

u p o r a b n a
INFORMATIKA

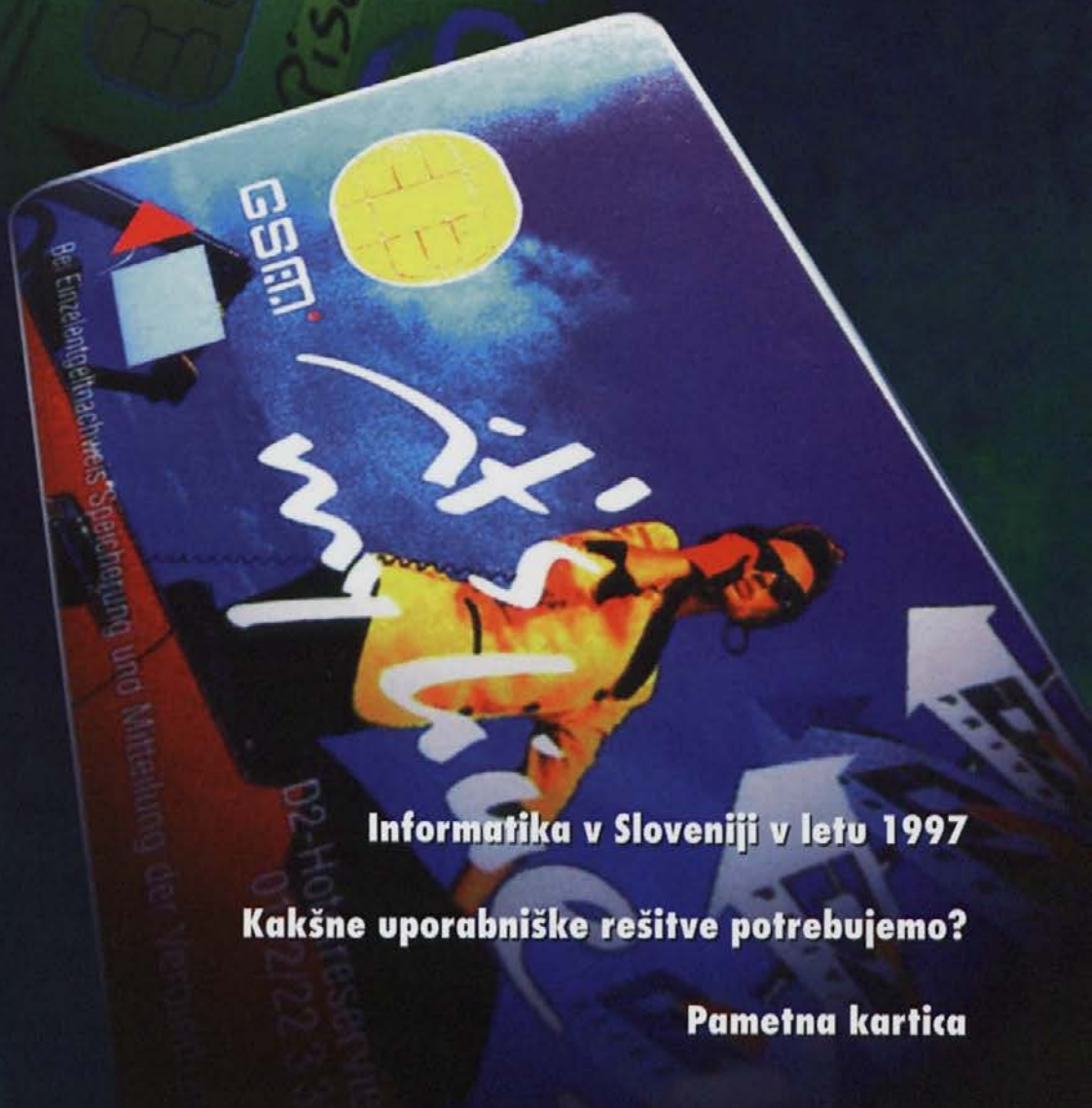
1997

ŠTEVILKA 1

JAN/FEB/MAR

LETNIK V

ISSN 1318-1882



Informatika v Sloveniji v letu 1997

Kakšne uporabniške rešitve potrebujemo?

Pametna kartica

Spoštovani bralke in bralci,

pričujoča številka Uporabne informatike bo izšla pred Dnevi slovenske informatike '97, posvetovanjem informatikov, ki smo ga podnaslovili **Informatika za tretje tisočletje**. Naj bo to priložnost za razmislek o informatiki nasploh in o nalogah, ki čakajo slovenske informatike. Mogoče bo kdo pomislil, da prehitavamo, pa smo prejkone celo nekaj malega v zaostanku in bi se na prihajajoče tisočletje morali začeti pripravljati že prej. Kaj je vendar tako posebnega na njem, da se s tretjim tisočletjem vsi tako prizadevno ukvarjamo? Saj bo prišlo tako ali tako, samo od sebe, pa če hočemo ali nečemo. To je seveda res, res je pa tudi, da so časovne prelomnice priložnost za pregled dosežkov in oceno dogajanja in čim bolj okrogla je obletnica, tem bolj se nam zdi pomembna. Prelomnica, ki prihaja, prihodnjih tisoč let ne bo nastopila in to je že razlog, da se ob njej pomudimo tudi z uvodnikom.

Prvo tisočletje je tisočletje rojstva zahodne civilizacije, ki je danes na višku razcveta. Drugo tisočletje je tisočletje stabilizacije evropskih in določanja svetovnih meja ter vzpostavljanja sedanjega gospodarske ureditve. Vse to je nastajalo v revizijah, ki jih danes poznamo kot stoletno vojno, reformacijo, tridesetletno vojno, protireformacijo, buržoazno revolucijo, pomlad narodov, prvo in drugo svetovno vojno, nastanek in zaton socializma. In kaj bo značilno za tretje tisočletje? Gotovo se zdi le to, da je danes znanih mnogo vprašanj in precej manj odgovorov. Lahko napovemo, da bo človeštvo tretje tisočletje preživljalo v znamenju informacije, ki bo takojšnja, poudarjena in vsem na razpolago. Tehnologija, ki bo to omogočala, bo povzročila tudi druge spremembe, ki jih lahko v obrisih že zaznavamo. Verjetno se osnovni mehanizmi delovanja družbe ne bodo spremenili, saj so, kakor je videti skozi zgodovino, stabilni. Spremenil pa se bo način njihovega delovanja, njihove pojavnosti oblike in način prilagajanja družbe. James Martin je že nekaj časa tega upeljal pojem "cybercorp", nove družbe, ki se obnaša anticipativno in kooperativno. Sistem izobraževanja bo moral upoštevati, da je poklic za celo življenje preteklost. Država bo morala končno postati to, za kar so jo državljani ustanovili: ne oblast za vsako ceno, temveč poceni in učinkovit servis.

Kdor ne pozna zgodovine, ne razume sedanjosti in je, namesto, da bi bil obdarovan s spoznanji, obsojen na ponavljanje. To velja za posameznike kakor za narode in za človeštvo v celoti. Razlog, da od tretjega tisočletja toliko pričakujemo, je intuitivno spoznanje, prej upanje, da vsaj največjih napak ne bi smeli več ponavljati. Katere so, je mogoče odvisno od meril, verjetno pa ne more biti dvoma, če kot merilo vzamemo posledice, ki so jih napake pustile na narodih in na človeštvu. Naj navedemo nekaj primerov ponovitev, ki morebiti niso usodne z vidika zgodovine, ki pa so vse po vrsti nesrečne za razvoj znanosti - geografije, astronomije, fizike, matematike.

V tem tisočletju je bila Amerika odkrita kar dvakrat. Najprej je tja okrog leta 1000 priplul Leif Eriksson, petsto let za njim pa še Krištof Kolumb. Odveč je pripominjati, da niti prvo niti drugo ni bilo odkritje za ljudi, ki so tam živeli. V štirinajstem stoletju je mongolski knez Ulug beg (1393 - 1449) določil za tisti čas daleč najbolj točne tabele poti planetov. Evropski astronomi bi jih bili neizmerno veseli, vendar o njih in o astronomu, ki jih je izračunal, niso niti slutili. Šele leta 1665, vendar že po tem, ko je Danec Tycho Brahe (1546 - 1601) izračunal točnejše, so jih predstavili v latinščino. Ni treba ugibati, koliko bi napredovala astronomija z natančnejšimi tablicami. Leonardo da Vinci (1452 - 1519) je izpeljal enačbe trenja, vendar njegovo delo ni imelo vpliva

u p o r a b n a
INFORMATIKA

999802923



na fiziko, ker svojih zapiskov ni objavil. Šele v 17. stoletju je zakone trenja ponovno odkril Guillaume Amontons (1663 - 1705), ki bi sicer lahko nadaljeval, kjer je končal Leonardo. Charles Augustin de Coulomb (1736 - 1806), eden od očetov moderne elektrotehnike, je odkril zakon o električnih silah, znan kot Coulombov zakon - ki pa ga je zgolj ponovno formuliral. Pred njim ga je spoznal Henry Cavendish (1731 - 1810), kar so našli v njegovih zapiskih šele pol stoletja po njegovi smrti in katerega znanstveno delo Coulomba ni doseglo. Samo ugibamo lahko, kaj bi bila pridobila fizika, če Coulomb ne bi poodkrival že odkritega, temveč bi svoj genij usmeril na v resnici še nerešene probleme. In za konec navajanja, pa ne kot konec primerov: še ne enaindvajsetletni francoski revolucionar Evariste Galois (1811 - 1832) je v noči pred zanj usodnim dvobojem napisal osnove teorije grup in zapiske zapustil prijatelju, ki pa ni razumel njihovega pomena. Teorijo grup je ponovno formuliral Leopold Kronecker (1823 - 1891). Koliko bi pridobila matematika, če bi lahko nadaljeval tam, kjer je Galois nehal! Vsem primerom je skupno to, da ni bilo informacije, kaj je na določenem področju v določenem trenutku znanega. Posledica je bila, da je bil napor raziskovanja, odkrivanja in študija podvojen.

V dobrih petdesetih letih razvoja elektronskih računalnikov so le-ti ostali vgrajeni v temelj, na katerem stoji zgradba nove stroke, ki združuje pridobitve vseh dosedanjih revolucij v prenosu informacije: govora, pisave in tiska, in ki tem trem dodaja še moč računalnikov in komunikacij. Informatika je disciplina, ki lahko omogoči, da nikomur več ne bo treba odkrivati že odkritega. Informatika naj bi, kakor mislijo nekateri, iz sveta naredila globalno vas. Verjamemo, da se ne bo zgodilo to, temveč nekaj dosti boljšega: globalno mesto. Razlika je tolikšna, da je ni treba posebej opisovati, ilustriramo pa jo lahko s splošnim pojmovanjem, da velja v civilizaciji mesto za središče, vas pa za obrobje dogajanja. Informatika bo pripomogla, da bodo vsem ljudem na razpolago poleg informacij in podatkov tudi vse dobrine, ki jih navadno asociiramo z mestom, pri čemer kulturne niso na zadnjem mestu.

Ob tem, ko razmišljamo o globalnem mestu, moramo pomisliti tudi na njegov sedanj in prihodnji del - Slovenijo, pravzaprav na dve: na to, ki je tukaj in zdaj, ki se prebija skozi različne tranzicije, in na tisto iz tretjega tisočletja. Obema mora biti informatika močna opora. Prvi pri prizadevanju, da opravi vse zahtevane izpite v prvem roku in brez ponavljanja, ker zgodovina ponavljavcem pač ni naklonjena, majhne države pa tudi nimajo virov, da bi ponavljanje lahko zdržale. Drugi pa pri prizadevanju, da bo mogla kljub majhnosti obdržati korak z velikimi. Globalna tranzicija, ki jo že vidimo in ki bo tudi slovenska, je prehod v postindustrijsko, informacijsko družbo. Izziv za slovensko informatiko, kako uspešno podpreti tudi to tranzicijo, je tu. Prav nič prekmalu ne bo, če se začnemo pripravljati že danes.

Da uvodnik ne bi izzvenel preveč evforično, naj opozorimo še na temnejšo stran odličja. Industrijska revolucija in avtomatizacija sta ukinjali delovna mesta; tudi informatizacija procesov jih še ukinja. Vprašanje je, ali informatika in informacijske storitve lahko ustvarjajo novo vrednost in nova delovna mesta. Končnega odgovora še ni, določena znamenja pa vendar vlivajo upanje.

Informacijske storitve gotovo ustvarjajo novo vrednost, vendar ob pogoju, da ostaja struktura gospodarstva uravnotežena tako, da se ne opustita niti kmetijska pridelava niti industrijska proizvodnja. V Združenih državah Amerike je bil pomemben argument za ponovno izvolitev Billa Clintona za predsednika prav zmanjšanje brezposelnosti za časa njegovega prejšnjega mandata.

Ta država, ki je mejo med industrijsko in informacijsko družbo že prestopila, dokazuje, da zmanjševanje števila delovnih mest ni nujno atribut informacijske družbe, čeprav nevarnosti splošne uporabe računalnikov za nekatere poklice ("job killer applications") še ne bi smeli podcenjevati. Lahko pa vendar pričakujemo, da bodo nastajali novi poklici informacijske družbe.

Naj še končamo s primerom iz zgodovine. Rimski cesar Konstantin je leta 391 prepovedal čaščenje egiptčanskih bogov. Posledica tega je bila, da se je branje hieroglifov, ki so bili dotlej še v splošni rabi, pozabilo tako rekoč čez noč. Zgolj naključje je pomagalo Jean-Françoisu Champolionu (1790 - 1832), da je po skoraj tisočletju in pol znova odkril dotlej pozabljeno znanje. Informatika je stroka, ki naj pripomore tudi k temu, da nečesa, kar je že spoznalo, človeštvo ne bi nikoli več pozabilo in nikoli več ponovno odkrivalo.

Niko Schlamberger

UVODNIK**AKTUALNO**

- 5** ■ ■ ■ ■ Informatika v Sloveniji v letu 1997 – razvojne dileme, vprašanja in naloge

STROKOVNE RAZPRAVE

- 8** ■ ■ ■ *ANDREJ KOVAČIČ:* Kakšne uporabniške programske rešitve potrebujemo?
- 16** ■ ■ ■ *IVAN VEZOČNIK:* Prenovitev informacijske podpore poslovnemu sistemu
- 23** ■ ■ ■ *BOGDAN VOZIČ:* Objektna usmeritev pri razvoju poslovnih in informacijskih sistemov
- 29** ■ ■ ■ *JURIČ, HERIČKO, ROZMAN:* Objektni model porazdeljenega procesiranja

NOVE TEHNOLOGIJE

- 37** ■ ■ ■ *ALEKSANDAR JURIŠIČ, ALENKA TROJAR:* Pametna kartica

POROČILA

- 46** ■ ■ ■ ■ Deseto posvetovanje sekcije za raziskovanje informacijskih sistemov

NOVE KNJIGE

- 47** ■ ■ ■ Marjan Pivka: Kakovost v programskem inženirstvu
- 48** ■ ■ ■ Borka Jerman Blažič: Internet
- 48** ■ ■ ■ Donald H. Taylor, G. William Glezen: Revidiranje - zasnove in postopki

DOGODKI IN ODMEVI

- 49** ■ ■ ■ ■ Serijska in butična proizvodnja programske opreme
- 49** ■ ■ ■ ■ 50 LET ACM

KOLEDAR PRIREDITEV

- 50** ■ ■ ■ ■

Izid te revije so finančno podprli:



SLOVENICA

zavarovalniška hiša d.d., Ljubljana

Za vašo varnost, vedno in povsod!

- življenjsko zavarovanje in življenjsko pokojninsko zavarovanje
 - prostovoljno pokojninsko zavarovanje in rentna zavarovanja
 - nezgodna zavarovanja, premoženjska zavarovanja
 - stanovanjsko zavarovanje
 - obvezna in kasko avtomobilska zavarovanja
- In še mnogo več.*

Ljubljana, Celovška 206, tel. (061) 159 73 86

Filiala Ljubljana, Celovška 91, tel. (061) 159 50 32 Poslovalnice in zastopstva: Ljubljana, Kranj, Domžale, Litija, Celje, Ribnica, Radovljica, Tržič, Metlika

Filiala Koper, Ljubljanska 3, tel. (066) 32 541 Poslovalnice in zastopstva: Portorož, Ilirska Bistrica, Postojna, Sežana, Izola, Divača

Filiala Maribor, Partizanska 37, tel. (062) 225 497 Poslovalnice in zastopstva: Rogatec, Murska Sobota, Lendava, Dravograd, Slovenj Gradec

Filiala Nova Gorica, Gregorčičeva 11, tel. (065) 27 070 Poslovalnice in zastopstva: Tolmin, Idrija, Ajdovščina, Logatec

Mrežni marketing: KD NET, Ljubljana, Stegne 21, tel. (061) 151 12 22

Ko previdnost postane modrost!

Informatika v Sloveniji v letu 1997

- razvojne dileme, vprašanja in naloge

Uvod

Leto 1997 je za informatiko v naši državi verjetno dokaj prelomno, če ne že tudi usodno. Veliko se sestajamo, posvetujemo, izmenjavamo mnenja, tudi o informacijski infrastrukturi se piše zelo veliko. Malo pa razumemo državo, njene naloge in vsebine, ki jih sodobna informatika podpira. Dobili smo novo vlado, oblikovali se bodo novi organizacijski zakoni, verjetno se bo pristojnost za informatiko razdelila ponovno - verjamemo, da manj neposrečeno kot je bila dosedaj. Zato je dobro razmisliti o tem, kje smo, kam gremo in kaj je treba narediti.

Evropski dokumenti, pa tudi raziskovanja informacijske infrastrukture, o čemer smo v Uporabni informatiki že pisali (št. 1/1995), so vedno bolj oddaljeni od nas. Naložbe v raziskovanje informatike v Ministrstvu za znanost in tehnologijo, nasploh pa organizacijski naporji za definiranje informacijske strategije ter definicije v tem ministrstvu pa stagnirajo.

Drugi se za informatiko kot horizontalno dejavnost ne čutijo pristojni, razen za določen del državne informacijske infrastrukture. Vsebine pa so obrnjene vase in tako "otočno" tudi informatizirane.

Informacijska infrastruktura se bo v Sloveniji tudi v nadaljevanju srečevala s problemi povezovanja in informatizacije vsebin in funkcij državne uprave. Ta daje povsod tudi razvojni ton in usmeritev. Javne in omrežne informacijske storitve, ki jih pospešeno izvaja z relativno malo sredstvi Center Vlade za informatiko, že udarjajo ob inštitucionalne probleme v ministrstvih, v upravnih enotah ter občinah. Nastajajo nova vsebinska ter strokovna nesoglasja, nepovezane uporabe in nove rešitve (davčni sistem, občine in obremenitve občanov ipd.).

Vendar je lahko taka informacijska infrastruktura tudi parcialno uspešna, čeprav nad njeno uporabo kmalu ne bo več dobrega pregleda in analiz. Ne vemo, koliko, poleg tega, kar je vgrajeno v infrastrukturo, stanejo še: vzgoja uporabnikov, izobraževanje in ali so organizacijske spremembe in projektni pristop za tak velik podvig tudi prisotne. Internet postopoma prehaja v državni internet. Srečali se bomo z novimi omrežnimi programskimi orodji, potrebno bo razmisliti o tem, na katerih področjih bomo ostali pri omrežnih osebniških (PC-jih), kje je še možno in smotrno uvesti omrežnik ali omrežen računalnik, kaj

je z novimi programskimi orodji in ali se jih bomo učili ali ne, pa tudi - ali bo sploh še kje delal kakšen star, tako imenovani neumni terminal.

Nujno je začeti reševati zakonske ureditve elektronskega podpisa, kar v ZRN in Evropi že počnejo, urediti elektronski notariat in preverjanje ter potrjevanje elektronskih podpisov. To na primer zadeva delovanje Agencije za plačilni promet, ki je trčila prav na problem dislociranega potrjevanja podpisov.

Nov problem bosta še vsebinska zaščita in identifikacija dokumentov. V državni upravi pričakujemo ustrezen standard. Zaščita dokumenta pa je potrebna v smislu vsebin, o čemer smo tudi večkrat govorili na naših dnevih slovenskih informatikov.

Zadeve so povezane tudi s kripto zaščito, ki spremlja vsa ta področja in pa seveda s konceptom omrežne rešitve za mlado državo ali malo državo, kot je Slovenija v celoti.

Očitno je, da so razmerja med Telekomom in velikimi uporabniki še nejasna, cene storitev v perspektivi nedoločene. Telematika vpliva na razne druge dejavnike in verjetno bo vprašanje tudi to, ali bo naša osrednja organizacija Telekom kdaj ter pod katerimi pogoji privatizirana.

Ali bo ob tem upoštevana predpostavka, da država Slovenija omogoči vsakemu državljanu svoj priključek ISDN ali ne, je posebno vprašanje. Bogati in veliki Nemci so si to zadali za nalogo, mi pa še vedno govorimo predvsem o cenah priključkov, ki so za vsako od teh nalog nekoliko premajhne.

Vsebine in njihova informatizacija

Nasploh bo treba poleg tehnologije govoriti še o konceptu in vsebini informacijskih storitev, ki so v Sloveniji še zelo nerazvite, govoriti o dodani vrednosti, o zaščiti pravic ter obračunavanju takih storitev tudi za upravne ali državne organe ter določiti: ali popolnoma neprofitno in zaščitno ceno na ameriški način, ali dodano vrednost zajeti skozi ceno v uporabi v omrežju ali na druge načine.

Ne nazadnje tudi arhivski in drugi problemi so tu. Nastaja veliko baz podatkov, ki bodo v določenem času zgodovinsko popolnoma izgubljene in bo treba najti tudi način arhiviranja ali dokumentacije. Na primer, sedaj delujočih okrog 22 računalniško podprtih skupin registrov, vezanih na poslovni register Slovenije, se dnevno spreminja in registrirana stanja v podjetjih tudi. Kdo hrani skupno stanje?



Zakoni, ki urejajo področje informatike ali informiranje

Veljavna zakonodaja za informatiko in zlasti vladna ureditev na tem področju nista posrečeni. Kot rečeno, nekatere stvari so napisane napak, nekatere določene napačnim ministrstvom, nekatere pa nikomur.

Verjetno nove evropske klasifikacije dejavnosti ob novi organizaciji ter delitvi ministrstev ne bodo upoštevane. Če bo ostal 1. člen zakona o delovanju in pristojnostih ministrstev, ki določa medsebojno sodelovanje na področju evidenc, izmenjavo podatkov ter ostalo, bo še vedno ostalo vprašanje: "Ali bo novo nastali Svet za informatiko sposoben izvrševati to koordinacijo?"

Vprašanje so občinske ter izpeljane upravne ureditve. Občine imajo sicer dovolj pravic do sprejemanja in uporabe podatkov, nimajo pa pravice (vsaj tako je tolmačenje), da bi zbirale svoje na osebo vezane podatke. S temi podatki po členu 21. zakona o lokalni samoupravi delajo veliko različnih stvari. Med drugim obdavčujejo prebivalce po stanju registra iz julija v mesecu decembru in delajo druge napake. Ali se bodo uredile proti občanom in proti državi?

Pričakovati bi bilo, da bodo vsaj tiste lokalne skupnosti, ki jim je do tega, da bi stroške ter delo razdelile, kupile ali se organizirale v nek skupen informacijski servis, ki bi s programsko opremo ter ostalimi (tudi na razdaljo) omogočal, da svoje aplikacije izvršujejo racionalno zase in za občane. Delitev stroškov za skupen razvoj programske opreme in sedanjih 147 enot bi bil hitro racionalen.

Samozadostnost, resorskost ...

Težave, ki jih ima naša mlada država zaradi nedorečenosti, slabih definicij ter prepletujočih se definicij, so že velike. Informatizacija takih razmer s pomočjo raznih samozadostnih informacijskih sistemov te probleme sicer razkriva, jih pogloblja, ne pa rešuje. Še več, omrežene slabe rešitve hitro prenašajo napake na vse strani.

Zato je zelo važno horizontalno prizadevanje, da bi državno upravo uredili tudi okrog pristojnosti glede na posamezne baze podatkov, to so uvedli tudi v Evropi (projekt IDA - Interchange of Data between Administrations).

Projekti PHARE in drugi so to situacijo z dodatnimi finančnimi sredstvi prej poglobili, kot povezali, saj je bilo zelo očitno, da so si ministrstva pomagala predvsem z računalniki ter drugimi "informacijskimi" sistemi, manj pa s kompleksnimi in povezanimi rešitvami.

Zakoni in ureditve

Ogrožene so tudi zakonske ureditve ob pomanjkanju kapacitet parlamenta. Nacionalni program statističnih raziskovanj še ni sprejet, kar odpira široke probleme pristojnosti ter obveznosti dajanja podatkov. Dokument, v

katerem je zapisano okrog 400 nalog, ki jih morajo narediti razni pooblaščen organi, obravnavamo v parlamentu že v tretjem letu.

To nadalje odpira druge probleme kot so zakonske ureditve na področju varovanja osebnih podatkov, kjer so v zaostanku geodezija, zdravstvo, del socialnega zavarovanja ter vsi tisti, ki so prevzeto zakonodajo iz prejšnje države mislili urediti do 1.1.1995 do 1.1.1996 in še sedaj niso tega uredili. Pričakujemo pa nove zakone o že omenjenem notariatu, zaščiti dokumentov, o kriptu zaščiti ter še o čem.

Posebno vprašanje je definicija javne uprave v celoti. Ali bo še okrog 1.000 javnih zavodov in podobnih institucij "delalo svojo informatiko", ali se bodo priključili ministrstvom, ki so za njih odgovorni in prevzeli njihove standarde? Predpostavka je, da bodo ministri med sabo uredili zadeve ob pomoči Sveta. Ob vsem tem bo verjetno Evropa prisilila, da se uredimo najprej okrog **definicij javne uprave**, negospodarstvo prevedemo v javne delavce ter podobno, uredimo nekatere obračunske kategorije in šele potem uredimo tudi informatiko. Vendar - kasneje bo to urejeno, dražje bo. Informatika se nam vseeno tehnično in parcialno uvaja in kasnejše rešitve bodo izredno drage. Posebej tudi, če bo v parlamentu še vedno nesporazum okrog tega, da emšo ne ogroža prebivalca, marveč šele emšo omogoča zaščito tudi pred tem, da ni 50-krat povabljen pred državna okenca, če se slučajno preseli.

Država in veliki informacijski projekti

V državi bomo v tem času morali dokončati že napovedane velike informacijske projekte ali pa se od njih posloviti. Tako so informacijski sistemi za varstvo okolja, določeni v zakonu, praktično še vedno v predalih, podobno je z informatizacijo registra stanovanj, ki se parcialno izvaja, zakon, ki ga določa je sicer prodal veliko stanovanj, te informatike pa ni uredil. Veliko rezervacij za informacijske sisteme se srečuje z najbolj pomembno **informatizacijo evropskih prizadevanj**, ki niti ni koncipirana, uvažamo jo po koščkih. Formulacije pošiljamo po disketah - formatizirane v velikih bazah podatkov Evropske zveze.

Migracija plačilnega prometa iz APP v bančno okolje se je začela. Kaj pomenijo informacijski sistemi za izobraževanje, za šport in vzgojo, informacijski sistem za celovito obravnavanje kmetijstva, informacijski sistem nepremičnin? Vsi resorji gradijo svoje informacijske sisteme. Ali vemo, koliko so povezani?

Ali bo ob tem občan šel v občinski ali upravni urad in pri nekem okencu uredil vse zadeve, kot smo zapovedali in prosili v deklaraciji o informacijskih storitvah, ali pa bo tekal od okenca do okenca (od sistema do sistema), ki bodo sicer vsa informatizirana. Ali bo zakon o

centralnem registru prebivalstva rešil tudi problem prostovoljnega prijavljanja ter izmenjave podatkov?

Posebno vprašanje pri teh velikih projektih pa je, ali bodo informatizirana, povezana ter drugače urejena tudi razmerja ter dejavnosti v koordinaciji za povezovanja v Evropsko unijo. Izredno problematično je, da v tem času še nimamo skupnega informatiziranega seznama vseh projektov PHARE, niti ne pregleda nad njihovo realizacijo, da ima država resne probleme s pregledom smernic in zakonov ter z izvrševanjem sklepov vlade. To vprašanje bo moralo biti rešeno verjetno na podoben način, saj se informatizacija postopkov vključevanja v Evropsko unijo, četudi se nekoliko zavleče, mora urediti takoj in ne zadeva samo majhnih skupin ali koordinacij, marveč državo v celoti.

Pri velikih projektih na koncu bi morali opozoriti tudi na vprašanje, ali bomo vendarle pričeli obdobjo popisovati računalniško opremo, mreže, priključke, načine uporabe, storitve ter ostalo in vsaj na tri, štiri, pet let ugotavljali, kje država Slovenija je tako v upravi kot v zasebnem sektorju, na kakšnih področjih smo dobro informatizirani, kakšne so naše perspektive. Za to dosedaj ni bilo ne volje, ne sredstev, ne metodologij. Ker pa Evropa že pripravlja nove metodologije mislimo, da bo sredstva potrebno najti. Seveda je še nekaj velikih projektov, ki niso vsi infrastrukturni, so pa informacijski.

Nova informacijska strategija

Kot že rečeno pričakujemo živahno aktivnost, ki pa se verjetno ne bo odvijala tako, kot pričakujemo. Zakon o varovanju osebnih podatkov je očitno odložen. Obljubljena strategija za informacijsko družbo poleg že več kot 20 strategij v državi ni napisana. Po proučevanju strategij ugotovimo, da informatika ali informacijska tehnologija, če ni slučajno konzumirana v kakšnem odstavku, v teh strategijah sploh ne obstaja. Še boljše je, da njen vpliv v teh strategijah, ki je lahko izredno pomemben in najmanj uničuje delovna mesta, če ne kaj drugega, niti ni prisoten. Vendar je to tudi vprašanje, kako celo v Slovenskem društvu informatika razumemo informatiko. Ali je to informacijska infrastruktura, ali še kaj drugega? Ne razumemo, kako to, da sektorske strategije in nacionalni

programi ne upoštevajo skupnega vpliva informatike in niti ne resornega vpliva.

Evropska zveza in Slovenija

O tem smo že pisali, vendar je Evropa pri svojih projektih že veliko napisala, kaj bo storila s pomočjo informatike. Ker smo informatiki o tem že pisali, seveda ne kaže ponavljati, koliko sredstev je Zveza vložila v to. Področje informatike je dobilo največ od vseh področij od združenih raziskovalnih sredstev in naslednje področje - okolje je dobilo 20 % toliko tega denarja kot informatika. Vendar kaže opozoriti na projekt IDA v katerem je okrog 60 projektov, ki zadevajo neposredno državo in njeno informatizacijo, opisanih vsak na svojih dveh ali treh straneh. Pri tem je že vprašanje, ali smo to prebrali in razumeli in kaj bomo storili doma. Mi nujno potrebujemo tudi vsebinsko urejen informatiziran pristop k evropskim prizadevanjem.

Nekaj od vprašanj smo navedli, kot so kriptografija, elektronski podpis, notariat, nekaj od tega pa ne, kar zadeva informatizacijo zdravstva, socialnih storitev ter podobnega. Vendar je resno vprašanje, ali bodo ministrstva, strokovni organi pa tudi zbornice in druge asociacije sploh lahko razumele ta izredna gibanja, ali se bomo v njih vključili, kaj dodali, kaj naredili sami, ali pa bomo zmanjšali svoja prizadevanja na izvoz nekaterih najbolj vitalnih, najbolj dobrih produktov države, vse ostalo pa kupovali? Bistveno pa je, ali se bomo lahko odrekli svojim resornim pristojnostim in se združevali in povezovali tudi horizontalno.

Tu pa ne moremo mimo vloge društva, ki je lahko majhna, da lahko kot civilna organizacija opozori na te probleme in seveda na vlogo države, ki naj bi vsaj v svetu za informatiko, ki bo verjetno v tem letu začel delati, začela vsaj odpirati te probleme.

Zaprta jih kmalu ne bo, vendar je dobro, da vemo, ne samo, kdo je za kaj pristojen, marveč tudi, kdaj bo kaj naredil in tudi vemo, kaj se mu zgodi in kdo bo naredil, če on ne bo naredil. To pa je osrednje vprašanje našega razvoja ne samo na področju informatike, marveč tudi drugje.

