh nivojev abstrakcije je danes splošno poznana in uporabljana. Tretja možnost ponovne uporabe pa pomeni novost, saj do sedaj nismo govorili o ponovni uporabi idej rešitev problemov. V situacijah, ko rešitev zadanega problema ne obstaja in ne najdemo niti delne rešitve je potrebno rešitev razviti samostojno. Vemo, da izkušnje razvojne skupine pomembno vplivajo na čas razvoja, kakovost rešitve in višino stroškov razvoja. Prispevek opisuje pristop k ponovni uporabi idej preizkušenih rešitev, ki lahko v veliki meri pripomore k zmanjšanju stroškov razvoja.

VZORCI

V vsakdanjem življenju opažamo, da vedno znova uporabljamo zelo podobne pristope k reševanju problemov. Vsakomur je jasno, da izkušnje pomagajo tudi pri reševanju novih problemov, saj človek poseduje sposobnost uporabe znanja v drugem kontekstu, kot si ga je pridobil (*prednost človeškega razmišljanja pred računsko močjo računalnika*).

Nezadostno število strokovnjakov z izkušnjami v informacijskih tehnologiji zahteva nov pristop k reševanju problema izkušenj. V prejšnjem poglavju smo videli, da je ena od možnosti ponovna uporaba. Pa ne samo ponovna uporaba implementacij konkretnih razredov in funkcij. Ponovno lahko uporabimo način, kako objekti med seboj komunicirajo, kako objekte poimenujemo, povezujemo v hierarhije ali združujemo v komponente. Ta način, kako ponovno uporabljati ideje rešitev, zagotavlja izkoriščanje nakopičenih izkušenj, zapisanih v obliki vzorcev. Vzorec je torej najsplošnejša možnost ponovne uporabe pridobljenega znanja.

Definicija vzorca

Področje vzorcev je danes v velikem razcvetu, kar posledično pomeni, da standardizirane definicije vzorca še ni. Vseeno sta v svetu in pri nas najbolj uveljavljeni naslednji definiciji:

- Vzorec je ideja rešitve, ki se je izkazala uporabna v določenem kontekstu in bi bila najverjetneje koristna tudi v drugih kontekstih. Vzorec predstavlja večkratne odločitve strokovnjaka, ki sicer vodijo do različnih rešitev, a vsebujejo določen nivo kakovosti [4].
- Vzorec opisuje problem, ki se večkrat pojavlja v našem okolju, podaja jedro njegove rešitve na tak način, da lahko idejo rešitve uporabimo v milijon različnih primerih, ne da bi pot od ideje do rešitve prehodili na enak način (Christopher Alexander, [3]).

Dokumentiranje vzorcev

Vzorci pokrivajo tako veliko definicijsko območje, da je njihova struktura zelo individualna. To je tudi razlog, da v svetu obstaja veliko število predlogov predstavitev vzorcev. Pri tem se pojavljajo tako popolnoma grafične predstavitve, kot tudi predstavitve v obliki spletnih strani.

V splošnem mora definicija vzorca vključevati naslednje elemente (bolj kot na obliko se osredotočimo na vsebino predstavitve vzorca):

- *Ime* - omogoča sklicevanje na vzorec z eno samo besedo ali kratko besedno zvezo.
- *Namen* - razloži problem, ki ga lahko s pomočjo vzorca rešimo - kdaj uporabimo vzorec, kakšne so njegove naloge in cilji, kaj z njim dosežemo oz. kakšen je njegov vpliv v določenem kontekstu.

- *Uporabnost* - podaja stanje oziroma predpogoje, kjer nastopi takšen problem, da je rešitev, ki jo vzorec nudi, ustrezna.
- *Strukturo, sodelovanje, udeležence* - podajajo način realizacije rešitve s pomočjo statičnih povezav in dinamičnih pravil. Opis lahko zajema slike, diagrame in besedilo, ki prikažejo strukturo vzorca, njegovo vlogo in delovanje, da opišemo postopek reševanja problema. Poleg statične strukture rešitev podaja tudi dinamično obnašanje.
- *Učinek* - podaja stanje ali konfiguracijo sistema po uporabi vzorca, vključno s posledicami, ki jih prinese uporaba vzorca, ter drugimi problemi in vzorci, ki se pojavijo v novem kontekstu. Dokumentacija stanja po uporabi vzorca pomaga pri primerjavi z začetnim stanjem drugega vzorca.
- *Implementacija* - podaja probleme, s katerimi se lahko srečamo pri sami implementaciji ideje rešitve.
- *Razlago korakov, pravil* - utemelji korake oziroma pravila uporabe vzorcev.
- *Sorodne vzorce* - opiše statične in dinamične povezave med vzorci. Medsebojno povezani vzorci si delijo različne učinke.
- *Primere uporabe* - beleži opis že znane uporabe vzorcev v obstoječih sistemih. Pomaga pri ugotavljanju primernosti vzorca za reševanje problema.

Poglejmo si primer vzorca za faze načrtovanja, ki pomaga pri ustvarjanju delne hierarhije razredov v našem sistemu. Uporaben je takrat, ko ima uporabnik (objekt) opravka z velikim številom različnih objektov. Vzorec s svojim opisom podaja idejo, da objekte združimo v hierarhijo in odpravimo razločevanje med skupkom objektov (kompozicijo) in atomarnim objektom (poimenovan tudi list). Tipično uporabo vzorca prikazuje slika 1.

Ena od pomembnih aktivnosti uporabe vzorcev je njihova klasifikacija. Klasifikacija pomaga tako uporabnikom kot tudi razvijalcem vzorcev pri iskanju vzorcev. Obstaja več različnih klasifikacij vzorcev. Kot osnovno klasifikacijo vzemimo razvojno fazo življenjskega cikla, v kateri je vzorec uporabljen. V tem primeru govorimo o vzorcih v analizi, načrtovanju, implementaciji, testiranju. Ta delitev se zdi naravna, a prihaja do nejasnosti pri vzorcih, ki so uporabni v več kot eni fazi razvojnega cikla. Pri tem pa ne gre za razliko v ideji vzorca, ampak za različno implementacijo vzorca v različnih fazah razvoja.

Vzorci lahko delimo tudi po namenu. Tako delitev vzorcev zasledimo tudi v [4, 7], kjer avtorja vzorce delita na strukturne (združevanje razredov in objektov v večje strukture z namenom modeliranja problemskega področja), vedenjske (vedenjski vzorci se ukvarjajo z algoritmi in dodelitvami odgovornosti med

<i>Ime</i>	KOMPOZICIJA
<i>Namen</i>	Združi objekte v drevesno strukturo za predstavitev delne hierarhije. Kompozicija dovoljuje uporabniku enako obravnavo posameznih in sestavljenih objektov.
<i>Uporabnost</i>	Vzorec uporabimo, če želimo predstaviti del hierarhije objektov, če želimo, da uporabnik ne bi razločeval med kompozicijo objektov in posameznimi objekti.
<i>Vpliv</i>	Nastane delna hierarhija in zato izvajanje dodatne kode, sicer je vpliv na izvajanje zanemarljiv.
<i>Udeleženci</i>	<i>Komponenta</i> (deklarira vmesnik za objekte v kompoziciji, implementira osnovno obnašanje vmesnika za vse razrede, deklarira vmesnik za dostop in upravljanje komponent naslednikov), <i>List</i> (predstavlja zadnje komponente v kompoziciji - listi nimajo naslednikov, definirajo obnašanje primitivnih objektov v kompoziciji), <i>Kompozicija</i> (definira obnašanje komponente, ki ima naslednike, implementira operacije, ki se nanašajo na naslednike), objekt <i>Uporabnik</i> (upravlja z objekti v kompoziciji preko vmesnika razreda <i>Komponenta</i>).
<i>Sodelovanje</i>	Uporabnik uporablja vmesnik razreda <i>Komponenta</i> za komunikacijo z objekti v kompozicijski strukturi. Če je sprejemnik <i>List</i> , potem se zahteva obravnavo takoj, če pa je sprejemnik <i>Kompozicija</i> , potem se zahteva posreduje naprej do naslednikov, po možnosti se dodatne operacije izvedejo pred ali za posredovanjem.
<i>Učinek</i>	Vzorec definira razredno strukturo primitivnih in sestavljenih objektov. Primitivni objekti se lahko združujejo v kompleksnejše

objekte. S tem poenostavi uporabnika združenih objektov, saj lahko le ta obravnava primitivne in sestavljene objekte enako. Omogoča enostavno dodajanje novih vrst komponent, saj novo definirani podrazredi (tipa *List* ali *Kompozicija*) neposredno podedujejo lastnosti svojih nadrazredov v strukturi.

Slabost vzorca je, da lahko naredi načrtovanje preveč splošno. Slabost enostavnega dodajanja novih komponent je ta, da težko omejimo število komponent celotne kompozicije.

Implementacija Določiti je potrebno zvezo s starši, saj povezave med komponento naslednikom in starši poenostavi prečkanje in upravljanje kompozicijske strukture. Pogosto je uporabno deliti komponente. Ena od nalog vzorca je omogočiti, da se uporabnik ne zaveda ali uporablja razrede tipa *List* ali *Kompozicija*. Za doseg te naloge mora razred *Komponenta* definirati čim več osnovnih operacij za razrede *List* in *Kompozicija*. Razred *Komponenta* ponavadi priskrbi osnovno implementacijo operacij, ki jih podrazredi predefiniirajo.

Razlaga korakov Poiščemo skupino objektov, katere največkrat uporabljamo skupaj in uporabimo vzorec. Pri tem nastane delna hierarhija, ki rešuje težave obvladovanja velikega števila objektov različnih tipov.

Sorodni vzorci Veriga odgovornosti, Dekorator, Zrno, Iterator, Obiskovalec.

Primeri uporabe Modeliranje prodajnih artiklov, organiziranje proizvodnje, vodenje evidence o zaposlenih.

Tabela 1: Vzorec Kompozicija



Slika 1: Tipična struktura sestavljenih objektov

objekti, pri tem pa opisujejo vzorce za komunikacijo med objekti in arhitekturne vzorce (zagotavljanje trdne in hkrati fleksibilne arhitekture celotnega sistema).

Poglejmo si predlog večnivojske delitve vzorcev. Kot primarno delitev predlagamo delitev vzorcev glede na razvojno fazo (vzorci analize, načrtovanja in implementacije). Drugi nivo delitve vzorcev je delitev glede na namen uporabe (kot osnovo uporabimo delitev [4, 7], kateri dodamo nove skupine vzorcev), v tretjem nivoju pa vzorce razdelimo glede na področje uporabnosti (razred, objekt). S tako delitvijo pridobimo preglednost in povečamo učinkovitost dostopa do vzorcev. Tabela 2 prikazuje uporabo delitve na vzorcih

faze načrtovanja, pri čemer tabela prikazuje le tri vrste vzorcev glede na namen, kar je prikazano kot vodovarna delitev. Navpično vzorce razdelimo po kriteriju področja uporabnosti.

	Ustvarjalni	Strukturni	Vedenjski
Razred	Tovarniška metoda	Prilagoditev	Interpreter
	Abstraktna tovarna Graditelj	Most Kompozicija	Iterator Posredovalec
Objekt	Prototip Edinec	Dekorator Fasada Namestnik	Opazovalec Strategija Obiskovalec

Tabela 2: Delitev vzorcev načrtovanja glede na namen in definicijsko območje

V tabeli 2 je potrebno razumevanje razlike med razredi (po definiciji predstavlja razred deklaracijo objektov, ki imajo enake lastnosti/atribute in vedenje/metode) in objekti (po definiciji predstavlja objekt primerek razreda z določenimi vrednostmi atributov, ima svojo identiteto in stanje, ki določa njegovo vedenje). Kriterij *področje uporabnosti* določa definicijsko območje uporabe vzorca (vzorec povezuje razrede ali objekte).

Vzorci, kot so Tovarniška metoda, Prilagoditev in Interpreter delujejo izključno na nivoju razreda. Tako vzorec Tovarniška metoda zagotavlja oblikovanje (kreiranje) primerkov razreda (objektov!) na podlagi definicije razreda in podanih podatkov (tako metodo praviloma imenujemo konstruktor). Vzorec Prilagoditev je razredni vzorec, ki z gradnjo vmesniških razredov (komponent) gradi strukturo sistema, medtem ko je vzorec Interpreter namenjen prikazovanju slovnice kot razredne hierarhije (prevajalnik je v tem primeru implementiran kot operacija nad primerki razredov. Vzorec uporabimo, kadar obstaja jezik za interpretacijo in lahko predstavimo njegove stavke kot abstraktna sintaksna drevesa).

Druga skupina vzorcev pa je skupina objektnih vzorcev, to je vzorcev, katerih področje uporabnosti so izključno objekti. Med temi vzorci najdemo vzorec *Namestnik*, ki podaja idejo o vpeljavi namestniškega objekta namesto izvornega (ker se nahaja v drugem delu računalniškega omrežja ali pa zahteva njegov prenos preveč računalniških sredstev). Uporabnost vzorca je danes poznana in se uporablja v porazdeljenih sistemih (CORBA, COM+), kjer aplikacija ne zahteva izvornega objekta, ampak le njegov namestniški objekt.

Poglejmo še nekaj osnovnih razlik med razrednimi in objektnimi vzorci. Razredni vzorci se ukvarjajo s povezavami med razredi in pripadajočimi podrazredi. Te povezave so pogosto povezane z dedovanjem, so statične in določene v času prevajanja. Objektni vzorci pa se ukvarjajo s povezavami objektov, ki so bolj

dinamične in se lahko spreminjajo v času izvajanja. Strukturni vzorci razredov uporabljajo dedovanje za združevanje razredov, medtem ko strukturni vzorci objektov opisujejo pristope za združevanje objektov. Razredni vzorci za obnašanje uporabljajo dedovanje za razlago algoritma in toka kontrole, medtem ko objektni opisujejo kako sodeluje skupina objektov, da dosežemo cilj, ki ga posamezen objekt ne more doseči.

RAZVOJ IN VZORCI



Slika 2: Pristop k razvoju informacijskih sistemov

Osnovno skupino aktivnost predlaganega pristopa k razvoju informacijskih sistemov tvorijo aktivnosti razvoja sistemov z uporabo vzorcev (slika 2). Vsebuje aktivnosti, ki zagotavljajo učinkovito in kvalitetno identifikacijo vzorcev in njihovo uporabo v procesu razvoja. Kot najpomembnejši aktivnosti se pojavljata identifikacija primernih vzorcev iz množice vseh dosegljivih vzorcev in aktivnost vrednotenja primernosti vzorca za uporabo v danem kontekstu. Pri tem se je potrebno zavedati dejstva, da vzorci pokrivajo celotno področje razvoja programskih proizvodov, od načina kodiranja programske kode do usmeritev in korektivnih ukrepov pri vodenju projektov programskega inženirstva. Pri tem je seveda potrebno identificirati vrste potencialnih uporabnikov vzorcev, saj s tem po eni strani ožimo izbor vzorcev za posamezne skupine, hkrati pa imamo možnost prilagajanja predstavitve vzorca ciljni skupini in stopnji izkušenj posameznikov.

Seveda prihaja v procesu razvoja do novih idejnih rešitev in s tem možnosti razvoja novih vzorcev (to so lahko nove ideje ali pa način uporabe obstoječih idej za reševanje problema v določenem kontekstu). Ti vzorci so še posebej pomembni, saj neposredno vsebujejo znanje in izkušnje poslovne organizacije, so *naravnani* na našo razvojno organizacijo in ne zahtevajo prilagajanja obliki projektne organizacije.

Drugo skupino aktivnosti tvorijo aktivnosti klasifikacije najdenih vzorcev, uvrstitve v določene razrede ter verifikacije in validacije obstoječih vzorcev. Pri validaciji preverjamo predvsem semantičen pomen vzorcev, medtem ko pri verifikaciji vzorcev preverjamo tako skladnost predstavitve kot tudi skladnost z drugimi vzorci.

Dinamika uporabe in razvoja vzorcev uvaja potrebo po aktivnostih upravljanja z vzorci. Aktivnosti te skupine skrbijo tako za pravilno uporabo vzorcev, kakor tudi za vzdrževanje konsistence predstavitve vzorcev. Vsebujejo aktivnosti nadzorovanja in sledenja uporabe vzorcev (vsaka uporaba vzorca pomeni nov vir informacij o koristnosti vzorca v novem kontekstu) in aktivnosti verifikacije vzorcev. Pri tem izvajamo tako sintaktično kot tudi semantično preverjanje predstavitve novega ali dopolnjenega vzorca.

Ob vseh prednostih uporabe vzorcev se moramo zavedati, da vsaka ideja v razvojnem procesu pomeni napredek, a postane splošno uporabna šele, ko tekom uporabe v različnih kontekstih preverimo njeno učinkovitost. Kljub delitvi posameznih aktivnosti obstaja močno prepletanje med uporabo in oblikovanjem vzorcev, saj se aktivnosti izvajajo vzporedno in so največkrat odvisne ena od druge.

Oblikovanje vzorcev

Oblikovanje vzorcev vsebuje skupek aktivnosti za detekcijo, zapis, verifikacijo in hranjenje idej, najdenih v času razvoja. Povedali smo že, da vzorce lahko najdemo kjerkoli v razvojnem procesu in pojavlja se vprašanja, kako vzorce najti. Sama ideja vzorca, vsebovanje strokovnjakovih izkušenj, daje delni odgovor na zastavljeno vprašanje. Vzorce praviloma najde strokovnjak problemskega področja, ki problemsko področje pozna in ima izkušnje z razvojem informacijskih sistemov. Vsak strokovnjak iz svojih izkušenj kuje rezultate svojega dela. In mi bi radi te izkušnje predstavili v obliki, da bi jih lahko uporabili tudi manj izkušeni razvijalci. Izkušnje kažejo, da je praviloma posameznik tisti, ki vzorce najde v razvojnem procesu, je pa zelo učinkovito periodično izvajati delavnice, kjer celotna razvojna skupina pregleda in po potrebi dopolni najdene vzorce.

Glede na definicijo vzorca se pojavlja vprašanje časa iskanja in zapisovanja novih vzorcev s knjižnico dosegljivih vzorcev. Zavedati se moramo, da mora vzorec predstavljati idejo rešitve, ki je vodila (vodi) k učinkoviti rešitvi problema. Torej je mogoče vzorce praviloma najti šele po rešitvi problema (kar pomeni po fazi implementacije in testiranja sistema za veliko večino vzorcev). Prav kreativne delavnice pa predstavljajo način, da se vzorec uvrsti v knjižnico že pred to verifikacijo.

Slika 3 shematično prikazuje kaskadni razvojni

proces in faze razvoja, v katerih lahko pričakujemo nove vzorce. Pomembno je, da poskušamo vzorce vpeljati zgodaj v razvojnem projektu in s tem olajšati razvojni proces. Na sliki so prikazane samo projektne aktivnosti razvojnega projekta, seveda pa lahko vzorce uporabimo tudi v predprojektih (konceptualizacija, študija izvedljivosti) in poprojektih aktivnostih.



Slika 3: Oblikovanje vzorcev

V fazi analize spoznavamo problemsko področje. V tej fazi najdemo vzorce, ki ponujajo ideje o modeliranju problemskega področja. Tipični vzorci so strukturni vzorci (primer organizacija – stranka, uslužbenec – transakcija, načrt – korak načrta...). Prednosti uporabe vzorcev v fazi analize je pokazal tudi pilotski projekt [2], kjer smo primerjali klasičen objekten pristop s pristopom z uporabo vzorcev. Rezultati projekta so bili popolnoma v skladu s pričakovanji in so prikazali prednosti uporabe vzorcev v fazi analize problemskega področja. Primerjava modelov problemskega področja je namreč pokazala boljšo strukturo in večjo stopnjo prilagodljivosti, pri tem pa je bilo porabljenih za oblikovanje modela s pomočjo uporabe vzorcev manj časa in sredstev.

Faza systemskega načrtovanja poskrbi za gradnjo močne, a fleksibilne arhitekture celotnega sistema. V kontekstu komponentne tehnologije se osredotočamo na večnivojske arhitekture poslovnih objektov, na povezovanje posameznih objektov v celote (pod-sisteme) in izbiramo način upravljanja s trajnimi podatki. Na tem področju najdemo celo vrsto vzorcev, ki zagotavljajo dobro arhitekturo sistema (strukturni in

arhitekturni vzorci), vzorce z idejami združevanja objektov v podsisteme in oblikovanja več nivojev poslovnih objektov ter vzorce z idejnimi rešitvami upravljanja s trajnimi podatki. Verjetno najbolj znan vzorec je vzorec Podatek-Pogled-Nadzor (*Model-View-Controller*), ki predstavlja osnovno upravljanje z večimi pogledi na enake podatke, pri tem pa zagotavlja ažurnost podatkov (vzorec najdemo v programski opremi, ki omogoča hkratno delo z večimi dokumenti in različnimi pogledi nad dokumenti, v sistemih za upravljanje s podatki...).

V fazi načrtovanja objektov na podlagi modela problemskega področja in skladno z definiranimi arhitekturnimi omejitvami oblikujemo končni model informacijskega sistema. Skupine objektov oblikujemo v enovite celote, ki omogočajo določeno funkcionalnost ter razrešimo nekatere težave (neskladnost vmesnikov objektov, način implementacije posameznih operacij, zagotavljanje konsistentnosti objektov).

Faza implementacije je faza, kjer je potrebno rezultate prejšnjih faz pretvoriti v programsko kodo. Če smo vzorce uporabili v prejšnjih fazah razvoja, lahko sedaj na podlagi primerov uporabe in opisov vzorcev določimo specifičnost posameznih programskih jezikov. Pri tem gre za enako implementacijo, ki pa je odvisna od specifičnosti programskega jezika.

Zadnja faza razvoja je testiranje. V tej fazi pride do izraza uporaba vzorcev v prejšnjih fazah, natančen opis pristopov olajša testiranje in podaja pristope k testiranju. Vzorce najdemo tako pri funkcionalnem testiranju (pri zapisu vzorca, ki smo ga uporabili je tudi opis načina testiranja implementacije tega vzorca, mogoče je pristop k testiranju opisati kot posamezen vzorec) kot tudi pri integracijskem testiranju (strukturni vzorci omogočajo hitro gradnjo modelov, vsebujejo pa tudi pristope za verifikacijo).

Zavedati se moramo, da prikazane faze predstavljajo le institucionalizacijo razvojnega procesa, brez aktivnosti upravljanja projekta programskega inženirstva. Ne prikazujejo faz konceptualizacije, študije izvedljivosti projekta in vseh poprojektov aktivnosti. A tudi v teh fazah se pojavlja množica vzorcev, množica pristopov k reševanju problemov, ki pa imajo namesto razvoja informacijskega sistema namen učinkovito implementirati razvojni proces in njegove aktivnosti.

Upravljanje vzorcev

Upravljanje vzorcev je skupek aktivnosti, ki zagotavljajo stalno kakovost vzorcev in omogočajo njihovo učinkovito izkoriščanje. K tem aktivnostim štejemo aktivnosti hranjenja vzorcev v repozitoriju, ki je skupen celotni razvojni organizaciji (projektne organizaciji). Prva faza uporabljanja vzorcev je njihova

identifikacija v repozitoriju. To pomeni, da mora upravljanje vzorcev zagotavljati učinkovito iskanje vzorcev v repozitoriju. Identificirane vzorce moramo ovrednotiti, da izberemo res najprimernejši vzorec, ki ga nato uporabimo. To zahteva, da so merila vrednotenja shranjena skupaj z vzorci, da so vzdrževana in po potrebi obnovljena. Tekom razvoja uporabljamo vzorce in vsako uporabo vzorcev moramo zabeležiti tudi v repozitoriju, saj pomeni verifikacijo primernosti vzorca in daje dodatne informacije o vzorcu (odkriti problemi, novi pristopi...). Vzorec, katerega najdemo v času razvoja, je potrebo opisati, klasificirati in shraniti v repozitorij. Pri tem mora upravljanje vzorcev vsebovati verifikacijske postopke, ki po potrebi zagotavljajo pravilnost opisa vzorca. Komerčnih orodij za upravljanje vzorcev na trgu še ni, zato bodo v nadaljevanju opisane osnovne funkcionalnosti in ideje takega orodja. Funkcionalnost orodja za podporo uporabe vzorcev lahko razdelimo na vsaj tri velika področja. Prvo področje je dokumentiranje vzorcev. Dokumentiranje mora uporabniku omogočati opisovanje vzorcev, pri tem pa mora vršiti preverjanje opisa. Takega preverjanja ne bi bilo težko implementirati, če ne bi bila struktura vzorcev precej individualna in zahtevala različne predstavitve.

Različne predstavitve postavljajo tudi visoke zahteve za omogočanje uporabe vzorcev. Identifikacija in vrednotenje vzorcev v različnih predstavitev je praviloma zapletena in časovno potratna. A objektna tehnologija ponuja s svojimi pristopi k modeliranju problemskih področij možnosti tudi na tem področju.

Nadzor uporabe vzorcev predstavlja funkcionalnost, ki je po svoji naravi močno povezana z upravljanjem konfiguracije. Modul je namenjen predvsem sledenju uporabe vzorcev, dopolnjevanju predstavitve vzorcev s primeri uporabe in s tem dodatnim pojasnilom in dopušča uporabljanje funkcionalnosti obstoječih orodij za upravljanje konfiguracije.

Seveda pa je razvoj kompleksnih informacijskih sistemov v veliki meri pogojen z uporabo orodja, ki spremlja celoten življenjski cikel. Gre za orodja CASE (*Computer Aided Software Engineering*), ki zagotavljajo sledljivost razvoja in višjo produktivnost razvojne skupine. Torej bi bilo ekonomsko upravičeno nadgraditi taka orodja tudi s podporo vzorcem. Pomemben prispevek uporabi vzorcev je tudi vključitev vzorca kot osnovnega koncepta razrednega diagrama metodologije UML (*Unified Modeling Language*).

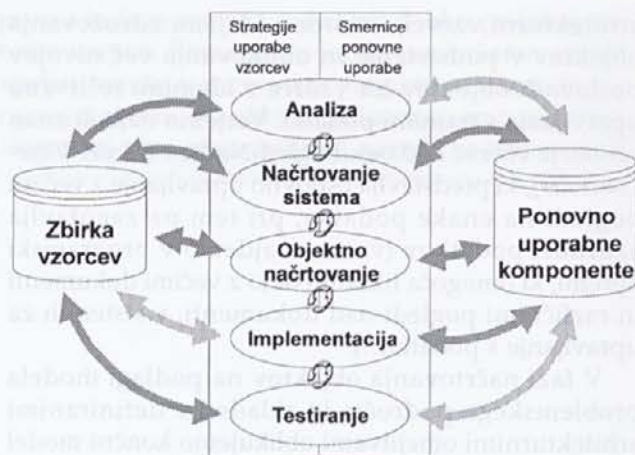
Raziskave kažejo, da zaposleni s srednjo in dolgo delovno dobo v podjetju pogosto migrirajo in s seboj odnašajo mnogo izkušenj. Tako podjetje izgublja informacije o idejah rešitev, ki so v preteklosti vodile

do uspehov, pa tudi neuspehov. Te izkušnje se pridobivajo skozi daljše časovno obdobje, hkrati pa to pomeni, da mladi člani z razmeroma malo izkušnjami le stežka v krajšem roku nadomestijo ljudi, ki so odšli. V takih situacijah se vzorci izkažejo kot odlično sredstvo hranjenja informacij, prenosa znanja in posredovanja izkušenj med zaposlenimi [7, 8], pod pogojem, da so dokumentirani na dovolj formalen način in dostopni vsej razvojni skupini. Izkušnje kažejo, da uporaba vzorcev v razvojnem procesu močno olajša komunikacijo v razvojni skupini, saj poenoti besednjak in s tem zmanjšuje dvoumnost naravnega jezika.

Uporaba vzorcev

Kot prikazuje slika 4, so temelj za uporabo vzorcev tako smernice ponovne uporabe kot tudi strategije, ki zagotavljajo potrebno znanje. Strategije pri uporabi vzorcev praviloma delimo na funkcionalne strategije (določevanje zahtevanih funkcionalnosti sistema), strategije organizacije sistema (delitev kompleksnega sistema na podsisteme), strategije identifikacije objektov in modeliranja dinamike (zagotavljajo poznavanje problemskega področja in omogočajo uporabo strukturnih arhitekturnih in ustvarjalnih vzorcev) [1]. Osnovni namen strategij je kakovostno zajemanje znanja o problemskem področju, z namenom lažjega identificiranja vzorcev. Prav identifikacija vzorcev je najpomembnejši del razvoja z uporabo vzorcev. Poglejmo si nekaj korakov za iskanje primerne vzorca, ki zagotavljajo učinkovito identifikacijo in izbor vzorcev določenega problemskega področja:

- Premislimo, kako vzorec rešuje problem na podlagi spoznanja, da vzorci pomagajo identificirati in specificirati vmesnik ter vlogo objektov.
- Pregledamo povzetke vzorcev, da ugotovimo kateri vzorci vsaj malo ustrezajo našemu problemu. Nato se posvetimo tem vzorcem še natančneje.



Slika 4: Razvoj z uporabo vzorcev

- Preučimo relacije med vzorci, saj opisi vzorcev ali slike povezovanja vzorcev grafično prikazujejo relacije med vzorci. Preučevanje odvisnosti pomaga pri izbiri ustreznega vzorca ali skupine ustreznih vzorcev.
- Preučimo, kaj naj bo spremenljivo v našem razvoju, s tem da razmislimo, kaj naj bi bilo na voljo za spremembo, ne da bi bilo potrebno spreminjati celoten razvoj. Osredotočimo se na ograevanje spreminjajočih konceptov.

Tabela 3 prikazuje koncepte nekaterih vzorcev načrtovanja, ki jih lahko spreminjamo, ne da bi bilo potrebno ponovno izvesti načrtovanje. To omogoča večjo fleksibilnost razvojnega procesa in dodatno zmanjšuje rizik razvoja.

Ko določimo vzorec, ki je primeren za reševanje zadanega problema, ga moramo pravilno uporabiti. Sledijo navodila kako učinkovito uporabiti vzorec izbrani vzorec:

- Natančno spoznamo vzorec, še posebej se osredotočimo na uporabnost in vplive vzorca, da se prepričamo, ali vzorec ustreza našemu problemu.

Namen	Vzorec	Vidiki, ki se lahko spremenijo
Ustvarjalni	Graditelj	Kreiranje sestavljenih objektov
	Tovarniška metoda	Podrazredi objektov, ki so kreirani (oblikovani)
	Prototip	Razredi objektov, ki so kreirani (oblikovani)
	Prilagoditev	Vmesnik objekta
	Most	Implementacija objekta
	Strukturni	Kompozicija; struktura in kompozicija objektov
	Dekorator	Odgovornost objekta brez tvorbe podrazredov
Vedenjski	Namestnik	Način dostopa do objekta; njegova lokacija
	Interpreter	Slovnica in interpretacija jezika
	Iterator	Dostop ali iskanje po elementih
	Posredovalec	Kako in kateri objekti sodelujejo med seboj
	Spomin	Katere informacije so shranjene izven objekta
	Opazovalec	Število objektov, ki je odvisno od drugega objekta; način zagotavljanja ažurnosti objektov

Tabela 3: Spremenljivi koncepti v definicijah vzorcev

- *Še enkrat preučimo strukturo, udeležence in sodelovanje vzorcev.* Prepričamo se, da razumemo razrede in objekte v vzorcu, in odvisnosti med njimi.
- *Izberemo imena za udeležence v vzorcu, ki so smiselna glede na kontekst,* saj so imena udeležencev v vzorcih ponavadi preveč abstraktna, da bi jih neposredno uporabili v aplikaciji.
- *Definiramo razrede.* Deklariramo vmesnike, relacije dedovanja in definiramo primerke spremenljivk. Identificiramo obstoječe razrede v naši aplikaciji na katere vpliva vzorec, in jih po potrebi modificiramo.
- *Definiramo ustrezna imena operacij v vzorcu.* Prvi del imena operacije naj vsebuje ime vzorca, drugi del imena operacije pa naj opisuje funkcijo, ki jo operacija izvaja.
- *Implementiramo aktivnosti, ki skrbijo za zanesljivost in sodelovanje znotraj vzorca.* Sekcija implementacije v vzorcu ponuja različne namige pri implementaciji. Tudi sekcija s primeri lahko pripomore pri implementaciji.

Zaključek

Vzorce uporabljamo že vse svoje življenje, le da se tega ne zavedamo. Tudi pri razvoju programske opreme hočemo uporabo vzorcev formalizirati s tem, da vzorce konsistentno opišemo in v razvojnem procesu definiramo aktivnosti, ki so potrebne za izkoriščanje obstoječega znanja. Vzorci so tehnika, ki ob pravilni uporabi pripomore k uspešnejšemu razvoju informacijskih sistemov. V prispevku so opisani osnovni koncepti predstavitve vzorcev in vplivi uporabe vzorcev na razvojni proces.

Ugotovili smo, da uporaba vzorcev uvaja nove aktivnosti v proces razvoja, hkrati pa ponuja nove možnosti ponovne uporabe. Smer razvoja področja vzorcev danes ponuja možnosti integracije vzorcev in ogrodij, združevanje vzorcev v jezike vzorcev za določena področja (paralelno programiranje, nadzorni sistemi, porazdeljeni sistemi) in vključitev vzorcev v metodologije razvoja in podporna orodja.

Pojavlja pa se vprašanje, kje dobiti zbirke vzorcev, če nimamo dovolj velike množice lastnih vzorcev. V svetu obstaja veliko število knjižnic vzorcev, ki pa so praviloma nedostopne širšemu krogu razvijalcev iz ekonomskih razlogov in varovanja poslovnih skrivnosti. Zato bo v prihodnosti potrebno najprej spremeniti mnenje o uporabi rešitev nekoga drugega (kompleks lastnega razvoja) in usmeriti vse moči v poenotenje predstavitve vzorcev. Prav orodja za upravljanje z vzorci ponujajo razmeroma preprosto rešitev, saj s seboj nehote prinašajo poenotenje predstavitve.

Literatura

- [1] COAD Peter, Object Models - Strategies, Patterns, & Applications, Yourdon Press, 1995
- [2] DOMAJNKO Tomaž, KOREN Silvo, CELCER Borut, DRVARIČ Ivan, Uporaba objektnega pristopa na primeru izgradnje informacijske podpore poslovne funkcije v zavarovalništvu, zbornik OTS'97 Objektna tehnologija v Sloveniji, str. 120-130,
- [3] FOWLER Martin, Analysis Patterns - Reusable Object Models, Addison-Wesley, 1996
- [4] GAMMA Erich, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
- [5] JACOBSON Ivar, Object-Oriented Software Engineering - A Use Case Driven Approach, Addison-Wesley Publishing company, 1992
- [6] JACOBSON Ivar, The Object Advantage - Business Process Reengineering with Object Technology, Addison Wesley Publishing Company, 1994
- [7] PRINS Robert, Developing Business Objects - A framework Driven Approach, McGraw-Hill, 1996
- [8] Rumbaugh J. et al., Object-Oriented Modeling and Design, Prentice Hall, 1991
- [9] Vatovec - Krnac Evelin, Vloga ponovne uporabe pri razvoju programske opreme, Uporabna informatika, št. 2/1997, str. 28-35,
- [10] WILKIE George, Object-Oriented Software Engineering - The professional Developer's Guide, Addison-Wesley, 1993
- [11] <http://www.cs.wustl.edu/~schmidt/>

Tomaž Domajnko je diplomiral leta 1996 na FER Maribor, kjer nadaljuje študij po enovitem doktorskem programu kot mladi raziskovalec. Področje njegovih raziskav je objektna tehnologija, s poudarkom na pristopih k razvoju modernih programskih sistemov.

Mag. Marjan Heričko je asistent na FER Maribor. Njegovo raziskovalno delo obsega vse vidike objektno tehnologije, s poudarkom na zagotavljanju kakovosti objektnega razvoja programskih sistemov.

Dr. Ivan Rozman je diplomiral na Fakulteti za elektrotehniko v Ljubljani, magistriral in doktoriral pa na Tehniški fakulteti Univerze v Mariboru. Je redni profesor Univerze v Mariboru in ustanovitelj Laboratorija za informacijske sisteme, ki ga vodi še danes. Je avtor številnih publikacij in vodja večih raziskovalnih projektov.

Dr. József Györkös je diplomiral, magistriral in doktoriral na Tehniški fakulteti Univerze v Mariboru. Je izredni profesor Univerze v Mariboru in avtor številnih publikacij objavljenih tako doma kot po svetu. Področja njegovih raziskav so informacijska družba, sistemi za podporo odločanju in upravljanje procesov razvoja informacijskih sistemov.

IZBIRNI POSTOPEK ZA VPIS NA FAKULTETE IN SAMOSTOJNE VISOKOŠOLSKE ZAVODE V SLOVENIJI

Tomaž Mohorič
Fakulteta za računalništvo in informatiko, Univerza v Ljubljani
tomazm@fri.uni-lj.si

Povzetek

Zadnja tri leta se vpis na obe slovenski univerzi in samostojne visokošolske zavode izvaja centralizirano, pri čemer je izbirni postopek naravnano tako, da imajo pri izbiri zaželenega študijskega programa prednost boljši kandidati. Prispevek obravnava postopke za optimalno razporejanje kandidatov na razpisana študijska mesta.

Abstract

For the last three years a centralized enrollment has been used at Slovenian universities and other schools of higher education. While choosing the desired study program, the selection procedure gives preference to candidates with better grades. The article deals with the selection and optimal order of candidates according to student quotas pertaining to each study program.



1. Uvod

Vsako leto se želi na fakultete ljubljanske in mariborske univerze ter na samostojne visokošolske zavode (v nadaljevanju fakultete) vpisati vse več kandidatov - bodočih študentov. In vsako leto ugotovljamo, da je prijav več, kot pa je na voljo študijskih mest, oziroma, da so želje kandidatov usmerjene predvsem v nekatere popularnejše študijske programe, med tem ko manj popularni programi ostajajo nezasedeni.

V preteklosti so za izbor in vpis kandidatov v svoje študijske programe skrbele kar posamezne fakultete in njihove vpisne službe. Kandidat, ki je sodil, da njegove možnosti za vpis na zaželeno fakulteto niso najboljše, se je moral prijaviti še na kakšno drugo fakulteto, morebiti še celo na več fakultet, če je želel biti kolikor toliko prepričan, da ga bo jeseni čakal sedež v predavalnici vsaj na eni izmed njih.

Tako večkratno prijavljanje pa je skrivalo v sebi temeljno slabost, namreč to, da fakultete vse do jesenskega vpisa niso vedele, kateri kandidati se bodo dejansko vpisali pri njih, in kateri kandidati so si jih izbrali le za rezervno varianto - izhod v sili. In tako ni bilo jasno ali bodo imele kandidatov preveč ali premalo. Ali bo potrebno kandidate selekcionirati ali pa razposlati naokrog obvestila, da so pri njih še prosta študijska mesta.

Vse to dogajanje v zvezi z vpisom je bilo do pred treh let pretežno strnjeno v obdobje konec septembra - začetek oktobra in ni bilo posebej prijetno ne za visokošolske zavode, ne za kandidate. Vpisati se na zaželeno fakulteto je bila bolj stvar organizacijskih sposobnosti kandidata in informacij, ki jih je uspel zbrati, kot pa njegovih študijskih sposobnosti - npr. doseženega uspeha pri maturi.

Nato pa je prišlo do manjše revolucije na področju vpisovanja v prve letnike fakultet in samostojnih visokošolskih zavodov. Leta 1993 je bil sprejet Zakon o visokem šolstvu, ki je predvidel centralizirano prijavljanje kandidatov v okviru vse Slovenije, priprave nanj so potekale vse leto 1994, njegovi učinki pa so se prvič pokazali pomladi in poleti 1995, ko se je ta novi način prijavljanja pričel tudi izvajati.

Razlog za uvedbo centraliziranega prijavljanja je preprost. Po eni strani naj bi bili kandidati razbremenjeni skakanja po različnih fakultetah in mrzličnega iskanja prostega mesta, po drugi strani pa naj bi postalo prijavljanje preglednejše, predvsem pa naj bi omogočilo boljšim kandidatom prednost pri izbiri zaželenega študijskega programa. Centralizirano prijavljanje je potekalo v študijskem letu 1996/97 že tretje leto.