Uredništvo revije

KAKŠNE UPORABNIŠKE PROGRAMSKE REŠITVE POTREBUJEMO?

ANDREJ KOVAČIČ
PRIS Consulting, Ljubljana

POVZETEK

V svetu ocenjujejo, da obstoječe uporabniške programske rešitve prehoda v novo stoletje oziroma tisočletje ne bodo preživele. Zamenjale jih bodo rešitve, ki se na tržišču že pojavljajo in katerih poglobljena značilnost je objektna, z odzivom na poslovne dogodke krmiljena zasnova. Te rešitve podpirajo informatizacijo podjetij v smislu sodobne, procesno usmerjene poslovne prenove. Prispevek obravnava odgovore na vprašanja, ki jih moramo upoštevati pri prilagoditvi naših rešitev na nove potrebe in izzive.

ABSTRACT

Recent studies show that the existing user applications won't live through the end of the millennium. They will be substituted by object-oriented event/response applications, the type we can already find on the market. These applications support informatization as a modern process-oriented renovation of business. The article provides answers to some of the critical problems that show up when we adapt existing applications to the new needs and challenges.



1 Ali potrebujemo drugačne uporabniške programske rešitve?

Uvodoma izhajamo iz že splošno znane in na področju obravnave poslovnih procesov uveljavljene ugotovitve, da bodo v prihodnosti preživela le podjetja, ki bodo sposobna masovno proizvodnjo in splošni marketinški pristop nadomestiti s prožnim, prilagodljivim delovanjem in iskanjem tržnih niš za svoje projekte (izdelke in storitve) ter s tem individualizirati in personalizirati svojo ponudbo. Potreba po prožnem prilagajanju, individualizacija in personalizacija, ki sta usmerjeni predvsem v znanega in zadovoljnega poslovnega partnerja, bodo seveda drastično spremenili izgled, predvsem pa obseg obravnave ter vsebinske možnosti obstoječih uporabniških programskih rešitev in zahtevali bistveno drugačen, bolj poglobljen pristop uporabnikov k razvoju in uporabi novih rešitev.

Splošno lahko uporabniško programsko rešitev ali aplikacijo opredelimo kot celoto uporabniških programov za obravnavanje podatkov določene vrste in določenega delovnega procesa ali reševanje določenega problema in vseh organizacijsko tehničnih navodil in pripomočkov za učinkovito in uspešno uporabo računalnika pri tem (Pojmovnik poslovne informatike, 1987). Pri opredeljevanju sodobnih rešitev je smiselno obstoječo opredelitev preurediti in dopolniti na področju obravnave podatkov in procesov. Izhodšče sodobnih rešitev je uporaba skupnih, celovito, na nivoju podjetja obravnavanih podatkov ter predhodno opredeljenih in urejenih poslovnih procesov.

Ugotavljamo, da je dosedanji razvoj rešitev potekal v skladu s klasičnimi poslovnimi in organizacijskimi izhodišči in spoznanji ter možnostmi, ki jih je nudila informacijska tehnologija. Slednja je na osnovi pretiranih pričakovanj na področju osebnih računalnikov in lokalnih mrež prispevala še k dodatni nenačrtosti razvoja rešitev. Tako posamezni organizacijski deli poskušajo sami reševati svoje informacijske potrebe brez ustrezne povezljivosti rešitev in podpore informacijskih potreb celotnega podjetja. Čeprav tak pristop ob relativno nizkih vlaganjih prinaša neposredne koristi posameznim okoljem (podjetjem, službam, ...) znotraj podjetij, pa je na drugi strani v mnogih okoljih vse bolj prisotna težnja po celovitem obravnavanju informacijske podpore poslovanja.

Ugotavljamo, da trenutna informacijska podpora izvajanju poslovnih procesov v večini naših podjetij in ostalih organizacij ni ustrezna. Zaradi izolirane obravnave poslovnega procesa po posameznih organizacijskih enotah (oddelkih) in odsotnosti zahtev po celoviti informacijski podprtosti se pojavljajo večkratne vzporedne evidence in neažurnost podatkov. Tako obdelani podatki niso primerni za pripravo izhodišč za poslovno odločanje. Spremembe in dopolnitve obstoječih obdelav so zaradi njihove konceptualne zasnove in nivoja uporabljene informacijske tehnologije časovno in stroškovno izredno zahtevne.

Odgovor na zastavljeno vprašanje je torej nedvoumen.

Potrebujemo uporabniške programske rešitve, ki bodo celovito obravnavale posamezne poslovne procese, ki potekajo v podjetju. Prav tako nedvoumno pa moramo poudariti, da v večini primerov takšen prehod ob predhodni prenovi ali prilagoditvi poslovnih procesov pomeni tudi zamenjavo ali drugačno uporabo obstoječe računalniške platforme ter informacijske arhitekture in orodij in ne le nadgradnjo obstoječih rešitev.

2 Kaj naj zajemajo nove rešitve ?

Uporabniške programske rešitve so po svoji naravi rezultat procesa izvajanja načrtovanih in medsebojno povezanih aktivnosti. V našem primeru gre za proces informacijske prenove, ki je zlasti usmerjena v prilagodljivost in preglednost poslovnih procesov. Izhaja iz uvodoma opisane potrebe po konkurenčnosti podjetij, ki zahtevajo informacijsko podporo, oziroma uporabniške programske rešitve, vsebinsko bistveno drugačne od obstoječih programskih rešitev.

Ob tem velja poudariti splošno ugotovitev, ki izhaja tudi iz naše vsakdanje prakse. Informacijska tehnologija igra sicer ključno vlogo pri prenovi poslovnih procesov, vendar zgolj z njenim vključevanjem v avtomatizacijo posameznih postopkov največkrat dosežemo, globalno gledano, slabe, če ne celo negativne rezultate. Parcialni pozitivni rezultati zamegljujejo priložnosti in prednosti informatizacije celotnega, primerno prenovljenega poslovnega procesa in infrastrukturno vlogo informatike v tem procesu.

Zato moramo, tako kot na drugih področjih, obstoječe poslovne procese najprej ugotoviti in analizirati predvsem s stališča njihove dosledne opredeljenosti in celovitosti ter primernosti za učinkovito informacijsko podporo. Od informatike (izogibajmo se izrazu informacijski sistem) oziroma uporabe sodobne informacijske tehnologije pričakujemo dvig kakovosti, znižanje stroškov in skrajševanje časa izvajanja tako ugotovljenih poslovnih postopkov, oziroma posameznih aktivnosti znotraj teh postopkov.

Ker smo po tradiciji vse od legendarnega Adama

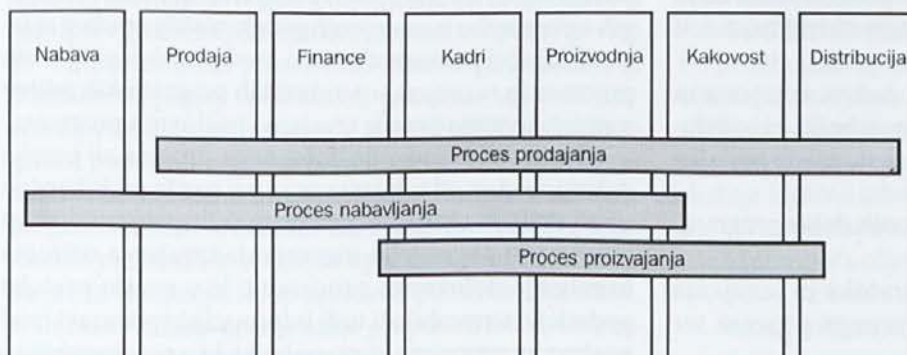
Smitha naprej obremenjeni s funkcionalno, oddelčno sliko delovanja in organiziranosti podjetja, moramo najprej preseči takšno razmišljanje. Ugotavljamo, da skozi tako imenovani funkcijski silos podjetja potekajo trije temeljni procesi, ki jih prikazuje model, ki ga podajamo v nadaljevanju (slika 1).

Model prikazuje pogled na poslovne procese in obenem nakazuje potrebo po drugačni zasnovi uporabniških programskih rešitev, ki izhaja iz spremenjenih informacijskih uporabnikov. Obstoječe oddelčno usmerjene rešitve postajajo neuporabne. Klasične aplikativne rešitve "Skladiščno in materialno poslovanje", "Saldakonti kupcev in dobaviteljev", "Glavna knjiga", ... zamenjujejo tri samostojne, vendar s skupnimi objekti oziroma skupno bazo podatkov povezane objektno zasnovane in na osnovi odzivov na poslovne dogodke prožene, procesno usmerjene uporabniške programske rešitve **nabavljanja**, **prodajanja** oziroma spremljanja naročil in **proizvajanja** oziroma pretvarjanja vhodnih veličin v izhodne.

Seveda se postavljata vprašanji, kaj naj zajemajo oziroma informacijsko pokrivajo nove rešitve in kakšna je možnost in umestnost prenove obstoječih rešitev v procesno usmerjene rešitve.

Nove uporabniške programske rešitve, ki se pojavljajo že sedaj v praksi skupaj s prenovno poslovanja, so procesno usmerjene. Posamezne procese pokrivajo celovito. Tako mora na primer rešitev na področju procesa nabavljanja informacijsko podpirati vse skupine aktivnosti, ki se izvajajo v procesu. Prične pri načrtovanju in ugotavljanju potreb po materialih in storitvah, se nadaljuje skozi naročanje in prevzemanje materialov in storitev, potrjevanje, likvidiranje in plačevanje računov dobaviteljev ter konča z obračunom nabavnega procesa in podatki v "glavni knjigi".

Le takšen, infrastrukturni pogled na informatizacijo, predstavlja ob standardizaciji podatkov in poslovnih pravil (objektov, kot so naročilo, račun,...), osnovo za povezovanje podjetij v smislu računalniške izmenjave podatkov (RIP, angl. EDI - Electronic Data Interchange; več o RIP-u v prispevkih, ki jih je pri nas objavil J. Gričar; npr. (1)). Lahko pričakujemo, da bodo tudi naša podjetja primorana v bližnji prihodnosti (za nekatera je to verjetno že pozno) prenoviti ne samo procese in programske rešitve na področju prodajanja in nabavljanja znotraj podjetja, temveč tudi način povezovanja s svojimi ključnimi kupci in dobavitelji. Že v prihodnjih nekaj letih predvidevamo na tržišču pojav z objektnimi orodji podprtih



Slika 1: Funkcijski silos in temeljni poslovni procesi

standardiziranih povezovalnih programskih rešitev (okvirov, angl. frameworks), ki bodo omogočili podjetjem informacijsko "priključitev" in povezovanje. Podjetjem, ki notranje tehnološko in poslovno na ta izziv ne bodo pripravljena, bo nova možnost povezovanja pomenila prej motnjo kot pa priložnost.

Prenova obstoječih rešitev v smislu njihove medsebojne procesno usmerjene povezave je možna in umešna le v primeru, da so bile obstoječe rešitve sodobno načrtovane, razvijane in delujoče na sodobnih informacijskih orodjih. To pomeni, da so ustrezno dokumentirane (funkcionalni in podatkovni model) in delujoče na integriranem relacijsko ali objektno usmerjenem orodju in bazi podatkov. V nasprotnem primeru se podjetju ponuja idealna priložnost za skladno in sočasno prenovno poslovanja in informacijske podpore oziroma programskih rešitev.

3 Kakšne naj bodo nove, sodobne rešitve?

Prenova poslovanja v smislu procesne usmerjenosti in nova infrastrukturna vloga informacijske tehnologije pogojuje na eni strani spremenjeno vlogo informatike podjetja (računskega centra), po drugi strani pa tudi drugačno vlogo uporabnikov pri uporabi informacijskega sistema podjetja oziroma uporabniških programskih rešitev.

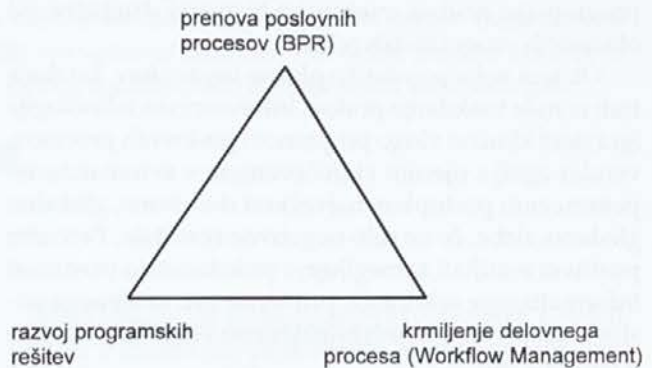
V preteklosti so bili delavci računskega centra tisti, ki so zagotavljali zajem in obdelavo podatkov, v velikem številu primerov pa so bili dolžni zagotavljati kakovost podatkov. Kakovost podatkov se je ugotavljala in korigirala na koncu poslovnega procesa, največkrat v računovodskem sektorju, v katerega se stekajo vsi finančno ovrednoteni podatki.

Opisani način obvladovanja poslovnih procesov vsebuje več hib, ki jih želimo s prenovo uporabniških programskih rešitev odpraviti. Obstoječe hibe oziroma področja bodoče obravnave najlažje opišemo z naslednjimi dejstvi:

1. enkratni zajem podatkov in sočasen, sproten nadzor podatkov v procesu,
2. zajem in kontrolo podatkov v posameznem delu procesa prevzame uporabnik, ki vsebinsko v celoti obvladuje obravnavano problematiko,
3. čimveč podatkov o posamezni zadevi se zajema in preverja na začetku procesa; uporabniki, ki sodelujejo v kasnejših aktivnostih procesa, zajete podatke le dopolnjujejo,
4. nadzorovana uporaba posameznih delov programskih rešitev in podatkov,
5. zagotovitev avtomatizacije pretoka in izvajanja posameznih aktivnosti poslovnega procesa ter sprotni nadzor izvajanja.
6. informatiki podjetja zagotavljajo izobraževanje in

7. pomoč pri uporabi programskih rešitev ter ugotavljanju novih informacijskih potreb uporabnikov, računovodstvo namesto odgovornosti za kakovost podatkov prevzema računovodski nadzor izvajanja poslovnih procesov.

Podrobnejša analiza navedenih ukrepov kaže prepletanje že poznanih ugotovitev na področju razvoja programskih rešitev ter prenove poslovnih procesov in krmiljenja pretoka poslovnih aktivnosti ali delovnega procesa (Workflow Management). Zbliževanje treh različnih tehnoloških usmeritev in njihovo soodvisnost pri razvoju uporabniških programov, pri kateri prenova poslovnih procesov (angl. Business Process Reengineering - BPR) prevzema vlogo strateškega usmerjevalca razvoja rešitev in njihove integracije s sodobnimi orodji, prikazuje slika 2:



Slika 2: Soodvisnost sodobnih usmeritev pri razvoju programskih rešitev

Medtem ko smo pristop k prenovi poslovnih procesov in njegov pomen in vlogo pri informatizaciji poslovanja oziroma pri razvoju poslovnega modela že podrobneje obravnavali (2), v nadaljevanju predstavljamo pomen tehnologije workflow-a pri integraciji in avtomatizaciji poslovnih procesov in uporabniških programskih rešitev.

Tehnologija workflow-a oziroma njena orodja sestavljajo ogrodje ali hrbtnico in vmesnik ali posrednik povezovanja uporabniških programskih rešitev in drugih uporabniku namenjenih programskih orodij v celoto. So ključni povezovalni člen med prenovo poslovnih procesov in razvojem uporabniških programskih rešitev v smislu avtomatizacije izvajanja poslovnih procesov.

Orodje za krmiljenje delovnega procesa ali poteka dela (nobeden od teh izrazov se pri nas še ni dokončno uveljavil) je skupina programskih proizvodov in storitev, ki zagotavlja usmerjanje izvajanja oziroma krmiljenja delovnega procesa ne le v smislu pretoka podatkov, temveč služi tudi informacijski povezavi med poslovnim procesom in uporabniki, ki v procesu oblikujejo podatke in uporabljajo informacije. Namenjeno je

torej usmerjanju in preoblikovanju ključnih poslovnih procesov in s tem izboljšanju njihovih rezultatov in značilnosti.

Orodja za krmiljenje delovnega procesa po svojem poreklu izhajajo s področja obvladovanja pretoka dokumentov. V zadnjih nekaj letih je ta orodja spremljal nagel, metodološko različno usmerjen, vendar v večini primerov uspešen razvoj. Tako lahko trdimo, da so orodja, ki so trenutno komercialno na razpolago, seveda v različni meri, vendar vseeno v večini primerov pomembna, če že ne ključna tehnologija pri razvoju sodobnih, v poslovni proces vpetih uporabniških programskih rešitev. Uporabnikom - odločevalcem v procesu zagotavljajo samodejno proženje in nadzor izvajanja posameznega delovnega procesa, pridobivanje vseh informacij, potrebnih za odločanje v procesu, prenos rezultatov izvršene faze procesa v naslednjo fazo in dinamično spremljanje pretoka oziroma stanja in celovitosti izvedbe del v procesu.

4 Kako do sodobnih programskih rešitev?

Dilema o nakupu ali lastnem razvoju ostaja. Prednosti in slabosti nakupa že izdelanih aplikativnih rešitev so znane v literaturi in vsakdanji praksi. Velja, da z nakupom močno skrajšamo čas razvoja in znižamo nivo tveganja o ustreznosti končnega rezultata, ki smo mu priča pri lastnem razvoju. Pridobimo tudi morebitna tuja znanja z obravnavanega aplikativnega področja, ki jih vsebujejo kakovostne uporabniške programske rešitve. Slabosti nakupa se kažejo v relativno visoki ceni nakupa in osnovnega prilagajanja rešitev. Še bolj pa se poudarjajo skozi problematiko uvajanja oziroma prilagajanja informacijskim potrebam uporabnikov in prenosu vseh znanj, potrebnih za vzdrževanje in nadaljni razvoj, na informatike v podjetju.

Odločitev o nakupu posameznih modulov ali o njihovem lastnem razvoju se lahko izvaja le na osnovi podrobno opredeljenih ter z modelom podatkov formaliziranih in prikazanih informacijskih potreb izvajanja postopkov znotraj poslovnega procesa. Velja pravilo, da je ob normalnih tržnih pogojih smotrna odločitev o nakupu v primeru, da aplikativna rešitev pokriva vsaj 80 % informacijskih potreb obravnavanega področja. Z normalnimi pogoji mislimo ob ustreznih ceni tudi razpoložljivost ustreznih rešitev v izvorni obliki in pripravljenost ponudnika za sodelovanje pri uvedbi in prilagajanju rešitve. Ocena o ustreznosti odločitve o nakupu programske rešitve nikakor ni enostavna. Nikakor je ne smemo prepustiti le bodočim uporabnikom. V primeru, da gre za obsežnejši sklop poslovno pomembnih rešitev, je smotrno takšno ocenjevanje prepustiti neodvisni instituciji ali svetovalcu. Le-ta mora v ta namen predhodno analizirati informacijske potrebe podjetja na

obravnavanem področju, jih formalizirati z modelom izvajanja postopkov in podatkov in ugotovitve primerjati z možnostmi ponujene programske rešitve. Na ta način se lahko izognemo razočaranju in nepričakovanim stroškom, saj prilagajanje in uvajanje neprimerne rešitve, če sploh uspe, povzroča neprimerno večje stroške od stroškov, povezanih s samim nakupom rešitve.

V praksi opažamo sicer pestro ponudbo programskih rešitev, ki pa konceptualno in vsebinsko žal izhajajo iz "zgodovinsko" poznanih programskih paketov. Največkrat gre pri tem le za "šminko" v okolju Windows in za uporabo krmilnega sistema baze podatkov oziroma baze podatkov na način, ki smo mu bili priča v preteklosti. Ker je "prvi vtis", ko si uporabniki ogledajo rešitev, ponavadi pozitiven, je kasnejše razočaranje ob potrebi po prilagajanju rešitev za učinkovito obvladovanje poslovnega procesa še toliko večje. Posebno izrazita in boleča so odstopanja od pričakovanih ob uvedbi raznih direktorskih sistemov na tako razvitih podatkovnih strukturah (nežurnost in ne celovitost podatkov).

Zavedati se moramo, da pomenijo podjetju ravno učinkovite in prilagodljive uporabniške programske rešitve ključno primerjalno prednost na področju informatike. V bodočnosti bo ta pogled v pridobivanje prednosti pred konkurenco za podjetja še kako pomemben. Zato se bo razmišljanje o vzdrževanju obstoječih, največkrat podedovanih rešitev na eni strani, ali pa o lastnem, cenemem in hitrem parcialnem razvoju rešitev moralo korenito spremeniti. Napredna podjetja ugotavljajo, da je potrebno rešitve sproti dopolnjevati in dinamično prilagajati stanju informacijske kulture podjetja oziroma informacijskim potrebam uporabnikov. Zato se v svetu, pa tudi pri nas ob ponudbi modulov sodobnih procesno usmerjenih programskih rešitev vse bolj uspešno pojavljajo in uveljavljajo svetovalne ali pa inženiring skupine in podjetja, ki so sposobni hitro in učinkovito, skupaj z informatiki podjetja ali pa samostojno, razviti ali prilagoditi uporabniške programske module in povezave potrebam podjetja.

5 Kakšni so kriteriji razvoja ali izbire sodobne rešitve?

Po predhodno opisanih aktivnostih v smeri formalizacije informacijskih potreb, in pred izbiro tehnološke platforme, ki jo podajamo v nadaljevanju, je vredno pri izbiri uporabniške programske rešitve upoštevati naslednja ključna izhodišča:

- da je izdelana s sodobnim informacijskim orodjem, ki omogoča objektiven pristop k razvoju in uporabi rešitve oziroma z odzivom na poslovne dogodke krmiljeno zasnovano,
- rešitev ne pogojuje platforma operacijskega sistema (odprti sistemi - npr. UNIX)

- da je razvita, dokumentirana in vzdrževana z ustreznim orodjem CASE (procesni in podatki),
- rešitev je na voljo v izvorni kodi,
- rešitev uporablja skupno in celovito bazo podatkov podjetja, iz katere izhaja,
- s stališča uporabe enotna rešitev (tipizirana uporaba funkcij, specifikacije rešujejo različni uporabniški vmesniki)
- v primeru nakupa rešitve je dobavitelj pripravljen sodelovati pri ugotavljanju informacijskih potreb podjetja in uvedbi rešitve,
- v primeru nakupa rešitve dobavitelj zagotavlja zadostno število izvajalcev ter fiksne pogoje za vzdrževanje rešitve.

Odločitvene kriterije, na osnovi katerih ocenimo ustreznost izbire posamezne alternativne ponudbe, delimo na tehnično - tehnološke, vsebinske in naložbene.

Tehnično - tehnološki kriteriji opredeljujejo potencialno alternativno odločitev z različnih zornih kotov ali stališč. To so zanesljivost in doslednost delovanja, celovitost in povezanost modulov, učinkovitost in funkcionalnost izvajanja, prožnost in prilagodljivost spremembam ter odzivnost vnosa in pridobivanja podatkov.

Vsebinski kriteriji. Kot sodilo o nakupu rešitev, na osnovi katerega lahko analiziramo in ocenjujemo ustreznost rešitev, ki jih predstavljajo posamezni ponudniki, mora biti prvenstveno uporabljen poslovni model podjetja. Na osnovi modela poslovnih procesov preverjamo funkcionalno podprtost in celovitost rešitve (v kolikšni meri rešitev podpira obravnavani proces ali del procesa). Medsebojna primerjava modelov podatkov (modela podatkov podjetja in obravnavane rešitve) pa kaže na stopnjo oziroma obseg, v katerem rešitev pokriva informacijske potrebe obravnavanega področja.

Naložbeni kriteriji so usmerjeni zlasti v oceno kakovosti naložbe v smislu njenega povratka (merljivih in nemerljivih rezultatov in oportunitetnih stroškov), pogojev in dinamike financiranja in ne nazadnje cenovne sprejemljivosti alternativne rešitve.

6 Kaj pa tehnološka platforma in koncepti?

Še hitreje kot razmišljanje o potrebi po prenovi uporabniških programskih rešitev se pojavlja potreba po reviziji naših pogledov na tehnološko platformo in koncepte, na katerih naj delujejo rešitve. Še pred nekaj leti smo prisegali na okolje odjemalec-strežnik (angl. Client/Server) in koncept porazdeljenih baz podatkov. Internet se nam je zdel le pomožno sredstvo, namenjeno posameznim informacijskim gorečnejšim. Na področju uporabe operacijskih sistemov in uporabniških vmesnikov so bila mnenja deljena. Nekateri so prisegali na odprte sisteme (UNIX) in znakovno okolje programskih

rešitev, drugi pa na mrežne koncepte (Novell in kasnejše NT) ter grafično okolje (Windows). Tudi na področju uporabe informacijskih orodij, namenjenih snovanju in razvoju za podjetje pomembnih uporabniških programskih rešitev smo naleteli na različne poglede. Nekateri so zagovarjali (žal mnogi od njih še vedno) uporabo kombinacije najustrežnejših orodij različnih proizvajalcev, drugi pa le uporabo celovitih, integriranih, na skupni bazi podatkov delujočih orodij.

Praktične izkušnje in napovedi svetovnih trendov kažejo na prilagoditev, spremembo ali pa sožitje različnih tehnoloških platform in konceptov. Neodvisni napovedovalci (3) predvidevajo, da bosta s približno enakim obsegom na trgu na začetku naslednjega stoletja prevladovala UNIX in Microsoft NT. UNIX je fazo zrelosti, ki se kaže v učinkovitosti in zanesljivosti delovanja, že dosegel in se uporablja ter se bo pretežno uporabljal v povezavi s strežniki podatkovnih baz na nivoju podjetja. NT bo dozorel do konca tega stoletja, ko bodo predvidoma odpravljene nezanesljivosti, ki izhajajo pretežno iz okolja Windows. Ostali operacijski sistemi, kot so MVS, VMS, OS, ki so vezani na posamezne proizvajalce računalniške opreme (so zaprti) in zgodovinsko obremenjeni (4), po pričakovanjih hitro izgubljajo tržni delež. Tudi s stališča enostavnosti in učinkovitosti delovanja ter uporabe in prenosljivosti sodobnih informacijskih orodij in z njimi razvitih programskih rešitev ti sistemi niso primerni kot gradnik v projektu prenove uporabniških programskih rešitev.

Koncept porazdeljenih baz podatkov je žal v praksi padel na izpitu, saj obstoječa tehnološka platforma (pa tudi tista, ki bo uporabljena v bližnji bodočnosti) zaradi ranljivosti uporabe v mnogih primerih ne nudi zanesljivosti delovanja. Tudi osnovni koncept odjemalec/strežnik se je izkazal, kljub drugačnim pričakovanjem, dražji od centraliziranega (vedno zmogljivejše in dražje delovne postaje - osebni računalniki, prepustnost in zanesljivost lokalnih omrežij, razmišljanja o prenosu skupnih računalniških resursov na osrednji računalnik, iz Interneta izhajajoče potrebe po zamenjavi ekranskih terminalov z mrežnimi računalniki, enostavnejšimi in cenejšimi od osebnih računalnikov (angl. Internet PC, Network PC, ...)).

Vse te ugotovitve in izkušnje kažejo, da je pri prenovi in prehodu na za podjetje bolj ustrezne uporabniške programske rešitve smiselna na eni strani uporaba osrednjega, na UNIX-u delujočega strežnika baze podatkov, na uporabniški oziroma mrežni strani pa nabava uporabniških rešitev, delujočih v okolju Microsoft NT - Windows. Osrednjemu računalniku podjetja prepuščamo torej skrb za podatke, omrežja osebnih računalnikov pa so namenjena "uporabniško prijaznemu" povezovanju z uporabniki.

Tudi odločitev o informacijskem orodju oziroma o razvojnem okolju na področju informatike pomeni za

podjetje eno ključnih strateških odločitev. Tudi v naši praksi se je zlasti pri zahtevnih programskih rešitvah pri velikih količinah podatkov pokazala kot izključno upravičena usmeritev v integrirana informacijska orodja. V primeru posamezne izbire nepovezanih orodij se je pokazala kot kritična predvsem povezava med orodjem za razvoj programov in krmilnim sistemom baz podatkov, kar je v mnogih okoljih pripeljalo do kočljivih okoliščin in dvomov o smiselnosti in upravičenosti uporabe sodobnih informacijskih orodij. Na tem področju izhaja veliko problemov in razočaranj, ki se kažejo na koncu v neakovostnih podatkih podatkovne baze, iz nekritične uporabe povezovalnih vmesnikov med različnimi informacijskimi orodji (npr. ODBC). Priporočamo, da povezovalnih vmesnikov ne bi uporabljali v primeru, ko z njimi neposredno spreminjamo vsebino podatkov v bazi podatkov (npr. SQL ukazi INSERT, UPDATE in DELETE). Takšne aktivnosti lahko brez posebnih nevarnosti opravimo le z integriranimi orodji.

Z ekonomskega stališča je prehod na nova informacijska orodja, gledano le s stališča informatizacije poslovanja, težko upravičljiv. Vložek v novo tehnologijo pa se obrestuje skozi poenostavitev, racionalizacijo in standardizacijo poslovanja in na tej osnovi razviti objektivno usmerjeni bazi podatkov in uporabniških rešitev. Baza podatkov ob samih podatkih zajema tudi vsa poslovna pravila, kar skrajšuje razvoj ter poenostavlja in bistveno znižuje obseg vzdrževanja aplikativnih rešitev. To dejstvo ocenjujemo kot posebno pomembno, saj je v življenskem obdobju razvoja in delovanja IS vzdrževanje udeleženo z deležem 70 - 90 % vseh stroškov IS.

Ob prenovi rešitev je potrebno izbrati informacijsko orodje, ki pokriva celoten cikel načrtovanja in razvoja rešitve (od poslovnega modeliranja do razvoja baze podatkov in pripadajočih procedur ter izdelave programov), za katerim stoji proizvajalec z relativno velikim svetovnim tržnim deležem in je predvsem kot razvojno orodje baz podatkov in programskih rešitev v našem okolju širše uporabljeno (na voljo ustrezna podpora in praktične izkušnje uporabe). V našem okolju je v tem trenutku primerno predvsem informacijsko orodje Oracle, ki skoraj v celoti ustreza navedenim pogojem.

7 Kakšni so ciljni rezultati projekta prenove rešitev?

Prenova uporabniških programskih rešitev je, kot vidimo, rezultat projekta, ki se mora opredeliti širše in si že v svoji zasnovi zastaviti cilje, ki bodo vplivali na razvoj podjetja tako v poslovnem kot v tehnološkem pogledu. Prenova mora potekati postopno, saj bi bila v nasprotnem primeru neobvladljiva in kritična prevsem ob prehodu iz obstoječega stanja na uvedbo novega načina dela in nove rešitve. V prvem koraku zajamemo ključne poslovne procese, ki s stališča poslovnega rezultata za-

jemajo do 80% vseh poslovnih aktivnosti podjetja.

S **poslovnega vidika** se morajo naložbene aktivnosti prvega koraka prenove povrniti skozi:

- integracijo in preново poslovnih procesov, potrebno za učinkovito izvajanje procesov,
- modernizacijo načina dela v delovnih okoljih v podjetju, ki naj izboljša kakovost dela posameznikov, skupin in podjetja na delovnih področjih in nalogah,
- enostavno organiziranje, vodenje in načrtovanje ter vzdrževanje skupinskega dela (Groupware) in krmiljenja delovnega procesa in pretoka dokumentov (Workflow Management),
- zagotavljanje celovitega pregleda nad vsemi resursi in cenami podjetja, ključnimi kupci in dobavitelji, konkurenti ter ostalim okoljem,
- celovito obvladovanje in spremljanje izvajanja poslovnega procesa,
- sprotne informacije o posledicah poslovnih odločitev,
- pregled in sprotno zagotavljanje najustreznejših finančnih virov,
- agregacijo in pripravo ažurnih podatkov za poslovno odločanje.

Tehnološki vidik se kaže v zagotavljanju osnovnih izhodišč sodobnega trenutnega in bodočega razvoja informatike skozi:

- lahek in enostaven ter s stališča uporabnika transparenten pristop ter prenos podatkov in uporabniških programov v okviru celotnega podjetja med računalniškimi sistemi različnih nivojev in proizvajalcev,
- enostavnejše medorganizacijsko povezovanje različnih podjetij med seboj ter z lokalnim in mednarodnim okoljem,
- varnost naložbe v informatiko, prehod in postopno uveljavljanje novega pristopa k uporabi informacijske tehnologije,
- razvoj prilagodljivih in povezljivih rešitev,
- višjo stopnjo specializacije kadrov, ki niso več obremenjeni z usmeritvami in rešitvami najrazličnejših proizvajalcev opreme in orodij,
- uporabo splošnih svetovno znanih in priznanih informacijskih storitev in razvoj ter nudenje takšnih storitev svojim poslovnim partnerjem,
- uporabo standardnega uporabniškega vmesnika oziroma okolja in orodij (Windows; Microsoft, Oracle).