Vsakoletni vpis se prične z Razpisom za vpis v prvi letnik [1, 2, 3], ki ga objavijo obe univerzi in samostojni visokošolski zavodi. V razpisu so navedeni vsi študijski programi in vpisni pogoji zanje, postopek prijavljanja, pravila v zvezi z omejitvijo vpisa, merila za izbiro, katera dokazila o izpolnjevanju vpisnih pogojev se zahtevajo za posamezne kategorije kandidatov in kakšen je izbirni postopek.

V nadaljevanju bo v prispevku podrobneje obravnavan predvsem izbirni postopek.

Namen izbirnega postopka je na razpisana študijska mesta razporediti kandidate, ki so študij uspešno zaključili z maturo ali zaključnim izpitom in izpolnjujejo morebitne posebne pogoje fakultet in visokih šol. Razvrstitev kandidatov, ki lahko izrazijo do tri različne želje za študij po določenem študijskem programu, mora biti pravična, kar pomeni, da naj ima kandidat z boljšim uspehom (izračunanim v skladu z obrazcem posamezne fakultete) pravico izbirati študijski program pred kandidatom s slabšim uspehom.

Obrazci fakultet so se v teh treh letih, kar je centralizirani vpis v veljavi, že precej poenotili. Tipični obrazec, ki prevladuje na večini fakultet, vrednoti srednješolski uspeh za vpis na univerzitetni študij takole: 60 % splošni uspeh pri maturi + 40 % splošni uspeh v 3. in 4. letniku, za vpis na visokošolski strokovni študij pa: 60% splošni uspeh pri zaključnem izpitu ali maturi + 40 % splošni uspeh v 3. in 4. letniku. Nekateri fakultete vključujejo v obrazec še uspeh pri kakšnem za stroko pomembnem predmetu, predvsem umetniške fakultete pa zahtevajo tudi preizkuse posebnih sposobnosti (npr. umetniške nadarjenosti), katerih utež lahko znaša do 90 % in celo 100 % uspeha – pravim umetnikom matura ni potrebna.

Manjši odstop od načela pravičnosti predstavlja z zakonom postavljeno pravilo, da se v primeru, ko ostaja posamezni program neomejen (zanj ni sprejet sklep o omejitvi vpisa), vanj uvrstijo vsi kandidati, ki so študij po tem programu izrazili s svojo prvo željo, ne glede na število doseženih točk. Pogoj je le, da so uspešno opravili maturo ali zaključni izpit.

2. Podatki za izbiranje

Izbirni postopek temelji na podatkih, ki prihajajo s fakultet, od kandidatov za vpis na fakultete, od Državnega izpitnega centra (uspeh pri maturah) in s srednjih šol, na katerih se opravljajo zaključni izpiti.

(1) Prvi vir podatkov je razpis, ki obsega podatke o številu razpisanih študijskih mest za posamezne študijske programe na fakultetah. Predlog zanj pripravijo fakultete, sprejmejo ga senata obeh univerz in senati samostojnih visokošolskih zavodov ob soglasju vlade Republike Slovenije. Razpis je sestavljen na osnovi predlogov

posameznih fakultet in samostojnih visokošolskih zavodov. Poglavitna sestavina razpisa je število razpisanih mest za posamezne študijske programe, vpisni pogoji in kriteriji za izbor kandidatov v primeru omejitve vpisa.

- (2) Drugi vir so kandidatove izražene želje po študijskem programu. Kandidati podajo eno do tri po prioriteti urejene želje, v katerih navedejo študijski program, ki ga želijo vpisati in študirati.
- (3) Tretji vir je morebitni sklep o omejitvi vpisa za posamezne študijske programe. Če število prijav v posamezni program (upoštevaje le prve želje) bistveno presega število v razpisu navedenih mest, je lahko sprejet sklep o omejitvi vpisa (predlagajo ga fakultete, soglasje daje vlada Republike Slovenije). Tak program imenujemo v nadaljevanju "omejeni program".
- (4) Četrti vir pa so podatki o uspehu kandidatov pri maturi oziroma zaključnem izpitu in podatki o uspehu v zadnjih dveh letnikih srednje šole. Podatke o študijskem uspehu kandidatov, ki so opravili maturo, posreduje Državni izpitni center, podatke o uspehu pri zaključnem izpitu pa posredujejo ustrezne srednje šole. Če je za vpis v program zahtevan tudi preizkus posebnih sposobnosti, se le-ta izvaja na matični fakulteti, vsi omenjeni podatki pa se nato zberejo pri visokošolski prijavno-informacijski službi. Ker se pri ocenjevanju uporabljajo različne merske lestvice, se le-te za potrebe razvrščanja predhodno normalizirajo - izračuna se število točk za vsakega kandidata, ki se izrazi z lestvico 0 - 100 točk.

Vsi omenjeni podatki se po predhodnem temeljitem preverjanju strnejo v dveh tabelah – podatki o razpisu in podatki o omejitvah v tabeli RAZPIS, podatki o željah in podatki o uspehu kandidatov pa v tabeli PRIJAVA.

Razpis. RAZPIS je tabela, ki vsebuje podatke o številu razpisanih mest za posamezne programe. V nekaterih primerih je število razpisanih mest podano za skupino programov, kar pomeni, da razdelitev števila mest za programe znotraj skupine ni vnaprej določeno in se mesta lahko razdele na posamezni fakulteti kasneje, v odvisnosti od želja kandidatov. Ključ tabele tvorijo atributi ID_VIS_ZAVOD (identifikacijska številka visokošolskega zavoda), ID_NACIN_STUDIJA (redni študij: 1, izredni študij: 2) in ID_ELEMENT_RAZPIS (program ali skupina programov, ki je predmet razpisa). Programi so lahko omejeni ali neomejeni in so lahko eno-predmetni ali dvopredmetni. Shema tabele RAZPIS in del njene vsebine (naslovi stolpcev so nekoliko okrajšani) je predstavljena na sliki 1.

Tabela želja. V tabeli PRIJAVA so zbrane vse želje vseh kandidatov za študij na fakultetah (slika 2). Vsaka

Name	Null?	Type
ID_VIS_ZAVOD	NOT NULL	NUMBER(2)
ID_NACIN_STUDIJA	NOT NULL	NUMBER(1)
ID_ELEMENT_RAZPIS	NOT NULL	VARCHAR2(4)
RAZPISANIH_MEST		NUMBER(5)
OMEJEN		VARCHAR2(1)
DVOPREDMETNI		VARCHAR2(1)
OPIS_RAZP_MESTA		VARCHAR2(70)

ID_VIS_ZAVOD	ID_NACIN_STUDIJA	ID_ELEMENT_RAZPIS	RAZPISANIH_MEST	OM_DV	OPIS_RAZP_MESTA
1	1	AJ10	120	D	N RAZREDNI POUK...
1	1	AJ30	30	N	N RAZREDNI POUK...
1	1	AK00	30	D	N SOCIALNA PEDA...
1	1	AL00	20	D	N LIKOVNA PEDAG...
1	1	AM00	45	N	N MATEMATIKA IN...
1	1	A000	30	N	N MATEMATIKA IN...
63	1	HB00	150	N	N RACUNALNIŠTVO...
63	1	L200	200	N	N RACUNALNIŠTVO...
64	1	H500	250	N	N ELEKTROTEHNIK...
64	1	L000	300	N	N ELEKTROTEHNIK...
71	1	2400	50	N	N MIKROBIOLOGIJ...
75	1	1412	60	N	N GOSPODARSKO I...
75	2	1412	20	N	N GOSPODARSKO I...
75	1	1512	70	N	N GOSPODARSKO I...
75	1	1612	40	N	N GOSPODARSKO I...
75	2	1712	10	N	N GOSPODARSKO I...
85	1	1500	30	D	N KRAJINSKA ARH...

Slika 1: Shema tabele RAZPIS in del njene vsebine

vrstica v tabeli predstavlja po eno željo. Ob prejemu prijave z željami dodeli prijavno-informacijska služba vsakemu kandidatu svojo identifikacijsko številko (ID_PRVA_PRIJAVA). Vsaka od želja posameznega kandidata je oštevilčena (ID_RUBRIKA $\in [1,2,3]$). Želja kandidata se lahko nanaša na enopredmetni ali pa dvopredmetni program. Želja po enopredmetnem programu je v tabeli PRIJAVA predstavljena z eno vrstico, pri čemer ima atribut ID_POLOVICA vrednost 1. Želja po dvopredmetnem programu je podana z dvema vrsticama, v prvi ima atribut ID_POLOVICA vrednost 1, v drugi pa vrednost 2. Atribut ID_VIS_ZAVOD je numerična oznaka visokošolskega zavoda, ki študijski program razpisuje. Atribut ID_NACIN_STUDIJA opredeljuje redni (= 1) oziroma izredni študij (= 2). Atribut ID_ELEMENT_RAZPIS označuje posamični program ali skupino programov, za katero je navedeno število razpisanih mest v razpisu. V primeru, da je razpisana skupina programov, podaja atribut ID_PROGRAM oznako konkretnega programa, v katerega se kandidat prijavlja, v nasprotnem primeru pa imata atributa ID_ELEMENT_RAZPIS in ID_PROGRAM enako vrednost. Atribut SK_TOCK podaja število točk, ki jih je kandidat dosegel, upošteva maturo oziroma zaključni izpit ter uspeh v srednji šoli. Ključ tabele tvorijo atributi, ki so na sliki 2 označeni z NOT NULL. To so ID_PRVA_PRIJAVA, ID_RUBRIKA in ID_POLOVICA (Slika 2).

3. Izbirni postopek

Izbirni postopek se izvaja v treh korakih. Prvi korak obsega inicializacijo osnovnih delovnih datotek in določene prilagoditve, pri čemer so nekatere od njih (verjetno) veljavne le v tekočem letu. V drugem koraku se izvede uvrstitev kandidatov, ki so se s prvo željo prijavi v neomejeni program, v tretjem koraku pa se izvede razvrščanje preostalih kandidatov glede na dosežene točke in še prosta mesta v študijskih programih (Slika 3).

V skladu s pravili uvrščanja oziroma razvrščanja se prijave kandidatov dele v dve skupini. Prva skupina so enopredmetne želje, s katerimi kandidati kandidirajo v posamezen program, v drugi skupini pa se nahajajo želje, ki se nanašajo na dvopredmetne programe, kot jih srečujemo na Filozofski fakulteti v Ljubljani in Pedagoški fakulteti v Mariboru. Pri slednjih se kandidatove želje nanašajo na dva iz množice razpisanih enakovrednih programov. Ker se izvajata uvrščanje in razvrščanje za ti dve skupini po različnih postopkih, se tudi želje v nadaljevanju ustrezno razdele v različne tabele (slika 4).

Globalni pogled na razvrščanje prikazuje slika 4. Vsebinska tabela PRIJAVA se najprej razdeli v enopredmetne (RUB-N) in dvopredmetne želje (RUB-P). Iz tabele RUB-N se nato v tabelo RUB-N9 izločijo vse prve želje

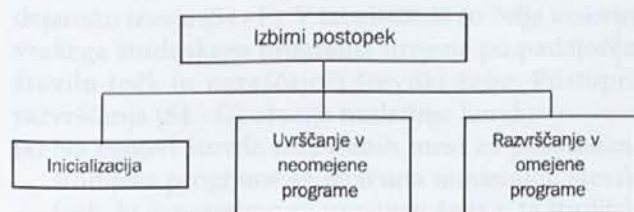
Name	Null?	Type
ID_PRVA_PRIJAVA	NOT NULL	NUMBER(5)
ID_RUBRIKA	NOT NULL	NUMBER(1)
ID_POLOVICA	NOT NULL	NUMBER(1)
ID_VIS_ZAVOD		NUMBER(2)
ID_NACIN_STUDIJA		NUMBER(1)
ID_ELEMENT_RAZPIS		VARCHAR2(4)
ID_PROGRAM		VARCHAR2(4)
SK_TOCK		NUMBER(10,4)

ID_PRVA_PRIJAVA	ID_RUBRIKA	ID_POLOVICA	ID_VIS_ZAVOD	ID_NACIN_STUDIJA	ID_E	ID_P	SK_TOCK
2328	2	1	2	1	6600	6600	86
2346	3	1	2	1	6100	6100	79
2348	2	1	2	1	6600	6600	65
2348	3	1	2	1	6300	6300	65
2351	2	1	2	1	7100	7100	87
2354	2	1	2	1	6300	6300	66
2438	3	1	2	1	7200	7200	49
2497	2	1	2	1	6100	6100	46
2543	2	1	2	1	7100	7100	83
2579	3	1	2	1	6300	6300	60
2596	3	1	2	1	6100	6100	83,2
2621	3	1	2	1	0300	0300	59
2651	3	1	2	1	6300	6300	80
2482	2	1	2	1	63D0	63D0	72,5
2482	2	2	2	1	79D0	79D0	72,5
2581	2	1	2	1	79D0	79D0	64,5
2581	2	2	2	1	80D0	80D0	64,5
2591	3	1	2	1	65D0	65D0	77
2591	3	2	2	1	66D0	66D0	77
2596	2	1	2	1	65D0	65D0	83,2
2596	2	2	2	1	80D0	80D0	83,2
2652	2	1	2	1	79D0	79D0	60
2652	2	2	2	1	80D0	80D0	60
2653	3	1	2	1	65D0	65D0	98,5
2653	3	2	2	1	80D0	80D0	98,5
2736	2	1	2	1	70D0	70D0	69
2736	2	2	2	1	68D0	68D0	69

Slika 2: Shema tabele PRIJAVA in del njene vsebine

v neomejene programe. Podobno se iz tabele RUB-P izločijo v tabelo RUB-P9 vse prve želje, pri katerih sta oba programa neomejena. Po omenjenem izločanju se v tabeli RAZPIS število razpisanih mest za posamezne programe zmanjša za število že uvrščenih prijav v tabelah RUB-N9 in RUB-P9. Vse ostale želje (druge in tretje) že uvrščenih kandidatov se iz tabel RUB-N in RUB-P

izbrišejo. Sledi razvrščanje v tabelah RUB-N1 oziroma RUB-P1, v kateri se prijave prenesejo iz tabel RUB-N in RUB-P. Razvrščanje poteka v več krogih. Najbolje uvrščene želje kandidatov v posameznem krogu razvrščanja se shranijo v tabelah RUB-N8 in RUB-P8. Po vsakem krogu razvrščanja je nato potrebno poiskati kandidate, katerih želje so se uvrstile tako v skupini enopredmetnih želja kot tudi v skupini dvopredmetnih želja. Veliko kandidatov namreč kandidira z obema vrstama želja. Bolje rangirana želja ostaja v tabeli, slabše rangirana želja pa se izloči v tabelo IZLOČENI-N oziroma IZLOČENI-P. In postopek razvrščanja v tabelah RUB-N1 in RUB-P1 se ponavlja toliko časa, dokler se v preseku pojavlja še kakšen kandidat s svojimi željami. Ob zaključku razvrščanja se v tabelo UVRŠČENI prepisejo neposredno uvrščeni kandidati iz tabel RUB-N9 in RUB-P9 ter tabel RUB-N8 in RUB-P8. Vsebina te tabele predstavlja rezultat



Slika 3: Faze izbirnega postopka

razvrščanja. V nadaljevanju so podrobneje predstavljene posamezne faze izbirnega postopka.

3.1 Inicializacija izbirnega postopka

V okviru inicializacije izbirnega postopka se kreirajo delovne tabele, izvedejo pa se tudi nekatere prilagoditve, ki običajno veljajo le za tekoči vpis in se lahko iz leta v leto spreminjajo.

Ena takih prilagoditev so tako imenovani dvopredmetni programi, katerih ena polovica sodi v okrilje ene fakultete, druga pa v okrilje druge fakultete. Tako so bili v šolskem letu 1996/97 sestavljeni v dvopredmetne programe *teologija B* s Teološke fakultete v Ljubljani in poljuben program s Pedagoške fakultete v Mariboru oziroma poljuben program s Filozofske fakultete v Ljubljani. Prav tako sta se v dvopredmetni program povezovala programa *politologija B* in *novinarstvo B* s Fakultete za družbene vede s poljubnim programom s Filozofske fakultete v Ljubljani. Povezovanje programov različnih fakultet se je izvedlo z navidezno selitvijo omenjenih programov z njihovih matičnih fakultet na Pedagoško fakulteto v Mariboru oziroma Filozofsko fakulteto v Ljubljani.

Vse možne povezave pri dvopredmetnih programih pa vendarle niso dopustne. Tako so v dveh tabelah *PREPOVEDANI_1* in *PREPOVEDANI_2* navedeni programi, katerih paré se ne dá (ne sme) vpisati kot dvopredmetni program. Kateri pari so dopustni in kateri ne, je pojasnjeno v razpisih posameznih fakultet, tako da so kandidati o tem pravočasno obveščeni.

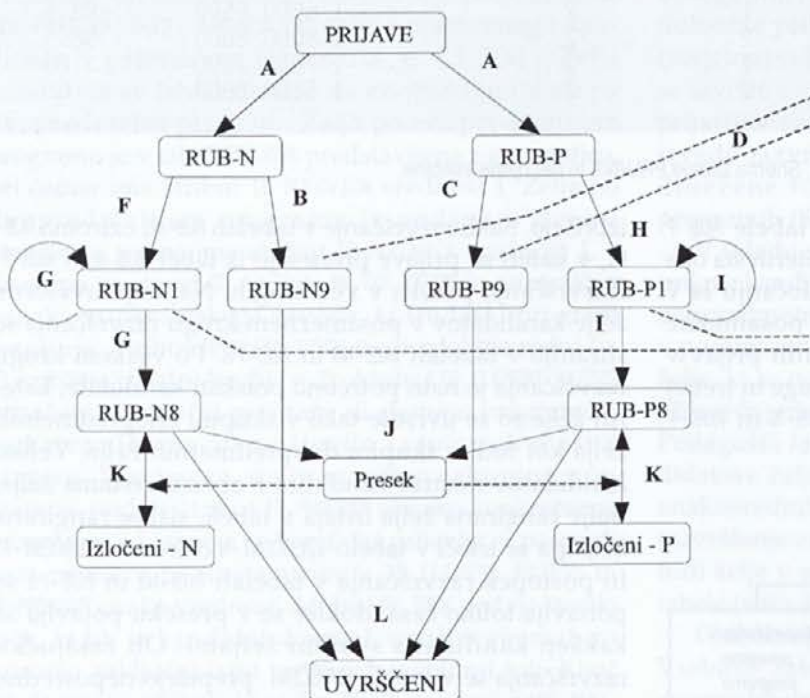
Uvrščanje in razvrščanje se pri enopredmetnih oziroma dvopredmetnih programih razlikujeta do te mere, da je bilo za vsako vrsto programov potrebno izdelati poseben algoritem, zato se pred razvrščanjem želje kandidatov razvrstijo v dve tabeli RUB-N in RUB-P (S4 - A). V prvi se nahajajo enopredmetni programi, v drugi pa dvopredmetni programi.

3.2 Uvrščanje želja pri neomejenih programih

Kot uvod v uvrščanje je potrebno tabelo enopredmetnih želja RUB-N kot tudi tabelo dvopredmetnih želja RUB-P najprej dopolniti z atributom OMEJEN, ki se izvirno nahaja v tabeli RAZPIS. Na ta način je vsaka želja opredeljena s svojim vrstnim redom (prva, druga, tretja želja) in kot omejena ali neomejena želja.

Enopredmetne želje. V tabeli RUB-N poiščemo po vrsti vse prve želje, ki se nanašajo na neomejene programe (OMEJEN = 'N' in id_rubrika = 1). Vse prve in hkrati neomejene želje kandidatov se prenesejo v tabelo RUB-N9 (S4 - B), preostale želje teh uvrščenih kandidatov pa se iz obeh tabel RUB-N in RUB-P izbrišejo.