8 Katere so projektne faze in aktivnosti prenove rešitev?

Projekt prenove uporabniških programskih rešitev poteka v večji ali manjši meri v povezavi ali celo v

okviru projekta prenove poslovnih procesov. Slednji je delno opredeljen v predhodnih poglavjih, podrobneje pa v prispevkih s tega področja. V nadaljevanju podajamo pregled vsebine in poteka projekta prvega koraka prenove in informatizacije poslovanja katerega rezultat so, med ostalim, tudi programske rešitve. Projektne faze so naslednje:

- Načrtovanje projekta
- Snovanje prenove
- Priprava prehoda
- Izvedba

V fazi načrtovanja projekta opravimo splošne aktivnosti v zvezi s postavitvijo projekta in projektne tima, opredelimo področje, cilje in pričakovane rezultate prenove in pridobimo podporo vodstva podjetja. Projekt in njegove možnosti izvedbe ocenimo s tehničnega, ekonomskega, strateško političnega in kadrovske psihološkega stališča. Skratka, v tej fazi opredelimo področje prenove (proces in postopke) ter ocenimo in zagotovimo za to potrebne resurse.

Faza snovanja prenove se prične z načrtovanjem (taktičnim), ugotavljanjem postopkov in pristopov k prenovi. V tej fazi se člani projektne tima, katerega se običajno razširi z dodatnimi izvajalci posameznih nalog, pripravi na spremembe in zagotovi njihova podpora in aktivno sodelovanje. Pri tem je ključna ugotovitev glavnega uporabnika nove rešitve in zagotovitev njegove vodilne vloge pri projektu. Projektne tim in vse izvajalce seznanimo z metodološkimi izhodišči, metodami in tehnikami dela pri projektu. Izdelamo poslovni model, ki na nivoju aktivnosti opredeljuje trenutno izvajanje poslovnega procesa. Podrobneje, v smislu ciljev in možnosti prenove obravnavanega poslovnega procesa, izdelamo tehnološka, ekonomska, strateško politična in kadrovske psihološka izhodišča, opredeljena v predhodni fazi. Oblikujemo predlog prenove in taktični načrt prehoda (naslednje faze prenove).

V fazi priprave prehoda se posvetimo vprašanju, kako izvesti željene spremembe, zasnovane v predhodni fazi. Največkrat ta faza izhaja in je pogojena z možnostmi uporabe sodobne informacijske tehnologije. Opredelimo vse spremembe, potrebne pri izvajanju že obstoječih in prenovljenih postopkov. V projektne tim vključimo nove člane s specifičnimi tehnološkimi znanji, potrebnimi v tej fazi projekta (informatiki). Izvedemo tudi izobraževanje bodočih razvijalcev in uporabnikov rešitev v podjetju. Koristna je tudi pilotska postavitev bodoče rešitve, ki se izvede v omejenem obsegu in je namenjena prototipnemu razvoju v naslednji fazi izvedbe. Na tej osnovi se izdelata dokončni predlog in načrt izvedbe in uvedbe nove rešitve.

Izvedba pomeni razvoj in uvedbo v fazah snovanja prenove in priprave prehoda ugotovljenih izhodišč. Izdelamo podroben načrt izvedbe in uvedbe novih or-

ganizacijskih in tehnoloških (informacijskih orodij in uporabniških programskih rešitev) rešitev. Ob tem izdelamo načrt preizkušanja (testiranja) zlasti novih avtomatiziranih postopkov in načrt izobraževanja uporabnikov. Instaliramo orodja ter izdelamo, dokumentiramo in uvedemo uporabniške programske rešitve, preizkusimo njihovo delovanje in izvedemo vsa potrebna izobraževanja za njihovo uporabo. Na koncu ocenimo izvršeno prenovo v smislu v fazi načrtovanja prenove opredeljenih ciljev.

Projektne aktivnosti prvega koraka prenove uporabniških programskih rešitev oziroma procesa informatizacije lahko glede na njihov potek strnemo v naslednjih točkah:

1. Projektne aktivnosti informatizacije so pogojene s predhodno zagotovitvijo funkcije skrbništva podatkov in baze podatkov, izbiro in instalacijo ustreznega orodja CASE ter izdelavo enotnega modela prenovljenih poslovnih procesov. Le-ta je osnovno izhodišče, iz katerega izhaja aktivnost izdelave podrobnega modela podatkov, ki opredeljuje informacijske potrebe podjetja ter pomeni, ob primerjavi s podatkovnimi modeli posameznih tržno razpoložljivih programskih rešitev, osnovno izhodišče za odločitev o nakupu, prilagajanju ali lastnem razvoju programskih rešitev.
2. Model podatkov je načrt in izhodišče projektne aktivnosti zasnove in izdelave baze podatkov podjetja.
3. Vzporedno s temi aktivnostmi gredo aktivnosti izbire, nabave in uvedbe nove tehnološke platforme, informacijskih orodij, orodij za krmiljenje delovnega procesa (Workflow) in uporabniških programov.
4. Sledijo izvedbeni projekti postopne zamenjave obstoječih uporabniških programov in orodij z novimi ter preureditev podatkov in prenos v novo bazo podatkov.
5. Vzporedno mora potekati tudi izobraževanje informatikov in uporabnikov v podjetju za uporabo nove informacijske tehnologije. Izobraževanje lahko poteka s pomočjo tečajev oziroma z izvedbo delavnic.

Kakovost in uspešnost izvedbe projektne aktivnosti priprave izvedbenih projektov je odvisna od kakovosti modelov posameznih poslovnih procesov in podatkovnega modela podjetja. Ker je ta aktivnost posledično najvplivnejša, ji je potrebno posvetiti posebno pozornost.

9 Katera so ključna pravila prenove rešitev?

Ugotovljamo, da postaja vloga informatike pri prenovi poslovanja vse bolj pomembna. Tudi problematika, ki jo

pri tem v procesu informatizacije obravnavamo, postaja vse kompleksnejša. Tudi nove, sodobne uporabniške programske rešitve, njihovo načrtovanje, razvoj ali prenova, posebno pa prilagajanje informacijskim potrebam in uvajanje, postajajo vse zahtevnejše. Zato namesto zaključka podajamo tri pravila, ki jih je pri prenovi uporabniških programskih rešitev vredno upoštevati. Izhajajo sicer iz pravil obvladovanja projektov Nase, na našem obravnavanem področju pa bi se lahko glasila (5):

Pravilo št. 1: *Bodimo prepričani, da vemo, kaj bi radi naredili.*

Mnoge odločitve so sprejete brez predhodne analize in razumevanja dejanskih ciljev. V praksi je pri nas to pravilo največkrat kršeno pri razvoju ali nakupu odločevalskih (beri: direktorskih) informacijskih sistemov in nakupu celovitih, "vseobsegajočih", največkrat tujih (beri: uvoženih) programskih rešitev. V obeh primerih je napačna odločitev storjena na začetku; je rezultat napačne, necelovite in nedosledne opredelitve dejanskih potreb. Na splošno pa za razvoj in prenovu programskih rešitev velja, da so vse resne napake storjene prvega dne.

Pravilo št. 2: *Med možnimi izberimo razvojno pot, katere posledice bodo v primeru neuspeha najmanj boleče.*

Ponavadi na projektih informatizacije težko ocenimo vrednost ali dodano vrednost projekta, saj je večina rezultatov v tem smislu težko merljiva. Vseeno pa v večini primerov lahko ocenimo, kako se bo posamezna razvojna pot odvijala v najboljšem in najslabšem primeru in kakšne bodo posledice odločitve. Zavedajmo se, da ima neuspeh projekta informatizacije oziroma prenove rešitev, ki je soodvisen in pogojuje prenovu poslovanja, lahko katastrofalne posledice za prihodnjo poslovno uspešnost podjetja. Največ napak na tem področju je bilo storjenih, ker načrtovalci rešitev niso predvideli postopkov reševanja posledic neuspeha ali nepredvidljivosti. Včasih je prihodnost težko ali

nemogoče predvideti, vendar jo je neodgovorno prezreti.

Pravilo št. 3: *Skrčimo kompleksnost obravnavane problematike.*

Znižanje kompleksnosti, z razdelitvijo področja obravnave na manjše, še obvladljive dele, omogoča analizo in spremljanje posledic naših posameznih odločitev. Tako poslovne procese s pomočjo dekompozicije razdelimo na postopke in te na neposredno obvladljive aktivnosti, ki jih povezuje podatkovni tok in so predmet postopka informatizacije ter razvoja rešitev. Pri slednjem največkrat velja pravilo, da so najenostavnejše rešitve največkrat najboljše ali v angleščini: KISS ("keep it simple, stupid" ali, bolj milo rečeno, "keep it short and simple").

REFERENCE

- (1) Gričar, Jože:
Izkušnje z uporabo RIP-a in elektronskega poslovanja v Kanadi, Ljubljana, Uporabna informatika, 3/1994, str. 31-32.
- (2) Kovačič, Andrej:
Organizacijska in informacijska prenova poslovnih procesov, Portorož, Zbornik, Dnevi slovenske informatike, 1996, str. 37 - 46.
- (3) McGuckin, Paul:
Server Vendor Selection - Tactical Decision, Strategic Milestone, Lake Buena Vista, Florida, Gartner Group, str. 1-17.
- (4) Kovačič, Andrej:
Ekonomski vidiki uporabe sodobne informacijske tehnologije, Portorož, Zbornik, Informatika '93, 1993, str. 18-25.
- (5) Elliott, Chris:
Managing the business system, v Heller, Robert (urednik): Managing 96, London, Sterling Publications Limited, 1996, str. 18-19.

◆

Doc. dr. Andrej Kovačič je zaposlen kot projektant in direktor podjetja PRIS Consulting, ter kot univerzitetni predavatelj predmetov Informatika na Visoki upravni šoli in Informacijska tehnologija in Podatkovne strukture in baze na Ekonomski fakulteti v Ljubljani. Je dolgoletni organizator in programski vodja portoroških srečanj informatikov ter ustanovni predsednik in član izvršilnega odbora združenja za informatiko in računalništvo pri GZS. V letu 1994 je opravil izobraževanje in izvajal koordinacijo mednarodnega projekta PHARE s področja prenove poslovnih procesov, v letu 1995 pa je pridobil tudi naziv veščaka s področja upravljanja in ravnateljstva, ki mu ga je podelila Zveza ekonomistov Slovenije.

◆

PRENOVITEV INFORMACIJSKE PODPORE POSLOVNEMU SISTEMU

Ivan Vežočanik
Razvojni center Celje, IRC d.o.o.

Povzetek

Prenovitev informacijskih sistemov (reinženiring, sestopanje, povratno inženirstvo) je relativno nova tehnologija, ki vpliva tako na proces vzdrževanja starih kot tudi na načrtovanje razvoja novih aplikacij. Večina obstoječih informacijskih sistemov je obremenjena z zastarelostjo in problematiko povezljivosti posameznih aplikacij. Rešitev je v prehodu v nova, sodobnejša informacijska okolja, v prenovitvi obstoječih in v razvoju novih aplikacij. V članku so predstavljeni model možne prenovitve in izgradnje aplikacije v naših poslovnih okoljih ter možnosti uporabe sodobnih tehnologij in tehnik.

Abstract

Information systems renovation (reengineering, downsizing, reverse engineering) is comparatively a new technology. It affects not only maintenance of old applications but also necessitates a new approach to application development engineering. Most of the existing user applications should be substituted by new, object-oriented / relational database based applications. The paper covers answers to some critical problems concerning system renovation with considering the present situation in the information technology use in Slovenia.

Ključne besede: prenovitev aplikacije, obrnjeno inženirstvo, CASE orodje, sestopanje, reinženiring, poslovni sistem, projektno vodenje, vizija razvoja informatike, strategija, arhitektura informacijskega sistema, kritični dejavniki uspeha.



Uvod

Cilj vsakega poslovnega okolja je v čimboljši informacijski podpori poslovnim procesom oz. poslovnemu sistemu kot celoti. Kakšna je raven te informacijske podpore, je odvisno od posameznega poslovnega okolja in njegovih dejanskih informacijskih potreb. Misel, da je potrebno informacijsko podpreti vsak poslovni proces (mogoče tudi zaradi prestižnega značaja), je ravno tako vprašljiva kot misel, da je informacijska podpora skoraj nepotrebna. Odgovor na vprašanje: "Kakšna informacijska podpora poslovnim procesom oz. poslovnemu sistemu kot celoti je potrebna?", da poslovni sistem funkcionira "optimalno" v vseh pogledih, ni napisan v nobeni knjigi. Ta odgovor si vsak poslovni sistem napiše sam, vendar do odgovora ne vodi enostavna pot, predvsem pa ne do realizacije zastavljenih ciljev.

Kakšno je pravzaprav stanje na področju informatike? Če so informatiki še nedavno veliko pričakovali od aplikativnih rešitev na osebnih računalnikih in omrežjih osebnih računalnikov, je takšno razmišljanje pustilo resne posledice pri načrtovanju informacijskih sistemov v smislu povezljivosti in podpore poslovnemu sistemu

kot celoti. Parcialno dobre aplikativne rešitve praviloma odpovedo pri povezovanju na ravni celotnega sistema in učinki za poslovni sistem kot celoto so pod pričakovani.

Da je zmeda še večja, so bila in so še vedno stališča informatikov tudi na tehnološkem področju (tehnološke platforme in koncepti) različna in neusklajena. Če smo še nedavno predstavljali koncept odjemalec-strežnik in porazdeljene baze podatkov kot edino pravilnega, če smo prisegali na odprte sisteme in koncepte omrežij osebnih računalnikov z grafičnim okoljem, da ne govorimo o uporabi informacijskih orodij, se sedaj po enakih načelih spet nakazujejo rešitve, ki so edino pravilne in neoporečne. Vse te spremembe oz. napovedi "pravilnih konceptov" povzročajo pri uporabnikih neodločnost in zmedo. Velikim pričakovanjem pri prenovi in razvoju informacijskega sistema oz. aplikacij sledijo še večja razočaranja.

Pozabiti ne smemo tudi ekonomskega vidika. Ne samo, da so vsa sodobna informacijska okolja draga, tudi stroški procesa prenovitve niso zanemarljivi.

Z vsemi temi dilemami in vprašanji je soočen uporabnik - poslovni sistem, ko je postavljen pred odločitev o prenovitvi informacijskega sistema oz. posameznih aplikativnih rešitev.

Na koncu je potrebno dati odgovor na ključno vprašanje: "Kdaj je informacijski projekt uspešno zaključen oziroma končan?" Obstaja vrsta metodologij in postopkov za ocenitev uspešnosti projekta in njegovega zaključka, vendar bomo v nadaljevanju privzeli naslednjo definicijo:

Informacijski projekt je uspešno zaključen oz. končan, ko je dosežen potreben pogoj: "dosežena skladnost z vizijo razvoja informatike oz. strategijo razvoja informatike", in ko je dosežen zadostni pogoj: "vsí neposredni udeleženci so zadovoljni".

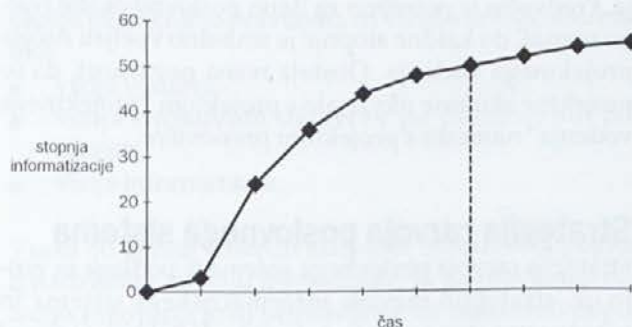
Stanje informatike v naših poslovnih okoljih

Brez dvoma lahko trdimo, da so se v naših poslovnih okoljih uspešno uveljavili novi, sodobni koncepti informatike in hkrati uspešno potrdile nove razvojne smeri. Res je, da v primerjavah in analizah na področju informatike z razvitim zahodnim svetom vedno potegnemo krajši konec kot tisto okolje, ki v smislu vlaganj v informacijsko tehnologijo in informatiko zaostaja za temi ekonomskimi sistemi. Brez pretiravanja pa lahko trdimo, da bolj ali manj uspešno korakamo v smeri sodobnih trendov informatike in da že razpolagamo s tovrstnim znanjem.

Obstoječe stanje informatike v naših poslovnih okoljih lahko orišemo takole:

Vsak poslovni sistem je do neke stopnje informatiziran. Z drugimi besedami, vsak poslovni sistem ima mesto nekje na krivulji informacijskega razvoja (slika 1).

Posamezno poslovno okolje praviloma razpolaga z dokaj pestro računalniško opremo: od velikih (main frame) računalnikov do sodobnih arhitektur računalniških omrežij in osebnih računalnikov.



slika 1: Informacijski razvoj

Do podobne ocene pridemo tudi pri sistemski programski opremi, za katero je značilna velika pestrost in razmeroma slaba izkoriščenost. Srečujemo se s tipičnimi okolji tretje generacije, jeziki četrte generacije, z relacijskimi podatkovnimi bazami in z uporabo CASE ter objektivne tehnologije.

Neurejenost nastopa tudi na področju aplikativnih rešitev: velikemu številu aplikacij se je življenjski cikel že zaključil ali pa se izteka, po drugi strani pa so uspešno realizirane tehnološko najsodobnejše aplikacije na arhitekturi odprtih sistemov, principih uporabnik-strežnik, v okolju jezikov četrte generacije in na relacijskih podatkovnih bazah.

Problem vzdrževanja starih aplikacij je pereč, nastopa problem povezljivosti posameznih aplikativnih okolij (problem integritete) in če temu dodamo še visoke stroške za informatiko ter upoštevamo nesporni zakon sprememb pri razvoju in vzdrževanju informacijskega sistema, potem smo dejansko pred resnim vprašanjem: "Kako naprej?"

Rešitev je potrebno iskati v smislu prenovitve obstoječega stanja informatike na osnovi jasno in realno zastavljenih ciljev. Izhodišče za prenovitev informacijskega sistema je v podpori temeljnemu smotrom in ciljem poslovnega sistema (povečanje konkurenčnosti, povečanje produktivnosti in zmanjšanje stroškov poslovanja ipd). To pomeni, da je prenovitev informacijskega sistema ključnega pomena tudi za prenovitev poslovnega sistema.

Pristop k prenovitvi informacijske podpore

Prenovitev informacijskega sistema mora temeljiti na realnih izhodiščih z jasno zastavljenimi cilji in v okviru finančnih zmožnosti poslovnega sistema. Vzroki za prenovitev aplikacije so lahko v manjkajoči dokumentaciji, prehodu iz enega v drugo tehnološko okolje, v ponovni ali večji uporabljivosti zastarelih, vendar v praksi uporabnih aplikacij, v racionalizaciji procesa vzdrževanja aplikacij, v prehodu z velikih (main frame) računalniških sistemov na delovne postaje in osebne računalnike oz. omrežja osebnih računalnikov (t.i. proces sestopanja), v prehodu na arhitekturo odprtih sistemov, na drugo programsko opremo (relacijske sisteme za upravljanje z bazami podatkov, distribuirane rešitve, interaktivno orientirane rešitve, ...), itd.

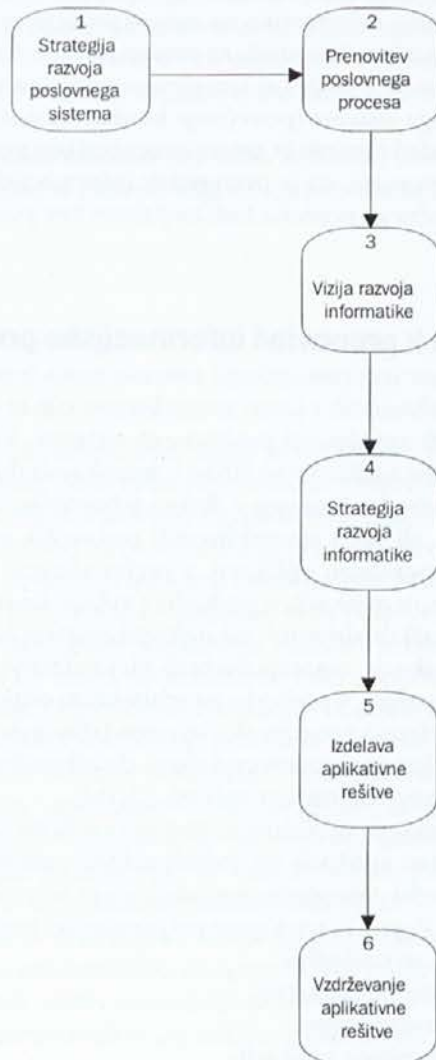
V praksi in literaturi nastopajo različne definicije prenovitve aplikacij oz. informacijskih sistemov s še vedno mehko definicijo metodologij in tehnologij prenovitve. Pojmi, ki nastopajo pri prenovitvi posameznih aplikacij, so naslednji:

- obratno inženirstvo,
- rekonstruiranje,
- reinženiring aplikacij,
- razvoj novih aplikacij.

Praksa je pokazala, da se pri prenovitvi kompleksnejšega aplikativnega okolja poslužujemo vseh zgoraj naštetih metodologij in tehnologij prenovitve, kar je odvisno od posameznega okolja. Pri tem se moramo zavedati, da je vsako poslovno okolje unikat zase in ne moremo računati na neke splošno preverjene rešitve.

S procesom prenovitve informacijske podpore želimo doseči naslednje cilje:

- reševanje problematike vzdrževanja aplikacij,
- zagotovitev dokumentacije in ponovne uporabljivosti aplikacije na višji tehnološki ravni,
- prehod iz ene na drugo strojno opremo,
- prehod iz enega na drugo sistemsko okolje (prehod na sodobna relacijska okolja, prehod iz centraliziranih na distribuirane rešitve, prehod na arhitekturo odprtih sistemov, prehod iz velikih računalniških sistemov na srednje sisteme, delovne postaje in osebne računalnike oz. t.i. sestopanje, ...),
- vključitev aplikacije v sodoben, povezan sistem aplikativnih rešitev z uporabo CASE orodja.



slika 2: Faze prenovitve informacijskega sistema

Proces prenovitve aplikacije je tesno povezan s prenovitvijo poslovnih procesov. Tako kot so v praksi deljena mnenja glede informacijske tehnologije in pravilnosti posameznih rešitev, obstajajo tudi pri prenovitvi aplikacije oz. informacijskega sistema različna mnenja glede vsebine in poteka prenovitve. Da se čimbolj izognemo predhodnim dvomom in pastem, je možna pot prenovitve, ki je opredeljena z naslednjo vsebino in fazami (slika 2):

- Strategija razvoja poslovnega sistema
- Prenovitev poslovnega procesa
- Vizija razvoja informatike
- Strategija razvoja informatike
- Izdelava aplikativne rešitve
- Vzdrževanje aplikativne rešitve



slika 3: Organizacija projekta

Ena izmed najpomembnejših in kritičnih aktivnosti prenovitve aplikacij je določitev projektne skupine za izvedbo prenovitve in njene organiziranosti. Organizacija projekta prenovitve je prikazana na sliki 3, kjer je smiselna trionovska delitev na:

- vodstvo projekta (sponzor projekta, vodja projekta),
- logično zasnovo projekta (odgovorne osebe za posamezna poslovna področja, skupine za izvedbo prenove za posamezna poslovna področja),
- operacionalizacijo (tehnične skupine),

z vsem instrumentarijem in načeli projektnega vodenja. Predvsem je potrebno za dano poslovno okolje realno oceniti, do kakšne stopnje je smiselno vpeljati načela projektnega vodenja. Obstaja resna nevarnost, da se projektne skupine ukvarjajo s projektom "projektnega vodenja" namesto s projektom prenovitve.

Strategija razvoja poslovnega sistema

Strategija razvoja poslovnega sistema je podlaga za vizijo oz. strategijo razvoja informacijskega sistema in njegove uspešne operacionalizacije. V kolikor poslovni sistem nima izdelane strategije razvoja, je zelo težko

govoriti o strategiji razvoja informatike. Z drugimi besedami, vsa izhodišča za uspešen razvoj in prenovitev informacijskega sistema imajo podlago v strategiji razvoja poslovnega sistema.

Prenovitev poslovnega procesa

Prenovitev poslovnega procesa in prenovitev informacijske podpore poslovnemu procesu sta tesno povezani aktivnosti, ki se velikokrat izvajata sočasno, čeprav prenovitev poslovnega procesa postavlja strateške podlage in okvire za prenovitev oz. razvoj informacijskega sistema.

Na osnovi praktičnih izkušenj v naših poslovnih okoljih je smiselno obravnavati prenovitev poslovnega procesa ločeno od prenovitve informacijskega sistema. S tem dosežemo, da pridemo do rezultatov pri obeh prenovitvah ločeno in da prenovitev informatike temelji na realnem poslovnem sistemu.

Vizija razvoja informatike

Vsako poslovno okolje ali poslovni sistem si ustvari lastno vizijo razvoja informatike, ki jo skuša realizirati bodisi z lastnimi močmi bodisi z zunanjo pomočjo. Pri tem nastopi v večini primerov vrsta nevšečnosti in nesporazumov. Zakaj?

Odgovor je v bistvu preprost. Ko pričnemo razmišljati o informatiki, njenem razvoju, njenem mestu v danem poslovnem okolju, njenih delovnih in ekonomskih učinkih, stroških in podobno, in če te kazalce primerjamo še s podobnimi (sorodnimi) ali informacijsko razvitejšimi poslovnimi okolji, potem si relativno hitro ustvarimo lastno mnenje oziroma želje o bodočem informacijskem sistemu: vodstvene strukture se navdušujejo za podporo odločanju in vodenju, tehnični strokovni delavci navadno razmišljajo o optimizacijskih problemih (optimizacija proizvodnega procesa, optimizacija proizvodne linije, ...), finančni delavci o idealnem finančnem sistemu, skladiščni delavci o optimizaciji zaloga, informatiki o lastni viziji sodobnega razvoja informatike.

Izkušnje so pokazale, da se v nekem poslovnem okolju izoblikujejo praviloma tri vizije razvoja informatike:

- vizija vodstva,
- vizija strokovnih delavcev po posameznih področjih,
- vizija informatikov.

Vsaka od zgoraj naštetih skupin si ustvari svoj pogled o nadaljnjem razvoju informatike, ki ga praviloma odločno zagovarja in ni pripravljena na nikakršno popuščanje oziroma sporazumevanje in usklajevanje. Kako razrešiti takšno situacijo?

Najprej je potrebno poizkusiti s sporazumevanjem in usklajevanjem na enakopravni ravni vseh udeležencev. Smiselno je, da vodenje vseh teh aktivnosti in delovnih skupin prevzame oseba s primernim statusom, z vodstvenimi pooblastili in osebnim ugledom. V kolikor takšen pristop ne rodi uspehov, potem je potrebno poklicati zunajo strokovno pomoč, ki izdelava ali pomaga izdelati vizijo nadaljnjega razvoja informatike. Vizija razvoja informatike naj bo izdelana v obliki pisnega dokumenta, ki je napisan realno in jasno, tako da ga razumejo vsi uporabniki oziroma sodelujoči delavci. V primeru, da takšen dokument izdelajo lastni strokovni delavci, ga je priporočljivo dati v preverjanje zunanjim strokovnjakom.

Ne samo, da pri izdelavi dokumenta "vizija razvoja informatike" obvezno sodelujejo strokovni delavci z vseh poslovnih področij, ta dokument oziroma vizijo morajo tudi sprejeti za svojo. Le s takšnim pristopom bo že v osnovi zagotovljen zdrav start projekta prenove informatike.

Strategija razvoja informatike

Strategija razvoja informatike podaja strateško vizijo prenovitve ali razvoja informatike in operativni plan aktivnosti na tem področju za dogovorjeno časovno obdobje (smiselno je planiranje za triletni horizont).

Strategija razvoja informatike je podlaga za pristop k prenovitvi ali razvoju integriranega poslovno - informacijskega sistema v danem poslovnem okolju, ki bo zagotavljal nemoteno operativno delovanje vseh poslovnih funkcij in zagotavljal informacije za odločanje na strateški in taktični ravni. Izhodišča za strategijo razvoja informatike so podana v "strategiji razvoja poslovnega sistema", v "prenovitvi poslovnega procesa" in v "viziji razvoja informatike". Strategija razvoja informatike obravnava celovit poslovni model od organizacijske sheme poslovnih funkcij in entitetnega modela do arhitekture aplikativnih rešitev in s planom definirane prioritete posameznih prenovitvenih in razvojnih aktivnosti. V okviru strategije je definirana tudi strojna in sistemska programska oprema in pa ekonomski vidiki prenovitve oz. razvoja.

Vsebina strategije razvoja informatike naj vsebuje naslednje sklope:

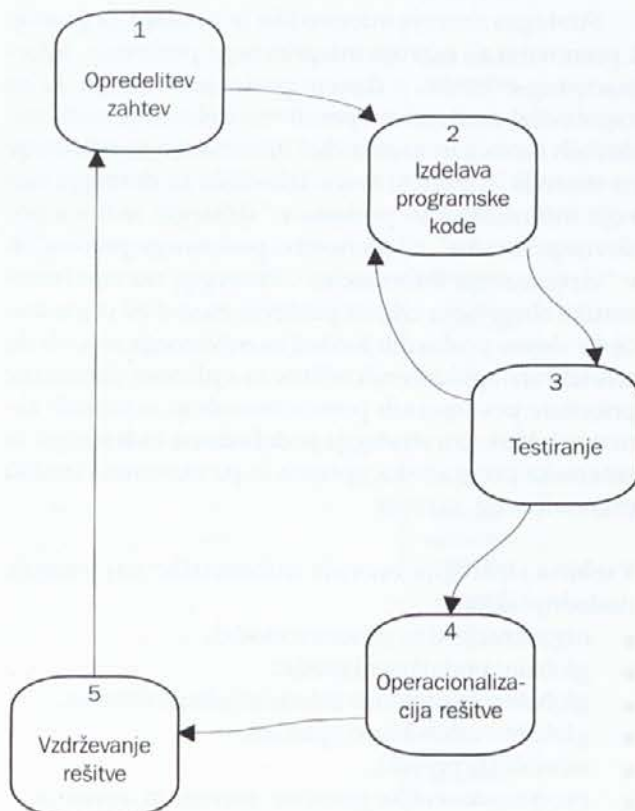
- organizacijski in procesni model,
- globalni podatkovni model,
- globalno arhitekturo informacijskega sistema,
- globalno arhitekturo aplikacij,
- tehnološki projekt,
- ekonomske vidike prenove, razvoja in uvajanja,
- plan prenovitve, razvoja in uvajanja,
- kritične dejavnike uspeha.

Pomembnejši kritični dejavniki uspeha, ki so pogoj za uspešno doseganje zadanih ciljev, so naslednji:

- vizija in realnost izvedbe projekta,
- organizacija poslovanja,
- obvladovanje poslovnih procesov,
- organizacija informatike,
- vodenje projekta prenovitve,
- vodenje uvajanja sprememb v poslovne procese,
- hitro doseganje prvih rezultatov,
- načrtovanje integriranih aplikativnih rešitev,
- poenotenje tehnološkega okolja,
- odzivnost in propustnost sistema.

Izdelava aplikativne rešitve

To je področje, kjer so pričakovanja vedno večja od doseženih rezultatov. Kljub mnogim novim pristopom, metodologijam in orodjem za razvoj novih oz. za prenovitev starih aplikacij se precejšnje število informatikov ukvarja z vzdrževanjem starih aplikacij. Iskanje najustrežnejših metodoloških pristopov, kakor je tudi razvoj orodij za povečanje "storilnosti informatikov", v začetku ni prineslo dramatičnih izboljšav. Te tehnologije so prvenstveno namenjene razvoju novih aplikacij, vendar jih lahko učinkovito uporabimo tudi pri prenovitvi starih aplikacij in vzdrževanju le-teh.



slika 4: Tradicionalni razvoj aplikacije

Življenski cikel razvoja starih aplikacij (t.i. "tradicionalni razvoj aplikacije") je predstavljen na sliki 4. Ključne značilnosti tradicionalnega razvoja aplikacije so naslednje:

- pomanjkljivo poznavanje cilja,
- nezadostna opredelitev zahtev uporabnika,
- pristop po funkcijski hierarhiji,
- pretirana usmeritev k posameznemu (enemu) problemu,
- tipično programsko okolje tretje generacije,
- avtoritativni položaj informatikov,
- nedokumentiranost rešitve,
- omejeno sodelovanje uporabnika.