Dvopredmetne želje. Nekoliko drugačen je pristop k uvrščanju dvopredmetnih želja in sicer se najprej označijo vse želje, ki ne morejo biti uvrščene. V tabeli RUB-P se označijo vse prve želje po uvrstitvi v omejene programe ter vse druge in tretje želje. Nato se dodatno označijo še prve želje, katerih ena polovica (eden izmed programov) je omejena. Neoznačene želje so



Slika 4: Globalni potek razvrščanja

tiste, ki se nato uvrstijo v tabelo RUB-P9 (S4 - C). Vse preostale želje uvrščenih kandidatov se iz tabel RUB-P in RUB-N izbrišejo (slika 4).

Ažuriranje razpisa. Po uvrščanju eno in dvopredmetnih želja se je določeno število razpisanih mest s tem že zasedlo. Število razpisanih mest v tabeli RAZPIS je zato potrebno zmanjšati za število že uvrščenih kandidatov. Kandidati, ki so kandidirali z enopredmetnimi željami, so zasedli vsak po eno mesto, kandidati, ki so kandidirali z dvopredmetnimi željami, pa so zasedli vsak po dve mesti. V ta namen se kreira začasna tabela, podobna tabeli RAZPIS, v kateri so navedena vsa že zasedena mesta. Od vsebine tabele RAZPIS se potem »odšteje« vsebina začasne tabele (S4 - D). V tabeli RAZPIS so tako navedena le še preostala nezasedena mesta. Le-ta so na voljo v okviru naslednje faze - razvrščanja v omejene programe.

3.3 Razvrščanje želja pri omejenih programih

Razvrščanje je postopek, s katerim se razporejajo kandidati na razpisana mesta omejenih programov ter preostala prosta mesta neomejenih dvopredmetnih programov in sicer glede na dosežene točke pri maturi oziroma zaključnem izpitu. Postopek razvrščanja poteka v naslednjih korakih:

- Razvrstitev kandidatov glede na enopredmetne želje.
- Razvrstitev kandidatov glede na dvopredmetne želje.
- Identifikacija kandidatov, ki so uvrščeni tako z enopredmetnimi željami kot tudi z dvopredmetnimi željami, in izločanje manj zaželenih prijav iz množice enopredmetnih želja oziroma iz množice dvopredmetnih želja.
- Zaporedje postopkov (a) – (c) se ponavlja toliko časa, dokler se ne pojavi noben dvojno uvrščeni kandidat več.

Pred samim razvrščanjem se k tabelama RUB-N in RUB-P kreirata pripadajoči tabeli RAZ-N in RAZ-P (S4 - E), v katerih so navedeni zgolj programi, na katere kandidirajo kandidati, katerih želje se nahajajo v tabelah RUB-N oziroma RUB-P.

3.3.1 Razvrščanje enopredmetnih želja

Tabeli RUB-N in RAZ-N služita za inicializacijo tabel RUB-N1 in RAZ-N1 (delovni razpis), v katerih se razvrščanje dejansko izvaja (S4 - F). V tabeli RUB-N1 so želje v okviru vsakega študijskega programa urejene po padajočem številu točk in naraščajoči številki želje. Postopek razvrščanja (S4 - G) obsega naslednje korake:

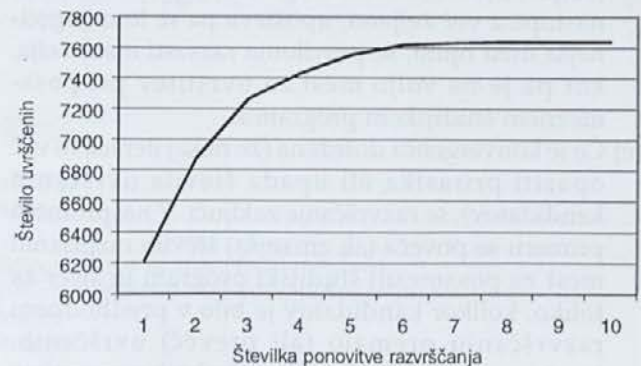
- Na osnovi števila razpisanih mest za posamezne študijske programe se izračuna minimalno število točk, ki je potrebno za uvrstitev želja v ta študijski program. Če se pojavi več želja z minimalnim

številom točk (želje v repu), se število razpisanih mest v RAZ-N1 za vsak študijski program ustrezno poveča. Zaradi hitrejšje konvergence se začetno število razpisanih mest v RAZ-N1 ob začetku razvrščanja pri vseh programih dodatno poveča za 20 %.

- V okviru vsakega študijskega programa se poiščejo tiste prve želje, ki dosegajo minimalno število točk. Označi se uvrščena želja posameznega kandidata, vse njegove preostale želje pa se označijo kot neveljavne.
- Postopek (b) se ponovi za vse druge in nato za vse tretje še neoznačene želje. Ker posamezni kandidat nastopa z več željami, upošteva pa se le najugodnejša med njimi, se praviloma razvrsti manj želja, kot pa je na voljo mest za uvrstitev pri posameznem študijskem programu.
- Če je konvergenca dosežena (že nekaj iteracij ni več opaziti prirastka ali upada števila uvrščenih kandidatov), se razvrščanje zaključi, uvrščene želje pa se prepisejo v tabelo RUB-N8. V nasprotnem primeru se poveča (ali zmanjša) število razpisanih mest za posamezni študijski program in sicer za toliko, kolikor kandidatov je bilo v predhodnem razvrščanju premalo (ali preveč) uvrščenih. Izračuna se tudi število točk, potrebnih za uvrstitev želja, število razpisanih mest pa se poveča še za število želja v repu. Oznake vseh želja se izbrišejo, izbriše se tudi vsebina tabele RUB-N8. Razvrščanje se nadaljuje s postopkom (b). Število uvrščenih kandidatov po prvem krogu razvrščanja enopredmetnih želja kaže diagram S5 (slika 5).

3.3.2 Razvrščanje dvopredmetnih želja

Pri razvrščanju dvopredmetnih želja obstaja posebnost, ki je pri enopredmetnih željah ni. Kandidat, ki nastopa v izbirnem postopku z več kot eno dvopredmetno željo, se uvrsti, če se uvrsti želja, sestavljena iz poljubnega para njegovih polovičnih želja. Veljavne dvopredmetne želje namreč niso le tiste, ki so jih kandidati eksplicitno navedli v prijavi, ampak tudi tiste, ki se dajo sestaviti iz polovic njegovih



Slika 5: Prvi krog razvrščanja enopredmetnih želja

dvopredmetnih želja. Izjema so le take sestavljenke, ki vsebujejo dve enaki polovični želji, in pa tiste sestavljenke, ki so navedene v tabelah PREPOVEDANI_1 in PREPOVEDANI_2.

Tabeli RUB-P1 in RAZ-P1 se ob pričetku razvrščanja inicializirata z vsebino datotek RUB-P in RAZ-P (S4 - H). Tudi v tabeli RUB-P1 so želje v okviru vsakega študijskega programa urejene po padajočem številu točk in naraščajoči številki želje. Postopek razvrščanja (S4 - I) obsega naslednje korake:

- (a) Na osnovi števila razpisanih mest za posamezni študijski program se izračuna minimalno število točk, ki je potrebno za uvrstitev polovičnih želja v ta študijski program. Če se pojavi več želja z minimalnim številom točk (želje v repu), se število razpisanih mest ustrezno poveča.
- (b) Preliminarno se uvrstijo polovične želje, ki dosegajo minimalno število točk, prav tako pa tudi vse neomejene in hkrati prve polovične želje, ki bi bile sicer uvrščene že v okviru uvrščanja neomejenih programov, če bi bila tudi njihova komplementarna polovica neomejena.
- (c) Iz tabele RUB-P1 se po vrsti berejo preliminarno uvrščene prve polovične želje. Prebere se polovična želja in iz nje se ugotovi, kateremu kandidatu pripada. V delovno tabelo se potem preberejo vse preliminarno uvrščene polovične želje tega kandidata. Množica želja se nato pomnoži sama s seboj (kartezijski produkt), kar nam da za rezultat množico dvopredmetnih želja. Iz te množice se izločijo vse neveljavne želje (želja, katere polovici sta identični; želja, ki je navedena v tabeli PREPOVEDANI_1 ali PREPOVEDANI_2). Preostale želje se uredijo po vrednosti para številki želja v naslednjem zaporedju: [1,1], [1,2], [1,3], [2,2], [2,3], [3,3]. Tako ima par polovičnih želja [1, 3] prednost pred parom [2, 2], saj se v prvem paru nahaja vsaj polovica najljubše želje. Najbolje uvrščeni par polovičnih želja prepisemo v tabelo RUB-P8, vse prijave kandidata v tabeli RUB-P1 pa se označijo kot neveljavne.
- (d) Postopek (c) se ponovi za vse druge in nato za vse tretje še neoznačene želje. Ker lahko kandidat nastopa z več željami, upošteva pa se le najugodnejša med njimi, se praviloma razvrsti manj želja, kot pa je na voljo mest za uvrstitev pri posameznem študijskem programu.
- (e) Če je konvergenca dosežena (že nekaj iteracij ni več opaziti prirastka ali upada števila uvrščenih kandidatov), se razvrščanje zaključi. V nasprotnem primeru se poveča (ali zmanjša) število razpisanih mest za posamezni študijski program in sicer za toliko, kolikor kandidatov je bilo v predhodnem razvrščanju premalo (ali preveč) uvrščenih. Izračuna se tudi število točk, potrebnih za uvrstitev želja, število razpisanih mest pa se poveča še za

število želja v repu. Oznake vseh želja se izbrišejo, izbriše se tudi vsebina tabele RUB-P8. Razvrščanje se nadaljuje s postopkom (b).

3.3.3 Presek med eno in dvopredmetnimi željami

Po razvrščanju eno in dvopredmetnih želja, se uvrščene enopredmetne želje kandidatov nahajajo v tabeli RUB-N8, uvrščene dvopredmetne želje pa v tabeli RUB-P8. Ker kandidirajo nekateri kandidati tako z enopredmetnimi kot tudi z dvopredmetnimi željami, so nakateri izmed njih uvrščeni s po eno željo ene in druge vrste.

- (a) Najprej se izračuna presek med vsebino obeh tabel (S4 - J), ki se shrani v delovno tabelo PRESEK. V njej se nahajajo vsi dvojno uvrščeni kandidati – kandidati, ki so se uvrstili z enopredmetnim in tudi z dvopredmetnim programom. Ker je ena uvrstitev bolj zaželena kot druga, se vse želje, povezane z manj zaželjeno uvrstitvijo izločijo iz tabele RUB-N1 (ali RUB-P1, odvisno od tega, s katero željo je bil kandidat manj uspešen) v tabelo IZLOČENI-N (ali IZLOČENI-P) in se shranijo za morebitno ponovno uporabo (S4 - K). S tem, ko se nekatere želje izločijo iz nadaljnega razvrščanja, se sprostijo mesta, ki jih lahko v nadaljevanju zasedejo doslej še neuvrščeni kandidati. Celoten cikel razvrščanja se nato ponovi.

V nadaljnjih ciklih razvrščanja se zaradi sproščenih mest uvrščajo želje, ki pred tem še niso bile uvrščene, ali pa se ne uvrstijo več želje, ki pa so bile pred tem že uvrščene. Tako se lahko med drugim pripeti, da se pri posameznem kandidatu tudi tista preostala želja ne uvrsti več. Na primer: kandidat se je najprej uvrstil tako z enopredmetno kot tudi z dvopredmetno željo. Manj zaželena želja (npr. enopredmetna) se je izločila, v naslednjem ciklu pa kandidat ni več uvrščen tudi z bolj zaželjeno željo (npr. dvopredmetno). Za tak primer smo uporabili izraz »izgubljeni« kandidat. Problem se reši tako, da se vse kandidatove želje vrnejo v ponovno razvrščanje.

- (b) Če se ne uvrsti več želja iz tabele RUB-N1, se v tabelo RUB-P1 vrnejo vse kandidatove želje iz tabele IZLOČENI-P, če se ne uvrsti več želja iz tabele RUB-P1, se v tabelo RUB-N1 vrnejo vse kandidatove želje iz tabele IZLOČENI-N in celoten cikel razvrščanja se ponovi (S4 - K).

Ko se noben kandidat ne uvrsti več z obema vrstama želja (presek je prazen), sledi prepis vseh uvrščenih želja v tabelo UVRŠČENI. Vanjo se prepisujejo neposredno uvrščene enopredmetne (RUB-N9) in dvopredmetne (RUB-P9) želje ter razvrščene enopredmetne (RUB-N8) in dvopredmetne (RUB-P8) želje (S4 - L).

Sledi še zadnja korekcija. Zaradi preseljevanja želja med datotekama RUB-N1 in IZLOČENI-N oziroma RUB-P1 in

IZLOČENI-P se lahko zgodi, da se znajde kandidatova bolj zaželena želja v datoteki izločenih želja, manj zaželena pa med uvrščenimi željami, pri čemer obe želji izpolnjujeta pogoj minimalnih točk za uvrstitve.

(c) Za vsakega kandidata, katerega želja se nahaja izločena v tabeli IZLOČENI-N oziroma IZLOČENI-P in ki dosega zahtevani minimum točk, se preveri, ali ni morda uvrščen z manj zaželjeno željo. Če je temu tako, se manj zaželena želja izloči iz tabele UVRŠČENI, vanjo pa se prenese bolj zaželena želja.

Postopek razvrščanja kandidatov je s tem končan. Vsi uvrščeni kandidati se predstavljene s svojimi optimalnimi željami nahajajo v tabeli UVRŠČENI.

3.4 Programska oprema

Programska oprema, ki podpira izbirni postopek, je napisana v programskih jezikih SQL in PL/SQL, podatkovni del pa je upravljan s podatkovnim strežnikom ORACLE V7.2.3.

V zvezi s tem, kako primerno napisati in organizirati programsko opremo, je bilo narejenih več poskusov. Prvi poskus z uporabo gostečega jezika ANSI C in vgnezenega SQL je pokazal, da je občutljiva točka prevajalnik za jezik C. S prevajalnikom generirano kodo je namreč potrebno prevesti še s prevajalnikom za C. Pokazalo se je, da prevajanje uspe le, če se zanj uporabi prav določena (in ne najnovejša) verzija prevajalnika, v nasprotnem primeru prevajalnik sicer ne sporoča napak pri prevajanju in prevedena koda se izvaja, žal pa so rezultati napačni. Zaradi te povezanosti z verzijo prevajalnika za C je močno ogrožena prenosljivost programske opreme iz verzije v verzijo pred-

prevajalnika. Drugi poskus je bil izveden z uporabo dinamičnega SQL v okviru jezika PL/SQL. Ta varianta sicer uspešno deluje, vendar se je pokazalo, da se obseg programske kode poveča za približno 100 % (kar velja tudi za vgnezeni SQL), njena preglednost in razumljivost pa seveda upade.

Ker se je programska oprema v zadnjih letih ves čas spreminjala (in se verjetno še bo), je bila izbrana varianta z mešanico jezikov SQL in PL/SQL. Po tej varianti je koda najmanj, je najpreglednejša in se jo da tudi preprosto vzdrževati, njena slabost pa je ta, da je realizacija zank in razvejitev možna le znotraj sekcij PL/SQL. Posledica je ta, da je rokovanje s tako napisanimi programi razmeroma okorno in daleč od vsake elegance. Predvidevamo, da bo potem, ko se bodo algoritmi razvrščanja ustalili – doslej so se še vsako leto nekoliko spreminjali – smiselno tudi programsko opremo spraviti v priročnejšo obliko. Prav zaradi nenehnega spreminjanja algoritmov in s tem povezanega testiranja programske opreme je le-ta napisana nekoliko redundantno. V končni obliki se bo lahko iz nje izpustila marsikatera tabela z vmesnimi rezultati, ki je bila potrebna zaradi preverjanja delovanja algoritmov.

Programska oprema je trenutno urejena v treh imenikih, v katerih se nahajajo posamezni programi (slika 6).

3.5 Izvedba izbirnega postopka

Izbirni postopek izvedemo s pomočjo v predhodnih razdelkih opisanih algoritmov oziroma programov. Najprej izvedemo programe iz imenika inic/, nato programe iz imenika neom/ in nazadnje programe iz imenika raz/.

Inic:		
1	BRISOBJ.sql	Brisanje starih delovnih tabel pred razvrščanjem
2	INIC.sql	Prilagoditve in kreiranje tabel eno in dvopredmetnih želja
Neom:		
1	INIC.sql	Kreiranje tabel za neposredno uvrščanje
2	NEOMN.sql	Uvrščanje enopredmetnih želja v neomejene programe
3	NEOMP.sql	Uvrščanje dvopredmetnih želja v neomejene programe
4	RAZPIS.sql	Ažuriranje števila prostih mest v razpisu
Raz:		
0	CR_IZLOCENI.sql	Kreiranje tabel za izločene želje
1	UVOD_N.sql	Priprava za enopredmetno razvrščanje
2	JEDRO_N.sql	Enopredmetno razvrščanje
3	KONEC_N.sql	Zaključek enopredmetnega razvrščanja
4	UVOD_P.sql	Priprava za dvopredmetno razvrščanje
5	JEDRO_P.sql	Dvopredmetno razvrščanje
6	PRESEK.sql	Iskanje dvojno uvrščenih kandidatov in izločanje manj zaželenih želja
7	IZGUBLJENI.sql	Iskanje kandidatov z izločenimi željami in njihovo vračanje v razvrščanje
8	ZDRUZENI.sql	Kreiranje tabele uvrščenih kandidatov
9	KOREKCIJA.sql	Korekcija pri izboru bolj zaželenih želja

Slika 6: Programska oprema za izbirni postopek, urejena po imenikih

A Inicializacija datotek

```
inic/ 1BRISOBJ.sql
      2INIC.sql
```

B Uvrščanje kandidatov na neomejene programe

```
neom/ 1INIC.sql
      2NEOMN.sql
      3NEOMP.sql
      4RAZPIS.sql
```

C Razvrščanje kandidatov po točkah

```
raz/ OCR_IZLOCENI.sql
     1UVOD_N.sql
     4UVOD_P.sql
     ponavljal
```

ponavljal

```
2JEDRO_N.sql
```

dokler se število uvrščenih ne spreminja več

```
3KONEC_N.sql
```

ponavljal

```
5JEDRO_P.sql
```

dokler se število uvrščenih ne spreminja več

```
6PRESEK.sql
```

```
7IZGUBLJENI.sql
```

dokler ni presek prazen

```
8ZDRUZENI.sql
```

```
9KOREKCIJA.sql
```

Iteracijski postopki so v zgornjem primeru realizirani z datotekami, v katerih se iterirani ukazi ponavljajo, saj izven jezika PL/SQL ni primernega ukaza za realizacijo programskih zank.

Programi, shranjeni v lokalnem računalniku – odjemalcu, se posredujejo v obdelavo podatkovnemu strežniku preko vmesnika SQL*Plus.

4. Zaključek

Programska oprema za izbirni postopek se je pisala in testirala v Laboratoriju za informatiko na Fakulteti za računalništvo in informatiko v Ljubljani. Od prvega razvrščanja v letu 1995 pa do lanskega leta je doživela veliko sprememb. V tem času se je spreminjal izbirni

postopek, spreminjale so se zahteve fakultet, zato so se spreminjali tudi programi. Vzroki niso bili zgolj zunanji, pač pa so se odpravljale slabosti in napake in izvajala se je tudi optimizacija. Zaradi programov, ki so se izvajali v dvojni zanki, je bilo zelo pomembno, kako so bili napisani kritični deli. Posebno pozornost je bilo potrebno posvetiti tudi indeksiranju tabel. Čas izvajanja izbirnega postopka se je postopoma skrajšal na približno 2 uri, vendar pa je močno odvisen od vrednosti vhodnih podatkov. V lanskem letu (1997) je bila struktura vhodnih podatkov naslednja:

	Vse želje	Enopredmetne želje	Dvopredmetne
Št. želja	41782	36013	5769
Št. kandidatov	16409	16085	2040

Izbirni postopek pa je uspešno prestalo naslednje število kandidatov:

Uvrščeni kandidati na	Št. kandidatov
- enopredmetne neomejene programe	5028
- dvopredmetne neomejene programe	235
- enopredmetne omejene programe	7563
- dvopredmetne omejene programe	562
Skupaj	13388

Pravilnost algoritmov izbirnega postopka se je preverjala z obsežnim testiranjem in v lanskem letu so testi pokazali, da nam je uspelo izvesti izbirni postopek povsem brez napak v nasprotju s predhodnimi leti, ko so bile potrebne po računalniškem delu razvrščanja še manjše ročne korekcije.