Aplikativna rešitev je bila v rokah informatikov (delno je to tudi posledica stopnje tehnološkega razvoja), ki so krojili tudi vsebinske in organizacijske komponente poslovnega sistema, v veliko primerih pa brez iniciative informatikov sploh ne bi bilo avtomatizacije. Zaradi nezadostnega sodelovanja uporabnika pri razvoju rešitve in zaradi pomanjkljive dokumentiranosti so se vse dodatne zahteve in spremembe prenašale na proces vzdrževanja aplikativne rešitve. Kljub vsem naštetim težavam in pomanjkljivostim ne smemo pozabiti, da te aplikativne rešitve še vedno operativno delujejo in da so v njih združena znanja organizacijskega, vsebinskega in informacijskega značaja.

Zato je smiselno pristopiti k prenovitvi aplikacije na podlagi izhodišč, podanih v poglavju "Pristop k prenovitvi informacijske podpore". Proces prenovitve aplikativnih rešitev dejansko obsega prenovitev zastarelih delujočih aplikacij (reinjening aplikacije), razvoj novih aplikacij in obrnjeno inženirstvo delujoče tehnološko sodobne aplikacije.

Proces prenovitve zastarele aplikacije je smiselno razdeliti v več razvojnih faz:

- celovita analiza stare delujoče aplikacije ter prenos znanja v zasnove sodobne aplikacije,
- upoštevanje novih dodatnih zahtev,
- uporaba znanja in izkušenj pri vodenju in vzdrževanju stare aplikacije,
- načrtovanje logičnega modela sodobne aplikativne rešitve,
- izdelava fizičnega modela v sodobnem tehnološkem okolju,
- operacionalizacija aplikativne rešitve.

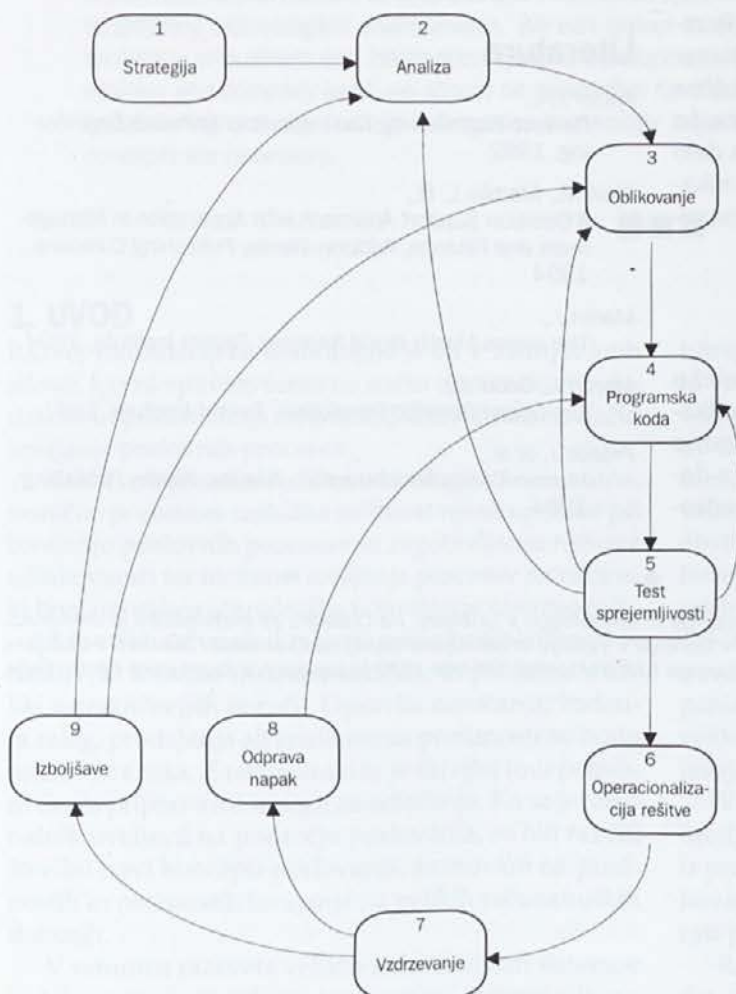
Življenski cikel razvoja (tudi pri procesu prenovitve) sodobne aplikativne rešitve je prikazan na sliki 5. Ključne značilnosti sodobnega razvoja aplikativne rešitve so naslednje:

- izdelana in potrjena strategija razvoja informatike,
- pristop po funkcijskih področjih,
- poudarek na analizi in oblikovanju informacijskega modela,

- poudarek na testu sprejemljivosti,
- aktivno vključevanje uporabnika v vseh fazah razvoja,
- uporaba CASE orodij,
- uporaba baze znanja,
- dokumentiranost rešitve,
- radikalno vzdrževanje aplikativne rešitve.

Funkcijsko orientirana aplikacija se preverja z vrednostnimi verigami (zaporedje opravil po različnih funkcijskih področjih, ki jih je potrebno opraviti, da pridemo do končnega rezultata). S tem dosežemo aktivno udeležbo vseh udeležencev po različnih funkcijskih področjih. Rezultat takšnega pristopa je zadovoljstvo uporabnika. Aplikativna rešitev ni več v rokah informatikov, uporabnik je intenzivno vključen v vseh fazah razvoja aplikacije in vzdrževanje aplikacije "se preseli" k uporabniku.

Strategija razvoja informatike je izhodišče za ves nadaljnji razvoj aplikativnih rešitev, kjer so nedvoumno podane smernice in cilji razvoja s terminsko, tehnološko in vsebinsko opredelitvijo.



slika 5: Sodobni razvoj aplikacije

Analiza: modeliranje entitetnega modela, funkcijskega modela in modeliranje toka podatkov.

Oblikovanje: zajema transformacijo entitetnega modela v podatkovni model (relacijski, objektni), definira modularno strukturo aplikativnih rešitev in definira logiko baze podatkov.

Test sprejemljivosti: namenjen je testiranju potrebne povezljivosti z ostalimi deli delujočih aplikativnih rešitev, po drugi strani pa z njegovo pomočjo testiramo zadostnost aplikativne rešitve. V primeru nezadovoljivih rezultatov testa sprejemljivosti se vračamo v predhodne faze razvoja aplikativne rešitve.

Operacionalizacija rešitve: to fazo sestavljajo naslednje aktivnosti:

- izobraževanje uporabnika,
- instalacija aplikativne rešitve,
- operacionalizacija potrebne povezljivosti z ostalimi operativno delujočimi aplikacijami,
- podatkovna ureditev,
- uglasitev aplikacije (optimizacija baze, performančne izboljšave, ...),
- operativni zagon aplikativne rešitve.

Vzdrževanje rešitve: je uspešno, če je opravljena delitev na operativno vzdrževanje aplikacije (varovanje podatkov, obnova podatkovnega sistema, adaptacija oz. spremembe zaradi sprememb systemske programske opreme ali strojne opreme, ...) in na vzdrževanje aplikativne rešitve, to je na popravljanje skritih napak in izboljšavo aplikativne rešitve.

Vzdrževanje je bilo vedno predstavljeno kot nujno zlo, čeprav pomeni večinsko dejavnost v vseh računalniških centrih. Tudi vzdrževanje aplikacij, ki so plod tradicionalne razvojne poti, ne pomeni samo popravljanje programskih napak, ampak gre velikokrat za pomembno prilagajanje spremembam v poslovnem sistemu, ki so ključne za njegovo delovanje.

Pri sodobnem razvoju aplikacije pa se funkcija vzdrževanja spremeni. Govorimo o radikalnem vzdrževanju aplikativne rešitve - izboljšavah in odpravljanju napak ter vračanjem v predhodne faze razvoja aplikacije (slika 5). Vzdrževanje obravnavamo kot kontinuiran proces razvoja aplikativne rešitve, ki je pod kontrolo uporabnika.

Uporaba CASE orodij

Ključno vlogo pri procesih prenovitve in načrtovanja novih aplikativnih sistemov imajo CASE orodja. Njihova uporaba je v podpori

celotnemu prenovitvenemu ciklu aplikacije, tako v primeru prenovitve zastarele delujoče aplikacije kot v primeru obrnjenega inženirstva ali razvoja nove aplikacije. Pri tem moramo zadostiti izhodišnim zahtevam:

- nov način mišljenja,
- poznavanje poslovnega področja,
- poznavanje metod in tehnik razvoja informacijskega sistema,
- projektni pristop,
- aktivno vključevanje uporabnika v vseh fazah razvoja,

da dosežemo želene rezultate:

- hiter razvoj oz. hitre rezultate,
- prototipni model,
- doseganje konstrukcije aplikativnega sistema do izdelave kode,
- prenovitev oz. razvoj je podprt z bazo znanja (enciklopedijo),
- dokumentiranost,
- lažja in učinkovita komunikacija z uporabniki.

Uporaba CASE orodja je pri prenovitvi aplikativnih rešitev oz. informacijskega sistema nujna, vendar morajo biti predhodno izpolnjene zgoraj naštetih izhodiščne zahteve, sicer uporaba CASE orodja ni učinkovita. In končno, CASE orodje je pripomoček, ki nam olajša delo in nas hitreje in kvalitetneje privede do rezultatov, nikakor pa nam ne rešuje vsebine problema ali celo postavlja rešitve.

Ekonomska upravičenost

Stroški vzdrževanja so največja postavka v deležu stroškov celotnega življenjskega cikla delovanja aplikacije. Zavzemajo delež nad 60%. Če temu neposredno merjenemu učinku (učinki merjeni neposredno z denarjem) dodamo še posredno merjene učinke (povečan-

je konkurenčne zmožnosti, kakovost, izboljšave, ...), potem je tudi s finančnega vidika očitno, zakaj je prenovitev informacijskih sistemov in aplikacij zanimiva

Zaključek

Proces prenovitve je prav gotovo ena najpomembnejših usmeritev na področju informatike. Tehnologija prenovitve je močno vezana na uporabo CASE orodij, medtem ko se med temeljnima aktivnostima, razvojem in vzdrževanjem izgublja meja.

Obvladljivost sprememb in hitri odzivni časi na zahteve za spremembe in dopolnitve so naslednji pomembni razlog, da stremimo h kar najbolj sodobnim in obvladljivim aplikativnim okoljem.

Motivov in razlogov za prenovitev aplikativnih rešitev je veliko in so vsebinskega, organizacijskega, tehnološkega in ekonomskega značaja

Zakovitost, ki še ni nikdar razočarala niti informatikov niti uporabnikov, je preprosta in večna: "Edina stalnica v informatiki so spremembe."

Literatura

- Frazer J.A.,
Reverse Engineering, The Institute of Software Engineering, 1992
- Klein M., Methlie L. B.,
A Decision Support Approach with Application in Management and Finance, Addison-Wesley Publishing Company, 1994
- Martin J.,
The James Martin World Seminar, Savant Institute, 1994
- Martin J., Oddel J.,
The Object Oriented Revolution, Savant Institute, 1991
- Preece J. et al.,
Human-Computer Interaction, Addison-Wesley Publishing, 1994

◆
Dr. Ivan Vezočnik je diplomiral na Fakulteti za naravoslovje in tehnologijo v Ljubljani, na Oddelku za matematiko in mehaniko, smer Tehnična matematika. Po diplomii se je zaposlil v Gorenju v Velenju in nadaljeval študij na Ekonomski fakulteti v Ljubljani, smer Operacijsko raziskovanje, kjer je magistriral in kasneje tudi doktoriral. Od leta 1979 je zaposlen v Razvojnem centru Celje, kot strokovni delavec in svetovalec za informatiko.

OBJEKтна USMERITEV PRI RAZVOJU POSLOVNIH IN INFORMACIJSKIH SISTEMOV

Bogdan Vožič
PRIS Consulting, Ljubljana, Slovenija

Povzetek

Objektna usmeritev je zelo blizu načinu kako človek vidi svet okoli sebe, zato je v tem trenutku objektna usmeritev največja pomoč pri razvoju poslovnih in informacijskih sistemov. Kot univerzalna tehnologija se širi v različna tehnološka okolja in pričakuje se, da bo ob koncu stoletja postala prevladujoča tehnika pri razvoju poslovnih in informacijskih sistemov. Pri uvajanju sodobne objektno tehnologije, pri organiziranosti ter poslovanju združb, se je potrebno zavedati, da za hiter in učinkovit razvoj ali prenovo poslovnih procesov ni zadosten porok sodobna informacijska (objektna) tehnologija, temveč so potrebni nova znanja, pristopi in koncepti.

Abstract

The object approach is very near the way how human beings see the world. That is why at present time object approach is the most helpful method at business and information systems development. As a universal technology, it is spreading to different technological environments. We can expect that at the end of the century it will become the dominant technique of business and information systems development. When implementing modern object technology, at organization and company level, we should be aware that for a quick and efficient development of business processes or reengineering not only modern information technology (object technology), but also new knowledge, approach and concepts are necessary.



1. UVOD

Razvoj informacijske tehnologije je bil v zadnjih letih silovit, kar ni vplivalo samo na način obravnavanja podatkov in pridobivanja informacij, temveč tudi na način izvajanja poslovnih procesov.

Kakorkoli je računalniška tehnologija fascinanta, resnično pozornost zaslužita možnost njene uporabe pri izvajanju poslovnih procesov in zagotavljanju njihove učinkovitosti ter možnost izvajanja procesov na načine, ki brez uporabe računalniške tehnologije niso mogoči.

Z uvedbo računalnikov se je zelo povečal obseg podatkov, ki se lahko shranijo, obdelajo in prikažejo v obliki najrazličnejših poročil. Opravila naročanja, vodenja zalog, prodajanja ali analiziranja profitnosti ne bodo nikoli več enaka. Z računalniki se je skrajšal tudi potreben čas za pripravo informacij za odločanje. Ko se je računalnik uveljavil na področju poslovanja, so bili razviti številni novi koncepti poslovanja, zasnovani na prednostih in možnostih izvajanja na velikih računalniških sistemih.

V vrhuncu razcveta velikih računalniških sistemov so bila osnovna merila za ocenjevanje računalniškega sistema hitrost in pomnilniške zmogljivosti. Večji in

hitrejši ko je bil računalnik, več informacij je lahko dal in več aplikacij je lahko izvajal. S tem argumentom, za večje računalnike, so nekateri dokaj nasilno nastopali do tistih, ki so zagovarjali elegantnejše rešitve problemov. Kadarkoli so potrebe poslovnega sistema narekivale obravnavo večjega števila podatkov, so nadgradili obstoječe računalnike ali jih zamenjali z večjimi in hitrejšimi. Večje in hitrejše je pomenilo vedno boljše in rešitev za vsak poslovni problem je bila v hitrejši rasti. Takšna usmeritev je bila na neki način prevara, saj je šlo v resnici za podpiranje rasti brez malenkostnega odstopanja od centraliziranega nadzora, in je vodila k razvoju velikih poslovnih birokracij, v katerih je vsako podjetje izvajalo večino poslovnih funkcij.

Veliki računalniški sistemi so bili zelo primerno orodje za izvajanje poslovne filozofije, ki je zagovarjala predvsem monolitne samozadostne združbe, in z njihovim stalnim razširjanjem so zadovoljivo obvladovali rast poslovnega sistema.

Razcvet centralizacije in omenjene poslovne filozofije, večji posel je boljši, je trajal več kot štiri desetletja. Zadnjih deset let so bile predpostavke, povezane s cen-

tralizacijo in razširjanjem poslovnih organizacij, pod velikim pritiskom. Nastale razmere in vnovična presoja o ustreznosti obstoječih oblik poslovanja so mnoge korporacije privedle do odločitve, da so opustile neučinkovite in arhaične poti izvajanja poslovanja. Tradicionalne korporacije s svojim nadzorom "top-down" in monolitno strukturo so se zdele kot omahujoči dinozaver, kateremu so šteti dnevi. Merilo, ki vključuje hitrost procesorja in velikost pomnilniških zmogljivosti, v devetdesetih letih ne velja več. Siloviti razvoj informacijske tehnologije v zadnjih letih ne vpliva samo na način obravnavanja podatkov in pridobivanja informacij, temveč tudi na način poslovanja. Nova tehnologija je z delovnimi postajami na delovnih mestih, povezanih v računalniško mrežo, omogočila decentralizirano obdelavo podatkov in s tem tudi vodenje poslovanja. Nova tehnologija z ustreznimi računalniškimi rešitvami in ob neprimerno nižji ceni zagotavlja učinkovito podporo poslovanju in odpira številne nove pristope pri obvladovanju poslovnega sistema.

Osební računalniki, majhne računalniške mreže, dva ali trije zmogljivi strežniki so dovolj za uspešno obvladovanje poslovnih procesov. Na vsakem osebnem računalniku lahko dobimo želeno informacijo v primeri obliki za izvajanje odločitvenega procesa. Ažurno lahko posredujemo informacije o zahtevah vsem zainteresiranim odločevalcem v podjetju ali zunaj njega.

Sinergija nove tehnologije in novih prijemov poslovanja se zdi idealna. Ali je res? Z uporabo osebnih računalnikov in možnostmi, ki nam jih nova tehnologija ponuja, smo preprosto pozabili na okolico domene osebnih računalnikov. Pozabili smo na velike računalnike, na katerih še vedno vsak dan poteka večina transakcij, vendar niso več v središču naše pozornosti. Uvedba nove tehnologije brez spremembe načina obravnavanja podatkov in novih orodij ne prinese pričakovanih rezultatov. Nastane celo dokajšnja zmeda pri zagotavljanju ustreznih informacij za uspešno izvajanje poslovnih procesov.

Ob tem spoznanju je treba bistveno spremeniti način razmišljanja in pristopa k analiziranju, oblikovanju, razvijanju in izvajanju poslovnih sistemov. Veliko novih konceptov, idej in zamisli (strukturna analiza in zasnova, integrirane baze podatkov, arhitektura odjemalce-strežnik, distribuirana obdelava podatkov, paralelno procesiranje, grafični vmesniki) je nastalo kot posledica nove tehnologije. Vsi ti novi pristopi skušajo spremeniti način oblikovanja, razvijanja, uporabe in prikazovanja sistemov. Toda ideje in koncepti niso obrzdali računalniške industrije pri razvoju obsežnih aplikacij, ki obdelajo veliko število podatkov v čim krajšem času. Izbira med nakupom hitrega procesorja in novimi načini obravnavanja podatkov se je pri veliki večini računalniških strokovnjakov končala z izbiro novega, večjega računalnika. Menili so, da povečanje hitrosti

obdelovanja podatkov vedno izboljša aplikativno rešitev, medtem ko novi pristopi obravnavanja podatkov prinašajo večje tveganje in so predvsem dražji. Zato so kljub novi tehnologiji resnične spremembe v našem temeljnem videnju računalniške obravnave podatkov počasne. Vzrok za nasprotovanje spremembam je treba iskati tudi v človekovi naravi.

2. RAZVOJ PRISTOPOV IN METODOLOGIJ GRADNJE INFORMACIJSKIH SISTEMOV

Uvajanje informacijske tehnologije v poslovanje podjetij sega v zgodnja šestdeseta leta. Ti začetki so bili omejeni na posamezne uporabniške rešitve za posamezne rutinske postopke, v povezavi z velikimi količinami podatkov. O metodologiji gradnje informacijskih sistemov oz. načrtovanju uporabniških rešitev še ni bilo mogoče govoriti. Proces gradnje informacijskih sistemov je bil z metodološkega vidika prvič obdelan šele sredi sedemdesetih let. Prvi poskusi uvajanja metodologij v načrtovanje uporabniških rešitev so bili usmerjeni v podporo najnižji ravni oziroma specifikiranju programske kode, medtem ko za fazi analize in načrtovanja informacijskega sistema še ni bilo ustreznih metod.

Razvoj računalniške opreme in večanje zahtev uporabnikov sta privedla do raziskav na področju gradnje informacijskih sistemov. Nastale so prve bolj ali manj celovite metodologije. Iz tega obdobja je znan izraz 'življenjski cikel', ki v grobem predstavlja postopnost procesa gradnje informacijskih sistemov. Praksa je pokazala, da so metodologije, ki temeljijo na filozofiji življenjskega cikla, mnogo manj prispevale k učinkovitosti gradnje, kot je bilo pričakovati.

V drugi polovici sedemdesetih let in v prvi polovici osemdesetih sta bila v več različicah razvita dva modela gradnje informacijskih sistemov. Prvi, tako imenovani programski inženiring, temelji na faznosti procesa gradnje, pri čemer so v prvih fazah analizirane funkcije sistema, za potrebe teh funkcij pa je izdelana fizična baza podatkov. Drugi model, imenovan informacijski inženiring, pa najprej analizira logični model podatkov in zanj izdelava posamezne funkcije. Na podlagi teh dveh temeljnih modelov je bila izdelana cela vrsta različnih metodologij, katerih skupna značilnost je zaporednost izvajanja faz gradnje informacijskega sistema.

2.1 Aplikativni pristop

Aplikativni pristop, imenovan tudi tradicionalni pristop, je osredotočen na parcialno obdelavo podatkov, potrebnih v posamezni organizacijski enoti organizacije, oziroma na obravnavo posameznih poslovnih procesov. Zahteve uporabnikov so se reševale z razvojem posameznih programskih rešitev, kot so plače, skladiščno poslovanje, glavna knjiga itd. Vsak uporabniški program, ki je bil razvit, je bil postopkovno usmerjen

in zasnovan tako, da je zadovoljil potrebe posameznega oddelka ali skupine uporabnikov. Ni bilo modela ali plana, na podlagi katerega bi potekal razvoj uporabniških rešitev.

Vedno nove zahteve uporabnikov po informacijah so sprožile razvoj novih programskih rešitev in prilaganje obstoječih. S tem so nastale nove podatkovne strukture, ki ostajajo domena posameznih programov ter se obravnavajo ločeno, neodvisno od drugih podatkovnih struktur, s katerimi so v logični povezavi. Podvajanje in nekonsistentnost podatkov postajata neobvladljiv problem, kakovost informacij pa dvomljiva.

Slabosti, ki izhajajo iz takšnega pristopa, so:

- nenadzorovano podvajanje podatkov,
- nekonsistentnost podatkov,
- neprilagodljivost podatkov,
- omejena možnost delitve podatkov,
- majhna možnost uvajanja standardov,
- nizka produktivnost programerjev,
- velika potreba po vzdrževanju uporabniških programov.

2.2 Podatkovni pristop

Spoznanje, da uvedba nove tehnologije brez spremembe načina obravnavanja podatkov ne prinaša pričakovanih rezultatov in so podatki eden bistvenih virov vsake organizacije, tako kot ljudje, material in drugi viri, ter je celotna organizacija odvisna od zmožnosti pravilnega načrtovanja in upravljanja teh virov, je privedlo do spremembe zasnove in gradnje informacijskih sistemov.

Danes postajajo informacije nepogrešljiva lastnina organizacije, računalniško zasnovani informacijski sistemi pa se gradijo s pomočjo modernih metodologij in tehnologij, pri čemer je zadovoljevanje informacijskih potreb usmerjeno na vse ravni v organizaciji: operativno, taktično in strateško.

Za zagotavljanje učinkovitega načrtovanja in upravljanja podatkovnih virov je bistveno vprašanje metodologije, ki bo omogočala jasno in nedvoumno definicijo podatkov in oblikovanje stabilnih modelov podatkov, ki pa bodo hkrati dovolj prožni, da bodo sledili dinamiki razvoja poslovnega sistema.

Za aplikativni pristop k izgradnji informacijskih sistemov je bilo značilno reševanje posameznih poslovnih procesov brez načrtovanja celotnega informacijskega sistema, medtem ko sta za podatkovni pristop potrebna temeljita analiza trenutnih in prihodnjih informacijskih potreb ter načrt oblikovanja ustrezne podatkovne baze, ki je vključen v načrt gradnje informacijskega sistema, ta pa je seveda bistveni element strateških načrtov organizacije.

Podatkovni pristop postavlja vprašanje, kakšna mora biti baza podatkov, da bo zadovoljila trenutne in

prihodnje informacijske potrebe organizacije. Rešitev je v razvoju struktur podatkov, ki se zelo malo spreminjajo v primerjavi s spremembami v poslovanju ali organizacijskih oblikah, katerim pa se morajo prilagajati programske rešitve.

Podatkovni pristop ima v primerjavi z aplikativnim pristopom vrsto prednosti. To so:

- minimalno podvajanje podatkov,
- konsistentnost podatkov,
- neokrnjenost podatkov,
- deljivost podatkov,
- uveljavljanje standardov,
- poenostavljeno razvijanje uporabniških rešitev,
- poenotena zaščita podatkov,
- zasebnost podatkov,
- podatkovna neodvisnost,
- zmanjšana potreba po vzdrževanju uporabniških programov.

3. Objektna usmeritev

Objektna usmeritev je poseben pristop k razvoju modelov kompleksnih sistemov, sestavljenih iz velikega števila entitet in dogodkov, obravnavanih kot objekti. Odnose med entitetami ali dogodki predstavljajo povezave med objekti. Dogodek, prek katerega objekt vpliva na drugi objekt, imenujemo povezavo med objekti. Lastnosti objektov so izražene z atributi objektov in so lahko statične ali dinamične.

Objektno usmerjen pristop je zelo blizu načinu kako človek vidi svet okoli sebe, zato je v tem trenutku objektna usmeritev največja pomoč pri razvoju dobrih modelov. Danes je veliko poskusov modeliranja informacijskih in tudi poslovnih sistemov z objektno usmerjeno tehnologijo.

Objektna usmeritev se pogosto obravnava kot revolucionarno nova tehnika, vendar to ne drži. Pred petindvajsetimi leti se je že uporabljala pri modeliranju različnih sistemov, med njimi tudi poslovnih. Objektiva usmeritev je univerzalna tehnika, ki se lahko enako uspešno uporablja pri modeliranju tehniških sistemov, poslovnih sistemov ali sistemov realnega časa. Kot univerzalna tehnologija se objektna usmeritev širi v različna tehnološka okolja in se vse bolj uporablja. Pričakuje se, da bo ob koncu stoletja objektna usmeritev v uporabi pri razvoju vseh vrst sistemov, ter da bo postala prevladujoča tehnika pri razvoju poslovnih in informacijskih sistemov.

Pri modeliranju poslovnih in informacijskih sistemov obstajajo podobne zahteve. Oba sistema sta kompleksna in pri obeh želimo priti do modela, ki je razumljiv, spremljiv, prilagodljiv in ponovno uporabljiv. V poslovnem inženirstvu je kreiranje takšnih modelov osnovna zahteva, in če rezultati modeliranja niso razumljivi

za vsakogar v organizaciji in zunaj nje, potem smo zagotovo zgrešili cilj razvoja poslovnega sistema. Ko enako tehniko uporabimo pri modeliranju organizacije in njenih poslovnih procesov ter pri gradnji informacijskega sistema, je prehod med obema aktivnostima enostaven in jasen.

Razvoj poslovanja je del velikega števila nalog v organizaciji, med katerimi sta tudi prenova poslovnih procesov in razvoj novih poslovnih procesov. Razvoj poslovanja ni običajen poslovni proces. Nima zunanjih udeležencev in ker se želimo izogniti izrazu 'proces', kot notranjemu organizacijskemu elementu, vidimo poslovni razvoj kot objekt ali še bolje, kot skupek objektov. Razvoj poslovnih procesov razvija poslovanje organizacije, njegovi vhodi vključujejo nove in spremenjene cilje, izhod pa je spremenjeno podjetje.

4. Objektno usmerjeni modeli

Poslovni model organizacije je prikaz procesov, ki jih organizacija izvaja, ter prikaz načina njihovega izvajanja. Poudarek je predvsem na strukturi, ločeno pa opisuje tudi potek izvajanja funkcij. V organizacijah se opravljajo različne dejavnosti, proizvajajo izdelki, opravljajo storitve itd. Vse to so dogodki, dovolj otipljivi, da imajo ceno; lahko jih prodamo in zadovoljstvo strank je merljivo.

Vsi ti dogodki so obravnavani kot objekti in ker je med dogodki realnega sveta in objekti v modelu tesna povezava, je vrzel med realnim in modelom zelo majhna. Gre za dogodke, ki so v modelu prikazani kot objekti, in odnose med dogodki, ki so v modelu prikazani kot povezave med objekti.

Ideja, da je naravno predstaviti statične strukture dogodkov in njihove povezave v objektnem modelu, je bila uporabljena pri podatkih in konceptualnem modeliranju informacijskih sistemov. Objektne usmeritev upošteva te razloge in gre korak dlje ter dovoli dogodke, ki se dinamično spreminjajo, oblikovati kot objekte. Objekt lahko torej predstavlja določeno vedenje v povezavi z okolico, lahko je aktiviran, lahko se spremeni, lahko aktivira druge objekte itd.

Objektne usmerjeni modeli so torej naravni in lahko razumljivi. Zelo malo je semantičnih vrzeli med resničnostjo in modelom. Objektne usmerjeni modeli imajo tudi veliko drugih pomembnih lastnosti, ki se kažejo v procesu modeliranja. Zato so ti modeli na splošno uporabni za opisovanje kompleksnih sistemov.

5. Objekt

Z vidika poslovanja je objekt za organizacijo pomemben element ali dogodek, ki ga želimo opisati v njegovem okolju. Besedo 'objekt' uporabljamo v različnih situacijah in primerih različno, njen pomen pa se v

skladu s tem spreminja. Bolj ko objekt ustreza nečemu, kar vemo in prepoznavamo v vsakdanjem življenju, bolj je uporabljen. Objekt kot entiteta ali dogodek vsebuje informacije o sebi, lahko vpliva na vedenje drugih objektov, na podlagi vedenja drugega objekta lahko spremeni informacijo o sebi ali pa spremeni informacijo o povezavi z drugim objektom.

Naštejmo nekaj primerov objektov v poslovnem sistemu. V sistemu prodaje so objekti: partner, račun, prodajni referent. V sistemu vodenja zalog sta objekta material in skladišče. Na prvi pogled bi lahko sklenili, da so objekti neke vrste skupki sorodnih podatkov, zapisani v bazi podatkov. Vendar ima objekt zmožnost vedenja, in prav to mu omogoča, da deluje neodvisno in na neki 'inteligenten način'.

Vsi elementi ali dogodki v okolju, ki se preučujejo, so potencialni objekti. Na primer: objekt 'avto' najdemo v registru avtomobilskih registracij in pri proizvajalcu avtomobilov. Postavlja se vprašanje, ali sta objekta enaka. Po pravilu nista, je pa med njima veliko podobnosti. Objekti se razlikujejo v bistvu v dveh pogledih:

Po različnih podatkih, povezanih z objektom. V registru avtomobilskih registracij objekt avto vsebuje podatke o lastniku, modelu, letu izdelave, barvi avtomobila. Pri proizvajalcu avtomobilov pa objekt avto vsebuje podatke o distributerju, datumu izdelave, barvi avtomobila. Te razlike izhajajo iz različnega namena objektov.

Po različnih načinih uporabe. V registru avtomobilskih registracij je objekt avto uporabljen za zapis sprememb o lastništvu, na drugi strani pa je objekt avto pri proizvajalcu uporabljen za fakturiranje distributerju.

5.1 Lastnosti objektov

Objekt je opisan z lastnostmi in z načinom oziroma namenom njegove uporabe. Seveda je to dvoje tesno povezano. Informacije o objektu so zajete z atributi, uporaba objekta pa z obnašanjem objekta oziroma z njegovim delovanjem na način, s katerim vpliva na druge objekte v svojem okolju. Postopek je sredstvo, s katerim objekt vpliva na drug objekt, in objekt je dosegljiv prek drugega objekta samo s postopkom.