Viri

- [1] Razpis Univerze v Ljubljani, Univerze v Mariboru in samostojnih visokošolskih zavodov za vpis v študijskem letu 1995/96, Ljubljana, 1995
- [2] Postopek razvrščanja kandidatov za vpis na fakultete in visoke šole, Ljubljana, 1995
- [3] Razpis Univerze v Ljubljani, Univerze v Mariboru in samostojnih visokošolskih zavodov za vpis v študijskem letu 1996/97, Ljubljana, 1996
- [4] Razpis Univerze v Ljubljani, Univerze v Mariboru in samostojnih visokošolskih zavodov za vpis v študijskem letu 1997/98, Ljubljana, 1997

Doc. dr. Tomaž Mohorič je zaposlen na Fakulteti za računalništvo in informatiko, kjer predava vrsto predmetov s področja podatkovnih baz in informacijskih sistemov. Raziskovalno se ukvarja s konceptualnim modeliranjem, načrtovanjem relacijskih in objektno usmerjenih podatkovnih baz ter sistemi za upravljanje podatkovnih baz.

INFORMIXOV OPTIMIZATOR POIZVEDBE SQL

Lilijana Mihelič

Povzetek

Informixov optimizator poizvedb SQL (Informix Cost-Based Optimizer) je del celotnega sistema, ki skrbi za delovanje podatkovne baze. Njegova naloga je določiti optimalen način poizvedbe. Članek opisuje delovanje Informixovega optimizatorja, ki je na voljo v Informixu, v izdaji 7 in v kasnejših izdajah.

V članku so podani povzetki tehnik, s katerimi povečamo učinkovitost poizvedb. Opisane so tri strategije, ki se uporabljajo pri izvajanju stika, značilnosti optimizatorja, algoritmi, ki jih uporablja optimizator, z namenom, da izvrši poizvedbo na najprimernejši način in nekaj tehnik pri pisanju poizvedb, s katerimi pomagamo optimizatorju, da poizvedbo izvrši hitreje.

Abstract

The Informix Cost-Based optimizer is the component of the database engine that decides how to perform a query. This article describes the optimizer used in Informix engines, version number 7 and higher.

The article reviews techniques for making queries more efficient. It covers the following topics: describes the three join strategies, the features of the cost-based optimizer, algorithms used by the optimizer to find the best method to process the query and an assortment of techniques to help optimizer choose the fastest way to accomplish a query.



1. Uvod

V današnjem času smo priče vse hitrejšega tempa na področju razvoja novih računalniških tehnologij in produktov. Kljub pojavitvi objektne tehnologije, porazdeljenih objektov, ki delujejo preko Interneta in podobnih novosti, se nas še veliko ukvarja z ne-najnovejšo programsko in strojno opremo, ki jo skušamo kar se da najbolje izkoristiti. V taki situaciji sem tudi sama, ko pišem aplikacije, ki delujejo na relacijskih podatkovnih bazah. Vsaka pridobitev pri hitrosti izvajanja poizvedb po bazi nam je zelo pomembna in dragocena. V ta namen sem preučila delovanje Informixovega optimizatorja poizvedb SQL.

Informixov optimizator nam pomaga pri pisanju čimbolj optimalnih poizvedb. Poizvedbe naj bi bile optimalne glede na ceno izvajanja. V nadaljevanju so opisani postopki in tehnike za pohitritev poizvedb in povečanje učinkovitosti, ob predpostavki, da ne moremo spreminjati strukturo obstoječe podatkovne baze.

Kadar izvajamo ukaz SQL nad tabelo, na katero je postavljeno več indeksov, sistem Informix sam odloči, s katerim indeksom dostopa do podatkov. Če za neko poizvedbo ni primerne indeksa, uporablja zaporedni dostop. V sistemu Informix moramo zapisati posebne ukaze, če želimo videti, kako se je dejansko izvedla neka poizvedba. Lahko se nam zgodi, da za neko poizvedbo mislimo, da se v njej dostopa do po-

datkov z indeksom, v resnici pa se podatki pregledujejo zaporedno. Take anomalije nam skuša odpravljati optimizator.

2. Strategije izvajanja operacije stik

Stik (join) dveh tabel vsebuje informacije iz obeh tabel, glede na relacijo, ki povezuje en ali več stolpcev iz vsake tabele.

Rezultat te operacije so lahko vsi, ali pa samo nekateri stolpci tabel.

Stik se lahko izvaja tudi nad večimi tabelami. V takem primeru rezultat operacije vsebuje informacije iz več tabel, relacije pa povezujejo en ali več stolpcev posameznih tabel.

Če v poizvedbi zamenjamo vrstni red naštetih tabel ali relacije povezav in pogojev, lahko dobimo isti rezultat, hitrosti izvajanja pa so lahko zelo različne.

Če želimo poizvedbo napisati tako, da se bo izvajala optimalno, lahko uporabimo Informixov optimizator, ki pregleda vse možnosti in nam vrne optimalno napisan ukaz.

Primer:

Če Informixov optimizator pri izvajanju stika dveh tabel ugotovi, da je bolje zamenjati vrstni red tabel, se

stik izvede tako, da se najprej indeksno ali zaporedno pregleda druga tabela in se ji, glede na filter, poišče pripadajoč zapis prve tabele.

Sistem Informix pri izvajanju stika uporablja tri strategije:

- STIK Z VGNEZDENO ZANKO
- STIK Z UREJANJEM IN ZLIVANJEM
- STIK Z UPORABO RAZPRŠENE TABELE

2.1. Stik z vgnezdjeno zanko

Stik z vgnezdjeno zanko deluje tako, da se po nekem vrstnem redu pregleda prva tabela in se, glede na filter, poišče pripadajoč zapis iz druge tabele. Najprej se vzame prvi zapis iz prve tabele. Polja tega zapisa, ki se uporabljajo pri stiku, se imenujejo ključ zapisa. Potem se pregledujejo zapisi iz druge tabele in se primerjajo z vrednostjo ključa zapisa. Če je ključ zapisa druge tabele enoznačen, se pregledovanje konča, ko se najde prva ustrezna vrednost, se pravi, prvi zapis druge tabele, ki ima enako vrednost ključa, kot zapis prve tabele.

Če v poizvedbi nastopa še tretja tabela, se za vsak ustrezen zapis iz druge tabele išče pripadajoč zapis iz tretje tabele. Proces se nadaljuje, dokler ne pregledamo vseh navedenih tabel.

Vrstni red pregledovanja katerekoli tabele, ki nastopa v poizvedbi, je določen z indeksom, ki se uporablja. Če ustreznega indeksa ni, se iskanje izvaja zaporedno.

V splošnem velja, da se za vsak zapis iz prve tabele prebere vsaj en zapis iz druge tabele in iz vseh naslednjih tabel, razen, kadar ustreznega zapisa ni. Ker ni vsak prebrani zapis pravi, se v večini primerov prebere več zapisov, kot je potrebno.

2.2. Stik z urejanjem in zlivanjem

Tabele, ki nastopajo v operaciji stika, se uredi glede na vrednost filtra. Zapise različnih tabel se zliva, kadar so vrednosti stolpcev, glede na katere se dela stik, enake.

Primer:

Imamo dve tabeli z imeni B in C, na katerih niso postavljeni indeksi. Izvaja se naslednji ukaz SQL:

```
SELECT * FROM B,C
WHERE C.x = B.w AND
      C.z = 17
```

Stik z urejanjem in zlivanjem naredi naslednje:

- izberejo se vsi zapisi iz tabele C, ki zadoščajo filtru, to je, za katere velja $C.z = 17$,
- ti zapisi se uredijo glede na vrednosti $C.x$,
- zapisi tabele B se uredijo glede na vrednosti $B.w$,
- urejeni zapisi tabele B in urejeni zapisi tabele C se zlijejo skupaj, kadar velja, da je $C.x = B.w$.

2.3. Stik z uporabo razpršene tabele

Uporaba stika z razpršeno tabelo je primerna, kadar je potrebno eno tabelo pregledati zaporedno ali kadar imajo tabele veliko število zapisov. Glede na rezultat zgostitvene funkcije pregledanih zapisov te tabele se zgradi razpršena tabela. Zapise iz drugih tabel se skuša povezati z vrednostmi v razpršeni tabeli.

Če ni potrebe, da bi se ena od tabel, ki nastopajo v operaciji stika, pregledala zaporedno, se razpršena tabela ustvari za tabelo, ki ima najmanj zapisov.

Primer:

Izvaja se poizvedba (prikazana na naslednji sliki) nad tabelami z imeni tabB in tabC. Za tabelo z imenom tabB, se ustvari razpršena tabela.

Za vsak prebrani zapis tabele tabB zgostimo njegovo vsebino z izračunom zgostitvene funkcije. Glede na vrednost zgostitvene funkcije se zapisi uvrstijo v določeno skupino.

Označba $h(\text{zapis})$ predstavlja vrednost zgostitvene funkcije za nek zapis. Ta vrednost se vpiše v razpršeno tabelo, v polje z imenom glava skupine. V neki skupini zapisov se nahajajo vsi zapisi tabele, ki imajo enako vrednost zgostitvene funkcije. Zapisi znotraj skupine niso urejeni.

Po izgradnji razpršene tabele se berejo zapisi tabele z imenom tabC. Tudi zanje se izračuna vrednost zgostitvene funkcije, da se ve, v kateri skupini razpršene tabele je možno najti ustrezen zapis.

Razpršena tabela se ustvari v skupnem pomnilniku sistema Informix, katerega velikost lahko določimo sami, ob prvotni postavitvi sistema. V kolikor tega prostora ni dovolj, se del razpršene tabele zapiše na disk, kar upočasni vse operacije, ki delajo z njo.

Velikost razpršene tabele za neko tabelo lahko izračunamo po obrazcu:

velikost razpršene tabele v zlogih =
 $(32 \text{ zlogov} + \text{dolžina zapisa tabele}) * \text{število zapisov v tabeli}$

2.4. Slikovni prikaz delovanja različnih načinov izvedbe operacije stik

Glej slike 1A in 1B.

3. Vpliv delovanja optimizatorja

Delovanje Informixovega optimizatorja lahko vključimo v izvajanje ukazov SQL v celoti, samo delno, ali pa ga sploh ne vključimo. Vpliv delovanja optimizatorja na izvajanje ukazov SQL določimo s konfiguracijskim parametrom, ki se imenuje OPTCOMPIND.

Vrednost OPTCOMPIND pove, ali naj optimizator medsebojno primerja opisane strategije stika za posamezni dve tabeli.

OPTCOMPIND = 0,

pomeni, da naj optimizator poišče ustrezen indeks, s pomočjo katerega dostopa do podatkov, ko izvaja operacijo stik. Če indeksa ni, se izvede stik z uporabo razpršene tabele.

OPTCOMPIND = 1,

pomeni, da optimizator deluje na dva načina:

- če se izvaja transakcija nad podatkovno bazo, ki je transakcijsko zaščiten in se zapisi med izvajanjem operacije zaklenejo za ostale uporabnike, se optimizator obnaša kot v primeru, ko je OPTCOMPIND postavljen na vrednost 0.

sicer:

- optimizator izračuna najcenejšo pot dostopa do podatkov, glede na uporabo opisanih treh strategij stika.

OPTCOMPIND = 2,

optimizator v vsakem primeru izračuna najcenejšo pot dostopa do podatkov, glede na uporabo opisanih treh strategij stika.

Primer:

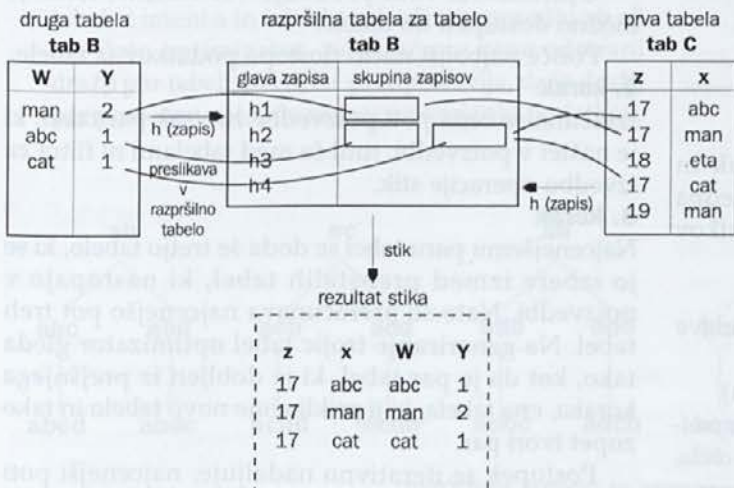
Imamo dve tabeli z imeni A in B ter indeks na tabelo B, na polje z imenom x.

Izvajamo ukaz SQL:

```
SELECT * FROM A,B WHERE A.x = B.x;
```

Optimizator ugotovi, da lahko izvede stik na dva načina. Prouči vrstni red tabel A, B in vrstni red tabel B, A. Glede na vrednost parametra OPTCOMPIND naredi sledeče:

- če velja OPTCOMPIND = 0: Izvede stik z vgnezdno zanko, z uporabo indeksa B.x v primeru, ko je vrstni red tabel A,B. Ko je vrstni red tabel B,A, uporabi stik z razpršeno tabelo.

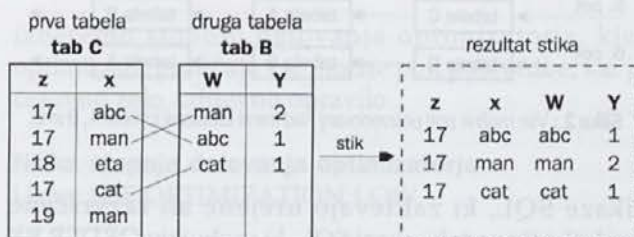


Slika 1B

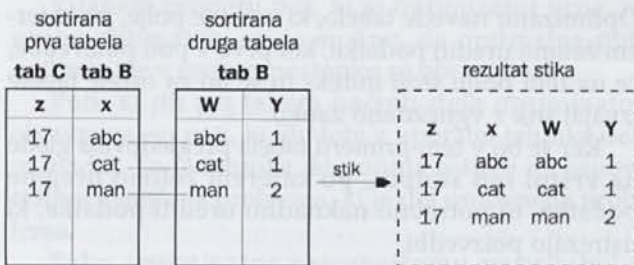
Izvajanje ukaza

```
SELECT * FROM tabB, tabC
WHERE tabC.x = tabB.w AND
tabC.z = 17
```

STIK Z VGNJEZDENO ZANKO



STIK S SORTIRANJEM IN ZLIVANJEM



Slika 1A

- če velja OPTCOMPIND = 2: Optimizator pri proučevanju vrstnega reda tabel B,A preučuje med možnostmi uporabe:
 - dinamično ustvarjenega indeksa na tabeli A,
 - stika z urejanjem in zlivanjem
 - stika z uporabo razpršene tabele.

3.1. Pot poizvedbe

Pot poizvedbe je neko zaporedje vseh tabel, ki nastopajo v ukazu SQL. Optimalna pot poizvedbe je neka permutacija vseh tabel, ki so našteje v ukazu SQL.

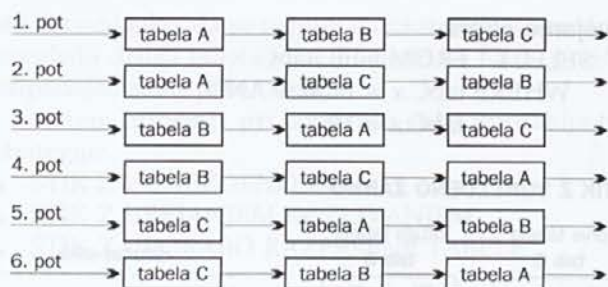
Optimizator izračuna ceno izvedbe ukaza SQL, glede na različno pot poizvedbe.

Primer:

Če v ukazu SQL nastopajo tri tabele, z imeni A,B in C, optimizator pregleda, kako bi se izvedel ukaz SQL, če bi si tabele sledile v naslednjem redu:

Pot poizvedbe vpliva na količino prebranih zapisov. Zato je na začetku poti poizvedbe priporočljivo imeti krajše tabele, ali pa tabele, na katere so postavljeni zelo omejujoči filtri (naprimer, post_stev = 1001). Na tak način se porabi manj časa za branje zapisov iz ostalih tabel, ki ne ustrezajo kriterijem poizvedovanja.

Z optimalno potjo poizvedbe se lahko izognemo postopku urejanja podatkov za



Slika 2.: Vse možne poti poizvedovanj nad tremi tabelami z imeni A, B in C.

ukaze SQL, ki zahtevajo urejene ali razvrščene podatke (to so taki ukazi SQL, ki vsebujejo ORDER BY ali GROUP BY).

Primer:

Optimizator navede tabelo, ki vsebuje polje, po katerem želimo urediti podatke, kot prvo v poti poizvedbe, če na tem polju stoji indeks in se bo za ostale tabele izvajal stik z vgnezdeno zanko.

Ker se bo v tem primeru tabela pregledovala glede na vrstni red stolpca, po katerem želimo urejene podatke, ni potrebno naknadno urediti podatke, ki ustrezajo poizvedbi.

3.2. Cena poizvedbe in optimalna pot

cena poizvedbe za posamezno pot poizvedbe = število dostopov do diska +

($W \cdot$ število procesiranih zapisov, ki ustrezajo poizvedbi)

ali zapisano drugače:

cena poizvedbe za posamezno pot poizvedbe = cena vhodno/izhodnih operacij +

($W \cdot$ cena delovanja centralne procesne enote)

W = faktor uteži, ki je trenutno 0.3; predstavlja razmerje cene delovanja centralne procesne enote proti ceni vhodno/izhodnih operacij.

Faktor uteži je določen s strojno opremo in njegove vrednosti ne moremo spreminjati.

Optimalna pot poizvedbe = najcenejša pot poizvedbe.

3.3. Cena urejanja podatkov

Postopek urejanja obremeni hitri pomnilnik in potrebuje tudi trdi disk. V kolikor neka poizvedba sproži urejanje podatkov, se cena urejanja podatkov prišteje v ceno poizvedbe.

cena urejanja =

cena obdelave v hitrem pomnilniku + cena obdelave na trdem disku

cena obdelave v hitrem pomnilniku = $c \cdot w \cdot n \log_2(n)$

c = število stolpcev zapisa iz tabele, ki jih je potrebno urediti, predstavlja ceno branja podatkov iz tabele in prepis podatkov v obliko, kjer so podatki urejeni glede na vrednosti ključa, po katerem se ureja

w = dolžina ključa, po katerem se ureja, od te dolžine je odvisno, koliko stane primerjava in prepis podatkov v urejeno obliko. Numerična vrednost tega podatka je odvisna od strojne opreme.

$n \log_2(n)$ = predstavlja potrebno število primerjav za urejanje tabele z n zapisi.

cena dela, ki ga opravi trdi disk = $2 \cdot n \cdot m$

m = število nivojev zlivanja, ki jih uporabi urejevalni algoritem.

Ta faktor zavisi od števila ključev, po katerih se ureja podatke.

Kadar se vse vrednosti ključev, po katerih se ureja podatke, lahko nahajajo v hitrem pomnilniku, je vrednost m enaka ena in tako je cena dela, ki ga opravi trdi disk enaka $2 \cdot n$.

Cena urejanja podatkov je odvisna od števila podatkov, ki jih urejamo.

Če ne moremo zmanjšati cene urejanja s tem, da urejamo manj podatkov, ceno lahko poskušamo zmanjšati tako, da urejamo po čim krajšem in enostavnejšem ključu, kar zmanjša faktorja c in w .

4. Optimizacija poizvedb

Proces optimizacije poizvedb je sestavljen iz treh korakov.

1. korak

To je postopek inicializacije, ki ga opravi optimizator, preden prične s postopkom izračunavanja cene poti.

Optimizator pregleda strukturo tabel, indeksov in filtra pri poizvedbi, proučuje nabor vrednosti zapisov, glede na filter in poskuša ugotoviti, kolikšna je propustnost filtra.