Do vpliva med objekti navadno pride tako, da prvi objekt pošlje drugemu sporočilo. Ker ima beseda sporočilo v kontekstu programiranja veliko pomenov, raje uporabimo besedo spodbuda. Ko objekt pošlje drugemu spodbudo, se objekt 'prejemnik' odzove tako, da odvisno od spodbude izpelje ustrezen postopek. Objekt prejemnik lahko izpelje več različnih tipov postopkov, zato ima vsaka 'spodbuda' ime, ki jasno opredeli postopek, ki naj se izvede. Med izvajanjem postopka lahko objekt sprejemnik bere in spreminja vrednosti atributov. V smislu vplivanja lahko pošlje spodbudo samemu sebi.

Atributi objekta so v objektu samem. Vrednosti atributov so povsem skrite okolici objekta in edini način za pridobitev informacij o objektu je prek postopka. Pred okolico objekta pa niso skriti samo atributi, temveč tudi struktura postopkov. Okolici objekta je znano le ime postopka, prek katerega je ta dosegljiv.

Informacije o imenu in tipu objekta imenujemo oznaka objekta, kombinacijo oznak pa protokol. Sodelovanje med objekti imenujemo **dogovor**. Ta določa spodbude, ki se izmenjujejo med objektoma. Ko je dogovor med objektoma vpeljan, lahko objekt pošiljatelj pošilja samo spodbude, ki so v dogovoru, in objekt sprejemnik mora nanje pravilno odgovarjati.

Objekt je tako opisan na dva različna, povezljiva načina. Prvi opis (zunanji opis) imenujemo **objektov protokol**. Z njim opišemo, kako je lahko objekt uporabljen in kakšne so njegove odgovornosti do drugih objektov. Drugi opis (notranji opis) imenujemo **objektna implementacija** in specificira objektove attribute ter izvedbo posameznih postopkov. S takšnim načinom opisa objekta omogočimo, da se notranja zasnova objekta lahko spremeni brez vpliva na okolico objekta. Tako dodani atribut ali sprememba pomembnosti atributa ne vpliva na uporabo objekta. Edino sprememba protokola ima učinek na objekte, ki ga uporabljajo.

Naslednja vrsta spremembe v modelu je vpeljava novega objekta, ki je podoben obstoječemu objektu v modelu, vendar se razlikuje od njega v enem ali več pogledih. Na vsak način želimo, če je mogoče, uporabiti obstoječe lastnosti objektov in opisati samo posebnosti novega objekta. V mnogih primerih lahko takšen objekt dodamo v model brez večjega vpliva na obstoječe objekte. To pomeni, da se model preprosto prilagaja novi uporabi.

Povezava objektov

Posamezen objekt še ni sistem, sistem predstavlja večje število med sabo povezanih objektov. Obstajajo različni tipi povezav med objekti, ki jih v modelu predstavimo z risanjem zvez med njimi.

Poglejmo primer ko ima klient v banki račun. V objektnem modelu definiramo objekt 'klient', ki ima zvezo z objektom 'račun'. Ta povezava pove, da ima klient v banki točno določen račun. Objekt 'klient' opišemo z atributi, kateri račun pripada določenemu klientu, pa povemo s povezavo na objekt 'račun'. Povezava vsebuje pravilo, ki velja med objektoma. Po obliki je statična in jo imenujemo 'poznansvena zveza'. Ob zapiranju klientovega računa in odpiranju novega, se opisana povezava spremeni. Povezava, ki kaže na star objekt 'račun' se briše, prav tako se briše stari objekt 'račun', kreira pa se nov objekt 'račun' in nova povezava, ki kaže nanj.

Naslednji tip povezave omogoča, da objekt uporabi postopek s katerim vpliva na drug objekt. Vzemimo

primer polog denarja na bančni račun. Postopek opišemo z objektom 'depozit', v objektu 'račun' pa uporabimo operacijo, ki poveča vsoto denarja na računu. Ko klient želi položiti denar, objekt 'depozit' pošlje spodbudo klientovemu računu z informacijo o znesku položenega denarja. Račun se poveča za položeni znesek. To povezavo smo oblikovali z definiranjem zveze med objektoma 'depozit' in 'račun' in se imenuje 'sporočilna zveza'. Povezava je po obliki dinamična in se uporablja samo takrat, ko objekt pošilja spodbudo drugemu objektu.

Objekti lahko oblikujejo skupke - agregate

Velikokrat želimo opisati objekte, ki so sestavljeni iz večih objektov. Na primer, v modelu bančnega sistema želimo oblikovati objekt 'inventar', ki je sestavljen iz različnih dogodkov oz. elementov v banki (pohištvo, računalniki, ...). Vsi pripadajo inventarju banke in v tem pomenu je objekt 'inventar' agregat, sestavljen iz velikega števila objektov.

Z opisom agregata določimo tudi pravila, ki jih lahko uporabimo pri obravnavi posameznega objekta. V našem primeru velja, da je računalnik del inventarja, zato inventar ne more biti obravnavan kot del računalnika. Prav tako velja, da je zaslon del računalnika in računalnik del inventarja. Zaslon je zato avtomatično del inventarja.

Objekti pripadajo razredom

V organizaciji je mnogo objektov istega tipa. Na primer v banki je veliko število klientov. Vsak izmed njih je objekt in če jih podrobno analiziramo vidimo, da so enaki. To pomeni, da je vsak objekt 'klient' opisan z enakimi atributi in vsebuje enake postopke. Kliente med seboj razlikujemo samo po vrednostih njihovih atributov.

Da ne opisujemo vsakega klienta posebej, je dovolj, da opišemo samo enega. Vsak posamezen klient tako pripada razredu klientov. Razred je definiran z imenom in opisom. Opis vsebuje vzorec, ki mu sledijo vsi objekti v razredu. Običajno obstaja več vzorcev, ki ustrezajo opisu posameznih razredov. Vzorcev dajo atributom imena, prek njih pa objekt dostopa do vrednosti atributov. Lahko zapišemo, da ima razred vzorec za attribute, ki naj jih objekt vsebuje, ali drugače povedano, objekt uporabi vzorec, da sestavi vrednosti svojih atributov.

Ker vsebujejo vsi objekti istega tipa enake postopke, je dovolj opisati postopke samo enkrat pri opisu razredov. Ko objekt prejme spodbudo za izvedbo postopka, se obrne k svojemu razredu, kjer identificira postopek, ki ga bo uporabil. S postopkom je določeno kaj mora objekt storiti, katere vrednosti atributov bodo uporabljene in kateri postopki, njegovi ali tisti, ki pripadajo ostalim objektom, se bodo izvršili.

Pri kreiranju objekta se opiše tudi razred, h kateremu

objekt pripada, vzorci razreda pa se prepisejo v objekt. Objekt dobi enovit identifikator in začetne vrednosti atributov. V času življenja objekta se vrednosti atributov lahko spreminjajo, ravno tako se lahko spreminjajo postopki, ki se izvršujejo na njem. Kreirani objekti prispevajo k opisu razreda in to pomeni, da so posamezni objekti pojavki razreda. V našem primeru velja, da so vsi klienti v banki pojavki razreda 'klient'.

6. ZAKLJUČEK

Za načrtovanje naših ciljev ter za usmeritev naših prizadevanj danes in v prihodnje je pomembno vedeti, kam vodi objektna usmeritev. Strokovnjaki si delijo mnenje, da bo prihodnji razvoj poslovnih in informacijskih sistemov potekal na podlagi idej, ki izhajajo iz objektno usmeritve. Kdo bo 'zmagovalec', bo pokazala prihodnost. Možnost, da bo v prihodnjih letih objektno usmeritev nadomestila kaka druga, je malo verjetna.

Bolj verjetno je, da bo skoraj na vseh področjih razvoja prizadevanje usmerjeno proti objektni tehnologiji.

7. UPORABLJENA LITERATURA

Cheryl Currid:

Computing Strategies for Reengineering your Organization, Prima Publishing, Rocklin, California, 1994

Michael Guttman, Jason R. Matthews:

The Object Technology Revolution, John Wiley & Sons, New York, 1995

Jacobson Ivar:

The Object Advantage, Addison - Wesley, ACM Press Books, 1995

Thomas H. Davenport,

Process Innovation, Reengineering Work through Information Tehnology, Ernst&Young, Boston, 1993

David R C Hill,

Object-Oriented Analysis and Simulation, Addison-Wesley Publishing Company, 1996

◆
Mag. Bogdan Vožič se je z računalništvom pričel ukvarjati leta 1979 v Iskri Delti, kjer je kot sistemski inženir najprej sodeloval pri razvoju prvih domačih računalnikov, kasneje pa se je ukvarjal z razvojem programskih generatorjev in sistemov za krmiljenje baz podatkov. Od leta 1986 je zaposlen v podjetju PRIS Consulting kot projektant in se ukvarja s projektiranjem in gradnjo informacijskih sistemov. Svoje praktične izkušnje in ugotovitve pri gradnji informacijskih sistemov in zasnovi podatkovnih baz je teoretično obdelal v magistrskem delu. V zadnjem obdobju se ukvarja tudi s prenovo poslovnih procesov in z revizijo informacijskih sistemov.
◆

Vabilo avtorjem

Uredniški odbor revije *Uporabna informatika* načrtuje razširitev obsega revije. Rezultati ankete med bralci revije so pokazali, da si želijo med drugim prispevke z naslednjih področij: ocena orodij in različnih rešitev, predstavitev primerov iz prakse, predstavitev rešenih informacijskih problemov, pregled stanja na nekaterih področjih v Sloveniji.

S tem vabilom se posebej obračamo na informatike v praksi, da s članki v naši reviji predstavijo svoje ugotovitve in izkušnje.

Navodila za prispevke objavljamo na zadnji strani revije.

OBJEKTNI MODEL PORAZDELJENEGA PROCESIRANJA

Matjaž B. Jurič, Marjan Heričko, Ivan Rozman

Povzetek:

Tehnologija porazdeljenih objektov omogoča izgradnjo programske opreme z integracijo komponent - objektov. Objekti se osvobodijo oklepa programskega jezika, v katerem so bili oblikovani in svobodno zaživijo v omrežju. Z zmožnostjo ponovne uporabe, prenosljivosti in povezljivosti v distribuiranem, heterogenem omrežju rešujejo večino težav, s katerimi se danes srečujejo razvijalci programske opreme in uporabniki računalnikov.

V prispevku opisujemo model *Common Object Request Broker Architecture* (CORBA). Temelji na posredniku zahtev objektov, ki ima vlogo neke vrste vodila in je zadolžen za zagotavljanje komunikacije med aplikacijami, objektnimi storitvami in skupnimi sredstvi. Predstavlja »de-facto« standard med distribuiranimi objektnimi modeli. Tehnologija porazdeljenih objektov bo sčasoma izpodrinila ostale tehnologije in nepovratno vplivala na način uporabe in izgradnje programske opreme.

Abstract:

Distributed object technology enables building software with integration of components - objects. Objects become independent of programming language they were designed in and can live in the network. With reuse, portability and interoperability in distributed heterogen network they address the majority of problems developers and end-users are dealing with today.

In this article the Common Object Request Broker Architecture (CORBA) is described. It is based upon an object request broker, which acts as a bus and is responsible for communication between applications, object services and common facilities. CORBA is "de-facto" standard for distributed object models. Over time distributed object technology will supersede other technologies and will have irreversible impact on the way software is used and built.



1. UVOD

Danes se srečujemo z množico različnih računalnikov, povezanih v omrežje. Najmočnejši super-računalniki izvajajo zapletene znanstvene izračune ali delujejo kot strežniki glomaznih podatkovnih baz. V sredini najdemo strežnike in namizne računalnike, ki nudijo kompleksno množico storitev. O najmanjših sistemih mnogokrat niti ne razmišljamo kot o računalnikih. To so televizijski aparati, termostati in alarmi, telefoni in elektronske beležnice. Vsi ti sistemi uporabljajo različne operacijske sisteme in programske jezike. Zelim si, da bi bili računalniki povezani ne glede na tip, velikost ali operacijski sistem.

Poglaviten problem današnjih računalniških omrežij, sestavljenih iz različnih računalnikov, operacijskih sistemov in aplikacij, je nezmožnost povezave. Le-ta preprečuje boljšo so-uporabo podatkov in večjo produktivnost uporabnikov. Vemo pa, da se je računalniška paradigma spremenila. Podjetja niso več zadovoljna z uporabo računalnikov zgolj za štetje denarja, ki so ga zaslužila. Računalnike želijo uporabiti kot orodje za služenje denarja.

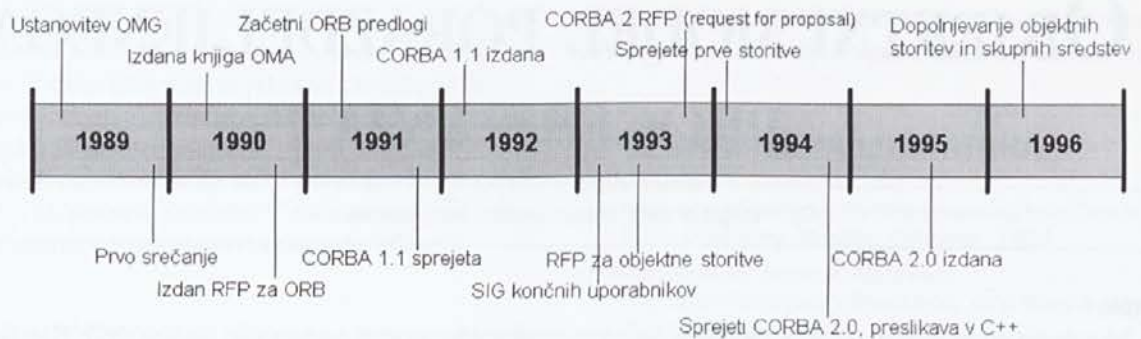
Poglejmo, kateri problemi izvirajo iz omenjenega okolja:

- težko je zagotoviti integracijo računalniške strojne opreme,
- še težje je zagotoviti integracijo računalniške programske opreme,
- razvoj programske opreme traja predolgo in stane preveč.

Potrebujemo nov način pogleda na celoten problem, od definicije problema in analize, skozi potek načrtovanja in implementacije do uporabe, vzdrževanja in dopolnjevanja. *Objektna tehnologija* je tak nov način. V prispevku si seveda ne bomo mogli ogledati celotne tematike. Osredotočili se bomo na področje implementacije.

1.1. Porazdeljeni objekti

V preteklosti je računalništvo temeljilo na *podatkih*, namesto na *njihovi uporabi*, in na *računalniških procesih*, namesto na *poslovnih procesih*. Uporabniki so se morali prilagajati računalnikom. Strojna oprema je v tem času postala veliko zmogljivejša. To nam daje možnost, da računalnike prilagodimo svojim zahtevam in ne



Slika 1: Kronologija dosežkov OMG-ja

obratno. Uporabniki ne želijo več velikih, monolitnih aplikacij, ki znajo vse, ampak iščejo majhne komponente - objekte, ki jih lahko sami združujejo in tako oblikujejo orodja, prilagojena njihovim tipičnim poslovnim potrebam. Objekti pa delujejo skupaj le, če so načrtovani in oblikovani na osnovi standarda.

Objekti so enote kode in podatkov, ki komunicirajo s pošiljanjem in sprejemanjem sporočil. Če so zgrajeni korektno, so v sistemu visoko povezljivi in relativno enostavno je zamenjati lokalne objekte z oddaljenimi in tako razširiti komunikacijo skozi omrežje.

Narava objektov radikalno spreminja način razmišljanja o tem, kje so naši podatki. Podatki, ograjeni v objekte, lahko pridejo tja, kjer so najbolj potrebni. Trenutno razmišljamo, da so dokumenti enostavno shranjeni na določenem trdem disku. Distribuirani objektni sistemi zavračajo to tezo in nudijo moč in fleksibilnost shrambe, neodvisne od lokacije.

Naš ultimativni cilj je integracija objektov. Za dosego tega cilja pa moramo najprej rešiti problem povezljivosti objektov v distribuiranih, heterogenih okoljih. Rešitve obstajajo v obliki modelov porazdeljenega procesiranja. V prispevku se bomo osredotočili na najpomembnejši objektni model, model CORBA (*Common Object Request Broker Architecture*).

Prehod na tehnologijo porazdeljenih objektov se ne bo zgodil čez noč; pot bo dolga in evolucijska. Pomembno pa je razumeti, kako tehnologije, ki so na voljo danes in tiste, ki bodo na voljo kmalu, pripravljajo uporabnike na življenje v svetu distribuiranih objektov.

1.2. Objektni modeli

Od leta 1989 obstaja konzorcij objektnih dobaviteljev. Imenuje se *Object Management Group*¹ (OMG). Ukvarja se s specifikacijo arhitekture odprtega programskega vodila, na osnovi katerega lahko povežemo objektne komponente različnih dobaviteljev prek omrežja. Or-

ganizacija OMG določa smernice in specifikacije upravljanja z objekti in določa okvir za razvoj aplikacij. Osnovni cilji so ponovna uporaba, prenosljivost in povezljivost objektno osnovane programske opreme v distribuiranih, heterogenih okoljih. Ujemanje s specifikacijami omogoča razvoj heterogenih aplikacij za vse pomembnejše računalniške platforme in operacijske sisteme.

Danes pomeni OMG-jeva specifikacija arhitekture upravljanja z objekti (*object management architecture - OMA*) »de-facto« standard med distribuiranimi objektnimi modeli [14]. OMA je konceptualna infrastruktura, na kateri temelji porazdeljeni objektni model CORBA (*Common Object Request Broker Architecture*).

Slika 1 prikazuje kronološki razvoj skupine OMG in njene pomembnejše dosežke.

Omenimo naj še obstoj drugih porazdeljenih objektnih modelov. Microsoftov OLE 2.0/COM (*Object Linking and Embedding/Component Object Model*) je objektni model, ki pa ni distribuiran. Microsoftov DCOM (*Distributed Component Object Model*) je ugledal luč sveta v Windows NT 4.0. IBM-ov SOM (*System Object Model*), bil je prvič uporabljen v OS/2, služi kot osnova *OpenDoc-u*. *OpenDoc* podpirajo Apple, Novell, Sun, Xerox, Oracle, IBM in Taligent. Zastavljen je bil neodvisno od operacijskega sistema, vendar je z razvojem daleč za konkurenco. Razširjeni model SOM omogoča dostop do distribuiranih objektov. DSOM (*Distributed SOM*) je skladen s specifikacijo CORBA. Vso funkcionalnost objektnih sistemov pa že nudi NextStep [15].

Od distribuiranih ne-objektnih modelov naj omenimo samo OSF DCE (*Open System Foundation Distributed Computing Environment*). Le-ta služi kot podlaga številnim CORBA implementacijam in kot osnova Microsoftovega *Distributed Component Object Model*a.

2. Arhitektura upravljanja z objekti

Referenčni model identificira in določa značilnosti komponent, vmesnikov in protokolov, ki sestavljajo arhitek-

¹ Od januarja 1996 je član OMG tudi Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko.

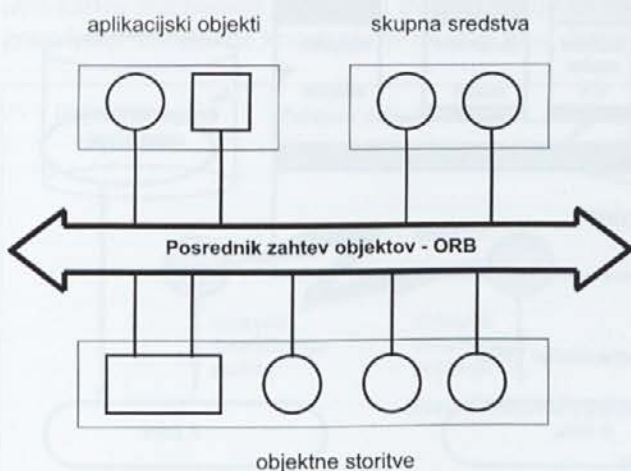
turo upravljanja z objekti, vendar jih natančno ne definira. Referenčni model [12] ima naslednje funkcije:

- Predstavlja ogrodje za pisanje specifikacij.
- Identificira komponente arhitekture upravljanja z objekti.
- Karakterizira funkcije, ki jih nudijo komponente.
- Razlaga relacije med komponentami in zunanjim operacijskim okoljem.
- Identificira protokole in vmesnike za dostop do komponent.

Aplikacija, skladna z arhitekturo upravljanja z objekti, je sestavljena iz množice povezljivih objektov, ki komunicirajo skozi posrednik zahtev objektov (*object request broker* - ORB). Referenčni model določa, kako objekti pošiljajo zahteve in sprejemajo odgovore, določa ključne operacije za vsak objekt in vmesnike objektov, ki nudijo skupna sredstva, uporabna v večini aplikacij.

Referenčni model je sestavljen iz naslednjih komponent (slika 2):

- *Posrednik zahtev objektov (object request broker)* je ključ do povezljivosti. Objektivno omogoča transparentno oblikovanje in sprejemanje zahtev in odgovorov v distribuiranem okolju. Je osnova za izgradnjo aplikacij iz distribuiranih objektov in za povezljivost med aplikacijami v heterogenih in homogenih okoljih.
- *Objektne storitve (object services)* so zbirka storitev (vmesnikov in objektov), ki podpirajo osnovne funkcije za uporabo in oblikovanje objektov. Storitve so potrebne za konstrukcijo vsake distribuirane aplikacije in so vedno neodvisne od aplikacijske domene.
- *Skupna sredstva (common facilities)* so zbirka storitev, skupnih mnogim aplikacijam, ki pa niso fundamentalne, kakor so objektne storitve.



- *Aplikacijski objekti* so izdelki posameznega dobavitelja. Aplikacijski objekti sovpadajo s tradicionalnim izrazom aplikacija in niso standardizirani s strani OMG. Aplikacijski objekti sestavljajo najvišji nivo referenčnega modela.

V splošnem so aplikacijski objekti in skupna sredstva aplikacijsko orientirani. Posrednik zahtev objektov in objektne storitve so orientirane na infrastrukturo sistema. Od štirih komponent referenčnega modela težijo tri k standardizaciji: posrednik zahtev objektov, objektne storitve in skupna sredstva. Četrta komponenta predstavlja aplikacijske objekte, ki so preveč specializirani za standardizacijo.

Pomembno je poudariti, da morajo aplikacije za sodelovanje v arhitekturi upravljanja z objekti zagotoviti samo vmesnike, skladne z OMA. Ni nujno, da je njihova konstrukcija objektivno orientirana. Na sliki 2 so take aplikacije prikazane v obliki pravokotnika, medtem ko krogi ponazarjajo objektivno orientirane aplikacije.

3. OBJEKTNI MODEL CORBA

Common Object Request Broker Architecture (CORBA) je konkreten objektivni model. Izpeljan je bil iz referenčnega modela arhitekture upravljanja z objekti. Sestavljen je iz jedra in komponent [4].

Objektivni model najprej opiše koncepte, razumljive odjemalcem, kot npr. oblikovanje objektov, zahteve in operacije, tipe in podpise. Nato opiše koncepte, povezane z implementacijo objektov strežnikov, med drugimi metode in izvršna okolja. Objektivni model je najbolj specifičen pri definiciji konceptov, razumljivih odjemalcem.

Ta objektivni model je primer klasičnega objektivnega modela, kjer odjemalec pošlje sporočilo objektu strežniku. Konceptualno gledano, objekt interpretira sporočilo in se odloči, katero storitev bo izvršil. V klasičnem modelu sporočilo identificira objekt in nič ali več dejanskih parametrov. Kot v večini klasičnih objektivnih modelov, tudi tukaj zahtevamo enoličen prvi parameter, ki identificira operacijo, ki jo je potrebno izvesti. Objekt interpretira sporočilo z izbiro metode, bazirane na specifični operaciji.

Objekt je lahko odjemalec, strežnik, ali pa oboje. V določenem trenutku uporablja storitve drugih objektov, v določenem trenutku nudi storitve drugim objektom. Za konkretno akcijo pa so vloge točno definirane.

Obstajati mora način specifikacije storitev, ki jih nudi objekt strežnik. CORBA zato definira jezik za definicijo vmesnikov (*OMG Interface Definition Language - IDL*). Definicija vmesnika specifičira operacije, ki jih je objekt pripravljen izvršiti, in vhodne ter izhodne parametre, ki jih potrebuje. Poleg tega so v definiciji navedene izjemne situacije, ki lahko nastopijo. Odjemalci lahko pristopajo do strežnikov samo prek vmesnika.

Slika 2: Referenčni model arhitekture upravljanja z objekti

Arhitektura CORBA torej strogo loči vmesnik objekta od njegove implementacije. Implementacijo objekta napišemo v poljubnem programskem jeziku, za katerega obstaja ustrezna preslikava.

Specifikacija modela CORBA [7] obsega:

- jedro - posrednik zahtev objektov,
- povezljivost,
- preslikave jezika za definicijo vmesnikov v C, C++, Smalltalk in Ado²,
- objektne storitve,
- skupna sredstva.

Minimalni zahtevi za sistem, združljiv s CORBA, sta upoštevanje specifikacij jedra in ene preslikave IDL-a v programski jezik. Vsaka nadaljnja preslikava v programski jezik je ločena, opcijska točka združljivosti.

3.1. Posrednik zahtev objektov

Odjemalec je entiteta, ki želi, da objekt izvrši operacijo. Implementacija objekta je kombinacija kode in podatkov, ki predstavljajo objekt. ORB je odgovoren za vse mehanizme, potrebne za iskanje implementacije objekta, za pripravo le-te za sprejem zahteve in za komunikacijo. Vmesnik, ki ga vidi odjemalec, je popolnoma neodvisen od tega, kje je objekt lociran, v katerem programskem jeziku je implementiran, ali od česarkoli drugega, kar se ne odraža v vmesniku objekta.

² V pripravi so preslikave za Java in COBOL.

Slika 3 predstavlja strukturo posameznega ORB. Vmesniki k ORB so prikazani v obliki škatel, puščice pa nakazujejo, ali je klican ORB, ali pa ORB izvaja klic navzgor v vmesniku.

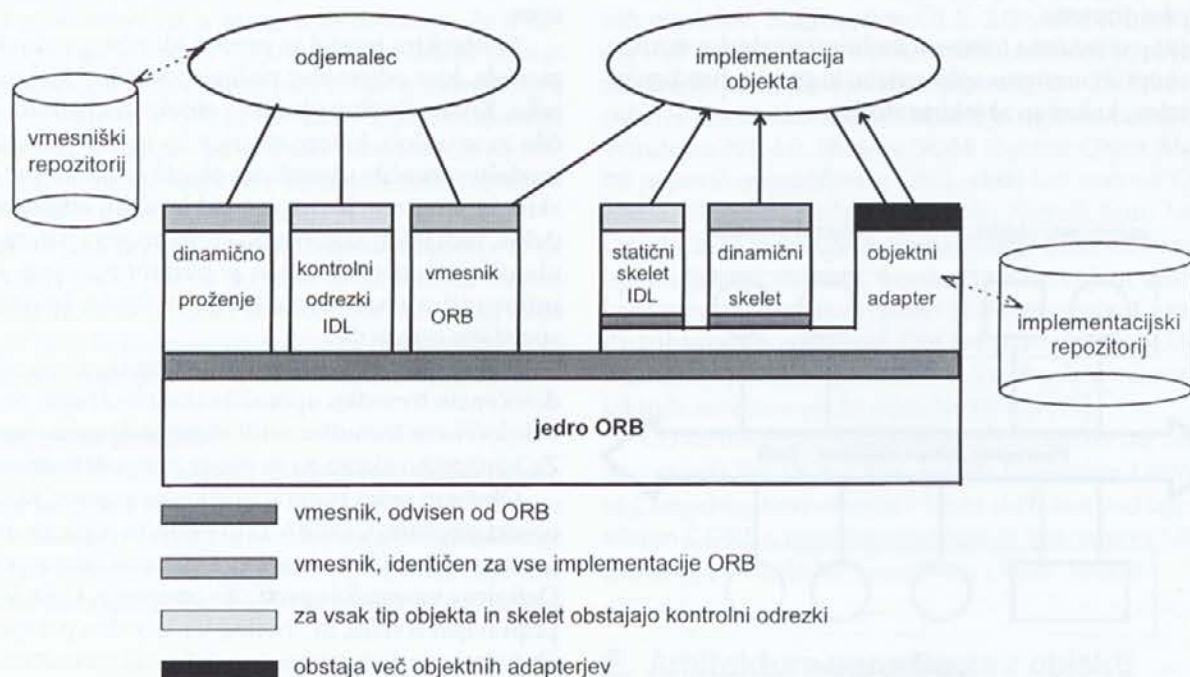
Za izvedbo zahteve lahko odjemalec uporabi vmesnik za dinamično proženje (vmesnik, neodvisen od vmesnika ciljnega objekta) ali pa kontrolni odrezek IDL (specifični kontrolni odrezek, odvisen od vmesnika ciljnega objekta). Za nekatere funkcije lahko odjemalec direktno komunicira z ORB skozi vmesnik ORB.

Implementacija objekta dobi zahtevo kot klic navzgor skozi skelet OMG IDL ali skozi dinamični skelet. Med procesiranjem zahteve ali drugače lahko kliče objektne adapter in ORB.

Vmesniki k objektom so lahko definirani na dva načina. Statično v IDL, kar imenujemo OMG IDL - jezik za definicijo vmesnikov. Ta jezik definira tipe objektov glede na operacije, ki jih lahko izvajamo nad njimi, in parametre teh operacij. Vmesniki so lahko dodani v vmesniški repozitorij (*interface repository*). Ta predstavlja komponente vmesnika kot objekte, kar dovoljuje dostop do njih v času izvajanja. V vsaki implementaciji ORB imata IDL in vmesniški repozitorij enako izrazno moč.

Odjemalec izvrši zahtevo. Pri tem ima dostop do reference objekta in pozna tip in zaželeno operacijo za izvedbo. Odjemalec inicira zahtevo s klicem rutin kontrolnega odrezka, specifičnih objektu ali z dinamično konstrukcijo zahteve.

Dinamični vmesnik in vmesnik kontrolnega odrez-



Slika 3: Struktura posrednika zahtev objektov

ka za proženje zahteve zadovoljijo isto semantiko zahtev, zato sprejemnik sporočila ne more ugotoviti, kako je bila zahteva prožena.

ORB locira ustrezno implementacijsko kodo, pošlje parametre in prenese kontrolo na implementacijo objekta prek skeleta IDL ali dinamičnega skeleta. Skelet je specifičen vmesniku in objektnemu adapterju. Pri izvršitvi zahteve lahko implementacija objekta uporabi storitve ORB-ja skozi objektni adapter. Ko je zahteva popolna, se kontrolne in izhodne vrednosti vrnejo odjemalcu.

V implementaciji objekta se lahko izbere, kateri adapter bo uporabljen. Ta odločitev bazira na tipu storitev, ki jih implementacija objekta zahteva.

Informacije o implementaciji objekta se shranijo med njegovo instalacijo v implementacijski repozitorij za uporabo med izvajanjem zahtev. Vmesniške in implementacijske informacije, shranjene v ustreznih repozitorijih, so na voljo odjemalcu in implementaciji objekta. Vmesnik definira OMG IDL oziroma vmesniški repozitorij. Definicija se uporablja za oblikovanje odjemalčevih kontrolnih odrezkov in skeletov implementacije objekta.

3.2. Povezljivost posrednikov zahtev

Arhitektura povezljivosti med posredniki zahtev objektov je okvir za definicijo elementov povezljivosti. Opisuje nove mehanizme in specificira konvencije za doseg povezljivosti med neodvisno izdelanimi ORB-ji.

Arhitektura povezljivosti jasno definira vloge različnih tipov domen za specifične informacije. Kjer sta ORB-ja v isti domeni, lahko komunicirata direktno. Navadno je to najbolj zaželen način.

Ko mora informacija proženja zapustiti domeno, pravimo, da gre prek mosta (slika 4). Vloga mosta je zagotoviti preslikavo vsebine in semantike iz oblike enega ORB-ja v drugo tako, da vsak ORB vidi samo ustrezno vsebino in semantiko.

Podpora mostovom med ORB-ji specificira ORB API (*application programming interface*) in konvencije za zagotavljanje enostavne konstrukcije mostov med dome-

nami ORB. Takšne mostove lahko razvije dobavitelj ORB-ja ali kdo drug. Slika 4 prikazuje splošen most med ORB-jema.

Razširitve, potrebne za podporo mostov med ORB-ji, so splošne in ne vplivajo na druge operacije. Uporabne so tudi v druge namene, kot npr. za čiščenje programov.

Most med ORB-ji je uporaben tudi za povezavo s sistemi, ki niso kompatibilni s CORBA, kot je Microsoftov *Component Object Model* - COM. Enostavnost takega početja je odvisna od teh sistemov in njihove podpore modela CORBA.

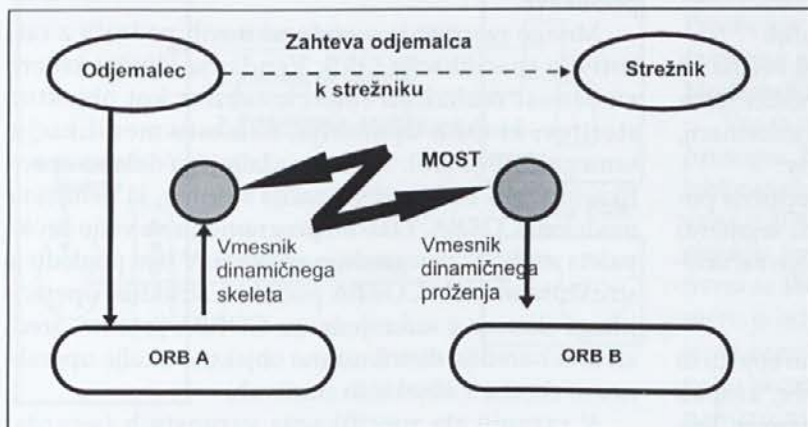
Protokoli za komunikacijo med ORB-ji so:

- *Splošen protokol med ORB-ji (General Inter-ORB Protocol - GIOP)* specificira sintakso prenosa na nizkem nivoju in množico oblik sporočil za komunikacijo med ORB-ji. Oblikovan je namensko za interakcije med ORB-ji in deluje preko vsakega transportnega protokola, ki ustreza minimalnim zahtevam. Protokol je enostaven, skalabilen in relativno enostaven za implementacijo. Omogoča prenosljive implementacije z majhno porabo pomnilnika in z glednimi zmogljivostmi, z minimalno odvisnostjo od podrejene programske opreme.
- *Internet protokol med ORB-ji (Internet Inter-ORB Protocol - IIOP)* specificira, kako se izmenjujejo GIOP sporočila z uporabo TCP/IP (*Transport Control Protocol/Internet Protocol*) povezav. IIOP specificira standardiziran protokol povezljivosti prek Interneta in omogoča povezljivost z drugimi kompatibilnimi ORB-ji. Uporaben je tudi kot protokol med polmostovi. Predstavlja osnovni protokol med ORB-ji za TCP/IP okolja.
- *Protokoli med ORB-ji, odvisni od okolja (Environment Specific Inter-ORB Protocol - ESIIOP)* slonijo na sredstvih, ki jih nudijo določena okolja. Podpirajo lahko specialna področja, kot npr. varnost in administracijo. Lahko so optimizirani za določeno okolje. Skladati pa se morajo s splošno arhitekturo povezljivosti

ORB-jev za omogočanje enostavne povezave z mostovi.

3.3. Objektne storitve

Posrednik zahtev objektov ne more zagotavljati povezljivosti na nivoju semantike aplikacij. ORB lahko primerjamo s telefonsko centralo, ki nudi osnovne mehanizme komunikacije, ne zagotavlja pa pomenske komunikacije med naročniki. Zato so potrebni dodatni vmesniki in protokoli zunaj telefonskega sistema kot so telefonski aparati, modemi in telefonski imeniki. Le-ti so ekvivalentni vlogi objektnih storitev.



Slika 4: Splošen most med ORB-jema

Specifikacija objektnih storitev je ponavadi sestavljena iz množice vmesnikov in opisa obnašanja storitev. Za opis vmesnika je uporabljen OMG IDL. Trenutno so standardizirane naslednje objektno storitve [6]:

- *storitve poimenovanja*, ki nudijo standardni pristop za oblikovanje imenskih kontekstov objektov,
- *dogodkovne storitve*, ki razgradijo komunikacijo med objekti,
- *storitve trajnih objektov* ponujajo skupen vmesnik za mehanizem, ki se uporablja za ohranjanje in upravljanje trajnega stanja objektov,
- *storitve življenjskega cikla* definirajo storitve in konvencije za oblikovanje, brisanje, kopiranje in premikanje objektov,
- *storitve kontrole hkratnosti* omogočajo koordiniranje dostopa več odjemalcev do deljenih virov,
- *storitve eksternalizacije* definirajo protokole in konvencije za eksternalizacijo in internalizacijo objektov
- *relacijske storitve* omogočajo ekspliciten prikaz entitet in relacij za modeliranje entitet realnega sveta,
- *transakcijske storitve* združujejo transakcijski in objektni vzor in naslavljajo probleme komercialnega procesiranja transakcij,
- *storitve povpraševanja* omogočajo uporabnikom in objektom sprožiti povpraševanja na zbirki objektov,
- *licenčne storitve* nudijo mehanizme za nadzor uporabe intelektualne lastnine proizvajalcev.

V prihodnosti pričakujemo razvoj novih objektnih storitev, kot npr. storitev trgovanja in varnostnih storitev.

3.4. Skupna sredstva

Skupna sredstva obsegajo specifikacije višje nivojskih storitev in vertikalnih tržnih področij. Nekateri splošni primeri skupnih sredstev so elektronska pošta, tiskanje, ipd. Ta sredstva so potrebna v večini aplikacijskih domen. Skupna sredstva so primerno področje za standarde pri poveztivosti med neodvisnimi dobavitelji programske opreme [5].

Skupna sredstva so razdeljena v dve kategoriji:

- *Horizontalna skupna sredstva*, ki si jih deli večina ali vsi sistemi. Obstajajo štiri poglavitne množice sredstev: uporabniški vmesnik, upravljanje informacij, upravljanje sistema in upravljanje poslov.
- *Vertikalna tržna sredstva*, ki podpirajo specifična področja, povezana z vertikalnimi tržnimi segmenti (npr. informacijske avtoceste, proizvodnja, računalništvo, idr.).

Ločnice med skupnimi sredstvi, aplikacijskimi objekti in objektnimi storitvami niso ostre in jasne, ampak odražajo evolucijo tehnologije objektnih sistemov. Trenutna postavitev ločnic predstavlja trenutne ukrepe

standardizacije OMG. Z večanjem izkušenj in zrelostjo aplikacij bomo odkrili in definirali nova skupna sredstva. Prav tako se bodo deli skupnih sredstev preselili v objektno storitve.

3.5. Praktična uporaba

CORBA 2.0 je ugledala luč sveta 15. avgusta 1995 na Object World-u v San Franciscu. Takrat je petnajst podjetij predstavilo svoje neodvisne implementacije CORBA, na katerih je delovala ena distribuirana aplikacija. Ta demonstracija je prikazala zrelost in dostopnost tehnologije. Pričakujemo, da bo do konca tisočletja uporabljalo arhitekturo CORBA prek devet milijonov računalnikov, kar bo predstavljalo nekako 75% trga distribuiranih objektov [3].

Pomembnejši dobavitelji posrednikov zahtev objektov, ki so skladni s CORBA, so Hewlett Packard, DEC, Sun, Iona, Expersoft in Visigenic.

Danes uporabljajo posrednike zahtev objektov, ki so skladni s CORBA, številna podjetja [9]: banke (*Bank of Boston, Chemical Bank, IBM Credit Corp.*), zavarovalnice (*Travelers*), letalstvo (*Allied Signal Aerospace Technology, Space Telescope Science Institute - Hubblov teleskop*), transport (*Narfolk Southern Railroad, Wheels*), telekomunikacije (*Bell SYGMA Telecom Solutions, British Telecom Labs, Time Warner Communications*) in mnogi drugi. V večini primerov se je z novimi aplikacijami povečala produktivnost in stroški so se znižali. Prav tako je tehnologija porazdeljenih objektov omogočila prehod iz velikih računalnikov na omrežje osebnih računalnikov.

4. Nadaljnji razvoj

Kljub splošnem sprejemu specifikacije CORBA delo OMG-ja ni končano. Trenutno poteka razvoj na standardizaciji jezikovnih preslikav za COBOL in Javo. Čeprav lahko jezikovno preslikavo oblikuje vsak, je zelo pomembna standardizacija. Tako model CORBA že uporabljajo z Objective C, Eiffel, Common Lisp, Scheme in drugimi jeziki, za katere še ne obstajajo standardne preslikave.

Mnogo razprav je o podpori novih področij z razširitvijo specifikacije ORB. Vendar je ključni kriterij zmožnost realizirati funkcionalnost kot objektno storitev, ki ORB uporablja, namesto modifikacije samega ORB-ja [13]. Večina nadaljnjega dela na specifikaciji OMG bo standardizacija storitev, ki delujejo s modelom CORBA. Tako bo programerju na voljo široka paleta sredstev za izgradnjo aplikacij. V tem pogledu je struktura modela CORBA podobna strukturi operacijskega sistema z mikrojedom: CORBA je jedro, sredstva, ki naredijo distribuirano objektno okolje uporabno, so zbrana v objektnih storitvah.

V razvoju sta specifikacija varnostnih (*security*) storitev [8] in storitev trgovanja (*trading*).

4.1. Povezava z objektnimi podatkovnimi bazami

Upravljanje distribuiranih objektov in objektne podatkovne baze sta trenutno vodilni tehnologiji objektne revolucije. Objektne podatkovne baze so komercialno na voljo že skoraj pet let in so našle uporabo v številnih aplikacijskih domenah. V povezavi z upravljanjem distribuiranih objektov jih lahko uporabimo za implementacijo unificiranih distribuiranih rešitev.

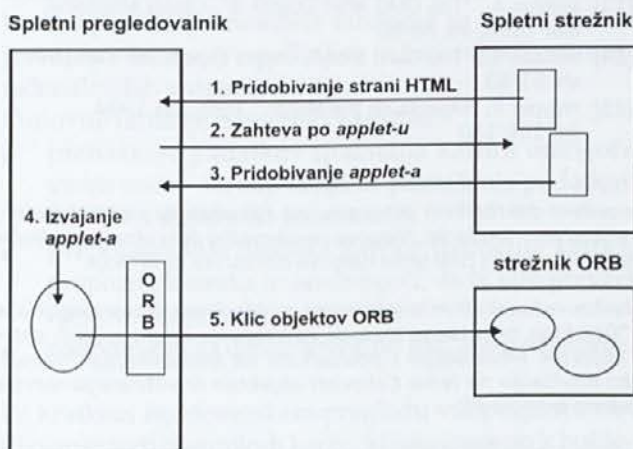
Zelo poenostavljeno lahko rečemo, da je objektna podatkovna baza odgovorna za podaljšanje življenjske dobe objekta onstran procesa, ki je objekt oblikoval. Rečemo lahko, da objektna podatkovna baza upravlja z lokalnimi objekti [1].

Naslednji stavek najlepše oriše pomensko razliko med posredniki zahtev objektov (ORB) in objektnimi podatkovnimi bazami: posrednik zahtev objektov izroči zahtevo objektu, medtem ko objektna podatkovna baza izroči objekt zahtevi.



Slika 5: ODBMS kot upravitelj objektov znotraj OMG arhitekture

Obstaja več načinov povezave ORB-jev z objektnimi podatkovnimi bazami. Katero bomo izbrali, je odvisno od zahtevane funkcionalnosti. V najenostavnejšem primeru lahko objektna podatkovna baza nudi trajnost



Slika 6: Integracija WWW z modelom CORBA in Java

objektom ORB[11]. Drugi način integracije je definicija objektnega adapterja za podatkovne baze (slika 5). Tretja možnost integracije je vmesnik IDL k API-ju objektne podatkovne baze ali h knjižnici razredov.

4.2. Porazdeljeni objekti in Internet

Danes sta pojma Internet in Svetovni splet (WWW) zelo popularna. Pomeni Svetovni splet smrt klasičnih arhitektur distribuiranih objektov, kot je CORBA? Odgovor je vsekakor *ne*. Pravzaprav je prav integracija spleta z arhitekturo CORBA idealno priložnost, da prevzame tehnologija distribuiranih objektov vodilno vlogo [2, 10].

CORBA je v osnovi strežniško orientirana arhitektura, ki veže stanje objekta in njegovo obnašanje s strežniškim procesom. Odjemalci pošiljajo zahteve objektom, ki so lahko kjerkoli. Odjemalci in strežniki lahko zamenjujejo vloge, kolikokrat želijo. Za določeno zahtevo pa so vloge odjemalca in strežnika strogo definirane.

Alternativa strežniško orientirani arhitekturi je takšna, kjer se objekti prosto premikajo med strežnikom in odjemalcem, s ciljem delovati lokalno. Odjemalno orientirani sistemi so uporabni pri aplikacijah, kjer samo beremo, ali je sprememb malo. Prednost je v tem, da lahko povezavo med strežnikom in odjemalcem prekinemo, kar je zelo ugodno za globalno distribucijo.

Največja pomanjkljivost pri implementaciji arhitekture CORBA je, da morajo biti odjemalcu metode objekta znane vnaprej. Z drugimi besedami, ko pridobivamo stanje objekta iz strežnika, mora biti nekaj kode na voljo lokalno, da lahko objekt uporabimo. Ravno tukaj pa vskoči WWW, ki zapolni praznino z Java, Visual Basicom in drugimi jeziki, ki omogočajo, da iz strežnika naložimo kodo in podatke in jo izvedemo lokalno (glej sliko 6). Razvijemo lahko splošnega odjemalca z minimalnim vgrajenim znanjem. Znati mora samo pridobivati oddaljene objekte in jih interpretirati. Istočasno lahko naložene skripte znova pokličejo strežnik, kar daje svobodo mešanja arhitekturnih stilov.

Za možnost integracije distribuiranih objektov in svetovnega spleta se zavzema SunSoft, ki želi standardizirati jezikovno preslikavo med OMG IDL in Java. Proces je v teku, medtem pa so številni dobavitelji izdali razvojne komplete, med drugimi: JavaSoft Joe, Post-Modern BlackWidow in Iona OrbixWeb.

Vsa ta orodja zagotavljajo integracijo na podobnem principu. Java program (*applet*), ki se naloži iz spleta, lahko vsebuje odjemalčeve kontrolne odrezke, oblikovane iz IDL. Na ta način nudi vmesnik odjemalca nazaj objektu, implementiranemu na strežniku ORB. Alternativno se lahko uporabi tudi JavaScript. V vsakem primeru je odjemalčevi strani na voljo podlaga za izvajanje, ki zagotovi povezavo kode Java z objekti, ki so na voljo skozi ORB. To, skupaj z integracijo s splošnimi namiznimi orodji (npr. spletnimi pregledovalniki), napravi Java idealnega odjemalca CORBA.

Kombinacija arhitekture CORBA in spleta torej obljublja veliko prilagodljivost za razvijalce, kakor tudi jasno načrtano pot k množični uporabi distribuiranih objektov.

5. Sklep

Tehnologija porazdeljenih objektov naslavlja ključne probleme razvoja programske opreme, kot so ponovna uporaba, prenosljivost in povezljivost programske opreme. Vodilo, v obliki posrednika zahtev (ORB), omogoča objektom življenje kjerkoli v omrežju in zagotavlja komunikacijo med njimi. Odjemalci dostopajo do njih s proženjem metod. Vmesnik in implementacija objekta sta ločeni, odjemalci so odvisni samo od vmesnika. Programski jezik in prevajalnik, uporabljena za oblikovanje porazdeljenih objektov, sta popolnoma transparentna za odjemalce. Odjemalcem ni potrebno vedeti, kje je distribuiran objekt, ali na kakšnem operacijskem sistemu se izvaja. Porazdeljeni objekti so »pametni« koščki programske opreme, ki lahko med seboj komunicirajo, ne glede na lokacijo.

Ko govorimo o porazdeljenih objektih, govorimo dejansko o neodvisnih programskih komponentah. Komponenta je torej objekt, ki ni vezan na določen program, programski jezik ali implementacijo. Komponenta je dovolj majhna, da jo lahko oblikujemo in vzdržujemo, dovolj velika, da jo uporabljamo in podpiramo ter ima standardni vmesnik za povezljivost. Objekti, zgrajeni kot komponente, so gradniki distribuiranih aplikacij naslednje generacije. Pomagajo nam zgraditi aplikacije, ki so boljše, hitrejše in uporabnejše od današnjih. Zgrajene so z manj napora in so zanesljivejše.

Objekti so del revolucije komponent, ki smo jo že doživeli pri strojni opremi. Predstavljajo programski ekvivalent mikro čipa in posledično osebnega računalnika [3]. Tehnologija komponent, kot nadaljevanje distribuiranih objektov, bo sčasoma izpodrinila vse ostale tehnologije, saj zna vse, kar znajo druge tehnologije, in to bolje.

Uporabljene kratice:

API	Application Programming Interface
COM	Component Object Model (Microsoft)

CORBA	Common Object Request Broker Architecture
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model (Microsoft)
DEC	Digital Equipment Corporation
DSOM	Distributed System Object Model (IBM)
ESIOP	Environment Specific Inter-ORB Protocol - protokol med ORB-ji, odvisen od okolja
GIOP	General Inter-ORB Protocol - splošen protokol med ORB-ji
HTML	Hyper Text Markup Language
IBM	International Business Machines
IDL	Interface Definition Language - jezik za definicijo vmesnikov
IIOIP	Internet Inter-ORB Protocol - Internet protokol med ORB-ji.
ODBMS	Object Database Management System - sistem za upravljanje z objektno podatkovno bazo
OLE	Object Linking and Embedding (Microsoft)
OMA	Object Management Architecture - arhitektura upravljanja z objekti
OMG	Object Management Group
ORB	Object Request Broker - posrednik zahtev objektov
OSF	Open System Foundation
RFP	Request for Proposal
SOM	System Object Model (IBM)
TCP/IP	Transport Control Protocol/Internet Protocol
WWW	World Wide Web - svetovni splet

Literatura:

- [1] Barry D., "ODBMSs and Distributed Systems", *Object Magazine*, August 1996, str. 28-30
- [2] Chauvet J. M., "ORB Spiders On The Web", *Object Expert*, Sept/Oct 96, str. 29-31
- [3] Guttman M., Matthews J. R., *The Object Technology Revolution*, John Wiley & Sons, Inc., USA 1995
- [4] Jurič M. B., *CORBA - objektna zasnova porazdeljenega procesiranja*, FER Maribor, oktober 1996
- [5] OMG, *CORBAfacilities: Common Facilities Architecture*, Revision 4.0, November 1995
- [6] OMG, *CORBAservices: Common Object Services Specification*, Revised Edition March 31, 1995, Updated: March 28, 1996
- [7] OMG, *The Common Object Request Broker: Architecture and Specification*, Revision 2.0, July 1995
- [8] OMG: "CORBA Security", *First Class*, Jun-Jul 1996
- [9] OMG: "Meeting the Challenge of Vertical Industries", *First Class*, Aug-Sep 1996
- [10] OMG: "Surfin' the Net with CORBA", *First Class*, Jan-Feb 1996
- [11] Reddy M., "ORBs & ODBMSs: Two complementary ways to distribute objects", *Object Magazine*, Jun 1995, str. 24-31
- [12] Soley R. M., Ph.D., *Object Management Architecture Guide*, OMG, John Wiley & Sons, Inc., Revision 3.0, Third Edition, Jun 15, 1995
- [13] Watson A., "The OMG After CORBA 2", *Object Magazine*, Mar 1996, str. 58-60
- [14] Watson A., "The OMG Story", *Object Expert*, Jan-Feb 1996, str. 51-53
- [15] Wayner P., "Objects on the March", *Byte*, Jan 1994, str. 139-150

Matjaž B. Jurič je diplomiral leta 1996. Študij nadaljuje po enovitem doktorskem programu na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru kot štipendist Slovenske znanstvene fundacije. Njegovo raziskovalno delo obsega področje objektnih tehnologij s poudarkom na porazdeljenih objektnih sistemih, kar bo tudi tema njegove doktorske disertacije.

Mag. Marjan Heričko je asistent stažist na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru. Naziv magistra je pridobil leta 1993 z uspešnim zagovorom magistrske naloge "Objektno orientirane metode načrtovanja informacijskih sistemov". Njegovo raziskovalno razvojno delo obsega vse vidike objektnih tehnologij, s poudarkom na metodologijah razvoja, orodjih CASE, razvojnih okoljih in metrikah. Pripravlja doktorsko disertacijo na temo Kakovost objektno orientiranega razvoja informacijskih sistemov. Aktivno sodeluje pri delu Centra za objektno tehnologijo.

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 več kot 200 publikacij, vodil je številne znanstveno raziskovalne projekte.

PAMETNA KARTICA

Aleksandar Jurišić, Department of Combinatorics and Optimization,
University of Waterloo, Waterloo, Canada, N2L 3G1 & Certicom Corp., Mississauga

Alenka Trojar, School of Business and Economics, Wilfrid Laurier, University Waterloo, Canada

Povzetek

Pametne kartice nudijo stopnjo varnosti, ki je potrebna, da računalniške mreže zares zaživijo, ter združijo telekomunikacije in računalnike. Po kratkem zgodovinskem pregledu razvoja kartic sledijo opisi treh pomnilnih kartic, ki pripomorejo k boljšemu razumevanju prednosti pametnih kartic. Z dvema primeroma so opisani vidiki varnosti in zaščite pametnih kartic. Seznanimo se tudi z osnovnimi komponentami in karakteristikami pametnih kartic. Sledi pregled sedanje uporabe pametne kartice in možnosti v prihodnosti.

Abstract

Smart cards provide the degree of security necessary to make computer networking truly viable. They unify telecommunications and computing. After a brief historical overview of the development of cards, we survey three basic types of memory cards, in order to better understand the advantages of smart cards. The security aspects of smart cards are illustrated by two examples. Applications of smart cards are surveyed and a view for their future possibilities is given.



1. Uvod

Večina ljudi ima danes vsaj eno kreditno kartico. V mnogih deželah uporabljajo tudi telefonske kartice, kartice za avtomate, zdravstvene kartice in še mnoge druge. Današnja tehnologija lahko vse te kartice nadomesti s kartico, ki ima različne funkcije in zagotavlja veliko varnost lastniku ter računalnikom, s katerimi kartica komunicira. Taka kartica se imenuje pametna kartica (Smart Card) in se poskusno uporablja že po vsem svetu. Ima enako velikost kot običajna kreditna kartica in vsebuje eno ali več tiskanih vezij s funkcijami procesorja, pomnilnika in vhodno-izhodne enote. Pametna kartica nam bo omogočila, da bomo kmalu nosili računalnik kar v žepu. S temi lastnostmi bo pametna kartica postala pomembna možnost za varno shranjevanje ter izmenjavo podatkov in bo izboljšala varnost računalniških sistemov.

Osnovne funkcije pametnih kartic so:

- prenašanje podatkov (pametna kartica omogoča varen način shranjevanja in prenašanja podatkov ter varen dostop do informacij)
- prepoznavanje lastnika kartice (pametna kartica prepozna lastnika in onemogoči, da bi kdo prevzel njegovo identiteto)
- nadomestilo za denar ter varno plačilno poslovanje.

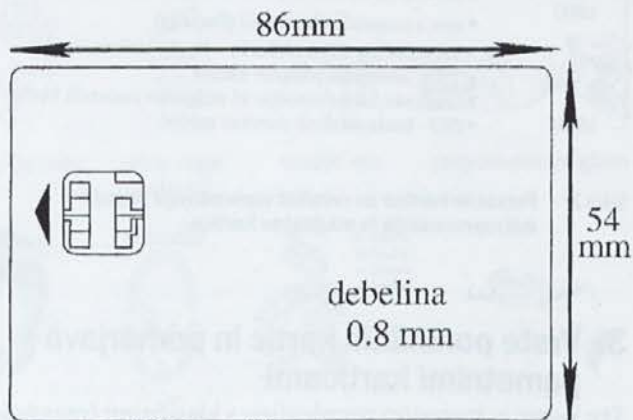
Po kratkem zgodovinskem pregledu v 2. poglavju sledijo opisi treh pomnilnih kartic, ki pripomorejo k boljšemu razumevanju prednosti pametnih kartic. V 4. poglavju so opisane prednosti pametne kartice na

področju varnosti in zaščite z dvema primeroma. V 5. poglavju pa se seznanimo z osnovnimi komponentami in karakteristikami pametnih kartic.

Tehnični del članka se zaključuje z opisom sistema pametne kartice. V 7. poglavju podajamo pregled mnogih načinov uporabe pametne kartice, v 8. pa njene možnosti v prihodnosti.

2. Kratka zgodovina

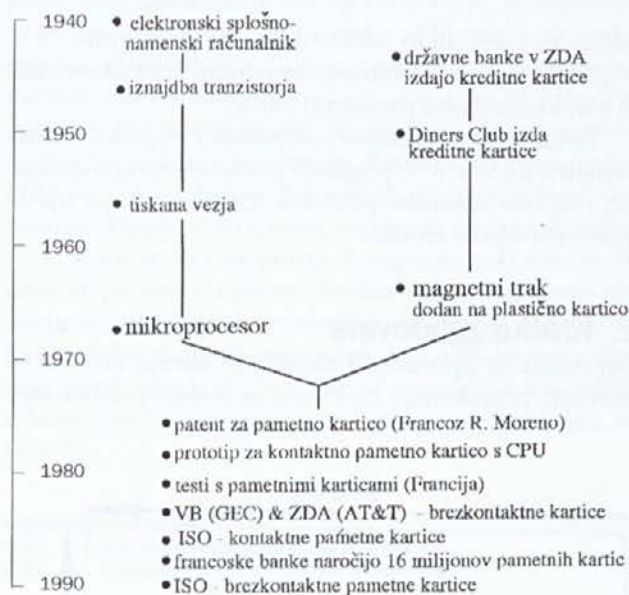
Desetletja so uporabljali papirnate kartice (vizitke) z imenom, priimkom in naslovom za osebno predstavitev



Slika 1: Dimenzije kartice in lega kontaktov so določeni s standardom ISO 7816.

in za krajša sporočila. Leta 1930 so se pojavile plastične kartice, ki jih je bančna industrija začela uporabljati kot kreditne kartice. Diner's Club in American Express sta predstavila prve take kartice v 50-ih letih. Leta 1967 so začeli plastičnim karticam dodajati magnetni trak. Kasnejši standardi pa so omogočili, da je uporaba magnetne kartice postala mednarodna.

Z razvojem računalnikov in komunikacijskih sistemov v zadnjih 20-ih letih je postala potreba po učinkovitem in varnem komuniciranju vedno nujnejša in pomembnejša. Konec 70-ih let je Francoz Moreno prijavil patent - čip, vgrajen v kreditno kartico. Tako pametno kartico je mogoče programirati za različne funkcije in jo usposobiti za kontrolo računalniškega povezovanja z zunanjo enoto. Sredi 80-ih let so v Angliji in ZDA razvili kontaktne in brezkontaktne pametne kartice in začeli razvijati njihove standarde. V svetu pametnih kartic pa je najbolj pomembna prav standardizacija. Kartice različnih proizvajalcev morajo biti usklajene za interakcijo z računalniki. V naslednjih petih, desetih letih pričakujemo nagel razvoj uporab pametne kartice, vse dokler ne bodo pametne kartice zares postale računalnik v našem žepu.



Slika 2: Pametne kartice so rezultat vzporednega razvoja mikroprocesorja in magnetne kartice.

3. Vrste pomnilnih kartic in primerjava s pametnimi karticami

Trg kartic je trenutno preplavljen s klasičnimi (magnetnimi) karticami ter različnimi novimi karticami. Da bi lažje razumeli prednosti pametne kartice in njeno po-

tencialno tržišče, bomo najprej opisali značilnosti pasivnih (pomnilnih) kartic, ki so trenutno najbolj v rabi.

Pasivne ali pomnilne kartice se uporabljajo predvsem za shranjevanje podatkov. Delimo jih v tri skupine: magnetne kartice, optične kartice in kartice s čipom.

(a) Magnetne kartice

Zaradi vse večje potrebe po avtomatizaciji v bančništvu so plastični kartici dodali magnetni trak. Le-ta deluje podobno kot avdio trak za snemanje; nekateri trakovi imajo tudi bralni del (*read-only*), kar pomeni, da lahko podatke samo beremo, ne pa tudi spreminjamo. Kljub temu pa je varnost te kartice pomanjkljiva, saj je magnetni trak izredno lahko posneti in kartico ponarediti. Pomanjkljivost kartice je tudi majhen pomnilnik (manj kot 900 zlogov). Vseeno pa je magnetna kartica trenutno najbolj razširjena zaradi nizkih stroškov proizvodnje ter pomnilne zmogljivosti, ki povsem zadošča za preproste vrste uporabe.

(b) Optične kartice

Glavna značilnost optične kartice je, da vsebine kartice ne moremo izbrisati. Te kartice delujejo po načelu "enkrat zapiši, večkrat preberi" (*write-once read-many times*). Za branje in pisanje na kartico se uporabljajo laserji, ki izžgejo milijon drobnih luknjic v tanek list optičnega traku (mesto z luknjico ali brez nje predstavlja stanje bita). Optična kartica ima izredno veliko pomnilno zmogljivost, saj nanjo lahko spravimo že od 2 do 8 MB podatkov. Zaradi te lastnosti se je kartica izkazala predvsem v zdravstvu in netiskanih publikacijah (CD-ROM, Internet). V primerjavi s pametno kartico ima optična kartica prednost predvsem zaradi velikega pomnilnika in nizke cene.

(c) Kartice s čipom

Razlikujemo dve vrsti kartic s čipom, pomnilne kartice ter pametne kartice. Pomnilne kartice s čipom imajo običajno manj pomnilnika kot pametne kartice ter lahko izvajajo manjše logične operacije s pomočjo integriranih vezij. Vendar pa te kartice nimajo logike v smislu procesorja in jih zato ne moremo ponovno programirati. Kartice s čipom se uporabljajo predvsem kot telefonske kartice.

Zadnja novost na trgu kartic je kartica s čipom obsežnega pomnilnika (do 32MB). Združenje *The Personal Computer Memory Card Industry Association* (PCMCIA) je uvedlo standard izdelave kartic in sicer v velikosti kreditne kartice, z izjemo debeline. Uporabljajo se predvsem za povečanje pomnilnika v prenosnih računalnikih ter za prenos podatkov.

Novi materiali (plastika) in značilnosti (magnetni trak, površina na optičnih karticah, čipi) so izredno povečali trg kartic. Po nekaterih ocenah se uporablja že

tri milijarde različnih kartic. Prednost varnega komuniciranja dela kartico vse bolj konkurenčno za uporabo pri denarnih zadevah. Pametna kartica stane od manj kot 1 ameriški dolar do 20 dolarjev (USD), medtem ko stanejo optične kartice od 4 do 8 USD, magnetne kartice pa stanejo od 10 do 50 centov, glede na to, ali vsebujejo sliko, hologram. Čeprav je cena pametnih kartic višja od cen drugih kartic, so jih nekatere dežele (Anglija, Japonska, Francija, Nemčija) že pričele uvajati. Uporaba pametnih kartic raste in ta trend se bo verjetno še povečeval, ko se bodo zniževali stroški proizvodnje.

4. Zakaj pametne kartice

Ker večina ljudi še vedno uporablja magnetno kartico, bomo podrobneje razložili, zakaj bo pametna kartica nadomestila magnetno kartico. Za razliko od pametnih kartic so magnetne kartice povsem pasivne (uporabljajo se v glavnem za hranjenje podatkov). Magnetna kartica dopušča različne oblike zlorabe, na primer prekoračitev računa, saj se večina nakupov ne zapisuje takoj v glavnem računalniku (nepovezani sistemi se vedno bolj uveljavljajo). Tudi ponarejanje in kopiranje magnetnih kartic ni redek pojav. Metode so ponavadi zelo rafinirane. Mogoče je kopirati informacije z magnetnega traku na prazno kartico in nanjo vtisniti podatke z originalne kartice

Kadar uporabljamo magnetno kartico v bankomatu, le-ta najprej "vpraša" lastnika kartice za geslo in ga pošlje glavnemu računalniku v banki skupaj z zakodiranim geslom z magnetnega traku. Računalnik zakodira še vtiskano geslo in opravi primerjavo. Geslo potuje po linijah nezaščiten in tako lahko nekdo, ki prisluškuje povezavi med bankomatom in računalnikom ali pa ima dostop do gesel na glavnem računalniku v banki, geslo ukrade.