Propustnost filtra je število, ki ima lahko vrednost med 0 in 1. Enako je razmerju med številom zapisov v tabeli in delom zapisov, ki ustrezajo filtru.

Optimizator potem ugotavlja, če obstoječi indeksi na tabelah omogočajo dostop do podatkov na način, ki ga pričakuje postavljeni filter in če se obstoječi indeksi na tabelah lahko uporabijo v operacijah urejanja in razvrščanja podatkov.

Optimizator tudi proučuje vse načine, kako je možno dostopati do tabele.

Poišče najboljši način dostopa podatkov iz tabele.

2. korak

Izračuna se cena poti poizvedbe za vsak par tabel, ki je naštet v poizvedbi, tudi če med tabelami ni filtra za izvedbo operacije stik.

3. korak

Najcenejšemu paru tabel se doda še tretjo tabelo, ki se jo izbere izmed preostalih tabel, ki nastopajo v poizvedbi. Nato se preračunava najcenejšo pot treh tabel. Na generiranje trojic tabel optimizator gleda tako, kot da je par tabel, ki je dobljen iz prejšnjega koraka, ena tabela, ki ji priključuje novo tabelo in tako zopet tvori par.

Postopek se iterativno nadaljuje; najcenejši poti poizvedbe se doda še eno tabelo izmed preostalih tabel

in se zopet preračuna najcenejšo pot. Postopek se konča, ko so v pot poizvedb zajete vse tabele, ki nastopajo v poizvedbi.

Na koncu optimizator pogleda, če ukaz SQL vsebuje določilo ORDER BY ali GROUP BY, kar pomeni, da mora ukaz SQL vrniti urejene podatke. Določene poti so takšne, da se izognejo postopku urejanja. Za take poti, kjer je potrebno še naknadno urediti podatke, optimizator prišteje še ceno urejanja podatkov.

Optimizator po vsem tem izbere pot, ki ima najmanjšo ceno. Tako izbrana pot je optimalna pot poizvedbe.

4.1. Postopek izbire optimalne poti poizvedbe

Primer gradnje poti za neko poizvedbo:

Imamo štiri tabele z imeni a, b, c in d. Vsaka tabela ima tri stolpce, z imeni a, b in c. Izvajamo ukaz SQL:

```
SELECT *
FROM a,b,c,d
WHERE a.a = b.a AND
      b.b = c.b AND
      c.c = d.c
```

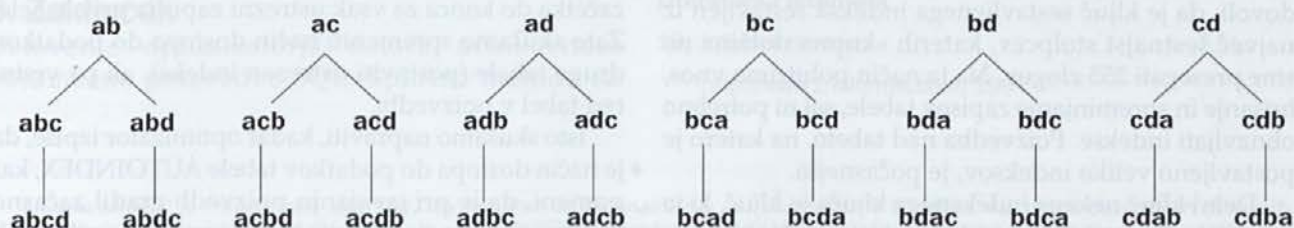
Možni pari tabel za opisan primer

ab ac ad ba bc bd ca cb cd da db dc

s črto so povezani pari tabel, kjer je en par tabel odvečen

Za vsak posamezen par tabel bo optimizator:

- poiskal najprimernejši način za izvedbo stika dveh tabel
- izbral najprimernejši način dostopa do podatkov za vsako tabelo
- izračunal ceno operacije stik
- iz postopka bo odstranil odvečen pare. Naprimer, tabeli z imeni a in b lahko združi v zaporedju ab ali ba. Zato optimizator po izračunu cene odstrani dražji par tabel, razen, če predpostavlja, da je dražji par potreben zato, da se izogne urejanju podatkov.



Nadaljevanje postopka po odstranitvi odvečnih parov tabel
(predpostavimo, da so ustrezni pari tabel tisti, ki so tudi po abecedi prvi)

4.2. Stopnje delovanja optimizatorja

Na nivoju izvajanja aplikacij lahko določimo, na kakšen način naj optimizator deluje.

Visoka stopnja delovanja optimizatorja

Z ukazom: SET OPTIMIZATION HIGH

izberemo stopnjo delovanja optimizatorja, kjer optimizator proučuje vse možne poti poizvedbe, kar je časovno zelo zahtevno opravilo.

Nizka stopnja delovanja optimizatorja

Ukaz: SET OPTIMIZATION LOW

pove optimizatorju, naj proučuje le določene poti poizvedb in izbere eno izmed njih.

V takem primeru pot, ki jo optimizator vrne, ni nujno optimalna, saj je možno, da optimalne poti optimizator v svojem postopku ni obravnaval.

Poti, ki jih pri takem načinu dela optimizator obravnava, so poti, ki jih dobi z uporabo tehnike deli in vladaj. To je tehnika, ki ponuja rešitev na osnovi rešitve manjšega problema, ki je del osnovnega problema.

Tako optimizator najcenejšemu paru tabel v postopku tvorjenja poti poizvedb doda preostale tabele, nato dodaja tabele najcenejši trojici. Na vsakem koraku postopek nadaljuje le za najcenejše zaporedje tabel, ki je hkrati že del končne dobljene poti.

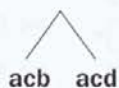
Kadar izberemo nizko stopnjo delovanja optimizatorja, je čas optimizacije poizvedbe (to je čas, ko optimizator proučuje poizvedbo) krajši, vendar se lahko poizvedba izvaja dalj časa, ker optimizator ni našel optimalne poti.

Primer gradnje poti poizvedbe pri izbiri nizke stopnje delovanja optimizatorja:

```
SELECT *
FROM a,b,c,d
WHERE a.a = b.a AND
      b.b = c.b AND
      c.c = d.c
```

Možni pari tabel in nadaljevanje postopka ob predpostavki, da je najcenejši par tabel ac in najcenejša trojica tabel acd

ab ac ad ba bc bd ca cb cd da db dc



acdb, to je pot, ki jo izbere optimizator

Izbira nizke stopnje delovanja optimizatorja je primerna, kadar so za nas pomembni odzivni časi poizvedb, ali, kadar je v poizvedovanju naštetih več kot pet tabel.

Nizka stopnja delovanja optimizatorja je najverjetneje primernejša od visoke stopnje delovanja optimizatorja, kadar imajo poizvedbe obliko:

- ena tabela je povezana z večimi drugimi tabelami (taki primeri so podatkovne blagovnice) ali
- v poizvedovanju je naštetih več tabel in vsaka naštetna tabela ima manj zapisov kot predhodno navedena, indeksi tabel pa so postavljeni na stolpce, preko katerih se povezujejo tabele.

V takih primerih je najverjetneje optimalna pot poizvedbe prav pot, ki jo optimizator najde tudi pri nizki stopnji delovanja.

V splošnem pa ni pravila, kdaj s katero izbiro pridemo do boljših rezultatov.

5. Pohitritev poizvedb

V splošnem velja, da je možno poizvedbe pohitrili:

- če nam uspe zmanjšati število zapisov, po katerih poizvedujemo,
- če se je možno izogniti postopku urejanja podatkov, ali če je možno izvesti urejanje po enostavnejšem ključu,
- če je možno dostopati do podatkov preko indeksa namesto zaporedno, ali, če so podatki primerno porazdeljeni po večih fragmentih in
- če se z uporabo sestavljenega indeksa ali iskanja preko delnega ključa lahko izognemo postavitvi večih indeksov na tabeli.

Sestavljen indeks je indeks, katerega ključ je sestavljen iz dveh ali več polj zapisa v tabeli. Sistem Informix dovoli, da je ključ sestavljenega indeksa sestavljen iz največ šestnajst stolpcev, katerih skupna dolžina ne sme presegati 255 zlogov. Na ta način pohitrimo vnos, brisanje in spreminjanje zapisov tabele, saj ni potrebno obnavljati indekse. Poizvedba nad tabelo, na katero je postavljeno veliko indeksov, je počasnejša.

Delni ključ nekega indeksnega ključa je ključ, ki je sestavljen iz prvih nekaj polj indeksnega ključa, ki so navedena v enakem vrstnem redu kot v indeksu. Preko indeksa se dostopa do podatkov tudi, kadar

poizvedujemo glede na vrednost njegovega delnega ključa. Nepotrebno je postavljati indeks na polja, ki so delni ključ nekega že obstoječega indeksa.

5.1. Postavitev testnega okolja, v katerem proučujemo poizvedbe

Izberemo poizvedbo, ki se nam zdi prepočasna in jo skušamo izvajati v takem okolju, kjer ponavljanje poizvedbe traja vedno enako časa. Edino tako bomo lahko proučili, ali bodo spremembe, ki jih bomo naredili kaj prispevale k hitrosti izvajanja poizvedbe.

Okolje, ki ni primerno za proučevanje poizvedb, je naprimer okolje, kjer so odzivni časi odvisni od delovanja virov, katerih obremenitev se kar naprej spreminja. Prav tako ni primerno proučevati poizvedbe znotraj nekega kompleksnega programa. V takem primeru skušamo poizvedbo izluščiti in jo izvajati interaktivno ali pa v okviru enostavnejšega programa.

Kadar imamo opravka s kompleksnimi poizvedbami, ki trajajo več ur, si za testiranje pripravimo bazo, v kateri so krajše tabele in na taki bazi izvajamo poizvedbe. Optimizator se v primerih majhnega števila zapisov lahko odloči za drugačno izvajanje, kot v primeru, kadar je teh zapisov veliko. V splošnem pa velja, da je čas izvajanja poizvedbe linearno odvisno od števila zapisov.

Tako lahko na osnovi časa izvajanja poizvedovanj v manjši testni bazi predvidevamo, koliko časa naj bi se izvajalo poizvedovanje v večji, pravi bazi.

5.2. Ponovno pisanje poizvedb, glede na izvajanje, ki ga izbere optimizator

Ko pregledamo, kako se je neka poizvedba izvedla, lahko ugotovimo, kako bi čas izvajanja poizvedbe pohitrili.

Nekaj primerov:

1. primer

Optimizator izpiše, da je pisal vmesne datoteke, s pomočjo katerih je kasneje izvajal urejanje podatkov. Število dostopov do diska je zato vsaj dvakrat večje, kot če se temu uspemo izogniti.

2. primer

Optimizator izpiše, da je do določene tabele, ki v poizvedbi nastopa kot druga, dostopal zaporedno, kar pomeni, da je to tabelo zaporedno pregledoval od začetka do konca za vsak ustrezeni zapis iz prve tabele. Zato skušamo spremeniti način dostopa do podatkov druge tabele (postaviti ustrezen indeks), ali pa vrstni red tabel v poizvedbi.

Isto skušamo napraviti, kadar optimizator izpiše, da je način dostopa do podatkov tabele AUTOINDEX, kar pomeni, da je pri izvajanju poizvedb gradil začasno indeksno drevo (tu je vključeno še urejanje indeksnih ključev), zato, da bi se izognil ponavljajočemu se zaporednemu dostopu do zapisov te tabele.

3. primer

Imamo tabelo in indekse:

```
table narocila
(
  st_narocila serial not null,
  st_stranke integer,
  ...
);
unique index o_num_ix on narocila (st_narocila);
index o_s_num_ix on narocila (st_stranke);
```

Poizvedba, ki povzroči zaporedni dostop do podatkov, kljub postavitvi indeksov:

```
SELECT * FROM narocila
WHERE (st_stranke > 1000 AND st_narocila < 1005)
OR st_narocila = 1008
```

To poizvedbo lahko pohitrimo na tak način, da jo pretvorimo v ukaz z uporabo združitve, kar pomeni, da se izvede poizvedba za posamezen del izraza, potem pa se dobljeni podatki združijo. Pri poizvedbi nad delom izraza se uporabljajo indeksi.

```
SELECT * FROM narocila
WHERE (st_stranke > 1000 AND st_narocila < 1005)
UNION
SELECT * FROM narocila
WHERE st_narocila = 1008
```

4. primer

Poizvedba:

```
SELECT * FROM stranka
WHERE koda[4,5] > "50"
```

pomeni, da želimo izpis tistih zapisov, kjer sta četrta in peti znak polja koda večja od 50. V takih primerih se ne dostopa do podatkov preko indeksa, tudi če indeks obstaja. Zato skušamo izraz zapisati z relacijskimi operatorji ali z uporabo izraza BETWEEN, na celotnem polju, kar omogoča indeksni dostop do podatkov.

Če tudi tako poizvedovanje ni hitrejše, je boljše poizvedovati na način:

```
SELECT * FROM stranka
IF koda[5] > "50" THEN ...
```

6. Zaključek

Delovanje optimizatorja določa učinkovitost posamezne poizvedbe SQL. Splošna metoda za

pohitritev poizvedb je torej poznavanje delovanja optimizatorja, ki se z uporabo ukaza SET EXPLAIN ON zapiše v datoteko. Nadalje je potrebno vedeti, katere operacije v izvajanju poizvedb so časovno potratne in kako se jim izognemo ali pa vsaj zmanjšamo njihov obseg.

Kadar nismo zadovoljni s hitrostjo izvajanja poizvedb neke aplikacije, uporaba optimizatorja ni prva stvar, na katero je potrebno pomisliti. Gledano na sistem s širšega vidika, je najprej potrebno skrbno proučiti nastali problem, razumeti delovanje aplikacije, ovrednotiti delovanje aplikacije z merili, poiskati ozko grlo delovanja, proučiti, koliko programov izvajamo, koliko računalnikov imamo na razpolago, koliko uporabnikov izvaja poizvedbe, itd. Vse to ima namreč lahko večji vpliv na učinkovitost izvajanja poizvedb, kot pa je prispevek optimizatorja. Kadar pa na take stvari nimamo vpliva, ali jih ne moremo spreminjati, nam optimizator lahko bistveno pomaga, zato je pomembno poznati njegovo delovanje.

Literatura

1. Advanced Programming in the UNIX Environment
W. Richard Stevens, Addison - Wesley Publishing Company 1992
2. INFORMIX - 4GL User Guide Database Tool
Informix Software, Inc., 1996
3. INFORMIX - Guide to SQL Tutorial
Informix Software, Inc., 1996
4. INFORMIX - NET and INFORMIX - STAR Configuration Guide
Informix Software, Inc., 1996
5. INFORMIX - OnLine Dynamic Server Databases
Informix Software, Inc., 1996
6. INFORMIX - OnLine Dynamic Server System Administration
Informix Software, Inc., 1996
7. IRIS InSight Library
Silicon Graphics OnLine Help, 1997
8. IRIS User's Man Pages
Silicon Graphics, Inc., 1994
9. Writing Solid Code
Stephen A. Maguire, Microsoft Press, 1993

Literatura z Interneta

10. Informix products
<http://www1.informix.com>, 1997

Lilijana Mihelič je magistrirala leta 1997 na Fakulteti za računalništvo in informatiko v Ljubljani. Na svojih dosedanjih delovnih mestih se je ukvarjala s tehnično podporo računalniškega omrežja, z administracijo Informix-ovega podatkovnega strežnika in z razvojem informacijskih sistemov. Trenutno je zaposlena v podjetju Telekom Slovenije, v Sektorju za informatiko, kjer sodeluje pri razvoju informacijskega sistema krajevnega omrežja, podprtega z grafiko.

TESTIRANJE PROGRAMOV - KAJ PA PODATKI?

Franc Žerdin

Kljub vsej naglici sodobnega poslovnega življenja je prav, da se od časa do časa ustavimo, se ozremo nazaj in ugotovimo, kaj smo in česa nismo storili na opazovanem področju.

Nekaj takega so storili uredniki revije Communications of ACM (april, 1997), ki so glavno temo (7 člankov) posvetili testiranju programov. Gostujoči urednik (M. Lieberman) uvodoma ugotavlja, da kljub vsemu napredku v zadnjih 30. letih - zmogljivejši in hitrejši računalniki, omrežja, grafični vmesniki in vse drugo - obstaja pri razvoju programske opreme nekaj neprijetnih, zoprnih dejstev, ki jih še vedno nismo odpravili. Večina teh zoprnih dejstev leti seveda na pomanjkljive tehnike testiranja in nezadostno uporabo avtomatskih orodij pri testiranju.

Tudi zagotavljanje zanesljivih in točnih podatkov je še vedno daleč od popolnosti. In vendar uporabniki niso preveč zainteresirani za razvoj programske opreme, dokler ta pravilno deluje, točnim in zanesljivim podatkom pa se nočejo in tudi ne morejo odpovedati. Pri obravnavanju velikih količin podatkov problemi ne le naraščajo, ampak postanejo kompleksnejši in težje obvladljivi. Pri tem obstaja še ena ovira, namreč, da so tovrstne izkušnje običajno zapisane le v internih poročilih posameznih družb.

Testiranje

Uporabniki še vedno prejemajo programsko opremo, ki ni temeljito stestirana (pravimo, da ima preveč uši ali hroščev). To povzroča pri uporabnikih nepotrebne frustracije in stroške, kar je prava nesreča. Vendar je še bolj presenetljivo dejstvo to, da programerji, ki so napisali program, dostikrat ne poznajo natančne poti, ki bi jim pokazala, kaj je bilo narobe, če je program odpovedal. Testiranje in odpravljanje programskih napak še vedno v pretežni meri uporablja metodo tipanja ("trial and error").

Glede na to koliko nejevolje, razočaranj in stroškov povzroča premalo preverjena programska oprema tako uporabnikom kot tudi razvojnemu osebju, je za računalniško znanost že pravi škandal (tako H. Lieberman), da je ta problem v tolikšni meri zanemarljiva. Sodobna programska orodja kot pomoč pri testiranju so le malo boljše kot tista izpred 30 let. Žalostno je ugotoviti, da še dandanašnji veliko programerjev navaja kot svojo tehniko testiranja ("debugging") "vstavljanje ukazov za tiskanje".

In vendar obstaja že veliko boljših načinov oz. metod, kako uspešno stestirati program. Vemo, da so programi kompleksni izdelki. Zakaj ne bi vključili računalnika tudi pri testiranju? Računalnik bo zlasti odigral aktivno vlogo pri razgrajevanju kompleksnosti. Računalniki so sedaj hitri, z velikim osrednjim pomnilnikom in velikimi zmogljivostmi na diskih. Zakaj ne bi uporabili teh prednosti, da bi programer prišel čimprej do informacij, ki mu bodo pomagale razumeti delovanje programa. Zakaj ne bi uporabili računalniške grafike, ki bi programerju predočila vedenje programa.

Veliko strokovnjakov dvomi v možnosti večjih izboljšav pri procesu testiranja programov. Menijo, da je to pač trdo delo. Mnogi programerji odklanjajo kakršnokoli orodje pri testiranju, češ dobri programerji tega ne potrebujejo. Resnica je drugačna. Dobra orodja lahko navidezno nedosegljivo napako (uš) prikažejo še tako izkušnemu programerju kot običajno in kot tako jo bo zlahka odpravil.

So tudi ljudje, ki zaupajo le formalnim tehnikam kot je programska verifikacija. Verjamejo, da tovrstne tehnike odpravljajo potrebo po testiranju. Vemo pa, da je nerealno

pričakovati popolno odpravo napak, čeprav se programska oprema razvija kar naprej, oziroma jo stalno dopolnjujejo. Orodja za testiranje so dobrodošla tudi pri dopolnjevanju programov.

Na podlagi ankete po internetu, ki jo je izvedel M. Eisenstadt v letih 1992 - 1993, je bilo ugotovljeno, da povzročajo več kot 50 % težav:

- velike časovne ali prostorske razlike med osnovnim vzrokom in začetnim simptomom
- napake, pri katerih testna orodja niso uporabna.