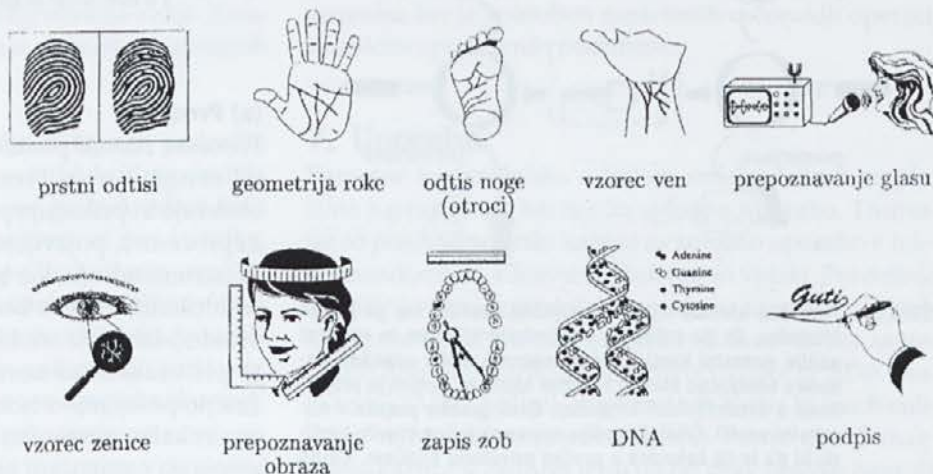
Pametna kartica ima dve značilnosti, ki onemogočata zgoraj omenjene zlorabe in ponarejanje. Ena od teh je obstojni pomnilnik, ki se ga ne da spreminjati in se ohrani tudi po prekinitvi napajanja. Ta pomnilnik lahko vsebuje tudi podatke, ki so bili zapisani po tem, ko je bila kartica izdana, in lahko zabeleži vsako transakcijo. S tem preprečuje lastniku, da bi prekoračil svoj limit. Druga značilnost pa je, da procesor kartice nadzira vse interakcije med različnimi zunanji napravami, ki berejo kartico in pišejo nanjo, in pomnilnikom pametne kartice. Ta je oblikovan tako, da so določeni deli

pomnilnika fizično in logično dostopni le izdajatelju kartice.

Naslednja dva primera kažeta, zakaj sta omenjeni značilnosti tako pomembni za zagotavljanje varnosti pri prepoznavanju. Le-to se opravlja v dveh delih: najprej se mora kartica prepričati, da jo uporablja njen lastnik, nato pa komunicira (varno) z glavnim računalnikom.

Oglejmo si bolj podrobno, kako lahko lastnik dokaže svojo identiteto kartici. Recimo, da podjetje hrani podatke v glavnem računalniku, do katerega je možen dostop s pametnimi karticami. Le-te so izdali zaposlenim, ki imajo dovoljenje za dostop do računalniškega sistema. Vsaka kartica je programirana z edinstvenimi informacijami, kot je na primer osebna prepoznavna številka PIN (*personal identification number*). Prepoznavna številka je zakodirana z enosmerno transformacijo (*hash function*) in shranjena v posebnem delu pomnilnika (v tajnem področju - *secret zone*), ki ga ni mogoče brati. Ko zaposleni želi dostop do računalniškega sistema, mora vstaviti kartico v vhodno-izhodno enoto in vtiskati PIN. Procesor pametne kartice izvede enosmerno transformacijo vtiskane številke ter jo primerja s shranjeno prepoznavno številko v tajnem področju. Ta primerjava poteka v procesorju kartice. Pomembno je, da PIN ne potuje prek nezanesljivih linij in se ne vpiše v delovni pomnilnik glavnega računalnika (ki bi ga lahko kdo opazoval). Če pametna kartica potrdi, da se prepoznavni številki ujemata, se prične drugi del prepoznavanja, ko pametna kartica komunicira z glavnim računalnikom in omogoči zaposlenemu dostop do računalniškega sistema.

Čeprav se osebna prepoznavna številka lokalno preverja, njena dolžina (štiri številke) ne zadostuje za varnost. To pomanjkljivost poskušajo odpraviti z omejitvijo števila poskusov vnašanja gesla ali pa z dodatnim preverjanjem, na primer prstnih odtisov, geometrije roke, podpisa, vzorca zenice, s prepoznavanjem glasu itd. (glej sliko 3).



Slika 3: Biometrični testi

Sedaj pa si podrobneje oglejmo še drugi del prepoznavanja. Ko smo uspešno opravili prvi del prepoznavanja in je kartica prepričana, da jo zares uporablja njen lastnik, prične kartica komunicirati z glavnim računalnikom. To lahko ponazorimo s hipotetično situacijo:

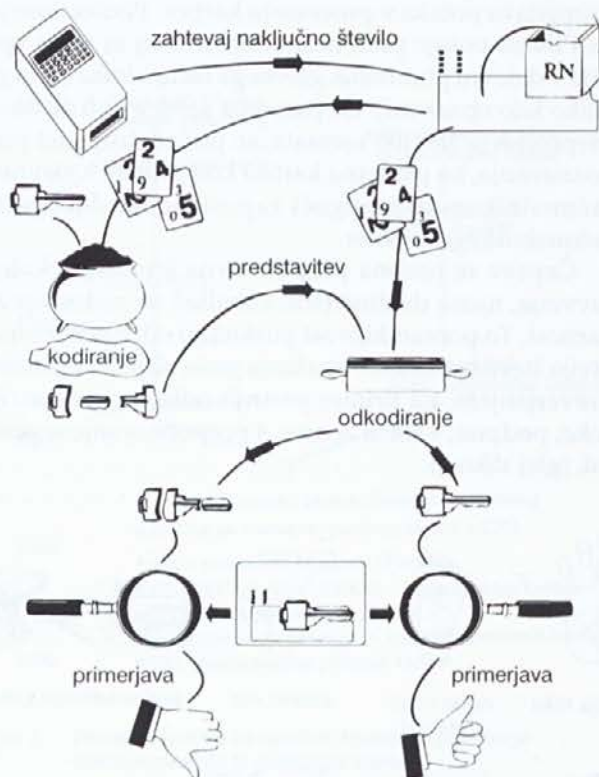
Temno je kot v rogu in po opravljeni diverziji v sovražnem taboru se vohun vrača v grad. Bližajoč se vratom, zasliši šepetajoč glas:

"Geslo ali streljam!"

Ali šepeta prijatelj ali sovražnik?

Kako lahko vohun prepriča stražarja, da pozna geslo, ne da bi ga pri tem izdal morebitnemu sovražniku-prisluškovalcu? McGeoch [17].

Vohunova dilema je vsakdanji problem v telekomunikacijah. Kadar kartica v bankomatu komunicira z banko, morata biti oba prepričana o pristnosti drug drugega. Iz tega sledi, da morajo biti elektronska gesla taka,



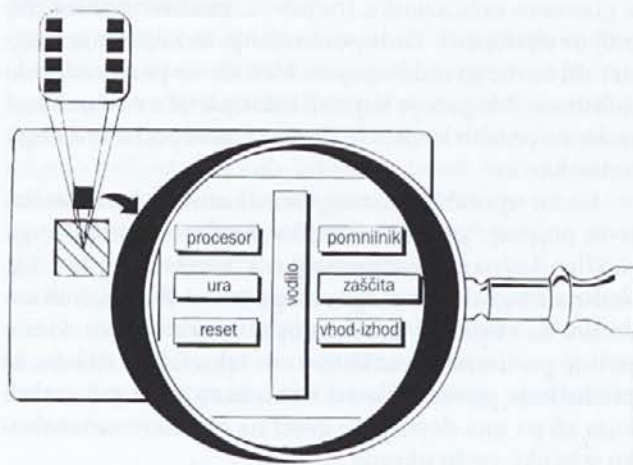
Slika 4: Pametna kartica ustvari naključno število, ter ga pošlje čitalniku. Ta ga zakodira s privatnim ključem in rezultat pošlje pametni kartici. Če pametna kartica uspešno dekodira naključno število z javnim ključem, potem je prepričana o avtentičnosti čitalnika. Enak proces poteka v nasprotni smeri. Čitalnik pošlje novo naključno število kartici, ki ga le-ta zakodira s svojim privatnim ključem. Če ga čitalnik uspešno dekodira, se lahko začne komunikacijski proces.

da jih ni mogoče ponarediti in da ne koristijo prisluškovalcu. Ena od metod za varno izmenjavo gesel v tem kontekstu se imenuje dokaz brez znanja (*zero-knowledge proof*), kar pomeni, da dokažeš (z odgovori na serijo vprašanj), da poznaš svoje geslo, a pri tem ne izdaš niti en sam bit gesla. Glej [17], [13] in [20].

Glede na pomembnost podatkov, ki jih varujemo, se odločimo za ustrezno obliko zaščite. Geslo oziroma PIN pomeni osnovno zaščito, DES (*Data Encryption Standard*) [5], [25] nudi srednji, shema javnih ključev (*Public Key Scheme - PKS*) [12], [8], [9, Ch.13,14], [15] pa visok nivo zaščite

5. Sestavine in vrste pametnih kartic

Mikroračunalnik pametne kartice je majhen računalnik, ki vsebuje vse tri osnovne mikroračunalniške sestavine: procesor, pomnilnik in vhodno-izhodno enoto (ne vsebuje pa vseh integriranih vezij kot osebni računalnik).



Slika 5: Mikročunalnik pametne kartice ima površino največ 5 mm x 5 mm, sicer bi ga lahko prožnost kartice poškodovala.

(a) Procesor

Procesor naredi pametno kartico "pametno", različno od drugih kartic. Procesor ima dve osnovni funkciji: obdeluje in prikazuje podatke. Trenutno so v rabi 8-bitni procesorji, pojavljajo pa se tudi že 16-bitni. Operacijski sistem odloča, kje bodo shranjeni podatki in v kakšnih okoliščinah se bo izvedel prenos informacij prek vhodno-izhodne enote. Kartica poskrbi za samouničenje pri vsakem nenormalnem stanju (segrevanje ali fizično poseganje v notranjost kartice zaznavajo varnostna stikala - *temper resistant switches*). Uporabniški programi pa omogočajo prepoznavanje, varnost poslovanja itd.

Nekatere vrste pametnih kartic vsebujejo tudi krip-to-koprosesorje za hitro kodiranje ali dekodiranje oziroma za izdelavo digitalnih podpisov (*digital signatures*), ki overovljajo poslovne informacije ali pa potrjujejo njihovo identiteto. Več o tem si lahko preberete v [2] in [4].

(b) Pomnilnik

Pomnilnik je lahko obstojen (*non-volatile*) ali začasen (*volatile*), glede na to, ali se podatki ohranijo ali izgubijo po prekinitvi napajanja, oziroma ko računalnik ugasnemo. Pametna kartica mora imeti obstojen pomnilnik, ki hrani podatke, kot so ime nosilca kartice, in uporabniške programe itd. Imeti mora tudi pomnilnik, kamor se vpisujejo sprotne informacije, na primer stanje po pravkar opravljenem poslu.

Na splošno ima pametna kartica tri vrste pomnilnika:

- bralni pomnilnik (ROM) - za shranitev operacijskega sistema
- bralno-pisalni pomnilnik (RAM) - za začasno shranjevanje podatkov
- programabilni bralni pomnilnik (PROM), ki ga lahko delimo v izbrisljivi programabilni bralni pomnilnik (EPROM) ali v električno izbrisljivi programabilni bralni pomnilnik (EEPROM).

EPROM se lahko uporablja v pametni kartici za trajno shranjevanje zapisov skozi njeno življenjsko dobo. EPROM ji omogoča večjo pomnilno zmogljivost kot drugi pomnilniki, vendar pa je podatke mogoče le zapisovati in ne brisati, tako da se pomnilnik po določenem času napolni in kartici se izteče življenjska doba. Prve pametne kartice so imele EPROM, sedaj pa se vse bolj uveljavlja uporaba EEPROM-a s kapaciteto od 100 zlogov do 64KB. EEPROM se uporablja za shranjevanje programov in podatkov, ki se periodično spreminjajo. Ker se EEPROM lahko izbriše z elektronskim signalom, kartica ne zapade, ko je pomnilnik poln. EEPROM ima manjšo pomnilno kapaciteto in je dražji kot drugi tipi pomnilnika ter potrebuje več integriranega vezja. Zaradi tega ni ustrezen za shranjevanje zapisov posameznih poslov.

PROM je razdeljen v tri področja. V tajnem področju (*secret zone*) so lahko shranjeni podatki, ki jih uporablja samo procesor, na primer lastnikovo geslo, kreditni limit itd. Drugi podatki, na primer priimek in ime lastnika, naslov, so shranjeni v odprtem področju (*open zone*), in jih lahko preberemo z različnimi čitalniki, a jih ne moremo spreminjati. Delovno področje (*working zone*) vsebuje zapis podatkov, ki se nanašajo na funkcijo kartice in jih je potrebno spreminjati (na primer zapisi nakupov). V delovnem področju se lahko piše (in bere) samo pri določenih pogojih, na primer če je kartica v čitalniku oziroma blagajni pri pooblaščenem prodajalcu.

(c) Vhodno-izhodna enota

Vmesnike pametne kartice delimo na kontaktne in brezkontaktne.

Kontaktna kartica se napaja in komunicira prek kovinskih kontaktov. Brezkontaktna kartica pa nima neposrednega kontakta in je ni potrebno nikamor vložiti. Podatke in energijo prenaša na več načinov, na primer z induktivnim sklopom, z optičnim sklopom zmogljivosti, z mikrovalovnim sklopom itd., in glede na to deluje na razdalji 1 mm ali pa nekaj metrov (na primer na avtocesti čitalnik zazna brezkontaktno kartico v avtu, ki vozi do 100 km na uro). Brezkontaktna kartica ima številne prednosti pred kontaktno kartico, saj je zanesljivejša, njena življenjska doba je daljša ter omogoča hitrejšo in bolj enostavno uporabo. Čitalna naprava nima reže, tako je manj možnosti za vandalizem (na primer lepilo, žvečilni gumi v reži).

6. Sistem pametne kartice

Sistem pametne kartice lahko vsebuje pisalno-čitalno enoto (*writer-reader unit - WRU*), tiskalnik in osebni računalnik.

Oglejmo si primer, ko se pametna kartica uporablja kot kreditna kartica (potovalni ček) in komunicira s čitalnikom v trenutku prodaje. Kaj se dogaja v čitalniku kartice? Lastnik vloži svojo pametno kartico v čitalnik, vtipka PIN, kartica in čitalnik pa preverita avtentičnost drug drugega. Kupec potrdi vrednost nakupa, čitalnik pa sporoči kartici, naj zabeleži nakup in zmanjša svojo vrednost za ceno prodanega blaga. Ko je transakcija zaključena, čitalnik izvrže kartico.

Čitalniki v trgovinah se občasno povezujejo z glavnim računalnikom v banki, da mu sporočajo opravljene transakcije in obnovijo seznam ukradenih kartic. Čitalnik pametne kartice imamo lahko tudi doma, kjer ga povežemo z računalnikom, tiskalnikom ali televizorjem, da si ogleđamo tekoče podatke o opravljenih nakupih in stanje našega računa v banki. Sistem pametne kartice poveča prilagodljivost in varnost v mnogih primerih uporabe, ker je sposoben zapletenih računskih operacij in zaščite spravljenih podatkov.

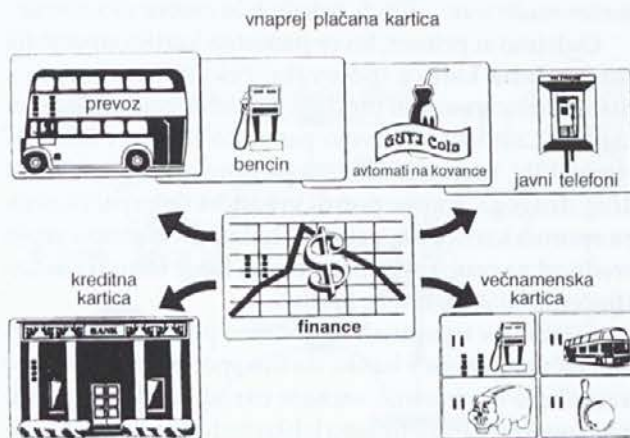
7. Uporaba

Pametne kartice lahko v širšem smislu delimo na plačilne kartice in na kartice za splošno uporabo. Trenutno so predvsem v rabi kartice za splošno uporabo v telekomunikacijah, zdravstvu, šolstvu in vojski. Precejšnje investicije v bankomate in že vpeljana uporaba magnetnih kartic v plačilne namene sta vsekakor vzroka za trenutno neenakomerno porazdeljeno tržišče (v ZDA imajo sedaj 13.000 čitalnikov pametnih kartic in kar 5 milijonov naprav za magnetne kartice). Porast zlorab magnetne kartice v zadnjih letih pa bo prav gotovo nagnila tehtnico v prid pametni kartici.

(a) Plačilne kartice

Za plačevanje se pogosto uporablja vnaprej plačana kartica (*prepaid card*). S tako kartico lahko plačamo toliko zneskov, kolikor denarja smo položili nanjo, in je zlasti uporabna za nepovezane sisteme (*off-line*). Vnaprej plačano kartico uporabljamo predvsem za plačevanje avtomatov (običajno na kovance), kot so javni telefon, fotokopirni in pralni stroj, parkirna ura, prodaja hrane in pijače. Raje uporabljamo nepovezane sisteme, saj so hitri in poceni. Čeprav so zneski plačila majhni, je pretok denarja velik, plačevanje pa izredno poenostavljeno. Državne banke pa opozarjajo, da gre v teh primerih za novo metodo tiskanja denarja brez ustaljenih določil, kdo garantira za izdano vrednost.

Pametne kartice se bodo kmalu uporabljale kot nadomestilo za denar in čeke, se pravi, da bomo namesto gotovine uporabljali elektronsko gotovino, namesto papirnatega čeka pa elektronski ček ali elektronski potovalni ček (vnaprej položen denar je mogoče porabiti



Slika 6: Plačilne, kreditne in večnamenske kartice, ki se uporabljajo na področju financ.



Slika 7: Področja v zdravstvu, kjer se uporabljajo pametne kartice

v katerikoli valuti). Pametna kartica bo v resnici postala elektronska denarnica in bo zagotavljala tudi kreditno sposobnost.

(b) Uporaba v zdravstvu

V zdravstvu nudijo pametne kartice celo paleto storitev, na primer prepoznavanje, vodenje administracije, kot so zapisi o izdanih receptih, boleznih, cepljenjih, pregledih, plačevanju, skratka vse zdravstvene informacije o lastniku kartice, njegovih zdravnikih (v urgentnih primerih) ter komunikacijo med zdravniki, pri tem pa ohranijo zasebnost podatkov. Poenostavljajo administracijo in omogočajo geografsko mobilnost (na primer bolnikom, ki potrebujejo dializo). Problem povzroča samo nezadosten pomnilnik, ki pa se ga da bolje izkoristiti z uporabo mednarodno uveljavljenih kratic za diagnoze, zdravila in podobno.

(c) Uporaba v šolstvu - na univerzah

Zapis na pametni kartici vsebuje informacije o študentu, tako akademske kot administrativne. Hkrati pa kartica omogoča prepoznavanje in vstop v knjižnice, laboratorije, športne ter druge objekte. Ker je univerza svet v malem, se je študentska pametna kartica prva približala večnamenski kartici, s tremi glavnimi funkcijami, te pa so podatkovna baza, nadzor dostopa in elektronska denarnica.



Slika 8: Uporaba pametnih kartic na univerzi

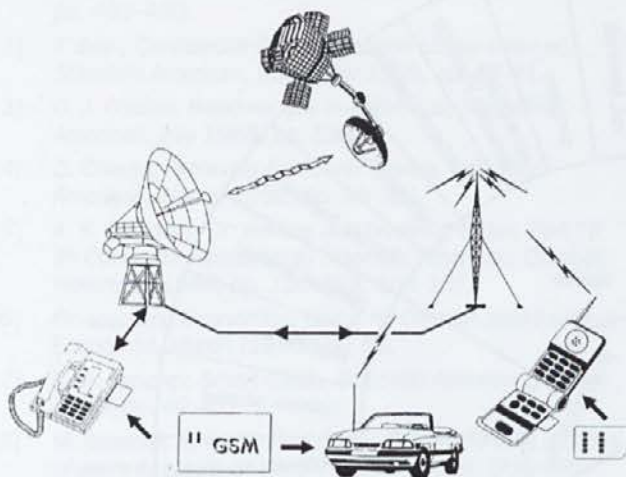
(d) Uporaba v telekomunikacijah

Javni telefoni so daleč največji uporabniki kartic s čipom. Večina telefonskih kartic so vnaprej plačane kartice (*prepaid cards*) in zaenkrat nimajo procesorja, marveč le nekaj logike v obliki integriranega vezja, z ne več kot 100 zlogov pomnilnika. So izredno zanesljive in poceni. Lahko hranijo tudi izbrane telefonske številke in omogočajo hitro klicanje teh števil (speed dialing).

Nekatera telefonska podjetja pa že nudijo telefone s pametnimi karticami.

Danes je mogoče dobiti številke, ki niso vezane na telefonski priključek, marveč na pametno kartico, že v 85-ih državah. V tem primeru je pametna kartica v bistvu sledni telefon" (*follow-me phone*), kar pomeni, da kjerkoli vložimo pametno kartico v telefon in vtipkamo osebno prepoznavno številko, lahko sprejmemo telefonski klic ali pa kličemo sami. Telefon je lahko osebna last ali pa je na razpolago v taksijih, letalih, ladjah, v sposojenih avtomobilih, javnih govornicah itd.

Ta sistem se je začel razvijati leta 1987 in se imenuje GSM (*Global System for Mobile Communications*). Gre za mednarodni mobilni telefonski sistem, ki temelji na digitalnem prenosu in značilnostih pametne kartice. Ima že prek 10 milijonov naročnikov in se bo kmalu razširil po vsem svetu. Uporabnikova kartica potrjuje njegovo identiteto (*Subscriber Identity Module - SIM*). Ko kartico vložimo v telefon, GSM uporabi podatke na kartici in prepoznavno številko za preverjanje naročnikove identitete, klic pa se obračuna v matični državi. Hkrati lahko pametna kartica omogoča tudi šifriran prenos in s tem preprečuje prisluškovanje.



Slika 9: GSM (*Globalni sistem za prenosno komuniciranje*)

(e) Varovanje podatkov in računalniškega sistema

Pametne kartice se lahko uporabljajo kot varnostna ključavnica za računalnike in diske. Če ne vtipkamo pravilnega gesla ali če pametna kartica ni prisotna, se tipkovnica oziroma sistem zaklene. Informacije na disku ali pa kakšnem drugem podatkovnem mediju je moč zakodirati s ključem, ki je bil dodeljen lastniku in spravljen v pametni kartici. Podobno se lahko zaščititi tudi komunikacija med različnimi računalniki.

(f) Plačani televizijski programi

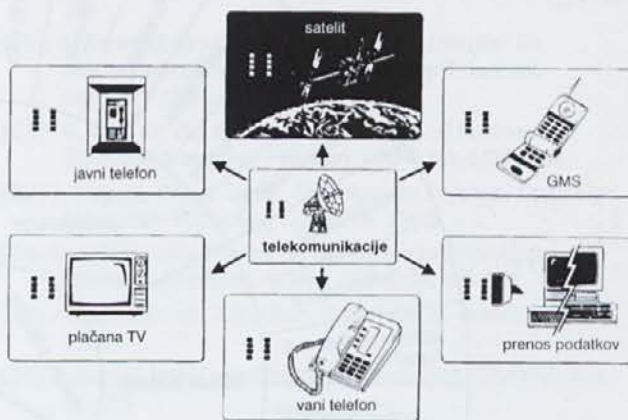
Določene televizijske programe je treba posebej plačati in so zakodirani. Če jih hočemo gledati, moramo plačati naročnino ter dobiti posebno napravo, ki zna program odkodirati. Da pa ne bi prišlo do ponarejanja teh naprav (predvsem kadar je na voljo samo enosmerna komunikacija od televizijskega podjetja do gledalca), je dobro uvesti gesla in jih pogosto zamenjevati, same naprave pa opremljati z ustreznimi kodami. Ta problem se da enostavno rešiti s pametno kartico, saj je veliko lažje menjavati gesla na kartici ali pa kartice kot pa same kodirne naprave.



spredaj

zadaj

Slika 10: Za plačane televizijske programe (*pay-TV*) se običajno uporablja pametna kartica v obliki ključa.



Slika 11: Uporaba pametne kartice v telekomunikacijah in uporabniški elektroniki

(g) Uporaba v vojski

Pametna kartica ima izredno pomembno vlogo v vojski, saj med drugim omogoča nadziranje gibanja, dostop do raznih objektov, uporabo orožja ter zagotavlja tajnost komunikacij.

(h) Druge vrste uporabe

Pametne kartice se uporabljajo na različnih področjih, tudi v transportu, hotelih, pri športu, na razstaviščih, črpalkah

in še marsikje. Kartica lahko vsebuje zapis podatkov ali rezultatov, na primer na borzi. V Angliji so nadomestili vstavljanje kovancev v posebne merilnike za uporabo kurjave in elektrike z vnaprej plačano pametno kartico.

Pametne kartice postajajo vse bolj popularne na vseh koncih sveta. Tako so jih s pridom uporabljali tudi na olimpijskih igrah v Atlanti leta 1996. Vsak športnik je dobil svojo pametno kartico, ki jo je uporabljal za prepoznavanje, kot elektronsko denarnico in za dostop v razne objekte. Več kot milijon vnaprej plačanih kartic so uporabljali v tisočih trgovinah.

Obstajajo tudi druge možnosti uporabe pametne kartice, ki pa zaenkrat se niso tako razširjene, na primer za dostop v zabavišče, kino, gledališče, kot potni list itd.

8. Zaključek

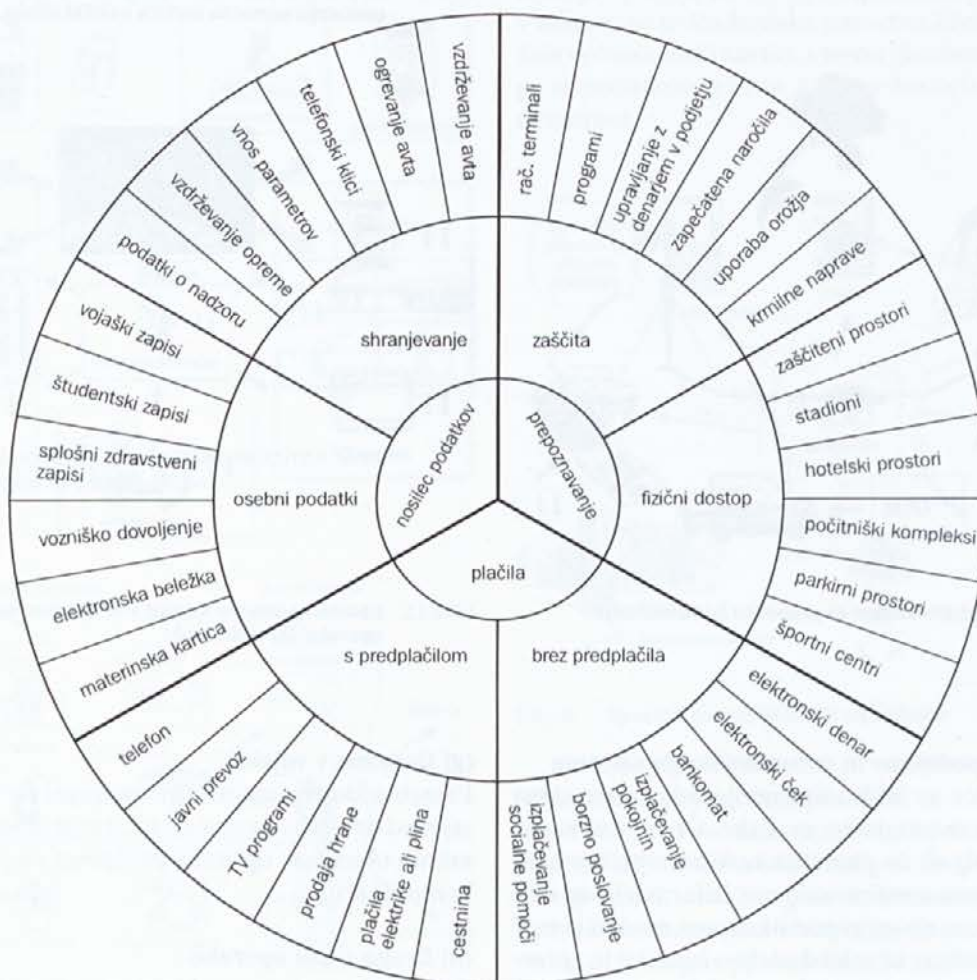
Danes si ne moremo predstavljati osebnega in poslovnega življenja brez kartic. Papirnato kartico, s katero smo se nekoč predstavljali, je z razvojem novih materialov zamenjala plastična kartica s funkcijo

kreditne kartice. Bančna industrija je zaradi potrebe po avtomatizaciji dodala plastični kartici magnetni trak. Premajhen pomnilnik in nezadostna varnost sta glavni pomanjkljivosti magnetne kartice. Precej večjo pomnilno zmogljivost ima optična kartica, največjo zaščito pa nam nudi pametna kartica. Pametna kartica nam omogoča predplačila, brezgotovinsko poslovanje, dostop do objektov, bazo osebnih in drugih podatkov ter zaščito le-teh, predvsem pa onemogoča zlorabo in kriminal ter poenostavlja administracijo.

Nadomestila bo kovance, bančne izpiske, čeke, identifikacijske dokumente, transportne karte, zdravstvene recepte, kreditne kartice, ključe itd, glej sliko 12.

Pametna kartica je računalnik v žepu, njen razvoj in možnosti pa še zdaleč niso zaključene. Na Japonskem in v ZDA so izoblikovali pametno kartico s tipkovnico in majhnim zaslonom (super pametna kartica - *Super Smart Card*), razvita pa je bila tudi že tako imenovana kombinirana kartica (*Combi Card ali Hybrid Card*), ki ima lastnosti kontaktne in brezkontaktne kartice.

Pred kratkim so se Europay, Mastercard in Visa



Slika 12: Področja uporabe pametnih kartic

(EMV) dogovorili za skupno specifikacijo pametne kartice, ki določa osnovne protokole za komunikacijo med kartico in čitalnikom. Ta se ravna po standardu RS-232 za komunikacijo med osebnim računalnikom in modemom.

Specifikacija je dovolj splošna, da lahko izmenjamo katero koli informacijo med računalnikom in programom. To pa je osnova za večnamensko pametno kartico (*Multipurpose Smart Card*). Glavni proizvajalci pametne kartice (Gemplus, Thompson - Francija, Philips - Nizozemska, Motorola - ZDA, Mondex - Anglija, Siemens - Nemčija) so dosegli prvo stopnjo njene tehnične zrelosti. Pomnilna zmogljivost ni več zavirajoč dejavnik, pojavljajo pa se nove dileme, kot so izbira med nivojem zaščite ter ceno kartice, katere informacije zaščititi in komu omogočiti dostop do njih. Prihodnost pametne kartice je odvisna tudi od potreb po zasebnosti, od gospodarskih in političnih razmer, od programskih možnosti in od drugih vplivov. Skratka, pametna kartica je tu in njene možnosti je treba izrabiti

Literatura:

- [1] S. Bassein, *A Sampler of Randomness*, *Scientific American*, *American Math. Monthly*, June-July (1996), pp. 483-490.
- [2] T. Beth, *Confidential Communication on the Internet*, *Scientific American*, December 1995, pp. 87-91.
- [3] G. J. Chaitin, *Randomness in Arithmetic*, *Scientific American*, July 1988, pp. 52-57.
- [4] D. Chaum, *Achieving Electronic Privacy*, *Scientific American*, August 1992, pp. 96-101.
- [5] A. K. Dewdney, *On making and breaking codes, Part I,II (in Computer Recreations)* *Scientific American*, October, November 1988, pp. 120-123, 104-107.
- [6] *Finance and Economics: Going for Olympic gold cards* *Economist*, March (1996), 67-68.
- [7] C. H. Fancher, *Smart Cards*, *Scientific American*, August 1996, pp. 40-45.
- [8] M. Gardner, *A new kind of cipher that would take millions of years to break, (in Mathematical games)*, *Scientific American*, August 1977, pp. 120-124.
- [9] M. Gardner, *Penrose tiles to trapdoor Ciphers*. W.H. Freeman and Company 1989.
- [10] M.E. Haykin and R.B.J. Warnar, *Smart Card Technology: New Methods for Computer Access Control*, NIST, Special Publication 500-157, 1988.
- [11] P.L. Hawkes, D.W. Davies and W.L. Price (eds.), *Integrated Circuit Cards Tags and Tokens*, BSP Professional Books, 1990.
- [12] M. E. Hellman *The Mathematics of Public-Key Cryptography*, *Scientific American*, August 1979, pp. 146-158.
- [13] Jean-Jacques, Myriam, Maurier and Michael Quisquater, Louis, Marie-Annick, Gaid, Anna, Gwenole, and Soazig Guillou (in collaboration with T. Berson, for the English version), *How to Explain Zero-Knowledge Protocols to Your Children*, *Advances in Cryptology - Crypto' 89*, *Lecture Notes in Computer Science* 435, Springer-Verlag Berlin, New York (1990), pp. 628-631.
- [14] D. Kahn, *Modern Cryptology*, *Scientific American*, July 1966, pp. 38-46.
- [15] B. Magajna, *O tajnopisih*, *Obzornik mat. fiz.*, 38 (1991), 9-18.
- [16] J. McCrindle, *Smart Cards*. IFS Publications/Springer-Verlag, 1990.
- [17] C.C. McGeoch, *Zero-knowledge proofs*, *American Math. Monthly*, Aug.-Sep. (1993), pp. 682-685.
- [18] R. McIvor, *Smart Cards*, *Scientific American*, November 1985, pp. 152-159.
- [19] D. Naccache, D. M'Raihi, Gemplus, *Cryptographic Smart Cards*, *IEEE Micro*, Vol. 16, No. 3, June 1996, pp. 14-24.
- [20] I. Steward, *Proof of Purchase on the Internet (in Mathematical Recreations)*, *Scientific American*, October? 1995, two pages.
- [21] G. Stix, *Dr. Big Brother (in Science and Business)*, *Scientific American*, February 1994, pp. 108-110.
- [22] J. Svigals, *Smart Cards, The Ultimate Personal Computer*. Macmillan Publishing Company, 1985.
- [23] P. Wallich, *Wire Pirates (in Trends in Communication)* *Scientific American*, March 1994, pp. 90-101.
- [24] J.L. Zoreda and J.M. Oton, *Smart Cards*. Artech house, 1994.
- [25] J. Zupan, *Nekaj o kriptografskih metodah*, *Obzornik mat. fiz.*, 25

Aleksandar Jurišič je diplomiral leta 1987 pri prof. Vrabcu (*Uporaba topologije v kombinatoriki*) na Fakulteti za matematiko, teoretična smer. Po letu dni magistrskega študija (mentor prof. Pisanski, *teorija grafov*) na domači univerzi je leta 1988 odšel nadaljevat študij na *Department of Combinatorics and Optimization*, University of Waterloo, Kanada, kjer je pod mentorstvom prof. Godsila (*algebraična kombinatorika*) 1990. leta magistriral, 1995. pa doktoriral (*Antipodal covers*). Trenutno opravlja postdoktorski študij iz kriptografije na *Department of Combinatorics and Optimization*, University of Waterloo in Certicom Corp., Mississauga.

Alenka Trojar je diplomirala leta 1988 na *Ekonomski Fakulteti* na plansko-analitski smeri in se zaposlila v Intertradu. Leta 1993 je odšla v Kanado, kjer je 1995. končala program *Diploma in Accounting* na *School of Business and Economics*, Wilfrid Laurier University. Sedaj vodi zastopniško podjetje Amoebius.

DESETO POSVETOVANJE SEKCIJE ZA RAZISKOVANJE INFORMACIJSKIH SISTEMOV

Grimšče - Bled, 13. - 14. februar 1997

Zamisel, ki se je porodila pred desetimi leti med informatiki v Zvezi ekonomistov, to je, da bi se občasno srečevali v ožjem krogu in razpravljali o temah, ki bi izhajale iz raziskav ali izkušeni prisotnih članov, se je uresničila v vsakoletnih srečanjih Sekcije za raziskovanje informacijskih sistemov. Pogoj za članstvo v sekciji je bil, da se posameznik ukvarja z raziskavami in ima kot stopnjo izobrazbe doktorat ali vsaj magisterij, oziroma se pripravlja nanj. Člani so praviloma organizirali razprave v okviru tematskih področij, na katere so vabili tudi ljudi iz prakse. Tako je delo sekcije pravzaprav povezovalo teorijo s prakso. Srečanja je v prvih letih spremljal tudi zbornik referatov. Bila so predvsem koristen način obveščanja, kdo kaj raziskuje v našem okolju, pa tudi učinkovit način izobraževanja tudi za vse udeležence, ki so v programu vselej aktivno sodelovali, bodisi z referatom, bodisi v razpravi.

Tako je bilo tudi na desetem posvetovanju. Vsak od 35 udeležencev je aktivno sodeloval v enem od tematskih področij.

Tematska področja so bila naslednja:

- I. Predstavitev raziskovalnih projektov na področju informacijskih sistemov
- II. Prenova in informatizacija poslovnih procesov
- III. Učenje na daljavo
- IV. Revidiranje in obvladovanje informacijskih sistemov
- V. Informatika in slovenski jezik

Razprava je razen tega tekla še v okviru treh aktualnih tem:

1. Univerza v informacijski družbi
2. Aktualni problemi uvajanja sodobnih informacijskih tehnologij in smeri njihovega reševanja: Izkušnje v državnem zboru Slovenije
3. Raziskovalni dosežki asistentov in mladih raziskovalcev

Večji del letošnjega programa je bil namenjen univerzitetnemu izobraževanju in pregledom raziskav, ki potekajo na obeh univerzah. To je bilo povsem v skladu z aktualno temo Univerza v informacijski družbi, ki so se je udeležili tudi rektor Ljubljanske univerze dr. Alojz Kraj in prorektorja obeh univerz, dr. Janek Musek in dr. Bruno Cvikel. Na vsa vprašanja, ki so se odpirala v času razprave, seveda ni bilo mogoče odgovoriti, bolj jih je bilo mogoče samo osvetliti z več vidikov. Vsekakor drži ugotovitev, da bo učenje na univerzitetni ravni potrebno korenito spremeniti in da pri tem lahko informacijska tehnologija v marsičem pripomore. Lep primer sodobnega pristopa k učenju je projekt Učenje na daljavo, ki ga je v letošnjem letu pričela izvajati Ekonomska fakulteta.

Zanimive in spodbudne so bile tudi predstavitve raziskovalnih projektov na obeh univerzah, prav tako tudi nastopi mladih raziskovalcev.

Koristno srečanje za vse, ki jih zanima, kaj se dogaja v Sloveniji na področju informacijskih sistemov in ki čutijo potrebo, da bi o svojih stališčih razpravljali v skupini kolegov.

K.P.

Dnevi slovenske informatike DSI '97 Informatika za tretje tisočletje

Portorož, 9.4. - 12.4.1997

OKVIRNI PROGRAM POSVETOVANJA

sreda 9. april

Otvoritev posvetovanja
Nagovor predsednika ZRIS
Nagovor predsednika DSI
Častni govornik
Podelitev društvenih priznanj
Vabljen referata

SEKCIJA A: Metodološki vidiki informatike
(vodja Ivan Rozman)

OKROGLA MIZA 1: Velikih sedem v tretjem tisočletju
(vodja Tomaž Banovec)

Družabnosti

četrtek 10. april

SEKCIJA B: Informacijska tehnologija in Internet
(vodja Stane Štefančič)

SEKCIJA C: Poslovne priložnosti informatike
(vodja Franci Mugerle)

OKROGLA MIZA 2: Informatika v izobraževanju
(vodja Vladislav Rajkovič)

Družabnosti

petek 11. april

SEKCIJA D: Informatika in infrastruktura
(vodja Franc Žerdin)

SEKCIJA E: Prenova in informatizacija poslovanja
(vodja Andrej Kovačič)

OKROGLA MIZA 3: Programski produkti:
domače znanje ali uvoz
(vodja Marjan Krisper)

Razglasitev priznanj avtorjem referatov
Družabnosti

sobota 12. april

Vabljen referata

SEKCIJA F: Informacijske rešitve
(vodja Ivan Vežočnik)

OKROGLA MIZA 4: Ugled poklica informatika
(vodja Niko Schlamberger)

Sporočilo posvetovanja
Izžrebanje udeleženca
Zaključek posvetovanja

Informacije: Slovensko društvo INFORMATIKA, fax: 302-370

Marjan Pivka:

KAKOVOST V PROGRAMSKEM INŽENIRSTVU

PRESKUŠANJE, CERTIFICIRANJE IN STANDARDI

Desk, 1996; 283 strani

Razvoj programske opreme se že dolgo ne nahaja več v sferi umetnosti, ko sta programerjev navdih in genialnost pogojevala kvaliteto programa ali aplikacije, pač pa že vrsto let obstajajo trdni temelji in smernice, kako se lotiti izdelave in končno tudi izdelati kvaliteten programski izdelek.

Delo dr. Marjana Pivke govori prav o tem. Razdeljeno je v dva dela. Prvi del obsega poglavja Programska oprema, Presoja sistema kakovosti, Certificiranje programske opreme, Preskušanje programskih izdelkov, Merjenje, razvrščanje in vrednotenje programske opreme, drugi del pa Karakteristike kakovosti in vrednotenje programskih izdelkov po ISO/IEC 9126 ter komentar k standardoma ISO 9000-3 in ISO 9127.

V prvem poglavju je avtor opredelil osnovne pojme s področja programskega inženirstva, okarakteriziral programski izdelek in proces izdelave programske opreme ter opisal njen klasični življenski cikel, razdeljen na faze in aktivnosti. Posebno pozornost je posvetil modelu zorenja, ki opisuje nivoje izdelave programske opreme. Le-ti se gibljejo od začetnega pa vse do optimiranega nivoja procesa izdelave programske opreme, karakteristična zanje pa sta način in vrsta upravljanja omenjenega procesa. Proces izdelave je potrebno čim skrbneje upravljanje zato, ker je v splošnem pričakovati, da bodo zato tudi izdelki, izdelani v okviru tega procesa, kvalitetni. V nadaljevanju je avtor definiral karakteristike kakovosti, ki se dele na notranje, povezane s procesom izdelave, in zunanje, ki so karakteristike izdelanega proizvoda. Konkretno se karakteristike kakovosti programskih izdelkov po standardu ISO 9126 dele na funkcionalnost, zanesljivost, uporabnost, učinkovitost, vzdrževalnost in prenosljivost, kar vse neposredno vpliva na zadovoljevanje potreb uporabnikov.

V drugem poglavju je avtor najprej definiral in opisal sistem kakovosti ter presojo kakovosti, vključno z vsemi njenimi komponentami. Presojno kakovosti je kategoriziral z ozirom na vrsto in vlogo udeležencev pri presojanju ter razložil motive in cilje presoje. Zelo podrobno je opisal izvedbo presoje od priprave nanjo, prek njene izvedbe, pa vse do rezultatov, ki jih od presoje pričakujemo. Poglavje je zaključil z opisom nadzora nad delovanjem sistema kakovosti.

V tretjem poglavju so opredeljeni osnovni koncepti pri certificiranju, kot so sistem certificiranja (v katerem nastopajo organ za certificiranje, proizvajalec in kupec), certificiranje ustreznosti, akreditiranje preizkuševalca oz. presojevalca, certifikacijski postopek ter izdaja certifikata ustreznosti. V četrtem poglavju se je avtor poglobil v področje preizkušanja programskih izdelkov, kjer je najprej razložil namen preskušanja, nato pa predstavil zahteve v zvezi s kakovostjo programskih izdelkov z ozirom na standard ISO/IEC 12119. Zahteve se nanašajo na opis izdelka, uporabniško dokumentacijo in kvaliteto programov in podatkov, pri čemer se le-ta ocenjuje v skladu z že omenjenimi šestimi karakteristikami kakovosti programskih izdelkov. Natanko je opredeljen tudi postopek preskušanja, ki se izvaja v treh fazah - priprava na preskus, izvedba preskusa in izdelava poročila o preskusu.

Peto poglavje se pričinja z razlago temeljnih pojmov na področju merjenja v programskem inženirstvu, ki je aproksimirano s sistemom s povratno zanko. Primerjava referenčnih

podatkov z rezultati procesa obsega merjenje in ocenjevanje rezultatov, razvrščanje rezultatov in njihovo vrednotenje. Podrobno so opisane tudi metrične skale, klasifikacija metrik (metrike procesa, izdelka, virov) in načini merjenja (posredni, neposredni) rezultatov procesa. V nadaljevanju je govora o karakteristikah programskega izdelka, ki so podvržene vrednotenju, in o statističnih obdelavah rezultatov meritev oz. ocenjevanj, ki se agregirajo v končni rezultat - vrednost kakovosti izdelka. Zaključek poglavja je posvečen tehnikam merjenja oz. ocenjevanja karakteristik programskega izdelka ter uvajanju programske metrike v prakso.

Šesto poglavje obsega komentirane smernice ISO 9000-3, ki se nanašajo na področje programske opreme. To so navodila, kako razumeti in uporabljati standard ISO 9001 pri razvoju, dobavi in vzdrževanju programske opreme.

V sedmem poglavju so predstavljene karakteristike kakovosti in vrednotenje programskih izdelkov po standardu ISO/IEC 9126. Najprej je predstavljen generični model kakovosti programske opreme, ilustriran s posebnim modelom, ki se nanaša na modularnost in metrike modularnosti. Po omenjenem standardu se kakovost ugotavlja na osnovi šestih, že v prvem poglavju omenjenih karakteristik. V nadaljevanju je predstavljen model postopka vrednotenja, opisane so faze postopka vrednotenja - definiranje zahtev, priprava in samo vrednotenje.

Osmo poglavje obsega standard ISO 9127, ki definira obseg in okvirno vsebino uporabniške dokumentacije kot sestavnega dela programskega izdelka. Tudi ta standard je, podobno kot v šestem poglavju, opremljen z obsežnimi avtorjevimi komentarji.

Delo dr. Marjana Pivke obravnava razmeroma zapleteno problematiko, vendar na zanimiv in razumljiv način, ki bralca pritegne, k čemer v veliki meri prispevajo tudi zgovorne ilustracije, brskanje po knjigi pa v veliki meri olajša tudi stvarno kazalo. Knjiga ima po moji sodbi trojni namen. Po eni strani naj služi kot učbenik in dodatno gradivo v okviru univerzitetnega študija, po drugi strani pa kot referenčno delo za proizvajalce programske opreme, katerih želja je pridobiti certifikate za svoje procese proizvodnje in seveda tudi končne izdelke. Delo pa je namenjeno tudi širšemu krogu bralcev, ki se žele seznaniti s problemom ugotavljanja in dokazovanja kakovosti izdelkov. Čeprav se delo posveča predvsem preizkušanju, certificiranju in standardizaciji programske opreme, je toliko splošno, da dobi bralec vpogled tudi v širšo problematiko ugotavljanja in zagotavljanja kakovosti ne glede na vrsto proizvodov ali storitev. Kot sklep naj zapišem - delo dr. Marjana Pivke bralcem tople priporočam.

T. Mohorič

Donald H. Taylor, G. William Glezen:

Revidiranje zasnove in postopki

Zveza računovodij, finančnikov in revizorjev Slovenije

Slovenski inštitut za revizijo je izdal prevod učbenika *Auditing: Integrated Concepts and Procedures* avtorjev Donald H. Taylorja in G. Williama Glezena, ki je v slovenščini izšel pod naslovom *Revidiranje - zasnove in postopki*. Založila ga je Zveza računovodij, finančnikov in revizorjev Slovenije.

Prevod tega učbenika je rezultat dveletnega dela prevajalke Alenke Černe, lektorice Mojce Luštrek in recenzenta prof. dr. Ivana Turka, ki je oblikoval slovensko strokovno izražje. Učbenik se "bere" in nimamo občutka, da je preveden. Seveda je prevod spremljalo veliko težav, povezanih z izrazi, in prevod je tudi dodaten prispevek k obogatitvi slovenskega jezika predvsem s slovenskim strokovnim izrazjem s področja revidiranja.

Na začetku vsakega poglavja so predstavljeni učni cilji oziroma zahteve, na katere bi bralec moral znati odgovoriti, ko bi prebral in preštudiral snov tega poglavja. Osrednji del je teoretična in tudi praktična predstavitev obravnavane snovi. Ob koncu vsakega poglavja so vprašanja za ponavljanje, ki bralcu oziroma uporabniku omogočajo ugotoviti, kako dobro je preštudiral snov. Vprašanja so prirejena po izpitih za naziv poo-

blaščeni revizor (*Certified Public Accountant, CPA*) in zahtevajo od bralca, da teoretično predstavljeno snov dobro preštudira in odgovori nanje. Zastavljena so tako, da bralca silijo k razmišljanju in logičnemu povezovanju teoretično predstavljene snovi s prigradki iz prakse. Učbenik je šolski primer dobrega učbenika, zato bi bil lahko zgled tudi nekaterim piscem domačih učbenikov.

Komu je učbenik namenjen? Gotovo bodo po njem pogosto posegali revizorji in pooblašeni revizorji ter vsi tisti, ki se za ta poklic usposablajo. Pri študiju ga bodo uporabljali študenti obeh univerz, pri svojem delu pa notranji revizorji, poslovodje, računovodje in še mnogi drugi. Ker pa je revizija meddisciplinarna stroka in so v učbeniku obravnavana še druga področja, jo priporočam tudi pravnikom, računalničarjem (predvsem programerjem), poslovnim finančnikom, vsem ravnem poslovodstvu, pa seveda vsem tistim, ki ste ali pa še boste imeli opravka z revizijo oziroma ste bralci revizijskih poročil.

M. Odar

Borka Jerman-Blažič:

Internet

Novi Forum, d.o.o., Ljubljana, 1996, 87 strani

Internet je, kot je zapisala avtorica v uvodu, dosežek devetdesetih let in kot kaže tudi najpomembnejši in najvidnejši znanilec prihajajoče informacijske družbe. Kljub temu, da uporaba Interneta eksplozivno narašča, pa o njem povprečen Slovenec še vedno relativno malo ve. To kažejo opravljene raziskave in o tem se lahko sami tudi prepričamo, če se nekoliko ozremo po svojem okolju. Večina uporabnikov Interneta izhaja iz akademske in raziskovalne sfere ter iz populacije šolajoče se mladine. Med podjetniki jih še vedno relativno malo misli, da bi lahko bil Internet, tukaj in zdaj, že tudi poslovno zanimiv. Večina gleda na Internet kot na modno novost, ki ji bo potrebno dati še precej časa, da bo zrasla v stičišče novih poslovnih priložnosti.

Knjiga, ki jo predstavljamo, bi zato lahko prispevala k boljšemu razumevanju tega, kaj Internet je in kakšne poslovne priložnosti nudi uporabnikom. Razdeljena je v pet poglavij, ki dokaj dobro zaokrožajo informacije in znanja, ki jih potrebujemo o Internetu, da si ustvarimo o njem grobo predstavbo.

V prvem poglavju knjiga na kratko predstavi zgodovinski razvoj računalniških omrežij, ki je v slabih treh desetletjih pripeljal do Interneta, kakršnega poznamo danes in opredeli, kaj pravzaprav Internet je. Drugo poglavje podrobneje opiše

osnove Interneta, njegovo konceptualno zasnovo, prenosne medije ter osnovne protokole. Tretje poglavje je za začetnike, ki še nič ne vedo, kaj vse nam Internet omogoča, morda najpomembnejše. Predstavi nam temeljne storitve Interneta od elektronske pošte, kot osnovne storitve globalnega omrežja, omrežnih novic, svetovnega spleta ter osnovnih orodij za delo na Internetu. Peto poglavje se ukvarja z možnostmi rabe Interneta v poslovne namene. Govori o trženju in oglaševanju prek Interneta, o elektronskem trgovanju in o problemih varovanja podatkov, ki pri tem nastajajo. Zadnje, to je peto poglavje, govori o Internetu v Sloveniji, o tem, kako se vključiti v Internet, ter o ponudnikih Internetovih storitev.

V knjigi nismo zasledili napotka, komu je knjiga pravzaprav namenjena. Menimo, da bo knjiga, ki ni preobsežna in je opremljena s številnimi ilustracijami, koristno in zanimivo branje predvsem za tiste, ki mislijo, da bi o Internetu, njegovem delovanju ter potencialnih možnostih morali vedeli nekaj več pa do zdaj še niso našli časa ali priložnosti, da bi se o tem konkretnije podučili.

M. Vintar

SERIJSKA IN BUTIČNA PROIZVODNJA PROGRAMSKE OPREME

V februarški številki revije Windows Magazine je ga. Cheryl Currid napisala odprto pismo proizvajalcem serijske programske opreme. Naštejmo nekaj poudarkov iz njenega pisma:

- Postanete lahko sam svoj sovražnik, ker ne nudite prave ali vsaj zadostne podpore niti pri instalaciji opreme niti pri konfiguraciji, da o podpori pri sami uporabi niti ne govorimo.
- Informatiki v podjetjih ali ustanovah ne želijo več kupovati programske opreme, ki je uporabniki ne morejo uporabljati brez obsežnega dodatnega učenja.
- Čeprav je konkurenca velika, čeprav ste v stalni nevarnosti, da vaša ideja pograbí kak SW Golijat, je premalo, če pripravite zgoščenko in jo pospremite le z dobrimi željami.
- Kako se torej lotiti zadeve? Bodite pošteni, zato naj vaše marketinško sporočilo navede pravi poslovni ali administrativni problem, ki ga skuša oprema rešiti. Oprema naj deluje podobno kot ljudje. Ne izgublajte preveč energije na vmesnikih (n.pr. na grafičnem), uporabite, kar je na voljo. Če hočete imeti programe res stestirane, angažirajte profesionalce. Nikar ne pošiljajte svoje beta verzije prijateljem in znancem, ker je malo verjetno, da boste dobili sistematično preverjen proizvod nazaj.

Ni kaj, sporočilo je jasno: Uporaba serijske programske opreme naj bo enostavna, četudi ima priloženo neposredno pomoč. Uporabniki namreč bežijo pred dolgotrajnim učenjem pred pričetkom uporabe, še zlasti, če ne vedo, ali jim bo oprema res v tolikšno pomoč kot jih ponudnik prepričuje.

Ob tem se človek vpraša, kakšno sporočilo naj bi naslovil na izdelovalce "butičnega softvera". Vendar je v tem primeru prav, da vključimo tudi naročnike. Brez njih ni naročil, ni specifikacij itn. In kakšno naj bo sporočilo, če so naročniki iz državne uprave? Saj vemo, da mora praktično vsak tovrstni projekt na javni razpis. Razpisne komisije za področje izdelave programske opreme pa praviloma "pozabljajo" na svojo profesionalnost in pristajajo na nedopustno nizko oceno potrebnega obsega del. Izdelovalci

opreme, ki se prijavimo na tak razpis, sicer vemo, da je vrednost del podcenjena, vendar nas različni razlogi (dostikrat preživetveni) silijo, da se prijavimo.

Po dokončanem izbirnem postopku pride do podpisa pogodbe. Pri tem naročnik praviloma pristane na predlagano metodologijo dela, izvajalec pa pristane na funkcionalnost sistema, ki bo znana šele z opredelitvijo zahtev.

Pri samem delu na projektu potem izvajalec ne more dobiti pravih naročnikovih strokovnjakov, ki dobro poznajo področje, ker so preveč zasedeni. Zato še tako ekspresni razvoj aplikacij odpove in se čas za opredelitev zahtev podaljša vsaj za dvakrat. Ko je prototip končno predstavljen, postanejo predstavniki naročnika bolj živahni, denarja na projektu pa je že zmanjkalo (zgodilo se mi je, da je predstavnik naročnika pred podpisom pogodbe obljubljal dodatna sredstva, pozneje pa je na to obljubo "pozabil"). Izvajalčev tim je dovolj profesionalen, da projekt dokonča, četudi visoko strokovni specialisti glede na porabljeni čas ne zaslužijo dosti več kot frizerski pomočnik.

Kakšno naj bo torej sporočilo kreatorjem butičnega softvera?

Kljub vsem naštetim težavam kot jih vidi predstavnik izvajalčeve strani, bi se sporočilo lahko glasilo:

Naročnik butičnega softvera naj v posebnem projektu da opredeliti zahteve, ki jim bo zadostil softver, če nima na voljo dovolj profesionalne ekipe, ki je sposobna to delo kvalitetno opraviti. Če pa je ekipa prava, bo skupaj z izvajalcem znala opredeliti zahteve in nato še objektivno oceniti potrebni obseg dela.

Naročnik in izvajalec se morata potruditi, da bo med njima obstajal ves čas dela na projektu partnerski in pošten odnos kot ga predvidevajo standardi o kakovosti. Potem bo delo opravljeno korektno, profesionalno in v obojestransko zadovoljstvo.

Če ni zadostnega zaupanja med njima, si lahko zagotovita korektno odnose v pogodbi, vendar sklicevanje na pogodbo gotovo ni znamenje zaupanja.

Franc Žerdin

50 LET ACM

Najstarejše računalniško združenje ACM (Association for Computing) je bilo ustanovljeno leta 1947. Je mednarodna znanstvena in vzgojna organizacija, ki je bila ustanovljena z namenom, da pospešuje umetnost, znanost, inženirstvo in uporabnost informacijske tehnologije. Podpira profesionalne in javne interese s tem, da goji odprto izmenjavo informacij in pospešuje profesionalne in etične standarde. Približno v tem smislu je združenje predstavilo uredništvo revije "Communications of ACM" v februarški številki, ki je v celoti posvečena 50. obletnici ustanovitve ACM. Uredništvo je pozvalo večje število eminentnih strokovnjakov s področja informacijske tehnologije (IT), naj sodelujejo v tej svečani številki. Odziv je bil enkraten. Februarška številka obsega kar 170 strani zanimivega, včasih kar provokativnega branja.

Večina avtorjev opisuje svoje videnje bodočega razvoja IT, vendar ne v obliki napovedovanja, ampak v obliki lastnih želja, kako naj bi razvoj izgledal, recimo čez 50 let.

Da bi lažje koordinirali prispevke, so jih razdelili v tri poglavja: sociološke spremembe, socialne dimenzije in v znanost bodoče tehnologije.

Poglavje socioloških sprememb je prvenstveno namenjeno medsebojni povezanosti med ljudmi in stroji - računalniki ter povečevanju te zveze s časom. Avtorji razpravljajo o različnih interakcijah ljudi in računalnikov in o vplivu teh interakcij na naše življenje. Večina jih je mnenja, da bo ta vez postajala s časom vedno močnejša.

V poglavju socialnih dimenzij avtorji razpravljajo o prihodnjih dekadah in kakšen vpliv bo imela tehnologija na kulturo, gospodarstvo in ekologijo. Lotevajo se tem kot so svoboda, človek-

ove pravice in zasebnost in kako naj bi se teh tem lotile bodoče generacije. Sugerirajo tudi, kaj bi se bodoče generacije morale naučiti iz naših napak.

Članki o poglavju Znanost bodoče tehnologije obravnavajo različna razvojna področja in nove tehnologije, ki bodo med temeljnimi silami napredka v prihodnosti. Opisujejo od kod prihajamo in kam gremo. Dotikajo se tem kot so programsko inženirstvo, vloga raziskav, nove razsežnosti telekomunikacij in vmesna prizadevanja.

Pa še kratek izvleček iz prispevka Gordona Bella (nekateri ga imenujejo "oče" miniračunalnikov, ki jih je razvil pri firmi Digital, sedaj pa je raziskovalec pri Microsoftu). Čez 50 let bo po njegovem mnenju vsa vsebina v kibernetnem prostoru (ta prostor je platforma in omrežna infrastruktura, ki vključuje vse komunikacije, video in računalniške naprave). Po napovedih bodo računalniki vsaj 100.000-krat zmogljivejši kot so sedanji. Če se bo strojna oprema še naprej razvijala po Moore-ovem zakonu, ki pravi, da se letno izboljša za faktor 1.6, bodo obstajali tudi računalniki, ki bodo milijonkrat in več zmogljivejši kot so sedanji. Računalniki, povezani v mreže, bodo prisotni povsod, v telefonih, stikalih, motorjih, zgradbah, na avtocestah, povsod. Edina omejitev, ki jo vidi avtor, bodo omrežja in naša sposobnost izdelati takšne računalnike, da bodo znali sodelovati z različnimi deli realnega sveta.

Vsem zainteresiranim bralcem želim čimveč zabave in novih spoznanj ob branju svečane februarške številke revije "Communications of ACM".

Franc Žerdin

**IFIP TC 11 Conference on Information Security:
Research and Business**

Copenhagen, DK, 14.-16.5.1997

**Organizatorja:
IFIP TC 11, DANFIP/DD**

Informacije:
e-pošta: coopjacl@inet.uni-c.dk, fax: +45 39 27 33 22



**10th International Bled Electronic Commerce Conference
Global Business in Practice**

Bled, 9.-10.6.1997,

**Organizator:
Fakulteta za organizacijske vede, Kranj**

Informacije:
e-pošta: gricar@uni-lj.si, fax: 064 221 224



**OTS '97 - Objektna tehnologija v Sloveniji
Prehod na objektno tehnologijo**

Maribor, 18. - 19.6.1997

**Organizator:
Center za objektno tehnologijo, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Inštitut za informatiko**

Informacije:
Marjan Heričko, Center za objektno tehnologijo, Smetanova 17, 2000 Maribor
e-pošta: cot@uni-mb.si, URL: <http://lisa.uni-mb.si/cot>, fax: (062) 225 013, tel: (062) 221 112



Symposium on Transportation Systems

Chania, GR, 16.-18.6.1997

**Organizatorja:
IFAC, IFIP TCS5**

Informacije:
e-pošta: tasos@ergasya.tuc.gr, fax: +30 821 69410



19th IFIP Conference on System Modelling & Optimatization

Cambridge, UK, 12.-16.7.1997

Organizatorji: IFIP TC7, DAMTP, University of Cambridge

Informacije:
e-pošta: m.j.d.powell@dampt.cam.ac.uk, fax: +44 1223 33 79 18



6th IFIP Conference on Human-Computer Interaction

Sydney, AUS, 14.-18.7.1997

**Organizator:
IFIP TC 13**

Informacije:
e-pošta: judy@socs.uts.edu.au, fax: +61 2330 1807

Conference on System Modelling and Optimization

Detroit, MI, USA, 22.-25.7.1997

Organizatorji:

IFIP TC7, Oakland University & Wayne State University

Informacije:

e-pošta: polis@vela.acs.oakland.edu, fax: +1 810 370 4261

**Informatics (Computer Science) as a discipline, and in other disciplines:**

What is in common?

Enschede, NL, 10.-15.8.1997

Organizatorji:

IFIP WG3.2

Informacije:

e-pošta: vdhoeven@cs.utwente.n

**Association for Information Systems Americas Conference**

Indiana, USA, 15.-17.8.1997

Informacije:

e-pošta: ljessup@indiana.edu, fax: 812 855 8679

**8th Intl. Conference on Production Engineering - Rapid Product Development**

Sapporo, J, 18.-20.8.1997

Organizatorji:

JSPE CIRP, IFIP WG5.2 and 5.3

Informacije:

e-pošta: isn00003@niftyserve.or.jp, fax: +81 3 33 67 0994

**Work Conference on Intelligent Networks**

Versailles, F, september 1997

Organizatorja: IFIP TC6, Telecom

Informacije:

e-pošta: guy.pujolle@prism.uvsq.fr

**First East-European****Symposium on Advances in Databases and Information System - ADBIS'97**

St. Petersburg, Russia, 2.-5.9.1997

Organizator:

Russian Foundation for Basic Research

Informacije:

e-pošta: adbis97@sigmod.ipi.ac.ru

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 **ASCII** formatu. Na disketi označite, kateri urejevalnik ste uporabili, in ime datoteke. Datoteko imenujte s svojim priimkom, n. pr. Novak.doc ali Novak.txt.

Slike, ki ste jih izdelali z grafičnim programom, označite podobno. Na natisnjem izvodu članka naj bo jasno vidno, kam sodi posamezna slika. Lahko priložite tudi originalne predloge, ki jih na hrbtni strani označite s številkami, tako kot v natisnjem besedilu.

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

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 7.200, za vsak nadaljnji izvod SIT 5.000.

Letna naročnina za posameznika SIT 4.000, za študente SIT 1.200.