Anketa je tudi pokazala, da je skoraj polovico pomanjkljivosti (uši) povzročala dobaviteljeva strojna ali programska oprema. Nedavno smo v RRC dobili od enega uporabnika pisno sporočilo, da neki naši programi ne delujejo več pravilno. Analiza programov na drugi strojni opremi pa je pokazala nasprotno. Poklicali so vzdrževalce strojne opreme, ki so porabili več kot 40 ur, da so našli napako in sicer na strojni opremi (začeli so jo seveda spet iskati v programih). Na dobavo strojne opreme RRC seveda ne more vplivati. Kljub temu so še sedaj nekateri ljudje pri uporabniku prepričani, da je bila krivda v RRC, ker preprosto ne morejo verjeti, da lahko pride do napak v delovanju tudi zaradi dobaviteljeve opreme.

Kdor hoče program dobro stestirati, ga mora znati pregledati z različnih vidikov. Da bi omogočili večjo berljivost in razumevanje programov, so razvili različne tehnike. Med najbolj učinkovite gotovo sodi vizualizacija izvorne programske kode. Sicer pa je na boljše razumevanje programov opozoril D. Knuth v svojem članku "Literate programming" (Comput. J. 27,2,1984), kjer apelira na avtorje programov, naj začno gledati na svoje izdelke kot literarne in naj jih dokumentirajo na tak način, da jih bodo ljudje razumeli.

In podatki?

V nekaterih oddelkih za informatiko večjih družb pogosto naletimo na prepričanje, da so z uspešnim prevzemnim testom rešili vse probleme v zvezi s predvideno aplikacijo. Iz-

kušnje pa nas učijo, da povzročča uporaba vsakega orodja večje ali manjše težave. To velja še zlasti za programe, ki obravnavajo večje količine podatkov. Pri tem ne smemo pozabiti na kakovost podatkov.

Novije študije (D.M. Strong, Y.W. Lee and R.Y. Yang: Data Quality in Context, Comm. of ACM, May 1997 / Vol.40, No. 5) v zvezi s kakovostjo podatkov namreč kažejo, da tovrstni problemi naraščajo. Slaba kakovost podatkov (KP) v podatkovnih bazah je draga tako v socialnem kot tudi v ekonomskem smislu. Organizacijske podatkovne baze (PB) so shranjene v širšem kontekstu informacijskih sistemov (IS). V tem širšem kontekstu zbirajo podatke iz večjih podatkovnih virov in jih shranjujejo v PB. Iz tako zbranih podatkov oblikujejo informacije za sprejemanje poslovnih odločitev.

V tem širšem kontekstu IS lahko nastanejo kjerkoli problemi kakovosti podatkov, čeprav nekateri smatrajo KP kot naravno lastnost podatkov, ki naj ne bi bila odvisna od proizvodnje in uporabe podatkov. V literaturi o kakovosti pa velja pravilo, da kakovosti ne smemo ocenjevati neodvisno od uporabnikov. Podobno pravilo velja za KP, ki jo je treba ocenjevati preko okvirov naravnih lastnosti (ne moremo torej gledati le shranjenih podatkov, ampak je treba vključiti tudi podatke v proizvodnih in uporabniških procesih) in pri tem upoštevati ljudi - uporabnike podatkov. Njihova ocena KP postaja vse pomembnejša, ker imajo vedno boljše kontrolo nad računalniškim okoljem in predvsem nad podatki, ki jih uporabljajo.

Da bi lahko kakovost podatkov bolje opredelili, jo razvrščajo v več kategorij:

- naravne lastnosti KP (točnost, objektivnost, možnost, uveljavljenost)
 - dostopnost KP (dostopnost, varnost dostopa)
 - povezanost (kontekstualnost) KP (relevantnost, dodana vrednost, pravočasnost, popolnost, količina podatkov).
- Običajne tehnike kontrole (n.pr. preverjanje, omejitve integritete PB, programska kontrola ažuriranja PB), ki zagotavljajo KP, so znatno izboljšale točnost podatkov. Vendar to ne

zadošča, kakor tudi ne sama kontrola nad shranjevanjem podatkov. Vključiti je treba revizijske tehnike, ki lahko nadzirajo procese, kjer nastajajo podatki.

Tudi dostopnost KP se razlikuje v tehničnem oz. uporabniškem smislu. Konvencionalni pristop obravnava dostopnost kot tehnično, računalniško zadevo. To pomeni, da so podatkovni skrbniki poskrbeli za dostop, če so strežniki povezani z uporabniki in delujejo, če ima uporabnik dovoljenje za uporabo podatkov itn. Uporabniki podatkov hočejo več, namreč enostavno manipulacijo s podatki, da zadostijo svojim potrebam. Tako so podatki iz posameznih avtonomnih sistemov tehnično dostopni, za uporabnike pa niso dostopni, ker so vrednosti enakih podatkovnih postavk različno definirane ali predstavljene. Velike količine podatkov so lahko tehnično dostopne, ne pa za uporabnike, ker so dostopni časi predolgi. Informatiki morajo razlikovati med tehnično dostopnostjo do podatkov, kar je njihova naloga, in med širšim pojmom dostopnosti, kakor jo pojmujejo uporabniki. Čim smo dojeli to razliko, lahko uporabimo druge tehnologije, n.pr. podatkovno skladiščenje, ki lahko oskrbi uporabnika z manjšo količino bolj relevantnih podatkov.

Uporabniki podatkov vrednotijo KP relativno glede na svoje naloge. Pri tem se značilnosti KP lahko s časom spreminjajo. Zato je razumljiva zahteva, da morajo zasnove zagotavljati prožne sisteme, ki vzpostavljajo kakovostne podatke (n.pr. enostavna manipulacija in združevanje podatkov), ker je sicer alternativa stalno vzdrževanje podatkov in sistemov.

Ugotovili smo, da je kakovost podatkov skrb tako informatikov kot tudi uporabnikov. Informatiki so dolžni zagotoviti tehnične pogoje, uporabniki pa morajo dobro poznati druge lastnosti KP, da ne prihaja do okvar podatkovnih baz. Kadar pa kreirajo ali ažurirajo PB pretežno s pomočjo javnih evidenc, je kakovost podatkov še posebej pomembna. Zato je nujno potrebno sodelovanje med informatiki in uporabniki, če hočejo proces zagotavljanja kakovosti čim bolj avtomatizirati.




Svoj izvod dobite na naslovu:

Kolapš, Kardeljeva ploščad 17, 1000 Ljubljana
 kolaps@uni-lj.si
<http://www.pasef.ef.uni-lj.si/kolaps/>

Iz zadnje številke:

Dosje: ZATON NACIONALNE DRŽAVE

dr. Frane Adam: NACIONALNO V KONTEKSTU GLOBALIZACIJE
 intervju: Milan Kučan, dr. Edmund Farhat in NSK

Tadej Mali: EKSPERIMENT ZA NOVO TISOČLETJE

dr. Ivan Ribnikar vs. Bernard Brščič: DRUGI STEBER IN PRIVATIZACIJSKA LUKNJA

Aljoša Valentinčič: DOGODKOVNE ŠTUDIJE

Matjaž Jager: NEKAJ O ENAKOSTI MED LJUDMI

Monika Osvald: ANDY WARHOL IN OSEMDESETA

Mark Sagoff: ŽIVALI KOT IZNAJDBE

Kolapš
 ECONOMIA - PHILOSOPHIA - IUS - SCIENTIA

Konference: takšne in drugačne

Delo na vsakem strokovnem področju zahteva nenehno dopolnjevanje znanja in bogatenje izkušenj. Ker smo kot posamezniki pri dopolnjevanju znanja izpostavljeni vsaj dvema ključnima omejitvama, to sta čas in rutinske obveznosti, poskušamo znanje in izkušnje poglobljati tudi v obliki druženja z ljudmi, ki rešujejo sorodne probleme kot mi sami. Tako se udeležujemo različnih konferenc in strokovnih srečanj. Vsaj štiri motivi izstopajo pri večini udeležencev konferenc:

- izpopolnjevanje znanja (izogibamo se zastaranju strokovnega znanja - "adaptivno vzdrževanje znanja"),
- izmenjava izkušenj (učni primeri, ki temeljijo na izkušnjah drugih, mnogokrat preprečijo nepotrebno preizkušanje alternativ za rešitev problema, ki ga imamo pred seboj),
- razširjanje obzorja (informatika zahteva multidisciplinaren pristop k reševanju problemov, saj je le izjemoma "sama sebi namen"),
- druženje (osebni stiki so kljub globalizaciji z uporabo informacijske tehnologije gonilo sodelovanja in nastopajo kot vzpodbujevalec vsebinskih motivov).

V grobem konference, simpozije in delavnice (v nadaljevanju jih bomo imenovali srečanja) po njihovi naravi delimo na znanstvene in strokovne (razpisi Ministrstva za znanost in tehnologijo Republike Slovenije govorijo o sofinanciranju *znanstvenih in strokovnih sestankov*). Tako kot vsaka delitev, tudi ta nosi v sebi negativen naboj hierarhičnega vrednotenja, ki ga pogojujeta zorni kot in naravnost opazovalca. Temu se bomo v pričujočem razmišljanju izognili in bomo pozornost usmerili v razmišljanje o optimalnem pristopu pri odločanju glede udeležbe na srečanjih, kar bomo podkrepili z opisom dveh pred kratkim izvedenih srečanj.

Udeležence srečanj lahko v grobem razdelimo v naslednje skupine: (a) akademski krog (izmenjuje izkušnje, se potrjuje), (b) strokovni krog, ki sprejema znanje in izmenjuje izkušnje, (c) strokovni krog, ki ponuja znanje in storitve in (d) opazovalci (mednje spadajo posamezniki, ki trenutno še ne vidijo specifičnega interesa na strokovnem področju, vendar slutijo njegovo morebitno aktualnost v bližnji prihodnosti). Pričakovanja posameznih skupin so ponekod diametralno nasprotna, kljub temu pa zgolj njihova "simbioza" lahko privede do uspehov pri vsestranskem prenosu znanja.

Ne glede na vrsto srečanja, njegov obseg po številu udeležencev, kakovosti referatov ali količini komercialnih predstavitev, prihaja skoraj povsod do naslednjih problemov, ki zmanjšujejo učinkovitost srečanja:

- urniki so preobremenjeni in če seje potekajo vzporedno, mora udeleženec izbrati predavanja na podlagi naslova referata ali imena predavatelja, saj je zbornik (večinoma) dobil šele pri registraciji istega ali prejšnjega dne,
- moderatorji posameznih sej so pogosto neučinkoviti; s snovjo in z nastopajočimi niso vnaprej seznanjeni, so

zgolj v vlogi napovedovalca ali pa preidejo v drugo skrajnost ter z interpretacijo in povzemanjem prenašajo pozornost s predavatelja nase,

- po zadnjem dnevu srečanja se izgubi stik med udeleženci; ohranijo se dopisovalni sezname, kar pa je premalo za kontinuiteto srečanja; izkušnje kažejo, da se ohranjajo tudi le redki osebni stiki, ki niso neposredno in kratkoročno interesno obarvani.

Kratka analiza dveh različnih konferenc

Nadaljevale te razmišljanja je subjektivno obarvano. V njem bom poskušal osvetliti dve srečanja, ki se ju nisem udeležil kot predavatelj. Razmišljanje je manj usmerjeno v vsebino dogodkov, temveč poskuša opredeliti vplive na posameznika - udeleženca srečanj, ki mora nenazadnje racionalno razpolagati s časom in sredstvi pri izpopolnjevanju znanja.

Simpozij GartnerGroup Itxpo97 (Cannes, Francija, 10.-13. november)

Organizatorji vsakega uspešnega srečanja morajo imeti pred očmi ciljni avditorij. Svetovalna firma GartnerGroup s tem seveda nima nobenih težav. Že podatki o kotizaciji in krajih, kjer se dogodki odvijajo, povedo, da je simpozij namenjen najvišjim vodstvenim strukturam, ki so usmerjevalci razvoja informacijske tehnologije v velikih sistemih, kot so npr. banke, zavarovalnice, telekomunikacijske firme in državna uprava. Simpozij je bil mozaik, sestavljen iz nastopov predsednikov vodilnih družb s področja informacijske tehnologije, predavanj, ki so jih izvajali analitiki GartnerGroup in individualnih srečanj tipa "one-to-one" med udeleženci in analitiki. Kot ključni kratkoročni izzivi so bili izpostavljeni: problem leta 2000, uvajanje omrežnega računalništva ter varovanje in zaščita podatkov v globalnih komunikacijskih sistemih.

Kot udeleženec lahko izpostavim naslednje pridobitve tovrstnega simpozija:

- V relativno kratkem času in na enem mestu si lahko bistveno razširimo obzorje z vpogledom v obstoječe tehnologije in oceno kratkoročnih smernic razvoja.
- Informacije so podane večplastno - pregledne predstavitev omogočajo spoznavanje neznane snovi oz. tehnologij, analitično poglobljene predstavitev in osebno svetovanje pa ponudita izhodiščne informacije in zadovoljivo vzpodbudo za poglobljeno obravnavo snovi oz. tehnologij.
- Udeleženec lahko ob poznavanju *lastnih potreb* pridobi natančne napotke za izbiro informacijske tehnologije v obliki primerjalnih analiz. Ob obilici ponujenih podatkov, analiz, izkušenj in nasvetov ostajata vsaj dve odprti vprašanju:
- Kako zapolniti manjkajoče tehnološko znanje, ki je potrebno pri odločanju in kasneje pri uvajanju sprememb in nove tehnologije?

- Kako oceniti tveganje, kadar zaupamo svetovalcu-analitično (paradoks o namembnosti svetovalca)?

Organizacijski vidik simpozija Itxpo'97 je bil na zavidljivi ravni, saj je vsa logistika v zvezi s 3300 udeleženci potekala brezhibno. S približno petdesetimi delovnimi postajami - informacijskimi stebri - je bilo poskrbljeno tudi za elektronsko komuniciranje ne le s svetom, temveč tudi med samimi udeleženci ter med udeleženci in izvajalci. Dopolnilne informacije o simpoziju Itxpo'97 so dosegljive na naslednjem naslovu na Internetu: <http://www.gartner.com/>

Mednarodna konferenca o prenovitvi informacijskih sistemov ReTIS'97

(Celovec, Avstrija, 4. in 5. december)

Usklajevanje med primerno znanstveno ravni srečanja in med njeno privlačnostjo za širši krog strokovnjakov je večna dilema organizatorjev srečanj. Zaključene znanstvene debate se običajno odvijajo na meduniverzitetnih kolokvijih, ki so mnogokrat nezanimivi za širšo javnost. Na področju informatike tudi redkokdaj dosežejo tradicionalnost, saj se, če so uspešni, prelevijo v srečanje odprtega tipa ali inicirajo razvojno - raziskovalne projekte. Mednarodna konferenca o prenovitvi informacijskih sistemov ReTIS (Re-Techniques for Information Systems) ima zanimivo preteklost. Nastala je kot združitev dveh vzporednih dejavnosti: (a) prva je bila izkustveno obarvano srečanje o uporabi orodij za računalniško podprto programsko inženirstvo (CASE) za razvoj novih in prenovitev obstoječih informacijskih sistemov, ki sta ga prvič leta 1991 priredila Laboratorija za informatiko na obeh slovenskih univerzah ob takratni izdatni tudi vsebinski podpori firme SRC in nekaterih drugih sponzorjev in (b) druga dejavnost pa je bila sodelovanje med univerzama v Mariboru in avstrijskem Celovcu v obliki teoretičnih kolokvijev in izmenjave podiplomskih študentov.

Svoje videnje prednosti, slabosti in perspektiv ReTIS-a bom zapisal kot sourednik spremljevalne knjižne izdaje referatov in kot eden izmed članov programskega sveta. Konferenca je letos dokončno prerasla slovensko - avstrijski okvir in postala mednarodna z udeleženci iz mnogih tudi izvenevropskih držav (Japonska, ZDA, Kitajska/Hong Kong in Brazilija). Vsebinsko je bil letos ReTIS usmerjen v *računalniško podprto pomoč pri vzdrževanju* informacijskih sistemov. Pokazalo se je, da obstaja mnogo delnih rešitev (npr. sistemi za podporo sodelovanja z uporabniki ("helpdesk"), translatorji podatkovnih shem, analizatorji kode, orodja za dokumentiranje), celostna podpora vzdrževanju pa je še vedno v veliki meri odvisna od kakovosti, to je popolnosti in ažurnosti, razvojne dokumentacije sistema, ki je predmet vzdrževanja. Razprave so bile v veliki meri usmerjene k migraciji informacijskih sistemov iz zaključenih okolij na Internet in k problemom varovanja in zaščite, ki sta ob problemu tehnološke izvedbe migracije najbolj pomembna.

Sklepne misli o uspešni kontinuiteti konference ReTIS je potrebno dopolniti tudi s problemom, ki je nastal z dvigom

znanstvene ravni konference, pri čemer je bila zaradi organizacijskih razlogov (omejenost virov) posvečena premajhna pozornost praktičnim predstavitev. Udeležencev iz industrije je bilo premalo, marsikaterega udeleženca iz Slovenije, ki je bil prisoten prejšnja leta, je po individualnih pogovorih zadržal preprost razlog - Karavanke, kar je paradoks, saj je bila z vidika vsebine pri istih sogovornikih letošnja konferenca ocenjena kot aktualnejša in bogatejša od prejšnjih izvedb.

Program konference, informacije o knjižni izdaji predstavljenih referatov in podatke o konferenci ReTIS'99, ki bo v italijanskem Salernu, lahko najdemo na Internetu: <http://lisa.uni-mb.si/conf/retis97/>. Na tem naslovu je zapisan tudi nagovor predsednika Slovenskega društva Informatika, ki je na konferenci izpostavil probleme integracije akademskega in praktičnega znanja ter mednarodnega povezovanja.

Zaključno razmišljanje

Konference bodo tudi v obdobju globalnih komunikacij nedvomno preživele, saj omogočajo neposredno navezovanje in ohranjanje stikov. Vpliv globalnega komuniciranja je lahko zelo pozitiven v smislu omogočanja temeljite priprave srečanj in ohranjanja njihove kontinuitete tudi po končanem dogodku. Vzpostavitev konferenčnega sistema za programski svet in spletna predstavitev za udeležence z možnostjo dialoga sta temeljni značilnosti, ki srečanju dajeta veliko mero privlačnosti in hkrati transparentnosti. Vsakoletna konferenca Dnevi slovenske informatike v tem na žalost zaostaja kljub pobudam, ki so jih udeleženci v zadnjih letih podajali na okroglih mizah in tudi v okviru samega društva Informatika, ki konferenco organizira. Večja agilnost društva na področju računalniško posredovane komunikacije s članstvom bi pripomogla k prirastku števila članov in k njihovemu aktiviranju ter k odpravljanju občutka o relativno zaprtem krogu moderatorjev društva.

Srečanj v tujini se udeležuje mnogo slovenskih strokovnjakov. Na ta način se vzpostavlja zahtevnejši avditorij tudi za srečanja, ki jih prirejamo v Sloveniji. Prirediteljem srečanj jemlje veliko truda potreba po nenehnem usklajevanju med strokovno-znanstveno poglobljenostjo in pozitivnim finančnim izidom prireditve, saj so dotacije Ministrstva za znanost in tehnologijo ter posameznih strokovnih združenj dokaj simbolične.

Prireditelji srečanj pogosto pozabljajo (mo) na javno promoviranje prireditve in se zadovoljijo z izjavo za javnost, objavljeno na četrti strani dnevnega časopisa v desnem spodnjem kotu. Srečanja bodo opravila svoje poslanstvo šele tedaj, kadar bodo tudi širši strokovni in "laični javnosti" (vedno bolj ozaveščenim državljanom informacijske družbe) večplastno predstavljeni dosežki ter pojasnjeni izpostavljeni problemi, kar pa pomeni, da znanstveno ali strokovno srečanje ne sme biti zgolj dogodek, temveč mora biti proces.

József Györköš

Konferenca Gartner Group

6. - 10. oktober 1997, Orlando, Florida, ZDA

Za svetovalsko družbo Gartner Group v Sloveniji nasploh in informatiki posebej do nedavnega niti nismo dobro vedeli, še manj pa za to, da vsako leto prireja kar nekaj dogodkov, med njimi tudi konferenco o trendih v informatiki. Šele nedavno tega se je nekaj domačih podjetij potrudilo razširiti o njej malo več informacij in eno od njih je tudi uradni zastopnik Gartner Group za Slovenijo. Naj navedemo le podatek, da se je konference v Orlando udeležilo več kot 7.500 poslušalcev. Že samo to dokazuje, da je v resnici pomemben svetovni dogodek informatike. Pomanjšana, vendar tudi pomembna konferenca je ponovitev ameriške v Evropi.

Gartner Group meri, ocenjuje, vrednoti in napoveduje prav kakor obljublja naslov konference. To počne na urejen in celo standardiziran način, kar se da videti že iz zgradbe predavanj. Vsa so urejena enako, enak je tudi raspored vsebine. Avtorji prispevkov dajo za napovedi tudi oceno verjetnosti uresničitve napovedi in čas, v katerem naj bi se to zgodilo. Vprašanje je seveda, ali je uresničitve napovedi posledica tega, da je bila napoved objavljena in da zato vpliva na obnašanje, ali gre za inteligentno uporabo modela napovedovanja in razpoložljivih podatkov ali pa zgolj za pravo mero zdrave pameti v kombinaciji s poznavanjem področja. Tisti, ki spremljajo napovedi dlje časa, so se lahko prepričali, da je dostop do informacije Gartner Group vreden svoje cene. Najmanj, kar lahko rečemo, je, da moramo vedeti, kam se obračajo vodilne družbe informacijske tehnologije in vodilna država informatike. Posebej zanimivo je, da Gartner Group na konferenci obravnava tudi informatiko v državni upravi, kar je na konferencah s področja informatike izjemno redko.

Skoraj nemogoče je podati popoln pregled o enem tednu dogajanja na konferenci, ki je potekala hkrati na treh do petih lokacijah in v do sedmih vzporednih sekcijah. Začetek je bila predkonferenca v nedeljo. Zbornik konference obsega 6 zvezkov formata A4 na skupaj nekaj tisoč straneh, ki merijo v višino 15 cm in tehtajo 6,70 kilograma. Ti podatki ilustrirajo obsežnost informacij, poudariti pa moramo, da količina ni mašilo in nadomestilo za vsebinsko in strokovno vrednost. Na srečo je vse zapisano tudi na laserski plošči, ki jo prejmejo udeleženci konference poleg natisnjenih gradiv. Ob konferenci so bile priložnostne prireditve: razstava informacijske tehnologije, predavanja ponudnikov ter razgovori s predstavniki Gartner Group in vodilnih podjetij s področja informacijske tehnologije in komunikacij. Udeležencem je bilo na razpolago blizu 100 v internet vključenih delovnih postaj, kjer so lahko odprli svoj poštni predal za elektronsko pošto, izhajal je tudi priložnostni dnevnik konference. V nadaljevanju naštevamo nekaj vročih tem, ki so jih podrobno predstavili večinoma vodilni zaposleni in vrhunski strokovnjaki Gartner Group in nekateri vabljeni predavatelji.

Internet - ključ za nove priložnosti

Sedanje telekomunikacije so za vse, kar je mogoče pošiljati preko interneta - podatki, glas, grafika in živa slika - prepočasne in njihove pasovne širine so preozke, vendar pričakovanja družb in posameznikov upravljavce komunikacij silijo v to, da nudijo vedno višjo tehnično raven storitev. Po drugi strani družbe internet obravnavajo kot element poslovnih strategij. Internet še ni nekaj stabilnega in tudi zato ga večina uporabnikov informatiziranih procesov ne razume prav dobro.

Elektronsko poslovanje

Elektronsko poslovanje je način poslovanja, ki se je začel že precej pred prihodom interneta, vendar je šele zdaj doživel neverjeten razmah. Za uspešno uvedbo mora obstajati strategija: obstajati mora cilj, izvedljiv način, kako ga doseči in dodeljeni morajo biti ustrezni viri za izvedbo.

Globalne telekomunikacije

Devetinšestdeset vlad od sto štiridesetih držav članic Mednarodne trgovne organizacije (World Trade Organization), med njimi seveda tudi ameriška, je februarja 1997 dalo soglasje k novi ureditvi telekomunikacijskih sporazumov, ki dovoljujejo več domače in tuje konkurence, več tujih vlaganj in zdaj prvič tudi tuje lastništvo telekomunikacijske infrastrukture, ki je bila do pred kratkim v glavnem monopol domačih podjetij. To bo dramatično vplivalo na politiko cen večine telekomov.

Intranet

Poenostavljeno bi lahko rekli, da je intranet uporaba novega medija znotraj organizacije. Navznoter omogoča nove načine delovanja in povečanje učinkovitosti. Lokalna mreža je potreben tehnični pogoj, vendar sama po sebi še ni intranet, če ni povezana z internetom.

Omrežni računalnik

Le zakaj ne bi imeli aplikacij in pomnilnikov na močnem strežniku? Pridrček do omrežnega računalnika je bolj čustvene narave, ker na videz oživilja koncept centralizirane obdelave podatkov - močnega računalniškega sistema in terminalov. Gre pa za dosti več, namreč za koncept stranka - strežnik v kombinaciji z v Javi napisanimi programi, ki se lahko izvajajo na kateremkoli računalniku pod kateremkoli operacijskim sistemom.

Oddajanje del

Za oddajanje del nasploh in tudi v informatiki je mnogo razlogov, glavni pa so ekonomski: konkurenčnost, razširitev ponudbe ob minimalnih stroških in zmanjšanje stroškov poslovanja. Praktično ni storitve, ki je ne bi mogli ponuditi specializirani izvajalci pod ugodnejšimi pogoji kakor domači zaposleni. Storitve, ki jih ne bi smeli oddati, pa so tiste, ki zadevajo ključne funkcije organizacije. Razen tega je postalo jasno da je oddajanje del praktično edina pot do poceni države. Prava strategija je očitno ta, da se izločijo vse funkcije, ki niso državi lastne.

Leto 2000 in evro

Presenetljivo je, da Gartner Group leto 2000 in evro, ki sta na prvi pogled drug od drugega neodvisna, obravnava tako rekoč v paketu kakor neke vrste neizogibno naravno katastrofo. Leto 2000 je potencialni problem za praktično vsa podjetja, ki imajo informatizirano katerokoli od svojih funkcij. Razumeti je treba, da je problem leta 2000 problem obstoja organizacije in s tem predvsem problem vodstva. Priprava na leto 2000 je za vse organizacije potreba: za majhne je vprašanje obstoja, za velike pa tudi moralna dolžnost, da svojega morebitnega nereda ne bi izvozile. Evro označujejo za kar apokaliptičen dogodek, ki se bo predvidoma zgodil v letu 2000. Priprave na evro naj bi potekale projektno in podobno kot priprave na leto 2000 in kdaj.

Niko Schlamberger

Vabilo k sodelovanju

Univerza v Ljubljani, Visoka upravna šola,
Ministrstvo za notranje zadeve in
Služba Vlade za lokalno samoupravo

organizirajo tradicionalno posvetovanje 'Dnevi slovenske uprave 98', ki bo letos osredotočeno na temo
'Reforma slovenske javne uprave in EU'.

Posvetovanje bo v Portorožu od 24-26 septembra 1998. K sodelovanju vabimo vse, ki bi želeli s svojo aktivno udeležbo, v obliki pisnega prispevka, prispevati k vsebinski obogatitvi srečanja ter sooblikovati nadaljni razvoj slovenske javne uprave. Pri izboru prispevkov bo programski odbor dal prednost predvsem (ne pa izključno) naslednjim temam:

- Razvoj slovenske javne uprave z vidika vključevanja Slovenije v EU
- Institucionalna in organizacijska preobrazba javne uprave
- Proračunska poraba in finančna decentralizacija uprave
- Davčna reforma: vsebinski in operativni vidiki
- Standardizacija javnih storitev
- Ekonomičnost, učinkovitost ter efekti javnih storitev
- Upravljanje javnega sektorja
- Prenova poslovanja ter informatizacija javne uprave
- Kakovost javnih storitev, standardi kakovosti, nagrade za kakovost
- Razmerja med lokalno upravo in samoupravo
- Razmerja med organi občine, naloge in financiranje občin
- Teritorialne spremembe v mreži občin ter oblikovanje širših lokalnih skupnosti

Pomembni datumi:

- 10.5.1998 rok za oddajo povzetkov na 1-2 straneh
- 10.6.1998 obvestila avtorjem o izboru prispevkov
- 15.8.1998 rok za oddajo prispevkov v obsegu 8-10 strani

Vse zainteresirane za sodelovanje prosimo, da pošljejo povzetke svojih prispevkov na naslov:

Visoka upravna šola
Kardeljeva pl. 5, 1000 Ljubljana
tel. 341-972
fax.168 62 04

pod oznako 'Dnevi slovenske uprave 98'. Vse podrobnejše informacije dobite pri g. Darinki Marguč tel. 341- 763

Predsednik programskega odbora:

Dr. Mirko Vintar

INTERNET ■ INTERNET ■ INTERNET ■ INTERNET ■ INTERNET ■ INTERNET

Vse člane in bralce revije obveščamo,
da lahko najdete domačo stran društva na naslovu:

<http://www.drustvo-informatika.si>

Za predloge in pripombe v zvezi z vsebino se priporočamo na naslov:

Niko.Schlamberger@stat.sigov.mail.si

INTERNET ■ INTERNET ■ INTERNET ■ INTERNET ■ INTERNET ■ INTERNET

DNEVI SLOVENSKE INFORMATIKE DSI '98

Z N A N O S T • S T R O K A • P O D J E T N I Š T V O

Kongresni center Grand hotel Emona, Portorož – 6. do 9. maj 1998

Okvirni program posvetovanja DSI '98:

torek 5. maj

15:00 - 18:00 prijava, recepcija

sreda 6. maj

08:00 - 18:00 prijava, recepcija

08:30 - 10:00 predkonferenca: Kaj je novega, Oracle?

10:00 - 10:30 odmor

10:30 - 12:00 Kaj je novega, Oracle? (nadaljevanje)

12:00 - 13:00 kosilo

13:00 - 15:30 Otvoritev posvetovanja

Nagovor predsednika ZRIS

Nagovor predsednika DSI

Častni govornik: rektor ljubljanske univerze

prof. dr. Jože Mencinger

Podelitev društvenih priznanj

15:30 - 16:00 odmor

16:00 - 17:30 Vabljeni referati (vodja Niko Schlamberger)

17:30 OKROGLA MIZA I:

Stroka kot most med znanostjo in podjetništvom

Oraclov večer

četrtek 7. maj

09:00 - 18:00 prijava, recepcija

09:00 - 10:30 SEKCIJA A (vodja Ivan Rozman)

09:00 - 10:30 SEKCIJA B (vodja Franc Žerdin)

09:00 - 10:30 SEKCIJA C (vodja Andrej Kovačič)

10:30 - 11:00 odmor

11:00 - 12:30 SEKCIJA A (nadaljevanje)

11:00 - 12:30 SEKCIJA B (nadaljevanje)

11:00 - 12:30 SEKCIJA C (nadaljevanje)

12:30 - 14:00 kosilo

14:00 - 15:30 SEKCIJA A (nadaljevanje)

14:00 - 15:30 SEKCIJA C (nadaljevanje)

14:00 - 15:30 SEKCIJA D (vodja Ivan Vezočnik)

15:30 - 16:00 odmor

16:00 - 17:30 OKROGLA MIZA 2:

Elektronsko poslovanje in povezovanje s svetom

Družabnosti

petek 8. maj

09:00 - 16:00 prijava, recepcija

09:00 - 10:30 SEKCIJA C (nadaljevanje)

09:00 - 10:30 SEKCIJA D (nadaljevanje)

09:00 - 10:30 SEKCIJA E (vodja Vladislav Rajkovič)

10:30 - 11:00 odmor

11:00 - 12:30 SEKCIJA C (nadaljevanje)

11:00 - 12:30 SEKCIJA D (nadaljevanje)

11:00 - 12:30 SEKCIJA E (nadaljevanje)

12:30 - 14:00 kosilo

14:00 - 15:30 SEKCIJA D (nadaljevanje)

14:00 - 15:30 SEKCIJA E (nadaljevanje)

15:30 - 16:00 odmor

16:00 - 17:30 OKROGLA MIZA 3:

Informatizacija in prenova poslovanja

16:00 - 17:30 DELAVNICA: Ocenjevanje programskih produktov (vodja Marjan Pivka)

Družabnosti

Razglasitev in podelitev priznanj avtorjem referatov

sobota 9. maj

09:00 prijava, recepcija

09:00 - 10:30 OKROGLA MIZA 4:

Prenova in informatizacija javne uprave

10:30 - 11:00 odmor

11:00 - 12:30 VABLJENA REFERATA (vodja Andrej Kovačič)

12:30 - 13:00 Sporočilo posvetovanja

Izžrebanje udeleženca

Zaključek posvetovanja

Prijave pošljite na naslov:

Slovensko društvo INFORMATIKA

DSI '98

Vožarski pot 12

1000 LJUBLJANA

Dnevi slovenske informatike - DSI 98

6.-9.5.98 Portorož Slovensko društvo Informatika ZRIS
 niko.schlamberger@stat.sigov.mail.si
 fax: 061 302 370
 Niko Schlamberger

INFOS Data Warehouse '98

19. - 20.5-98 Ljubljana INFOS
 A. Jus, B. Sotošek
 infos@infos
 tel.: 061 213 727, faks: 061 222 487

The Eleventh International Bled Electronic Commerce Conference

8. - 12.6.98 Bled Univerza v Mariboru, FOV Kranj
 Jože Gričar
 gricar@uni-lj.si
 http://ecom.fovret.uni-mb.si

Svetovni kongres Projektni management

10.-13.6.98 Ljubljana INFOS
 A.Jus, T. Končan
 infos@infos
 tel.: 061 213 727, faks: 061 222 487

OTS '98 - Objektna tehnologija v Sloveniji

17. - 18. 6. 98 Maribor Center za objektno tehnologijo
 Fakulteta za elektrotehniko, rač. in informatiko, Inštitut za informatiko
 Marjan Heričko
 cot@uni-mb.si
 tel.: 062 220 74 11
 faks: 062 211 178
 URL: http://lisa.uni.-mb.si/cot/ots98

Context-Sensitive Decision Support Systems

13. -15.7.98 Bled IFIP W.G. 8.3
 Univerza v Mariboru, FOV Kranj
 Slovensko društvo informatika
 Vladislav Rajkovič
 vladislav.rajkovic@ijs.si
 http://www-personal.umich.edu/~widmeyer/ifipwg83

6th International Conference on Information Systems Development - ISD '98

21. - 23. 9. 98 Bled Univerza v Mariboru, FOV
 Jože Zupančič
 faks (064)221-424
 ISD@fov.uni-mb.si

Internet v Sloveniji

23. - 24.9. 98 Ljubljana INFOS
 B. Sotošek
 infos@infos
 tel.: 061 213 727, faks: 061 222 487

INFOS '98

26.-3.10.98 Ljubljana INFOS
 A. Jus, R. Stojanovič
 infos@infos
 tel.: 061 213 727, faks: 061 222 487
 http://www.infos.si

Navodila avtorjem

Prispevke pošiljajte v predpisani obliki na naslov Slovensko društvo Informatika, 61000 Ljubljana, Vožarski pot 12, s pripisom za revijo Uporabna informatika.

Če je možno, naj bo članek lektoriran. V uredništvu bomo opravili korekturo in se po presoji posvetovali z avtorjem, da članek tudi lektoriramo.

Prispevek naj bo v obsegu največ avtorska pola (30.000 znakov) za strokovne članke in približno 2 do 3 tiskane strani za druge prispevke. Vsak strokovni članek naj ima na začetku povzetek v slovenskem in v angleškem jeziku. Na koncu dodajte kratek življenjepis.

Posljite ga na disketi in odtisnjene na papirju. Napisan naj bo v urejevalniku **WORD 6.0** oziroma v **Text ali RTF** formatu. Na disketi označite, kateri urejevalnik ste uporabili, in ime datoteke. Datoteko imenujte s svojim priimkom, n. pr. Novak.doc ali Novak.txt.

Slike, grafikoni, organizacijske sheme itd naj imajo belo podlago. Upoštevajte, da tiskamo v črno-beli tehniki s folije (ne iz filma). Priložite jih na posebni disketi, datoteka z besedilom naj bo brez njih.

Pišite v razmaku vrstic 1, brez posebnih ali poudarjenih črk ali podčrtovanja, za ločilom na koncu stavka napravite samo en prazen prostor, ne uporabljajte zamika pri odstavkih.

Za vsa vprašanja se obračajte na tehnično urednico Katarino Puc, 1000 Ljubljana, Ulica Gubčeve brigade 120 tel. 1271-579, elektronska pošta Katarina.Puc@uni-lj.si.

Članke in daljša sporočila pošiljajte na naslov: LiliJana.Jernejcic@stat.sigov.mail.si

Revija Uporabna informatika bo brezplačno objavljala v rubriki Koledar prireditev datume strokovnih srečanj, posvetovanj in drugih prireditev s področja informatike. Obvestila naj vsebujejo naslednje podatke: ime srečanja, datum in kraj prireditve, naziv organizatorja, ime in telefonska številka kontaktne osebe. Pošiljajte jih na naslov: Slovensko društvo Informatika, za revijo Uporabna informatika, rubrika: Koledar prireditev, 1000 Ljubljana, Vožarski pot 12. Objavljali bomo vsa obvestila, ki bodo prispela 30 dni pred objavo revije.

UPORABNA INFORMATIKA

ISSN 1318-1882

Ustanovitelj in izdajatelj:

Slovensko društvo Informatika, 1000 Ljubljana, Vožarski pot 12

Glavni in odgovorni urednik:

Mirko Vintar

Svet revije:

Ciril Baškovič, Andrej Cetinski, Ljubica Djordjevič, Franc Križaj, Ivan Žerko

Uredniški odbor:

Tomaž Banovec, Vladimir Batagelj, Ivan Vezočnik, Jože Gričar, Janez Grad, Andrej Kovačič,
Tomaž Mohorič, Katarina Puc, Vladislav Rajkovič, Ivan Rozman, Niko Schlamberger, Mirko Vintar, Franc Žerdin.

Tehnična urednica: Katarina Puc

Oblikovanje: Zarja Vintar, Dušan Weiss

Naslovnica: Zarja Vintar

Tisk: Prograf

Naklada: 700 izvodov

Revija izhaja četrtletno. Cena posamezne številke je 2.000 SIT.

Letna naročnina za podjetja SIT 7.800, za vsak nadaljnji izvod SIT 6.000.

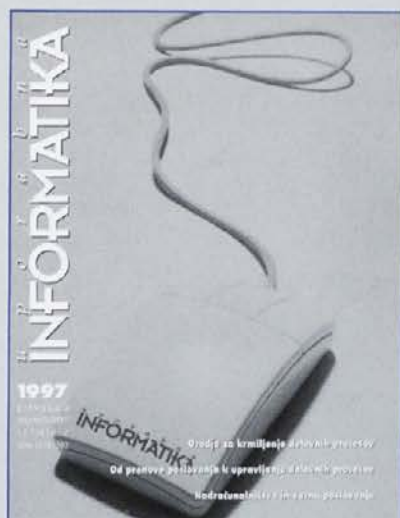
Letna naročnina za posameznika SIT 4.000, za študente SIT 2.000.

Za člane društva je cena revije vključena v članarini.

IZ VSEBINE REVIJE



UPORABNA INFORMATIKA V LETU 1997:



Andrej Kovačič:

Kakšne uporabniške rešitve potrebujemo?

Aleksandar Jurišič, Alenka Trojar:

Pametna kartica

Jože Gričar:

Elektronsko poslovanje: priložnosti za

gospodarske družbe, državno upravo in potrošnike

Tomaž Banovec:

Podatkovni model Republike Slovenije

Evelin Vatovec Krmac:

Vloga ponovne uporabe pri razvoju programske opreme

Ministrska deklaracija o globalnih informacijskih omrežjih

Bela knjiga: Elektronsko poslovanje malih in srednje velikih podjetij (posebna številka)

Lea Bregar:

Sodobna tehnologija-razvojni izziv izobraževanja

Boris Sobočan:

Metodologija ARIS

23
A 1388

